

Data Transfer Service

Data Migration

Product Documentation



Copyright Notice

©2013-2024 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



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

Data Migration

Databases Supported by Data Migration

Cross-Account TencentDB Instance Migration

Migration to MySQL Series

Migrating to MySQL

Migration from MySQL to TencentDB for MySQL

Migration from MariaDB/Percona to TencentDB for MySQL

Migration from TDSQL for MySQL to TencentDB for MySQL

Migrating to MariaDB

Migration from MySQL to TencentDB for MariaDB (MySQL/MariaDB/Percona)

Migration from TDSQL for MySQL to TencentDB for MariaDB

Migrating to TDSQL-C for MySQL

Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL

Migration from TDSQL for MySQL to TDSQL-C for MySQL

Migrating to TDSQL MySQL

Migration from TDSQL for MySQL to TDSQL for MySQL

Migration from MySQL/MariaDB/Percona/TDSQL MySQL to TDSQL for MySQL

Migrating to TDSQL TDSStore

Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDSStore)

Advanced Operations for MySQL Series Migration

Migrating Advanced Object

Creating Data Consistency Check Task (MySQL)

Account Migration

Database/Table Renaming

Migrating Online DDL Temp Table

Processing for MySQL Series Pre-Check Failure

Database Connection Check

Peripheral Check

Version Check

Source Instance Permission Check

Partial Database Parameter Check

Target Instance Permission Check

Target Database Content Conflict Check

Target Database Space Check

Binlog Parameter Check

Foreign Key Dependency Check

View Check

Advanced Object Check

Warning Item Check

TDSQL Proxy Check

TDSQL Kernel Check

DDL Ring Linkage Synchronization Detection for Same Database and Table Objects

DDL Synchronization Conflict Detection for the Same Target and Same Database and Table Object

Offline Migration of MySQL Data

Migrating to PostgreSQL

Migration from PostgreSQL to TencentDB for PostgreSQL

Migration from TDSQL-C PostgreSQL to TencentDB for PostgreSQL

PostgreSQL Physical Migration

Processing for PostgreSQL Series Pre-Check Failure

Database Connection Check

Object Dependency Check

Version Check

Source or Target Instance Permission Check

Incremental Migration Pre Check

Structural Compatibility Check

Structural Conflict Check

Account Conflict Check

Parameter Configuration Conflict Check

Plugin Compatibility Check

Multi Task Conflict Detection

Long Transaction Check

Migrating to MongoDB

Migration from MongoDB to TencentDB for MongoDB

Processing for MongoDB Series Pre-Check Failure

MongoDB Connection Check

Database/Table Content Conflict Check

Source Database Node Role Check

Oplog Check

Source/Target Database Account Permission Check

Database Version Check

Database Capacity Check

Target Database Load Check

ShardKey Check

- Source Database Balancer Check
- Time Series Collection Verification
- Compression Algorithm Verification
- MongoDB Data Consistency Verification
 - Description of Consistency Verification Function
 - Creating Data Consistency Check Task
 - Common Consistency Verification Issues
- Migrating to SQL Server
 - Migration from SQL Server to TencentDB for SQL Server
 - Processing for SQL Server Pre-Check Failure
 - Migration Parameter Check
 - Migration Network Check
 - Source Library Connectivity Check
 - Local Disk Space Check
 - Database Version Check
 - User Permission Check
 - Source Database Existence Check
 - Target Database Existence Check
- Migrating to TencentDB for Redis
 - Migration from Redis to TencentDB for Redis
 - Fix for Verification Failure
 - Network Connectivity
 - Source-Target Instance Version Compatibility
 - Source Instance Parameter Check
 - Target Instance Capacity Check
 - Target Instance Status Check
 - Source Instance Type (Master or Replica)
- Migrating to KeeWiDB
 - Migration Redis to KeeWiDB
 - Migration Tendis to KeeWiDB
- Task Management
 - Task Status Description
 - Viewing Task
 - Configuring Task
 - Verifying Task
 - Start Task
 - Retrying Task (MySQL)
 - Retrying Task (Redis)

Speed Limit (MySQL)

Complete Task

Terminating Task

Deleting Task

Pausing Task

Resuming Task

Changing Configuration

Terminating Task

Deisolating Task

Deleting Task

Managing Tag

Creating a Similar Task

Viewing Log

Data Migration

Databases Supported by Data Migration

Last updated : 2024-07-08 19:41:52

Scenario Overview

Data migration is a one-time task of replicating data from the source database to the target database at the database level. After the migration is completed, the business will be cut over to the new database.

DTS supports data migration from self-built databases, TencentDB databases, and third-party cloud databases.

Scenario	Description
Migration from a self-built database to the cloud	Migrating a self-built database in the local IDC or on CVM to a TencentDB database.
Migrating from third-party cloud vendors to Tencent Cloud.	Such as migration from an Alibaba Cloud or AWS database to TencentDB.
Migration between TencentDB instances	Database version upgrade, cross-region migration (including migration between regions inside China or outside China, or cross-border migration), and the migration between database instances under different Tencent Cloud accounts.

Feature Description

DTS supports access methods such as public network/self-built on CVM/DC/VPN access/CDB/CCN/VPC, with different deployment modes of databases offering various access options. Each access method requires specific network conditions. Please see [Preparations](#).

Self-built IDC databases/other cloud vendor databases: Access methods can include public network/DC/VPN access/CCN.

Self-built databases on CVM: Select CVM self-built as the access method.

CBD instances: Select CDB as the access method.

Migrating to MySQL

Note:

The version requirements for each database migration are as follows:

The data migration module does not support migrating out of cloud when the source is the TDSQL-C MySQL database type. If you have such cloud migration needs, please use the data sync module.

MySQL/TDSQL MySQL/MariaDB/TDSQL-C MySQL/TDSQL MySQL: The target database version must be equal to or later than the source database version, distinguished by the major version number. For example, 5.6.x can be migrated to 5.6.x, 5.7.x, or later versions.

MySQL > MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MySQL 5.5, 5.6, 5.7, 8.0 Self-built MySQL 5.5, 5.6, 5.7, 8.0 on CVM	TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0	Structure migration Full migration Full + incremental Migration	Migration from MySQL to TencentDB for MySQL
Third-Party Cloud Vendors Alibaba Cloud RDS 5.5, 5.6, 5.7, 8.0 Alibaba Cloud PolarDB 5.6, 5.7, 8.0 AWS RDS MySQL 5.6,5.7,8.0 AWS Aurora MySQL 5.6,5.7			
TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0 Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts			

MariaDB > MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MariaDB 5.5, 10.0-10.6 Self-built MariaDB 5.5, 10.0-10.6 on CVM	TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0	Structure migration Full migration Full + incremental	Migration from MariaDB/Percona to TencentDB for MySQL

TencentDB for MariaDB (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4) Sync between databases under the same Tencent Cloud root account.		migration	
---	--	-----------	--

Percona > MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC Self-built Percona 5.5, 5.6, 5.7, 8.0 Self-built Percona 5.5, 5.6, 5.7, 8.0 on CVM	TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0	Structure migration Full migration Full + incremental migration	Migration from MariaDB/Percona to TencentDB for MySQL

TDSQL MySQL > MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TencentDB for MySQL (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same root account	TencentDB for MySQL	Structure migration Full migration Full + incremental migration	Migration from TDSQL for MySQL to TencentDB for MySQL

Migrating to MariaDB

MySQL > MariaDB

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MySQL 5.5, 5.6, 5.7, 8.0	TencentDB for MariaDB (Database versions: MySQL 8.0, Percona	Structure migration Full migration	Migration from MySQL to TencentDB for MariaDB

Self-built MySQL 5.5, 5.6, 5.7, 8.0 on CVM	5.7, MariaDB 10.1, MariaDB 10.4)	Full + incremental migration	
TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0 Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts			

MariaDB > MariaDB

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MariaDB 5.5, 10.0-10.6 Self-built MariaDB 5.5, 10.0-10.6 on CVM	TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4)	Structure migration Full migration Full + incremental migration	Migration from MySQL to TencentDB for MariaDB (MySQL/MariaDB/Percona)
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4) Sync between databases under the same Tencent Cloud root account			

Percona > MariaDB

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built Percona 5.5, 5.6, 5.7, 8.0 Self-built Percona 5.5, 5.6, 5.7, 8.0 on CVM	TencentDB for MariaDB (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4)	Structure migration Full migration Full + incremental migration	Migration from MySQL to TencentDB for MariaDB (MySQL/MariaDB/Percona)

TDSQL MySQL > MariaDB

--	--	--	--

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same root account	TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4)	Structure migration Full migration Full + incremental migration	Migration from TDSQL for MySQL to TencentDB for MariaDB

Migrating to TDSQL-C for MySQL

MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MySQL 5.5, 5.6, 5.7, 8.0 Self-built MySQL 5.5, 5.6, 5.7, 8.0 on CVM	TDSQL-C for MySQL 5.7, 8.0	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL
Third-Party Cloud Vendors Alibaba Cloud RDS 5.5, 5.6, 5.7, 8.0 Alibaba Cloud PolarDB 5.6, 5.7, 8.0 AWS RDS MySQL 5.6,5.7,8.0 AWS Aurora MySQL 5.6,5.7			
TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0 Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts			

MariaDB > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
-----------------------------	-----------------------------	----------------	-------------------------

Self-Built Database IDC self-built MariaDB 5.5, 10.0-10.6 Self-built MariaDB 5.5, 10.0-10.6 on CVM	TDSQL-C for MySQL 5.7, 8.0	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4) Sync between databases under the same Tencent Cloud root account			

Percona > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built Percona 5.5, 5.6, 5.7, 8.0 Self-built Percona 5.5, 5.6, 5.7, 8.0 on CVM	TDSQL-C for MySQL 5.7, 8.0	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL

TDSQL MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TencentDB for MySQL (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same root account	TDSQL-C for MySQL 5.7, 8.0	Full migration Full + incremental migration	Migration from TDSQL for MySQL to TDSQL-C for MySQL

Migrating to TDSQL MySQL

MySQL > TDSQL MySQL

Source Database and Version	Target Database and	Migration	Reference Documentation
-----------------------------	---------------------	-----------	-------------------------

	Version	Type	
Self-Built Database IDC self-built MySQL 5.6, 5.7, 8.0 Self-built MySQL 5.6,5.7,8.0 on CVM	TDSQL for MySQL (database version MySQL 8.0)	Full migration Full + incremental migration	Migration from TDSQL for MySQL to TDSQL for MySQL
TencentDB for MySQL 5.6,5.7,8.0 Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts			

MariaDB > TDSQL MySQL

For homogeneous migration from MariaDB to TDSQL for MySQL (MariaDB), the target database version must be later than or equal to the source database version. For heterogeneous migration, currently only MariaDB 10.1 to TDSQL for MySQL (Percona 5.7) is supported.

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MariaDB 5.5, 10.0-10.6 Self-built MariaDB 5.5, 10.0-10.6 on CVM	TencentDB for MySQL (database version: MariaDB 10.1)	Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL for MySQL
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4) Sync between databases under the same Tencent Cloud root account			

Percona > TDSQL MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database	TencentDB for	Full migration	Migration from

IDC self-built Percona 5.5, 5.6, 5.7, 8.0 Self-built Percona 5.5, 5.6, 5.7, 8.0 on CVM	TDSQL for MySQL (database version: Percona 5.7)	Full + incremental migration	MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL for MySQL
---	--	------------------------------	--

TDSQL-C MySQL > TDSQL MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TDSQL-C for MySQL 5.7, 8.0	TencentDB for MySQL (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL for MySQL

TDSQL MySQL > TDSQL MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TencentDB TDSQL MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same root account	TencentDB for TDSQL MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	Structure migration Full migration Full + incremental migration	Migration from TDSQL for MySQL to TencentDB for MySQL

MySQL > TDSQL MySQL(TDStore)

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MySQL 5.5, 5.6, 5.7, 8.0 Self-built MySQL 5.5, 5.6, 5.7, 8.0 on CVM	TencentDB for TDSQL MySQL (TDStore)	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)
TencentDB for MySQL 5.5, 5.6, 5.7, and 8.0 Sync between databases under the same Tencent Cloud root account			

Sync between databases
under different Tencent
Cloud root accounts

MariaDB > TDSQL MySQL(TDStore)

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built MariaDB 5.5, 10.0-10.6 Self-built MariaDB 5.5, 10.0- 10.6 on CVM	TencentDB for TDSQL MySQL (TDStore)	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4) Sync between databases under the same Tencent Cloud root account			

Percona > TDSQL MySQL(TDStore)

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-Built Database IDC self-built Percona 5.5, 5.6, 5.7, 8.0 Self-built Percona 5.5, 5.6, 5.7, 8.0 on CVM	TencentDB for TDSQL MySQL (TDStore)	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)

TDSQL MySQL > TDSQL MySQL(TDStore)

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TencentDB for TDSQL MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	TencentDB for TDSQL MySQL (TDStore)	Structure migration Full migration Full + incremental	Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)

Migration between databases under the same root account		migration	
---	--	-----------	--

Migrating to PostgreSQL

PostgreSQL > PostgreSQL

Only versions 9.4 and above support incremental migration.

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
Self-built PostgreSQL Databases 9.3, 9.4, 9.5, 9.6, 10-15 IDC self-built Self-built on CVM	TencentDB for PostgreSQL 10, 11, 12, 13, 14, 15	Structure migration Full migration Full + incremental migration	PostgreSQL Migrating to PostgreSQL
Third-party cloud vendor PostgreSQL 9.3, 9.4, 9.5, 9.6, 10-15 Alibaba Cloud RDS PostgreSQL AWS RDS PostgreSQL Huawei Cloud RDS for PostgreSQL			
TencentDB for PostgreSQL 10-15 Migration between databases under the same root account Migration between databases under different root accounts			

TDSQL-C PostgreSQL > PostgreSQL

Source Database	Target Database	Migration Type	Reference Documentation
TDSQL-C PostgreSQL 10.17 Migration between databases under the same root account	TencentDB for PostgreSQL 10, 11, 12, 13, 14, 15	Structure migration Full migration Full + incremental migration	Migration from TDSQL-C for PostgreSQL to PostgreSQL

Migrating to MongoDB

MongoDB > MongoDB

Source Database	Target Database	Migration Type	Reference Documentation
Self-Built Database IDC self-built MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0 Self-built MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0 on CVM	TencentDB for MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.4, 5.0, 6.0	Full migration Full + incremental migration	Migration from MongoDB to TencentDB for MongoDB
Third-party cloud vendors (Alibaba Cloud) MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.4, 5.0, 6.0			
TencentDB for MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.4, 5.0, 6.0 Migration between databases under the same root account Migration between databases under different root accounts			

Migrating to SQL Server

SQL Server > SQL Server

Source Database	Target Database	Migration Type	Reference Documentation
Self-Built Database IDC self-built SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, 2022 Self-built SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, 2022 on CVM	SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, 2022 Supported by TencentDB for SQLServer	Full migration Full + incremental migration	Migration from SQL Server to TencentDB for SQL Server
Third-party cloud vendors			

(Alibaba Cloud, AWS) SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, 2022			
SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, 2022 Supported by TencentDB for SQL Server Migration between databases under the same root account Migration between databases under different root accounts			

Migrating to TencentDB for Redis

Redis > Redis

Data migration feature for AWS ElasticCache for Redis requires [Submit a Ticket](#) application.

Source Database	Target Database	Migration Type	Reference Documentation
Self-Built Database IDC self-built Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0 Self-built Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0 on CVM	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0	Full + incremental migration	Migration from Redis to TencentDB for Redis
Third-Party cloud vendors AWS ElasticCaChe for Redis 4.0,5.0,6.0			
TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0 Migration between databases under the same root account Database migration between different root accounts (requires Submit a Ticket application)			

Migrating to KeeWiDB

Redis > KeeWiDB

Source Database	Target Database	Migration Type	Reference Documentation
Self-Built Database IDC self-built Redis 4.0 Self-built Redis 4.0 on CVM	KeeWiDB 1.0	Full + incremental migration	Migration from Redis to KeeWiDB
All third-party cloud vendors Redis 4.0			
TencentDB for Redis 4.0 Migration between databases under the same root account Database migration between different root accounts (requires ticket submission)			

Tendis > KeeWiDB

Source Database	Target Database	Migration Type	Reference Documentation
TencentDB for Tedis 4.9 Migration between databases under the same root account Database migration between different root accounts (requires ticket submission)	KeeWiDB 1.0	Full + incremental migration	Migrating Tendis to KeeWiDB

Cross-Account TencentDB Instance Migration

Last updated : 2024-07-08 19:41:52

Overview

This document describes how to use the data migration feature of DTS to perform data migration between the source and target TencentDB instances under different root accounts.

Supported Scope

Data migration from TencentDB for MySQL/MongoDB/PostgreSQL.

Prerequisites

You have created the target database instance.

Notes

This operation involves multiple account information configuration items. The following lists the main configuration logic for easier understanding and configuration.

Data migration direction: Source database (database instance under another account) > target database (database instance under the current account).

The account executing the migration task can be the root account or a sub-account of the target database.

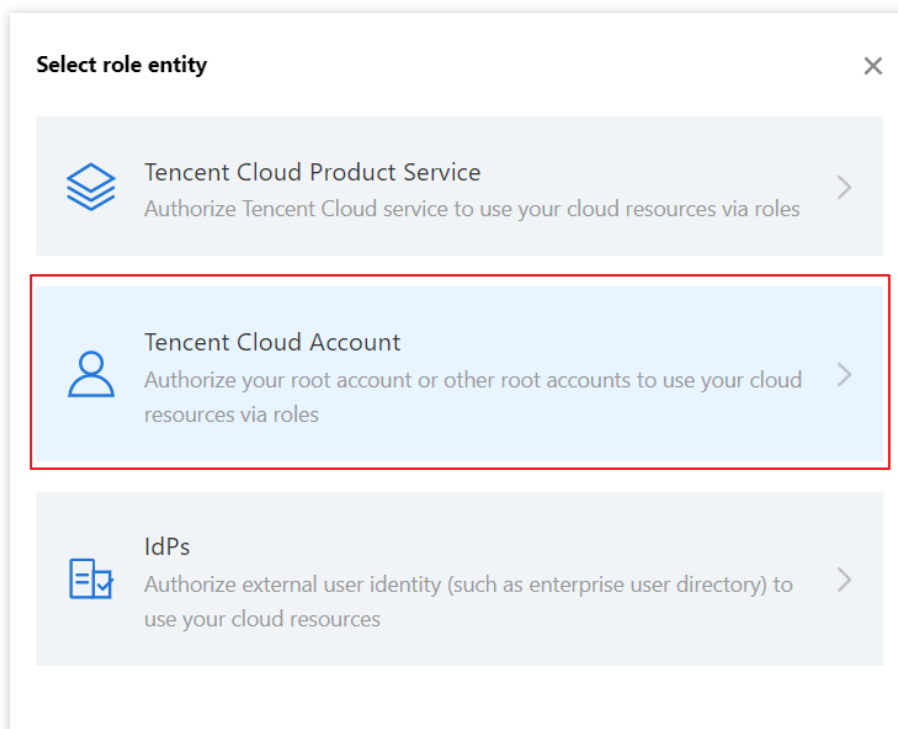
If you use a root account to execute the migration task, before executing the task, ask the root account of the source database to grant the root account of the target database access to the source database.

If you use a sub-account to execute the migration task, before executing the task, ask the root account of the source database to grant the root account of the target database access to the source database with a role, and then ask the root account of the target database to grant the sub-account access to the source database.

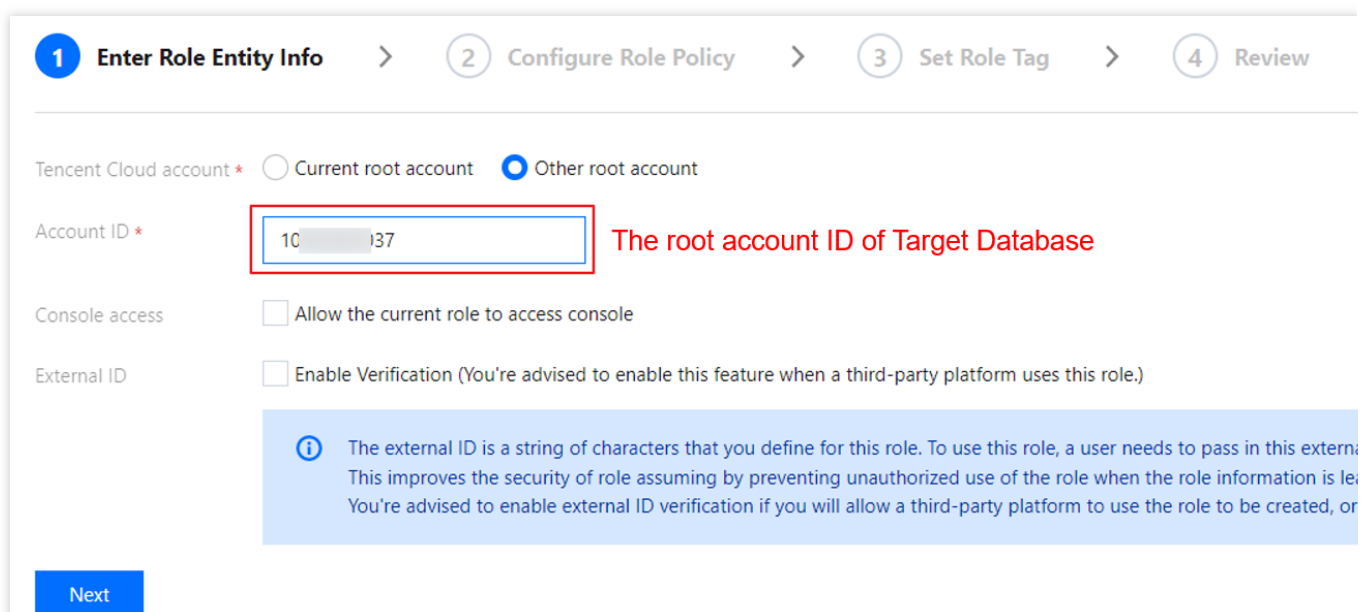
Authorizing an Account

To execute the migration task with a root account or a sub-account, follow steps 1–6 or steps 1–11 respectively.

1. Log in to the [CAM console](#) with the Tencent Cloud root account of the source database (if the sub-account has CAM and role permissions, you can also log in with the sub-account).
2. Click **Role** on the left sidebar to enter the **Role Management** page. Then, click **Create Role**.
3. On the **Select role entity** page, select **Tencent Cloud Account**.



4. On the **Enter Role Entity Info** page, configure the information and click **Next**.



Tencent Cloud account: Select **Other root account**.

Account ID: Enter the Tencent Cloud root account ID of the target database, which can be viewed in [Account Info](#).

Even if the target database instance is owned by a sub-account, you still need to enter the root account ID here.

External ID: You can enable it as needed.

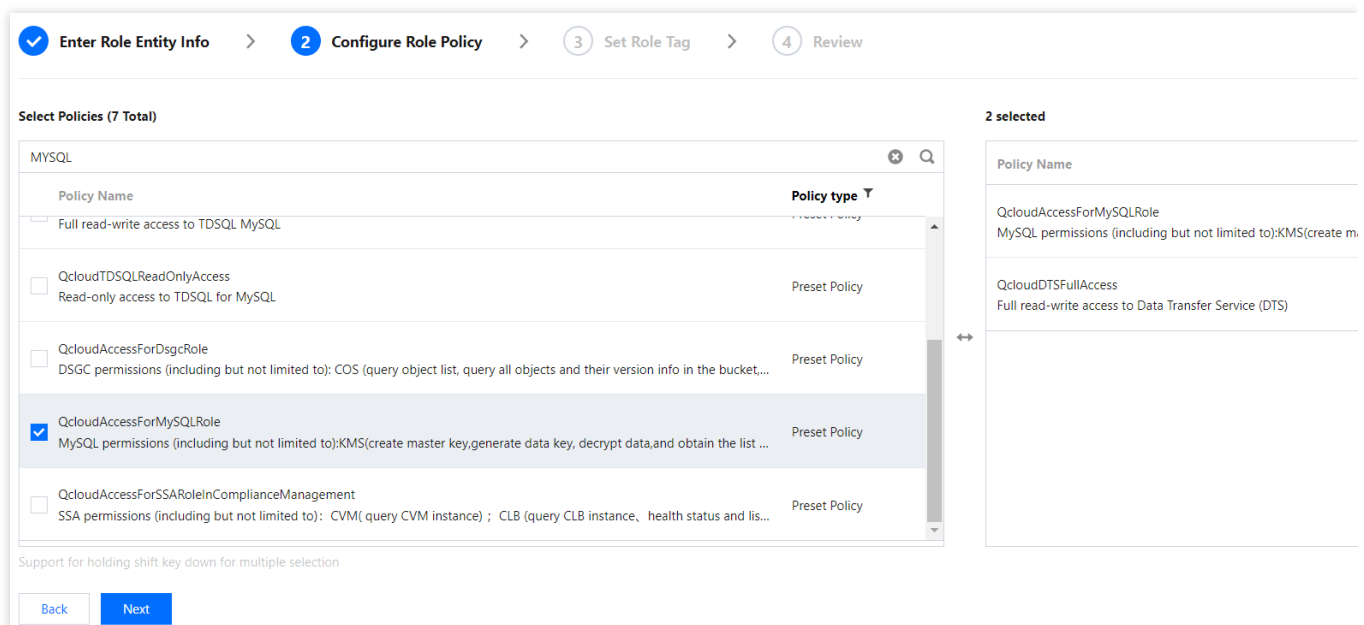
Note:

If an external ID is used, record and keep the ID on your own, as it cannot be queried in DTS.

5. On the **Configure Role Policy** page, select the corresponding policies of DTS and the source database and click **Next**.

For the DTS policy, select `QcloudDTSReadOnlyAccess` .

For the corresponding policy of the source database, select the policy of the Tencent Cloud service to which the source database belongs. For example, if the source database is TencentDB for MySQL or PostgreSQL, select `QcloudAccessForMySQLRole` or `QcloudPostgreSQLReadOnlyAccess` respectively. Select a policy based on the actual conditions.



6. Configure role tags. Then, on the **Review** page, set the role name and click **Complete**.

Note:

Record the configured role name, which needs to be entered when you create the migration task later.

✓ Enter Role Entity Info

>

✓ Configure Role Policy

>

✓ Set Role Tag

>

4

Role Name *

DTS-test

Description

Role Entity

Account-

Access Type

Programming access

Tag

No tag

Policy Name	Description
QcloudDTSFullAccess	Full read-write access to Data Transfer Service (DTS)
QcloudAccessForMySQLRole	MySQL permissions (including but not limited to):KMS(create master key,generate data key, decrypt

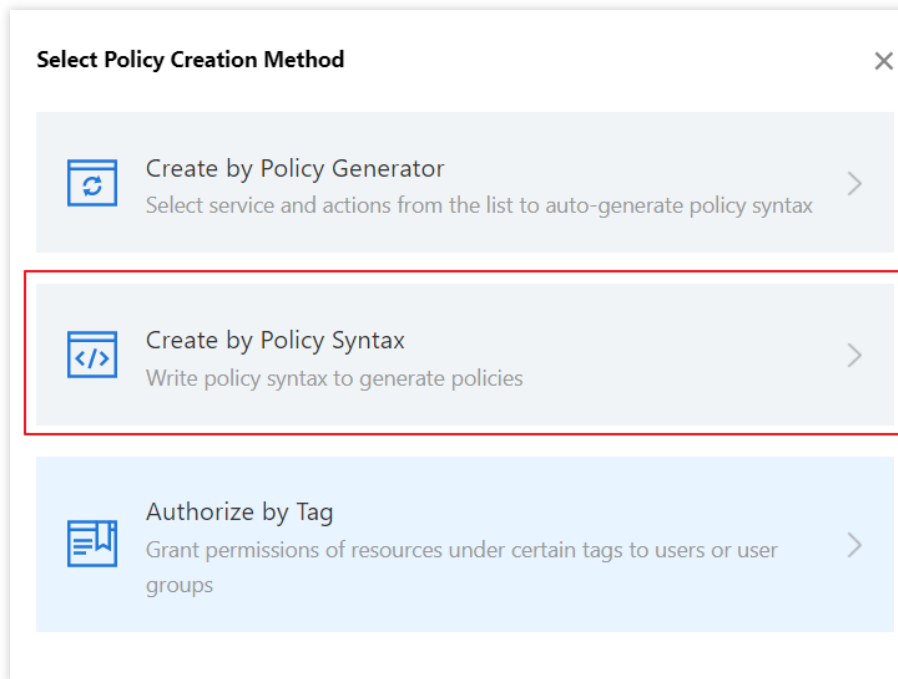
Back

Complete

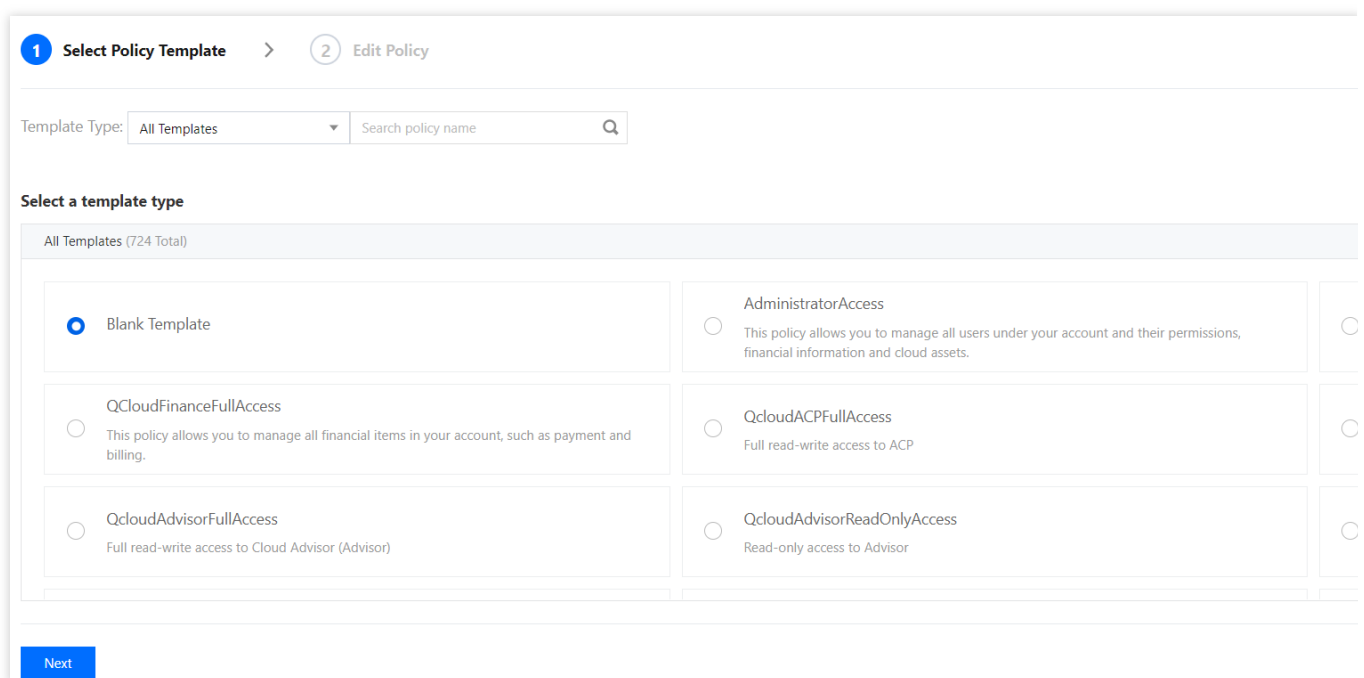
Note:

To execute a migration task with the root account, just follow the steps above; to execute a migration task with a sub-account, you also need to ask the root account to authorize the sub-account as follows:

7. (Optional) Log in to the [CAM console](#) with the Tencent Cloud root account of the target database and click **Policies** on the left sidebar. Then, click **Create Custom Policy** on the right and select **Create by Policy Syntax**.



8. (Optional) Select **Blank Template** and click **Next**.



9. (Optional) Create a policy and enter the policy name and description as needed. After copying the sample code to the **Policy Content**, replace the content in the red box with the actual information.

✓ Select Policy Template

>

2 Edit Policy

Policy Name *

DTS-policy

Description

Policy Content

[Use Legacy Version](#)

```
1  {
2    "version": "2.0",
3    "statement": [
4      {
5        "effect": "allow",
6        "action": ["name/sts:AssumeRole"],
7        "resource": ["qcs::cam::uin/1688:roleName/DTS-role"]
8      }
9    ]
10 }
```

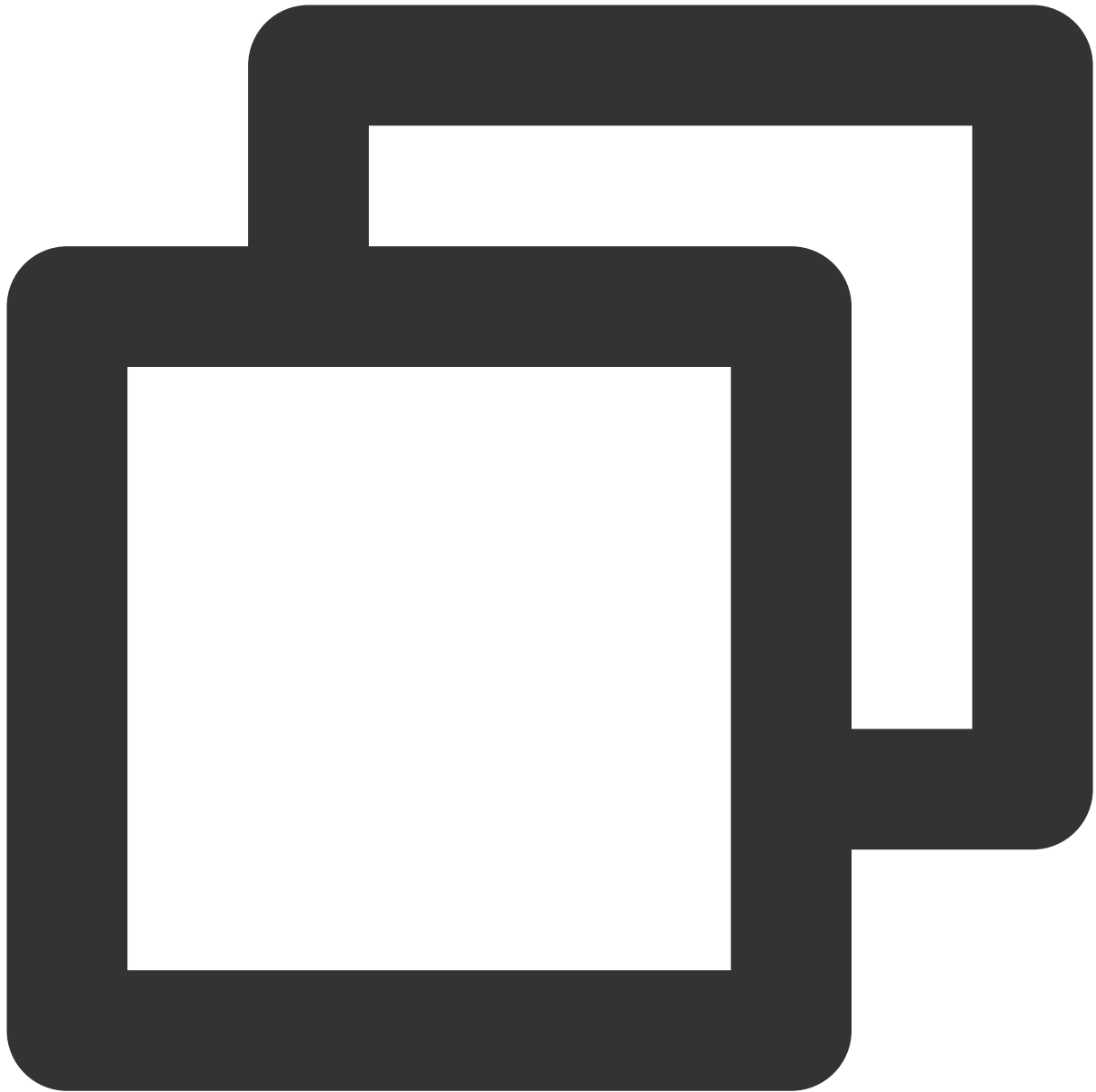
The root account ID of Source Database

Role name

Previous

Complete

10. Sample policy syntax:



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": ["name/sts:AssumeRole"],
      "resource": ["qcs::cam::uin/10*****8:roleName/DTS-role"]
    }
  ]
}
```

11. (Optional) Click **Complete**, return to the **Policy List** page, and click **Associate Users/Groups**.

Create Custom Policy Delete All Policies		
<input type="checkbox"/>	Policy Name	Service Type
<input type="checkbox"/>	DTS-policy	

12. (Optional) Select the sub-account of the target database instance (i.e., the sub-account executing the migration task) and click **OK**.

Associate User/User Group/Role

Select Users (16 Total)

Support multi-keyword search by user name/ID/SecretId/mobi

— User

Switch to User Group ...

☒ DTS-test

User

(1) selected

Name

DTS-test

Support for holding shift key down for multiple selection

OK

Cancel

Creating a Migration Task

1. Log in to the [DTS console](#) with the Tencent Cloud account of the target database instance.
2. Select **Data Migration > Create Migration Task** to purchase a migration task.
3. After making the purchase, return to the data migration task list and click **Configure** in the **Operation** column to enter the migration task configuration page.
4. On the **Set source and target databases** page, configure the source and target database information. The following takes data migration between two TencentDB for MySQL instances as an example.

1 Set source and target databases > 2 Set migration options and select migration objects > 3 Verify task

Task Configuration

Task Name *
stacy-test

Running Mode *

Immediate execution

Scheduled execution

Source Database Settings

Source Database Type *
MySQL

Service Provider *
Others

Access Type *

Public Network

Self-Build on CVM

Direct Connect

VPN Access

Database

CCN

Intranet

Ac

Please add the DTS IP addresses to the security group allowlist in advance so that the connectivity test can be quickly passed. For details, see

Cross-/Intra-Account *

Intra-account

Cross-account

Peer Account ID *
10 8

Peer Account Role Name *
DTS-role

Role External ID
Please enter the external ID of the role

Region
South China(Guangzhou)

Database Instance *
cdbro-evd3j8xt (truexu-dst1-ro1) ↻

Account *
root

Password *
Please enter password

Test Connectivity

Configure the key parameters for cross-account data migration as follows:

Access Type: Select **Database**, indicating that the source database is a TencentDB instance.

Cross-/Intra-Account: Select **Cross-account**.

Peer Account ID: Enter the root account ID of the source database.

Peer Account Role Name: The **role name** created in step 6 in [Authorizing an Account](#). For more information on roles, see [Role Overview](#) and [Cross-account Access Role](#).

External Role ID: This parameter is optional, and its value can be obtained from the previous section. For more information on roles, see [Role Overview](#) and [Cross-account Access Role](#).

Note:

After completing the above configuration, select the **Region** to get the instance list under the source database account. If an error occurs while getting the instances, the configuration may be incorrect, or no authorization has been performed. For more information, see [FAQs](#).

5. On the **Set migration options and select migration objects** page, set the data migration options, select migration objects, and click **Save and Go Next**.

6. On the **Verify task** page, complete the verification. After all check items are passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again.

7. Return to the data migration task list, and you can see that the task has entered the **Running** status.

FAQs

1. What should I do if the error "role not exist[InternalError.GetRoleError]" is reported when the instance list is pulled across accounts?

Check whether the **Peer Account ID** (the root account ID of the source database) and **Peer Account Role Name** (the **role name** created in step 6 in [Authorizing an Account](#)) have been correctly configured.

2. What should I do if the error `InternalError: InternalInnerCommonError` is reported when the database instance list is obtained?

The policy of the Tencent Cloud service to which the source database belongs hasn't been granted to the role. Grant it as instructed in step 5 in [Authorizing an Account](#).

3. What should I do if the error "you are not authorized to perform operation (sts:AssumeRole), resource (qcs::cam::uin/1xx5:roleName/xxxx) has no permission" is reported when the instance list is pulled across accounts?

Error cause: The account that you use to create the migration task is a sub-account without the `sts:AssumeRole` permission.

Solution:

Use the root account to create the migration task.

Ask the root account of the target database to authorize the sub-account as instructed in [Authorizing an Account](#) and set `resource` in the policy syntax to the field in blue in the error message.

Migration to MySQL Series

Migrating to MySQL

Migration from MySQL to TencentDB for MySQL

Last updated : 2024-07-08 19:41:52

This document describes how to use the data migration feature of DTS to migrate data from MySQL to TencentDB for MySQL.

The following deployment modes of the source database are supported:

Self-built MySQL.

TencentDB for MySQL.

MySQL in other clouds: Alibaba Cloud ApsaraDB RDS, Alibaba Cloud ApsaraDB for PolarDB, Amazon RDS, and Amazon Aurora.

Notes

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

Migration is implemented without locks by default, during which no global lock (the FTWRL lock) is added to the source database, and only tables without a primary key are locked.

When you [create a data consistency check task](#), DTS will use the account that executes the migration task to write the system database `__tencentdb__` in the source database to record the data comparison information during the migration task.

To ensure that subsequent data problems can be located, the `__tencentdb__` system database in the source database will not be deleted after the migration task ends.

The `__tencentdb__` system database uses a single-threaded connection wait mechanism and occupies a very small space, about 0.01%–0.1% of the storage space of the source database; for example, if the source database is 50 GB, `__tencentdb__` will be about 5–50 MB. Therefore, it has almost no impact on the performance of the source database and will not preempt resources.

Prerequisites

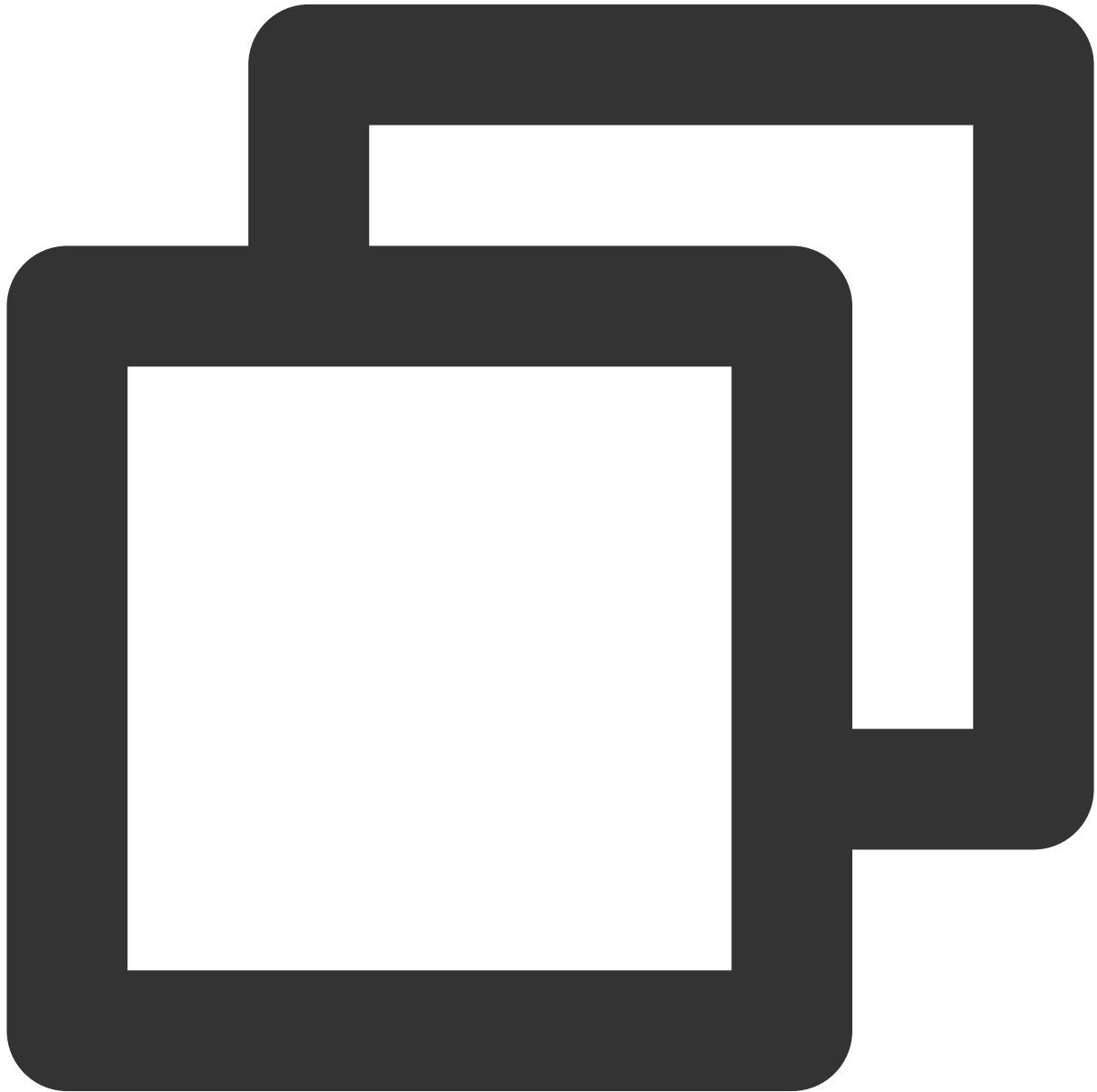
You have created a TencentDB for MySQL instance as instructed in [Creating MySQL Instance](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

The source database must have the following permissions:

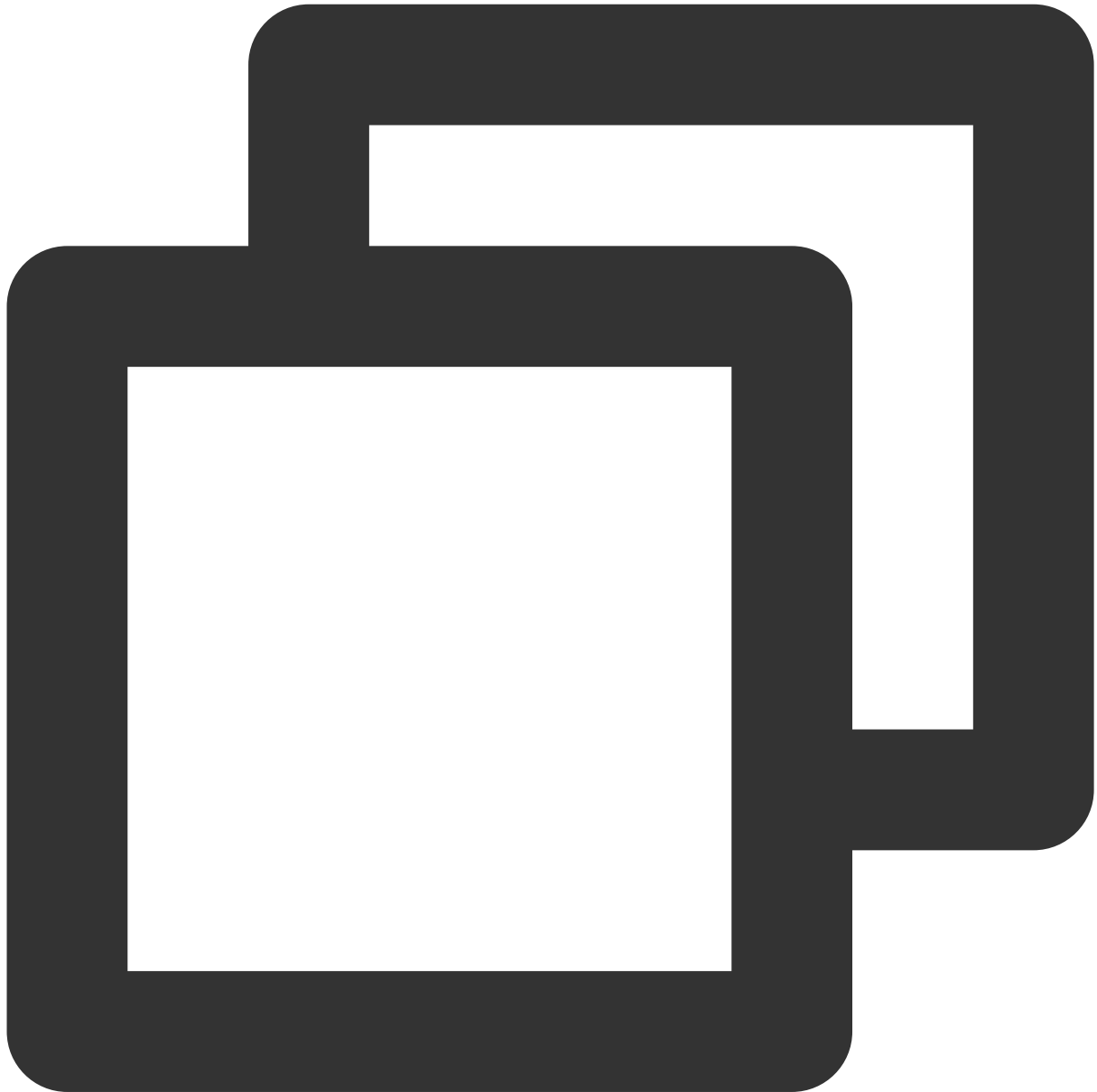
Migration of the entire instance:



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V
// If the source database is an Alibaba Cloud database, you don't need to grant the
```

```
// If you select to migrate triggers and events, you need grant both the `TRIGGER`  
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';  
GRANT SELECT ON *.* TO 'migration account';
```

Migration of specified objects:



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';  
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V  
// If the source database is an Alibaba Cloud database, you don't need to grant the  
// If you select to migrate triggers and events, you need grant both the `TRIGGER`  
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';  
GRANT SELECT ON `mysql`.* TO 'migration account'@'%';
```

```
GRANT SELECT ON database to be migrated.* TO 'migration account';
```

Permissions required of the target database: ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, and UPDATE.

Application restrictions

Basic tables, views, functions, triggers, procedures, and events can be migrated, while system tables such as

`information_schema` , `sys` , `performance_schema` , `__cdb_recycle_bin__` , `__recycle_bin__` , `__tencentdb__` , and `mysql` cannot.

When views, procedures, and functions are migrated, DTS will check whether `user1` corresponding to `DEFINER ([DEFINER = user1])` in the source database is the same as the migration account `user2` , and if not, DTS will change the `SQL SECURITY` attribute of `user1` in the target database from `DEFINER` to `INVOKER ([INVOKER = user1])` after the migration, and set the `DEFINER` in the target database to the migration account `user2 ([DEFINER = migration account user2])`. If the view definition in the source database is too complex, the task may fail.

If the source MySQL database is a non-GTID instance, DTS doesn't support HA switch for it. If it is switched, DTS incremental migration may be interrupted.

Only data with the following three database engines can be migrated: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during migration by default. If there is compressed TokuDB engine data in the source database, the target database must support the compression mode before migration can be performed; otherwise, the task will report an error.

Correlated data objects must be migrated together; otherwise, migration will fail. Common correlations include table referenced by views, view referenced by views, and tables correlated through primary/foreign keys.

During incremental migration, if the source database has distributed transactions or generates binlog statements in the `STATEMENT` format, the migration will fail.

In migration scenarios without locks, after the migration task enters the "source database export" step, DDL operations are not supported.

If the binlog of the source database has a GTID hole, it may compromise the performance of the migration task and cause the task to fail.

Scenarios that contain both DML and DDL statements in the same transaction are not supported and will trigger errors during task execution.

Geometry data types are not supported and will trigger errors during task execution.

The `ALTER VIEW` statement is not supported and will be skipped during migration.

Operation restrictions

During migration, do not perform the following operations; otherwise, the migration task will fail:

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers.

Do not run distributed transactions in the source database.

Do not write binlog data in the `STATEMENT` format into the source database.

Do not clear binlogs in the source database.

Do not run DDL operations of changing the database/table structure during database/table structure migration or full migration.

Do not delete the system table `__tencentdb__` during incremental migration.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.

If the source database is Alibaba Cloud ApsaraDB for RDS or PolarDB, it will add an additional primary key column to tables without a primary key or non-null unique key in the binlog. The added primary key column is invisible in the table structure and thus may not be recognized by DTS. Therefore, we recommend you not migrate tables without a primary key.

Supported SQL operations

Operation Type	Supported SQL Operation
DML	INSERT, UPDATE, DELETE, REPLACE
DDL	TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE VIEW: CREATE VIEW, DROP VIEW INDEX: CREATE INDEX, DROP INDEX DATABASE: CREATE DATABASE, ALTER DATABASE, DROP DATABASE <code>CREATE TABLE table name AS SELECT</code> is not supported.

Environment requirements

Notes:

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item](#)

[Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment Requirement
Requirements for the source database	<p>The source and target databases can be connected.</p> <p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p> <p>Requirements for instance parameters:</p> <p>The <code>server_id</code> parameter in the source database must be set manually and cannot be 0.</p> <p><code>row_format</code> for the source databases/tables cannot be set to <code>FIXED</code>.</p> <p>The values of the <code>lower_case_table_names</code> variable in both the source and target databases must be the same.</p> <p>The <code>connect_timeout</code> variable in the source database must be greater than or equal to 10.</p> <p>We recommend you enable <code>skip-name-resolve</code> to reduce the possibility of connection timeout.</p> <p>Requirements for binlog parameters:</p> <p>The <code>log_bin</code> variable in the source database must be set to <code>ON</code>.</p> <p>The <code>binlog_format</code> variable in the source database must be set to <code>ROW</code>.</p> <p>The <code>binlog_row_image</code> variable in the source database must be set to <code>FULL</code>.</p> <p>On MySQL 5.6 or later, if the <code>gtid_mode</code> variable is not <code>ON</code>, an alarm will be triggered.</p> <p>We recommend you enable <code>gtid_mode</code>.</p> <p>You cannot set filter conditions with <code>do_db</code> and <code>ignore_db</code>.</p> <p>If the source instance is a replica database, the <code>log_slave_updates</code> variable must be set to <code>ON</code>.</p> <p>We recommend you retain the binlog of the source database for at least three days; otherwise, the task cannot be resumed from the checkpoint and will fail if it is suspended or interrupted for longer than the time set for binlog retention.</p> <p>Foreign key dependency:</p> <p>Foreign key dependency can be set to only one of the following two types: <code>NO ACTION</code> and <code>RESTRICT</code>.</p> <p>During partial table migration, tables with foreign key dependency must be migrated.</p> <p>The migration precision of DTS for data in <code>FLOAT</code> type is 38 digits, and for data in <code>DOUBLE</code> type is 308 digits. You should check whether this meets your requirements.</p> <p>The environment variable <code>innodb_stats_on_metadata</code> must be set to <code>OFF</code>.</p>
Requirements for the target database	<p>The target database version must be later than or equal to the source database version.</p> <p>The size of the target database space must be at least 1.2 times that of the databases/tables to be migrated in the source database. Full data migration will execute INSERT operations concurrently, causing some tables in the target database to generate data fragments.</p> <p>Therefore, after full migration is completed, the size of the tables in the target database may be larger than that in the source database.</p> <p>The target database cannot have migration objects such as tables and views with the same name as those in the source database.</p>

The `max_allowed_packet` parameter of the target database must be set to 4 MB or above.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** to enter the **Create Migration Task** page.
2. On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Item	Description
Creation Mode	Select Create task .
Billing Mode	Only the pay-as-you-go billing mode is supported. After the purchase is completed, fees will be charged only in the incremental migration stage but not task configuration and full migration stages. However, due to Tencent Cloud's unified requirements for pay-as-you-go billing, the fees for one hour's usage will be frozen after the purchase. For detailed billing rules, see Billing Overview .
Source Instance Type	Select the source database type, which cannot be changed after purchase. Here, select MySQL .
Source Instance Region	This refers to the source region of the DTS service. If the source database is TencentDB, select its region here. If the source database is self-built or in another cloud, select the region closest to it, so that DTS can choose the optimal migration path to reduce the migration time.
Target Instance Type	Select the target database type, which cannot be changed after purchase. Here, select MySQL .
Target Instance Region	Select the target database region.
Specification	Select the specification of the migration link based on your business conditions. For the performance and billing details of different specifications, see Billing Overview .
Quantity	You can purchase up to 10 migration tasks at a time.

3. On the **Set source and target databases** page, configure the task, source database, and target database settings. After the source and target databases pass the connectivity test, click **Save**.

Notes:

If the connectivity test fails, troubleshoot as prompted or as instructed in [Database Connection Check](#) and try again.

1

Set source and target databases

>

2

Set migration options and select migration ol

Task Configuration

Task Name *

Running Mode *

Immediate execution

Scheduled execution

Source Database Settings

Source Database Type *

MySQL

Service Provider *

Others

AWS

Alibaba Cloud

Access Type *

Public Network

Self-Build on CVM

Direct Connect

VPN Access

Please add the DTS IP addresses to the security group allowlist in advance so that the connect

Cross-/Intra-Account *

Intra-account

Cross-account

Region

South China(Guangzhou)

Database Instance *

cdbro-

: (tr

1)

Account *

root

Password *

Please enter password

Test Connectivity

Target Database Settings

Target Database Type *

MySQL

Region

South China (Guangzhou)

Access Type *

Database

Database Instance *

cdb-1

3 (c

1)

▼

Account *

root

Password *

Please enter password

Test Connectivity

As there are many cross-scenarios of source database deployment modes and access types, the migration steps for different scenarios are similar. The following only provides configuration examples for typical scenarios. For other scenarios, configure by referring to the examples.

Example 1: Migrating a local self-built database to TencentDB through Direct Connect/VPN

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Automatic Retry	After this option is set, if the migration task is temporarily interrupted due to a network error, DTS will automatically retry and resume the task within the set time range. The supported time range is 5–720 minutes.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Service Provider	For a self-built database (such as a CVM-based one) or TencentDB database, select Others . For a third-party cloud database, select the corresponding service provider. In this scenario, select Others .
	Region	The region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Direct Connect or VPN Access , and you need to configure VPN-IDC interconnection as instructed in Direct Connect or VPN Access: Configuring VPN-IDC Interconnection . For the preparations for different access types, see Overview . Public Network: The source database can be accessed through a public IP.

		<p>Self-Build on CVM: The source database is deployed in a CVM instance.</p> <p>Direct Connect: The source database can be interconnected with VPCs through Direct Connect.</p> <p>VPN Access: The source database can be interconnected with VPCs through VPN Connections.</p> <p>Database: The source database is a TencentDB instance.</p> <p>CCN: The source database can be interconnected with VPCs through CCN.</p>
	VPC-based Direct Connect Gateway/VPN Gateway	Only VPC-based Direct Connect gateway is supported. You need to confirm the network type associated with the gateway. For VPN Gateway, select a VPN Gateway instance.
	VPC	Select a VPC and subnet associated with the VPC-based Direct Connect Gateway or VPN Gateway.
	Host Address	IP address or domain name for accessing the source MySQL database.
	Port	Port for accessing the source MySQL database.
	Account	Account of the source MySQL database, which must have the required permissions.
	Password	Password of the source MySQL database account.
	Connection Method	<p>Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link.</p> <p>Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.</p>
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.

	Password	Password of the target database.
	Connection Method	Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.

Example 2: Migrate between two TencentDB instances

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Automatic Retry	After this option is set, if the migration task is temporarily interrupted due to a network error, DTS will automatically retry and resume the task within the set time range. The supported time range is 5–720 minutes.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Service Provider	For a self-built database (such as a CVM-based one) or TencentDB database, select Others . For a third-party cloud database, select the corresponding service provider. In this scenario, select Others .
	Region	The source database region selected during data migration task purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database . For the preparations for different access types, see Overview . Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance. Direct Connect: The source database can be interconnected with VPCs through Direct Connect . VPN Access: The source database can be interconnected with VPCs through VPN Connections . Database: The source database is a TencentDB instance.

		CCN: The source database can be interconnected with VPCs through CCN .
	Cross-/Intra-Account	Intra-account: The source and target database instances belong to the same Tencent Cloud root account. Cross-account: The source and target database instances belong to different Tencent Cloud root accounts. The following uses migration under the same account as an example. For detailed directions on cross-account migration, see Cross-Account TencentDB Instance Migration .
	Database Instance	Source MySQL instance ID.
	Account	Account of the source MySQL database, which must have the required permissions.
	Password	Password of the source MySQL database account.
	Connection Method	Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database.
	Connection Method	Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.

Example 3: Migrating an Alibaba Cloud ApsaraDB for RDS instance to a TencentDB instance over the public network

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Automatic Retry	After this option is set, if the migration task is temporarily interrupted due to a network error, DTS will automatically retry and resume the task within the set time range. The supported time range is 5–720 minutes.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Service Provider	For a self-built database (such as a CVM-based one) or TencentDB database, select Others . For a third-party cloud database, select the corresponding service provider. In this scenario, select Alibaba Cloud .
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	For a third-party cloud database, you can select Public Network generally or select VPN Access , Direct Connect , or CCN based on your actual network conditions. In this scenario, select Public Network . For the preparations for different access types, see Overview Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance. Direct Connect: The source database can be interconnected with VPCs through Direct Connect . VPN Access: The source database can be interconnected with VPCs through VPN Connections . Database: The source database is a TencentDB instance. CCN: The source database can be interconnected with VPCs through CCN .
	Host Address	IP address or domain name for accessing the source MySQL database.
	Port	Port for accessing the source MySQL database.

	Account	Account of the source MySQL database, which must have the required permissions.
	Password	Password of the source MySQL database account.
	Connection Method	Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database.
	Connection Method	Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the source database first in this case.

4. On the **Set migration options and select migration objects** page, configure the migration type and objects and click **Save**.

Notes:

If you want to rename a table (for example, rename table A table B) during migration, you must select the entire database (or entire instance) where table A resides rather than only table A as the **migration object**; otherwise, the system will report an error.

1 Set source and target databases > 2 Set migration options and select migration objects > 3 Verify migration objects

Migration Type ⓘ *

Structural migrationFull migrationFull + Incremental migration

Data Consistency Check ⓘ *

Full checkNo check

If the source instance is read-only, data consistency check will be skipped after data migration. For details, see [Migration](#)

Data Check

Content check

Migration Object ⓘ *

Entire instanceSpecify object

Advanced Migration Object ⓘ

☐ Procedure☐ Function☐ Trigger☐ Event

Advanced objects can only be copied once, which means you cannot copy new objects once the task is started.

Migrate Account ⓘ *

☐

Source Database Object

Search database name, supporting fuzzy match

ⓘ 1 database in total, with 1 displayedMore

☐ db-dst

- ☒ Tables
- ☐ Views

RefreshSelect allClear

Selected Object ⓘ

Globally search for c

☐ db-dst

Unfold allFold allSe

ⓘ For migration notes, see [Migration FAQs](#)

PreviousSave

Configuration Item	Description
Migration Type	Select an option based on your scenario. Structural migration: Structured data such as databases and tables in the database will be migrated.

	<p>Full migration: The database/table structure and data of the entire database will be migrated. The migrated data will only be existing content of the source database when the task is initiated but not include the incremental data written to the source database after the task is initiated.</p> <p>Full + Incremental migration: The database/table structure and data of the entire database will be migrated. The migrated data will include the existing content of the source database when the task is initiated as well as the incremental data written to the source database after the task is initiated. If there are data writes to the source database during migration, and you want to smoothly migrate the data in a non-stop manner, select this option.</p>
Data Consistency Check	<p>If Full + Incremental migration is selected, you can perform data consistency check to carefully compare the data in the source and target databases after migration.</p> <p>If Full check is selected, when the migration task enters the "incremental sync" step, the time lag between them is 0 seconds, and the data gap between the target and the source databases is 0 MB, DTS will automatically trigger a data consistency check task.</p> <p>If Full check is not selected, you can manually trigger a check task when the task enters the "incremental sync" step. For more information, see Creating Data Consistency Check Task.</p>
Migration Object	<p>Entire instance: Migrate the entire database instance excluding the system databases such as <code>information_schema</code> , <code>mysql</code> , <code>performance_schema</code> , and <code>sys</code> .</p> <p>Specified objects: Migrate specified objects.</p>
Advanced Migration Object	<p>Procedures, functions, triggers, and events can be migrated.</p> <p>The migration of advanced objects is a one-time operation: only advanced objects already in the source database before the task start can be migrated, while those added to the source database after the task start will not be synced to the target database.</p> <p>Procedures and functions will be migrated during source database export. If there are no incremental migration tasks, triggers and events will be migrated when the task stops; otherwise, they will be migrated after you click Done, in which case the transition will take a slightly longer time.</p> <p>For more information, see Migrating Advanced Object.</p>
Migrate Account	<p>Select this feature if you want to migrate the account information of the source database.</p>
Selected Object	<p>Database/Table mapping (renaming) is supported. Hover over a database or table name, click the displayed Edit icon, and enter a new name in the pop-up window.</p> <p>We recommend you not rename tables when migrating advanced objects; otherwise, the migration may fail.</p>
Sync Online DDL Temp Table	<p>If you perform an online DDL operation on tables in the source database with the gh-ost or pt-osc tool, DTS can migrate the temp tables generated by online DDL changes to the target database.</p> <p>If you select gh-ost, DTS will migrate the temp tables (<code>_table name_ghc</code> , <code>_table name_gho</code> , and <code>_table name_del</code>) generated by the gh-ost tool to the target database.</p>

If you select **pt-osc**, DTS will migrate the temp tables (`_table_name_new` and `_table_name_old`) generated by the pt-osc tool to the target database.
For more information, see [Migrating Online DDL Temp Table](#).

5. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again. You can skip certain check items by blocking them after the verification fails. Then, you need to initiate the verification again to continue the task.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

If you select **Migrate Account**, the verification task will check the account information of the source database and migrate accounts meeting the requirements. DTS won't migrate ineligible accounts or will migrate them with fewer permissions. For more information, see [Account Migration](#).

Create Verification Task

Query Verification Result

- connect db check
- necessary check
- version check
- source instance privilege check
- simple instance param check
- target instance privilege check
- check if target instance has conflict content
- check if there's enough space in target instance
- source instance binlog param check
- foreign key constraint check
- view check
- warning param check

Previous Verify Again **Start Task** Start Later

6. Return to the data migration task list, and you can see that the task has entered the **Preparing** status. After 1–2 minutes, the data migration task will be started.

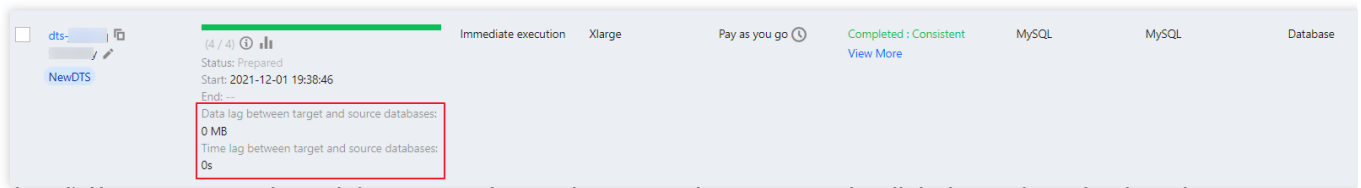
Select **Structural migration** or **Full migration**: Once completed, the task will be stopped automatically.

Select **Full + Incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data migration.

Manually complete incremental data migration and business switchover at appropriate time.

Check whether the migration task is in the incremental migration stage without any lag. If so, stop writing data to the source database for a few minutes.

Manually complete incremental migration when the data gap between the target and the source databases is 0 KB and the time lag between them is 0 seconds.



7. (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

8. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Migration from MariaDB/Percona to TencentDB for MySQL

Last updated : 2024-07-08 19:41:52

This document describes how to use the data migration feature of DTS to migrate data from MariaDB or Percona to TencentDB for MySQL.

The following deployment modes of the source database are supported:

Self-built MariaDB and TencentDB for MariaDB.

Self-built Percona.

Note:

TencentDB for MariaDB supports three kernels: MariaDB, Percona, and MySQL. You don't need to distinguish the kernel when using the service. If the source database is TencentDB for MariaDB, no matter which kernel is used, you need to select **MariaDB** as the source database type.

Notes

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

Migration is implemented without locks by default, during which no global lock (the FTWRL lock) is added to the source database, and only tables without a primary key are locked.

When you [create a data consistency check task](#), DTS will use the account that executes the migration task to write the system database `__tencentdb__` in the source database to record the data comparison information during the migration task.

To ensure that subsequent data problems can be located, the `__tencentdb__` system database in the source database will not be deleted after the migration task ends.

The `__tencentdb__` system database uses a single-threaded connection wait mechanism and occupies a very small space, about 0.01%–0.1% of the storage space of the source database; for example, if the source database is 50 GB, `__tencentdb__` will be about 5–50 MB. Therefore, it has almost no impact on the performance of the source database and will not preempt resources.

Prerequisites

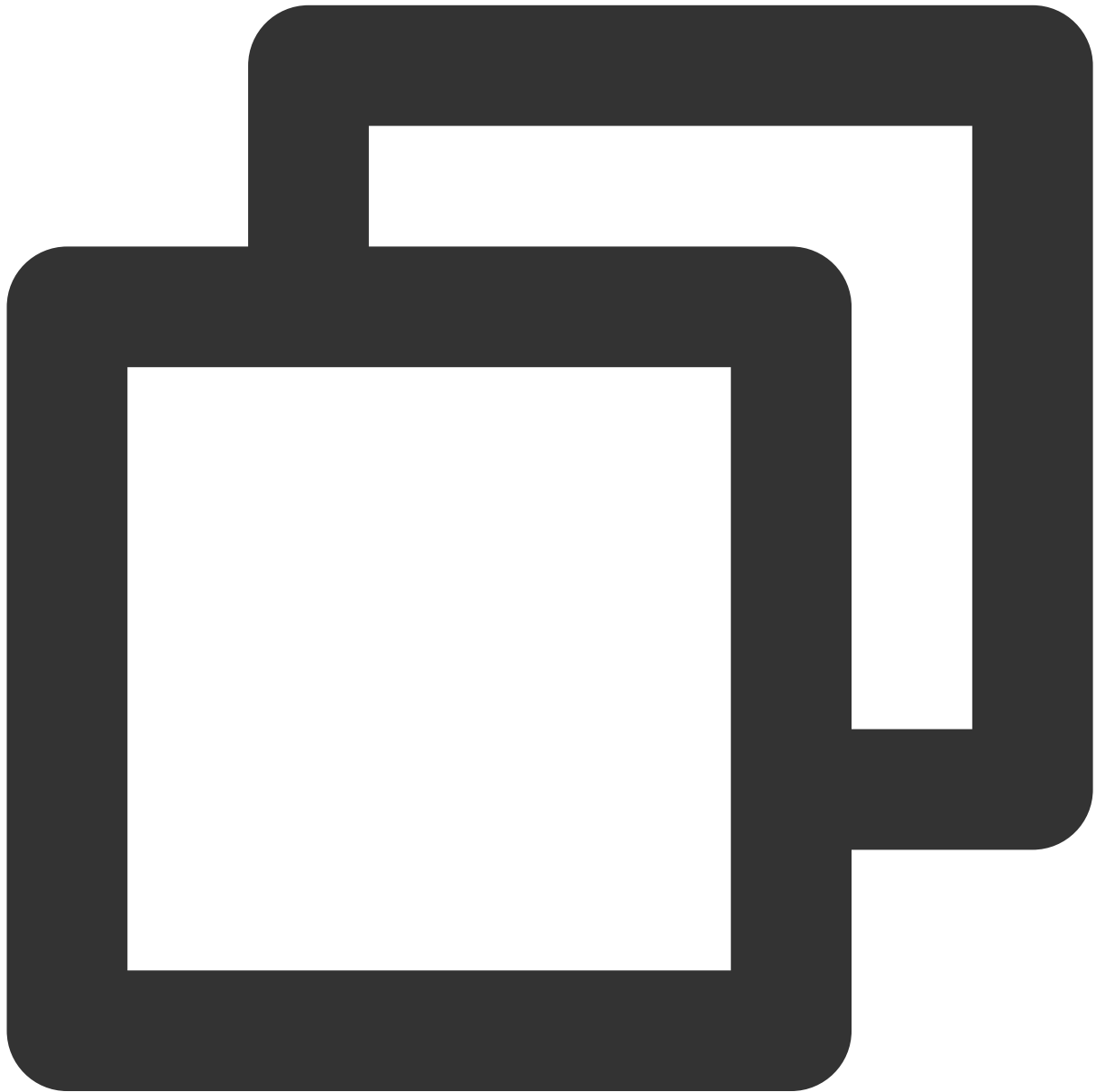
You have created a TencentDB for MySQL instance as instructed in [Creating MySQL Instance](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

The source database must have the following permissions:

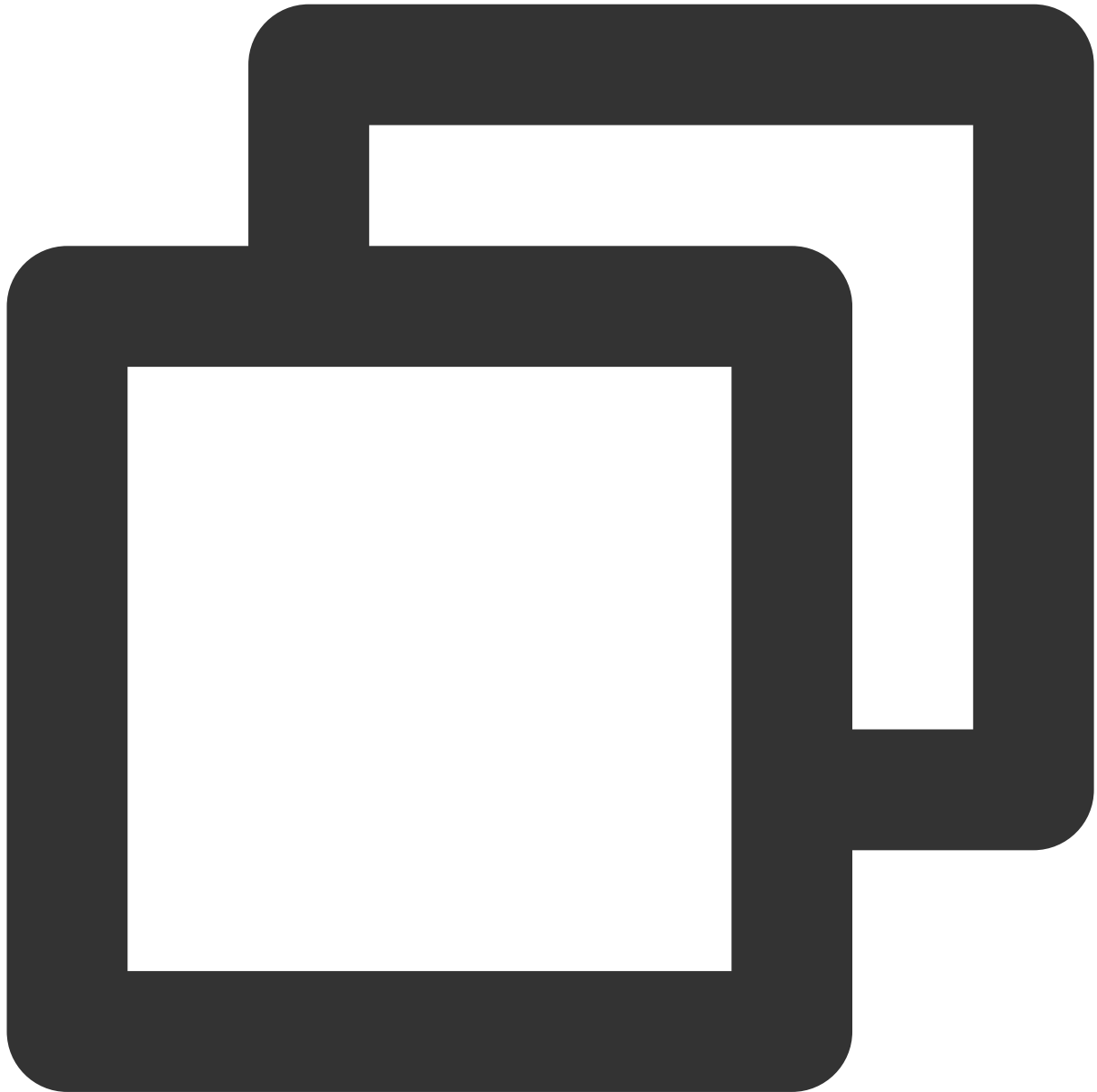
Migration of the entire instance:



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V
// If the source database is a TencentDB for MariaDB database, you need to submit a
// If you select to migrate triggers and events, you need grant both the `TRIGGER`
```

```
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';  
GRANT SELECT ON *.* TO 'migration account';
```

Migration of specified objects:



```
CREATE USER 'migration account'@ '%' IDENTIFIED BY 'migration password';  
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V  
// If the source database is a TencentDB for MariaDB database, you need to submit a  
// If you select to migrate triggers and events, you need grant both the `TRIGGER`  
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';  
GRANT SELECT ON `mysql`.* TO 'migration account'@'%';  
GRANT SELECT ON database to be migrated.* TO 'migration account';
```

If the source database is MariaDB 10.5 or 10.6, you also need the `SLAVE MONITOR` permission to run `show slave status`.

Permissions required of the target database: ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, and UPDATE (if the target database is TencentDB for MariaDB, you need to [submit a ticket](#) to authorize `RELOAD`).

Compatibility description for heterogeneous migration

During migration from MariaDB to MySQL, due to their slight differences in features, the following compatibility issues may occur:

1. Due to MariaDB's functional characteristics, some SQL statements are different from the returned result of `SHOW CREATE TABLE`, which may cause differences of the synced DDL statements in the target database.

Even if no default value is specified for the blob type in MariaDB, `SHOW CREATE TABLE` will still display the default value `DEFAULT NULL` after the table is created.

If the DDL statement of `datetime` type in the source database is `datetime NOT NULL ON UPDATE CURRENT_TIMESTAMP`, `SHOW CREATE TABLE` will display `datetime NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE current_timestamp()` after the table is created, and the DDL parsed by the target MySQL will be `datetime NOT NULL ON UPDATE CURRENT_TIMESTAMP`.

2. Some statements only supported by MariaDB (such as `CREATE OR REPLACE TABLE/PERIOD FOR/WITHOUT OVERLAPS`) may cause the migration task to report errors during full migration and will be ignored during incremental migration.

If the `PERIOD FOR/WITHOUT OVERLAPS` statement is executed before the migration task is started or during full migration (in the source database export and data import step), the migration task will fail; if it is executed during incremental sync, the target database will ignore it, and data cannot be synced to the target database.

As DDL operations that change the database or table structure cannot be performed during full migration, if the `CREATE OR REPLACE TABLE` statement is executed during full migration, the migration task will fail; if it is executed during incremental sync, the target database will ignore it, and data cannot be synced to the target database.

3. MariaDB allows default values for blob/text data, but MySQL does not. If there are SQL statements of these types, the migration task will report errors.

Application restrictions

Basic tables, views, functions, triggers, procedures, and events can be migrated, while system tables such as `information_schema` , `sys` , `performance_schema` , `__cdb_recycle_bin__` , `__recycle_bin__` , `__tencentdb__` , and `mysql` cannot.

When views, procedures, and functions are migrated, DTS will check whether `user1` corresponding to `DEFINER ([DEFINER = user1])` in the source database is the same as the migration account `user2` , and if not, DTS will change the `SQL SECURITY` attribute of `user1` in the target database from `DEFINER` to `INVOKER ([INVOKER = user1])` after the migration, and set the `DEFINER` in the target database to the migration account `user2 ([DEFINER = migration account user2])`. If the view definition in the source database is too complex, the task may fail.

If the source MySQL database is a non-GTID instance, DTS doesn't support HA switch for it. If it is switched, DTS incremental sync may be interrupted.

Only data with the following three database engines can be migrated: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during migration by default. If there is compressed TokuDB engine data in the source database, the target database must support the compression mode before migration can be performed; otherwise, the task will report an error.

Correlated data objects must be migrated together; otherwise, migration will fail. Common correlations include table referenced by views, view referenced by views, and tables correlated through primary/foreign keys.

During incremental migration, if the source database has distributed transactions or generates binlog statements in the `STATEMENT` format, the migration will fail.

If the source database is TencentDB for MariaDB, the following limits apply:

The DTS migration task requires that the values of the `lower_case_table_name` parameter (table name case sensitivity) of the source and target databases be the same. If the source database is TencentDB for MariaDB, as it allows modifying this parameter only during instance creation, you need to determine the case sensitivity rule when creating the source database and modify this parameter of the target database if the values are different during verification.

If the source database is TencentDB for MariaDB 10.4, the **Access Type** does not support the **Database** option, and you need to select **Public Network** or other types.

If the binlog of the source database has a GTID hole, it may compromise the performance of the migration task and cause the task to fail.

Scenarios that contain both DML and DDL statements in the same transaction are not supported and will trigger errors during task execution.

Geometry data types are not supported and will trigger errors during task execution.

The `ALTER VIEW` statement is not supported and will be skipped during migration.

Operation restrictions

During migration, do not perform the following operations; otherwise, the migration task will fail:

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers.

Do not run distributed transactions in the source database.

Do not write binlog data in the `STATEMENT` format into the source database.

Do not clear binlogs in the source database.

Do not run DDL operations of changing the database/table structure during database/table structure migration or full migration.

Do not delete the system table `__tencentdb__` during incremental migration.

If MariaDB's unique statements such as `CREATE OR REPLACE TABLE/PERIOD FOR/WITHOUT OVERLAPS` are included, the migration task may report errors during full migration and will ignore them during incremental migration.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.

Supported SQL operations

Operation Type	Supported SQL Operation
DML	INSERT, UPDATE, DELETE, REPLACE
DDL	TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE VIEW: CREATE VIEW, DROP VIEW INDEX: CREATE INDEX, DROP INDEX DATABASE: CREATE DATABASE, ALTER DATABASE, DROP DATABASE <code>CREATE TABLE table name AS SELECT</code> is not supported.

Environment requirements

Note:

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment Requirement
Requirements for the source	The source and target databases can be connected.

database	<p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p> <p>Requirements for instance parameters:</p> <p>The <code>server_id</code> parameter in the source database must be set manually and cannot be 0.</p> <p><code>row_format</code> for the source databases/tables cannot be set to <code>FIXED</code>.</p> <p>The values of the <code>lower_case_table_names</code> variable in both the source and target databases must be the same.</p> <p>The <code>connect_timeout</code> variable in the source database must be greater than or equal to 10.</p> <p>We recommend you enable <code>skip-name-resolve</code> to reduce the possibility of connection timeout.</p> <p>Requirements for binlog parameters:</p> <p>The <code>log_bin</code> variable in the source database must be set to <code>ON</code>.</p> <p>The <code>binlog_format</code> variable in the source database must be set to <code>ROW</code>.</p> <p>The <code>binlog_row_image</code> variable in the source database must be set to <code>FULL</code>.</p> <p>On MariaDB 10.2 or later and Percona 5.6 or later, if the <code>gtid_mode</code> variable is not <code>ON</code>, an alarm will be triggered. We recommend you enable <code>gtid_mode</code>.</p> <p>You cannot set filter conditions with <code>do_db</code> and <code>ignore_db</code>.</p> <p>If the source instance is a replica database, the <code>log_slave_updates</code> variable must be set to <code>ON</code>.</p> <p>We recommend you retain the binlog of the source database for at least three days; otherwise, the task cannot be resumed from the checkpoint and will fail if it is suspended or interrupted for longer than the time set for binlog retention.</p> <p>Foreign key dependency:</p> <p>Foreign key dependency can be set to only one of the following two types: <code>NO ACTION</code> and <code>RESTRICT</code>.</p> <p>During partial table migration, tables with foreign key dependency must be migrated.</p> <p>The migration precision of DTS for data in <code>FLOAT</code> type is 38 digits, and for data in <code>DOUBLE</code> type is 308 digits. You should check whether this meets your requirements.</p> <p>The environment variable <code>innodb_stats_on_metadata</code> must be set to <code>OFF</code>.</p>
Requirements for the target database	<p>The target database version must be later than or equal to the source database version.</p> <p>The size of the target database space must be at least 1.2 times that of the databases/tables to be migrated in the source database. Full data migration will execute INSERT operations concurrently, causing some tables in the target database to generate data fragments. Therefore, after full migration is completed, the size of the tables in the target database may be larger than that in the source database.</p> <p>The target database cannot have migration objects such as tables and views with the same name as those in the source database.</p> <p>The <code>max_allowed_packet</code> parameter of the target database must be set to 4 MB or above.</p>

Directions

Migration from MariaDB and Percona to TencentDB for MySQL follows the same steps of migration from MySQL to TencentDB for MySQL. For detailed configurations, see [Migration from MySQL to TencentDB for MySQL](#).

Migration from TDSQL for MySQL to TencentDB for MySQL

Last updated : 2024-07-08 19:41:52

The steps of data migration from TDSQL for MySQL to TencentDB for MySQL are basically the same as those of migration from TDSQL for MySQL to TDSQL for MySQL.

Migrating to MariaDB

Migration from MySQL to TencentDB for MariaDB (MySQL/MariaDB/Percona)

Last updated : 2024-07-08 19:41:52

This document describes how to use the data migration feature of DTS to migrate data from MySQL, MariaDB, or Percona to TencentDB for MariaDB.

The following deployment modes of the source database are supported:

Self-built MySQL and TencentDB for MySQL.

Self-built MariaDB and TencentDB for MariaDB.

Self-built Percona.

Note

TencentDB for MariaDB supports three kernels: MariaDB, MySQL, and Percona. You don't need to distinguish the kernel when using the service. If the source database is TencentDB for MariaDB, no matter which kernel is used, you still need to select **MariaDB** as the source database type.

This document describes how to migrate data from MariaDB to TencentDB for MariaDB. The requirements and steps of data migration from MySQL and Percona to TencentDB for MariaDB are basically the same.

Note

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

Migration is implemented without locks by default, during which no global lock (the FTWRL lock) is added to the source database, and only tables without a primary key are locked.

When you [create a data consistency check task](#), DTS will use the account that executes the migration task to write the system database `__tencentdb__` in the source database to record the data comparison information during the migration task.

To ensure that subsequent data problems can be located, the `__tencentdb__` system database in the source database will not be deleted after the migration task ends.

The `__tencentdb__` system database uses a single-threaded connection wait mechanism and occupies a very small space, about 0.01%–0.1% of the storage space of the source database; for example, if the source database is 50 GB, `__tencentdb__` will be about 5–50 MB. Therefore, it has almost no impact on the performance of the source database and will not preempt resources.

Prerequisites

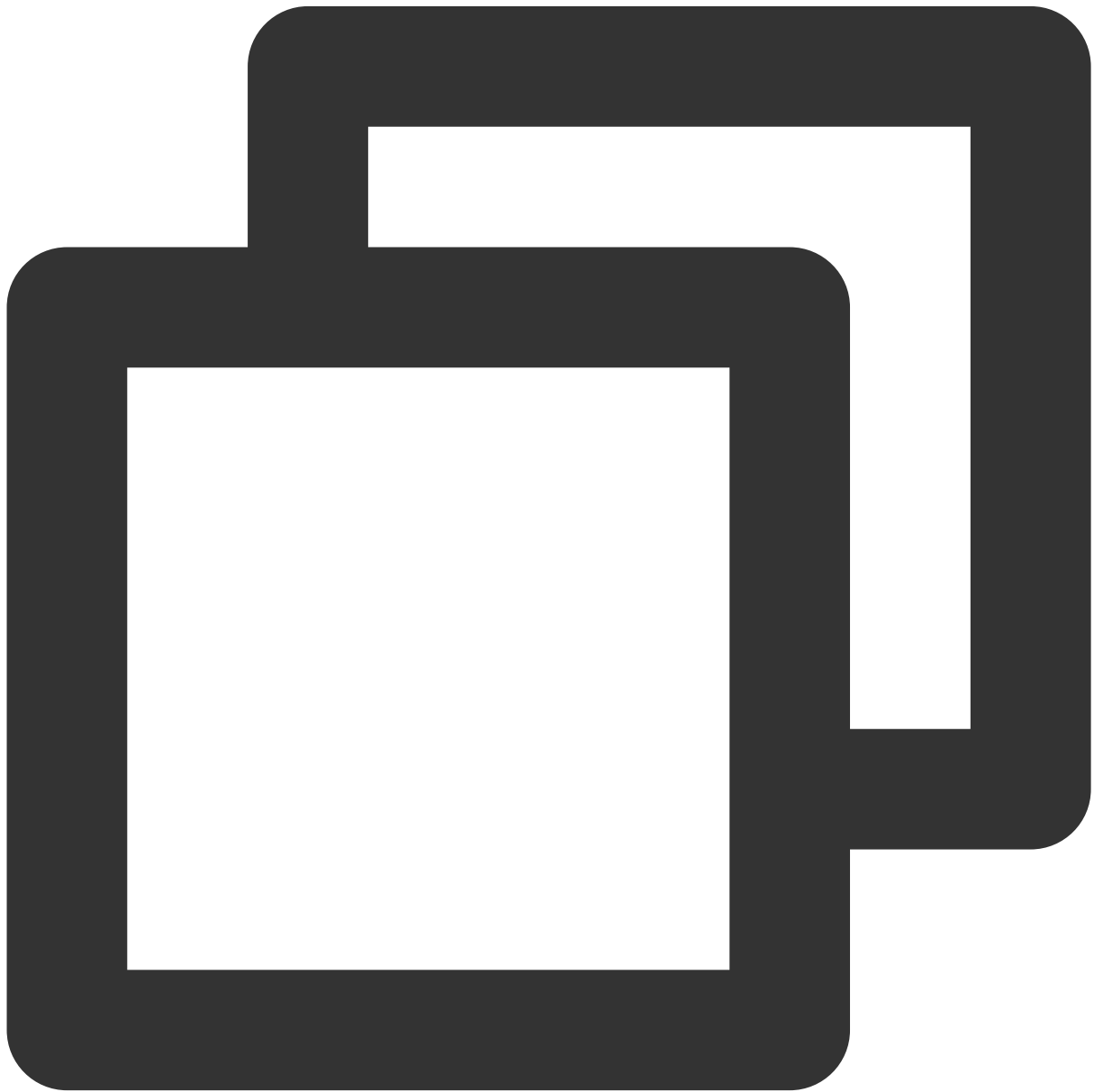
You have created a TencentDB for MariaDB instance as instructed in [Creating Instance](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

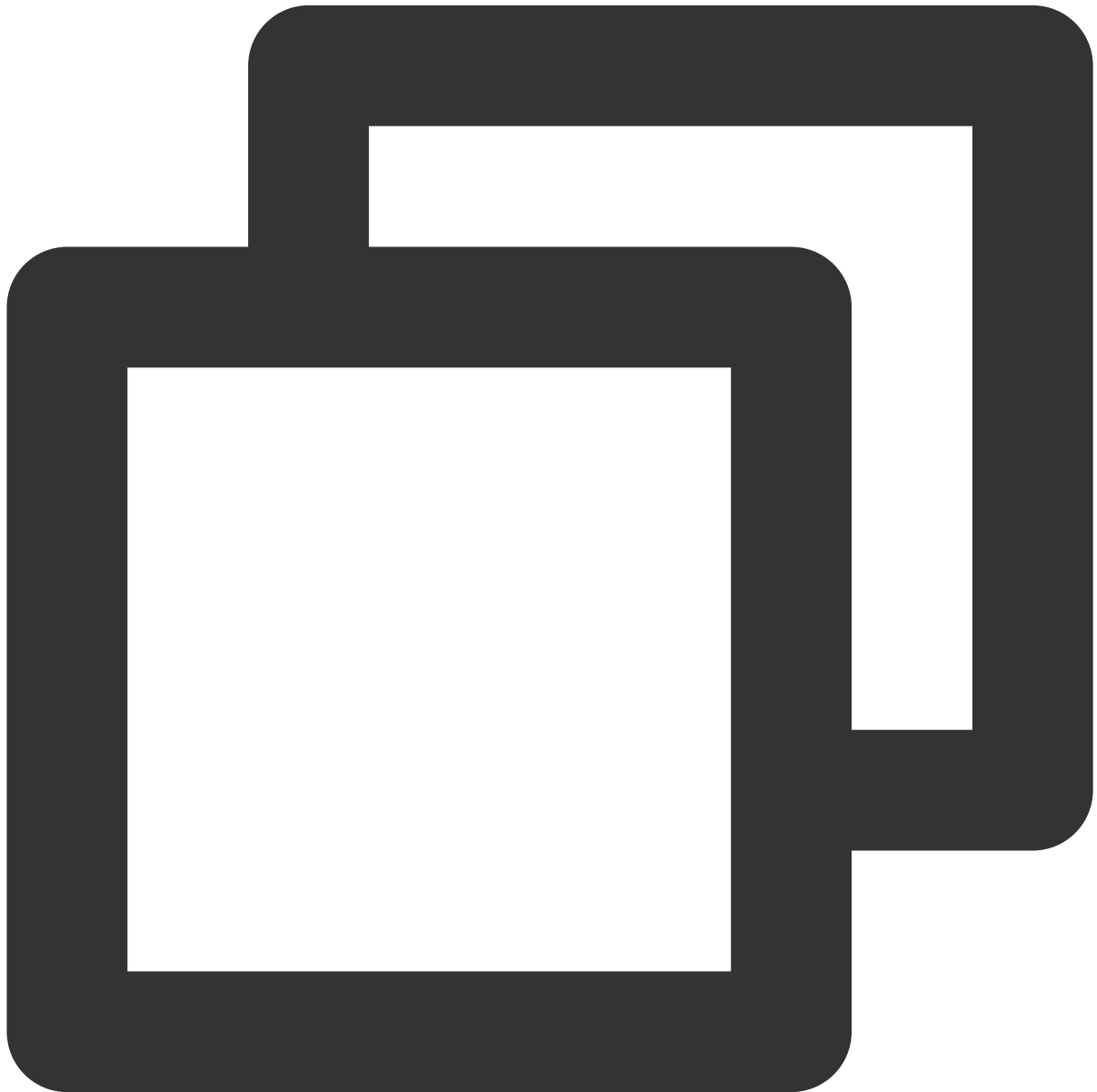
The source database must have the following permissions:

Migration of the entire instance:



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V
// If the source database is a TencentDB for MariaDB database, you need to submit a
// If the source database is an Alibaba Cloud database, you don't need to grant the
// If you select to migrate triggers and events, you need grant both the `TRIGGER`
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';
GRANT SELECT ON *.* TO 'migration account';
```

Migration of specified objects:



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';
```

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW V
// If the source database is a TencentDB for MariaDB database, you need to submit a
// If the source database is an Alibaba Cloud database, you don't need to grant the
// If you select to migrate triggers and events, you need grant both the `TRIGGER`
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';
GRANT SELECT ON `mysql`.* TO 'migration account'@'%';
GRANT SELECT ON database to be migrated.* TO 'migration account';
```

Permissions required of the target database: ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, and UPDATE.

Application Restrictions

Basic tables, views, functions, triggers, procedures, and events can be migrated, while system tables such as

`information_schema` , `sys` , `performance_schema` , `__cdb_recycle_bin__` , `__recycle_bin__` , `__tencentdb__` , and `mysql` cannot.

When views, procedures, and functions are migrated, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the migration account `user2` , and if not, DTS will change the `SQL SECURITY` attribute of `user1` in the target database from `DEFINER` to `INVOKER` (`[INVOKER = user1]`) after the migration, and set the `DEFINER` in the target database to the migration account `user2` (`[DEFINER = migration account user2]`). If the view definition in the source database is too complex, the task may fail.

If the source MySQL database is a non-GTID database, DTS doesn't support HA switch for it. If it is switched, DTS incremental sync may be interrupted.

Only data with the following three database engines can be migrated: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during migration by default. If there is compressed TokuDB engine data in the source database, the target database must support the compression mode before migration can be performed; otherwise, the task will report an error.

Correlated data objects must be migrated together; otherwise, migration will fail. Common correlations include table referenced by views, view referenced by views, and tables correlated through primary/foreign keys.

During incremental migration, if the source database has binlog statements in the `STATEMENT` format, the migration will fail.

If the source database is TencentDB for MariaDB, the following limits apply:

The DTS migration task requires that the values of the `lower_case_table_name` parameter (table name case sensitivity) of the source and target databases be the same. If the source database is TencentDB for MariaDB, as it allows modifying this parameter only during instance creation, you need to determine the case sensitivity rule when

creating the source database and modify this parameter of the target database if the values are different during verification.

If the source database is TencentDB for MariaDB 10.4, the **Access Type** does not support the **Database** option, and you need to select **Public Network** or other types.

In migration scenarios without locks, after the migration task enters the "source database export" step, DDL operations are not supported.

If the binlog of the source database has a GTID hole, it may compromise the performance of the migration task and cause the task to fail.

Scenarios that contain both DML and DDL statements in the same transaction are not supported. They will trigger errors during task execution.

Geometry data types are not supported. They will trigger errors during task execution.

The `ALTER VIEW` statement is not supported and will be skipped during migration.

Operation Restrictions

During migration, do not perform the following operations; otherwise, the migration task will fail:

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers.

Do not write binlog data in the `STATEMENT` format into the source database.

Do not clear binlogs in the source database.

Do not run DDL operations of changing the database/table structure during database/table structure migration or full migration.

Do not delete the system table `__tencentdb__` during incremental migration.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend that you select full + incremental data migration.

If the source database is Alibaba Cloud ApsaraDB for RDS or PolarDB, it will add an additional primary key column to tables without a primary key or non-null unique key in the binlog. The added primary key column is invisible in the table structure and thus may not be recognized by DTS. Therefore, we recommend you not migrate tables without a primary key.

Supported SQL Operations

Operation Type	Supported SQL Operations
DML	INSERT, UPDATE, DELETE, REPLACE

DDL	<p>TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE</p> <p>VIEW: CREATE VIEW, DROP VIEW</p> <p>INDEX: CREATE INDEX, DROP INDEX</p> <p>DATABASE: CREATE DATABASE, ALTER DATABASE, DROP DATABASE</p> <p><code>CREATE TABLE table name AS SELECT</code> is not supported.</p>
-----	--

Environment Requirements

Note

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment Requirements
Requirements for the source database	<p>The source and target databases can be connected.</p> <p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p> <p>Requirements for instance parameters:</p> <p>The <code>server_id</code> parameter in the source database must be set manually and cannot be 0.</p> <p><code>row_format</code> for the source databases/tables cannot be set to <code>FIXED</code>.</p> <p>The values of the <code>lower_case_table_names</code> variable in both the source and target databases must be the same.</p> <p>The <code>connect_timeout</code> variable in the source database must be greater than or equal to 10.</p> <p>We recommend that you enable <code>skip-name-resolve</code> to reduce the possibility of connection timeout.</p> <p>Requirements for binlog parameters:</p> <p>The <code>log_bin</code> variable in the source database must be set to <code>ON</code>.</p> <p>The <code>binlog_format</code> variable in the source database must be set to <code>ROW</code>.</p> <p>The <code>binlog_row_image</code> variable in the source database must be set to <code>FULL</code>.</p> <p>On MySQL 5.6 or later, if the <code>gtid_mode</code> variable is not <code>ON</code>, an alarm will be triggered. We recommend you enable <code>gtid_mode</code>.</p> <p>You cannot set filter conditions with <code>do_db</code> and <code>ignore_db</code>.</p> <p>If the source instance is a replica database, the <code>log_slave_updates</code> variable must be set to <code>ON</code>.</p> <p>We recommend that you retain the binlog of the source database for at least three days; otherwise, the task cannot be resumed from the checkpoint and will fail.</p> <p>Foreign key dependency:</p>

	<p>Foreign key dependency can be set to only one of the following two types: <code>NO ACTION</code> and <code>RESTRICT</code> .</p> <p>During partial table migration, tables with foreign key dependency must be migrated. The migration precision of DTS for data in <code>FLOAT</code> type is 38 digits, and for data in <code>DOUBLE</code> type is 308 digits. You should check whether this meets your requirements. The environment variable <code>innodb_stats_on_metadata</code> must be set to <code>OFF</code> .</p>
Requirements for the target database	<p>The target database version must be later than or equal to the source database version. The size of the target database space must be at least 1.2 times that of the databases/tables to be migrated in the source database. Full data migration will execute INSERT operations concurrently, causing some tables in the target database to generate data fragments. Therefore, after full migration is completed, the size of the tables in the target database may be larger than that in the source database.</p> <p>The target database cannot have migration objects such as tables and views with the same name as those in the source database.</p> <p>The <code>max_allowed_packet</code> parameter of the target database must be set to 4 MB or above.</p>

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** to enter the **Create Migration Task** page.
2. On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Items	Description
Creation mode	Select Create new task .
Billing Mode	Only the pay-as-you-go billing mode is supported. After the purchase is completed, fees will be charged only in the incremental migration stage but not task configuration and full migration stages. However, due to Tencent Cloud's unified requirements for pay-as-you-go billing, the fees for one hour's usage will be frozen after the purchase. For detailed billing rules, see Billing Overview .
Source Instance Type	<p>Select the source database type, which cannot be changed after purchase. Here, select MariaDB.</p> <p>If the source database is a self-built or third-party MariaDB, select MariaDB. If the source database is TencentDB for MariaDB (which has three kernel versions: MySQL, MariaDB, Percona), no matter which kernel is used, you still need to select MariaDB as the source database type.</p>
Source Instance	This refers to the source region of the DTS service. If the source database is TencentDB,

Region	select its region here. If the source database is self-built or in another cloud, select the region closest to it, so that DTS can choose the optimal migration path to reduce the migration time.
Target Instance Type	Select the target database type, which cannot be changed after purchase. Here, select MariaDB .
Target Instance Region	Select the target database region.
Specification	Select the specification of the migration link based on your business conditions. For the performance and billing details of different specifications, see Billing Overview .
Quantity	You can purchase up to 10 migration tasks at a time.

3. On the **Set source and target databases** page, configure the task, source database, and target database settings. After the source and target databases pass the connectivity test, click **Create**.

Note

If the connectivity test fails, troubleshoot as prompted or as instructed in [Database Connection Check](#) and try again.

Category	Configuration Items	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Automatic Retry	After this option is set, if the migration task is temporarily interrupted due to a network error, DTS will automatically retry and resume the task within the set time range. The supported time range is 5–720 minutes.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Service Provider	For a self-built database (such as a CVM-based one) or TencentDB database, select Others . For a third-party cloud database, select the corresponding service provider. In this scenario, select Others .
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Direct

		<p>Connect or VPN Access, and you need to configure VPN-IDC interconnection as instructed in Direct Connect or VPN Access: Configuring VPN-IDC Interconnection. For the preparations for different access types, see Overview.</p> <p>Public Network: The source database can be accessed through a public IP.</p> <p>Self-Build on CVM: The source database is deployed in a CVM instance.</p> <p>Direct Connect: The source database can be interconnected with VPCs through Direct Connect.</p> <p>VPN Access: The source database can be interconnected with VPCs through VPN Connections.</p> <p>Database: The source database is a TencentDB instance.</p> <p>CCN: The source database can be interconnected with VPCs through CCN.</p>
	VPC-based Direct Connect Gateway/VPN Gateway	Only VPC-based Direct Connect gateway is supported. You need to confirm the network type associated with the gateway. For VPN Gateway, select a VPN Gateway instance.
	Virtual Private Cloud (VPC)	Select a VPC and subnet associated with the VPC-based Direct Connect Gateway or VPN Gateway.
	Host Address	IP address or domain name for accessing the source database.
	Port	Port for accessing the source database.
	Account	Account of the source database, which must have the required permissions.
	Password	Password of the source database account.
Target Database Settings	Connection Method	<p>To try out the SSL connection feature, submit a ticket for application.</p> <p>Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link.</p> <p>Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the database first in this case.</p>
	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .

	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database account.
	Connection Method	To try out the SSL connection feature, submit a ticket for application. Secure Sockets Layer (SSL) refers to the secure connection between DTS and the database through SSL that encrypts the transfer link. Enabling SSL may slow down the connection response. In general, the Tencent Cloud private network is secure, so there is no need to enable this option. If public network or Direct Connect are used for data transfer, and your business requires a higher level of data security, enable it; however, you should enable SSL encryption in the database first in this case.

4. On the **Set migration options and select migration objects** page, configure the migration type and objects and click **Save**.

Note

If you want to rename a table (for example, rename table A table B) during migration, you must select the entire database (or entire instance) where table A resides rather than only table A as the **migration object**; otherwise, after the renaming operation, the data in table B will not be synced to the target database.

Unfold all Fold all SeePage 67 of 341

Migration Object	<p>Entire instance: Migrate the entire database instance excluding the system databases such as <code>information_schema</code> , <code>mysql</code> , <code>performance_schema</code> , and <code>sys</code> .</p> <p>Specified objects: Migrate specified objects.</p>
Advanced Migration Object	<p>Procedures, functions, triggers, and events can be migrated.</p> <p>The migration of advanced objects is a one-time operation: only advanced objects already in the source database before the task start can be migrated, while those added to the source database after the task start will not be synced to the target database.</p> <p>Procedures and functions will be migrated during source database export. If there are no incremental migration tasks, triggers and events will be migrated when the task stops; otherwise, they will be migrated after you click Complete, in which case the transition will take a slightly longer time.</p> <p>For more information, see Migrating Advanced Object.</p>
Selected Object	<p>Database/table mapping (renaming) is supported. Hover over a database or table name, click the displayed Edit icon, and enter a new name in the pop-up window.</p> <p>We recommend that you not rename tables when migrating advanced objects; otherwise, the migration may fail.</p>
Sync Online DDL Temp Table	<p>If you perform an online DDL operation on tables in the source database with the gh-ost or pt-osc tool, DTS can migrate the temp tables generated by online DDL changes to the target database.</p> <p>If you select gh-ost, DTS will migrate the temp tables (<code>_table name_ghc</code> , <code>_table name_gho</code> , and <code>_table name_del</code>) generated by the gh-ost tool to the target database.</p> <p>If you select pt-osc, DTS will migrate the temp tables (<code>_table name_new</code> and <code>_table name_old</code>) generated by the pt-osc tool to the target database.</p> <p>For more information, see Migrating Online DDL Temp Table.</p>

5. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

Create Verification Task

Query Verification Result

- connect db check
- necessary check
- version check
- source instance privilege check
- simple instance param check
- target instance privilege check
- check if target instance has conflict content
- check if there's enough space in target instance
- source instance binlog param check
- foreign key constraint check
- view check
- warning param check

[Previous](#) [Verify Again](#) [Start Task](#) [Start Later](#)

6. Return to the data migration task list, and you can see that the task has entered the **Preparing** status. After 1–2 minutes, the data migration task will be started.

Select **Structural migration** or **Full migration**: Once completed, the task will be stopped automatically.

Select **Full + Incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data sync.

Manually complete incremental data sync and business switchover at appropriate time.

Check whether the migration task is in the incremental sync stage without any lag. If so, stop writing data to the source database for a few minutes.

Manually complete incremental sync when the data gap between the target and the source databases is 0KB and the time lag between them is 0 seconds.

7. (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

8. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Migration from TDSQL for MySQL to TencentDB for MariaDB

Last updated : 2024-07-08 19:41:52

The steps of data migration from TDSQL for MySQL to TencentDB for MariaDB are basically the same as those of migration from TDSQL for MySQL to TDSQL for MySQL.

Migrating to TDSQL-C for MySQL

Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL

Last updated : 2024-07-08 19:41:52

The steps of data migration from MySQL, MariaDB, or Percona to TDSQL-C for MySQL are the same as those described in [Migration from MySQL to TencentDB for MySQL](#).

Migration from TDSQL for MySQL to TDSQL-C for MySQL

Last updated : 2024-07-08 19:41:52

The steps of data migration from TDSQL for MySQL to TDSQL-C for MySQL are basically the same as those described in [Migration from TDSQL for MySQL to TDSQL for MySQL](#).

Migrating to TDSQL MySQL

Migration from TDSQL for MySQL to TDSQL for MySQL

Last updated : 2024-07-08 19:41:52

This document describes how to use the data migration feature of DTS to migrate data from TDSQL for MySQL to TDSQL for MySQL.

The requirements for data migration in the following scenarios are the same as those for data migration from TDSQL for MySQL to TDSQL for MySQL. You can refer to this document for directions.

Data migration from TDSQL for MySQL to TencentDB for MariaDB

Data migration from TDSQL for MySQL to TencentDB for MySQL

Note

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

Migration is implemented without locks by default, during which no global lock (the FTWRL lock) is added to the source database, and only tables without a primary key are locked.

When you [create a data consistency check task](#), DTS will use the account that executes the migration task to write the system database `__tencentdb__` in the source database to record the data comparison information during the migration task.

To ensure that subsequent data problems can be located, the `__tencentdb__` system database in the source database will not be deleted after the migration task ends.

The `__tencentdb__` system database uses a single-threaded connection wait mechanism and occupies a very small space, about 0.01%–0.1% of the storage space of the source database; for example, if the source database is 50 GB, `__tencentdb__` will be about 5–50 MB. Therefore, it has almost no impact on the performance of the source database and will not preempt resources.

Prerequisites

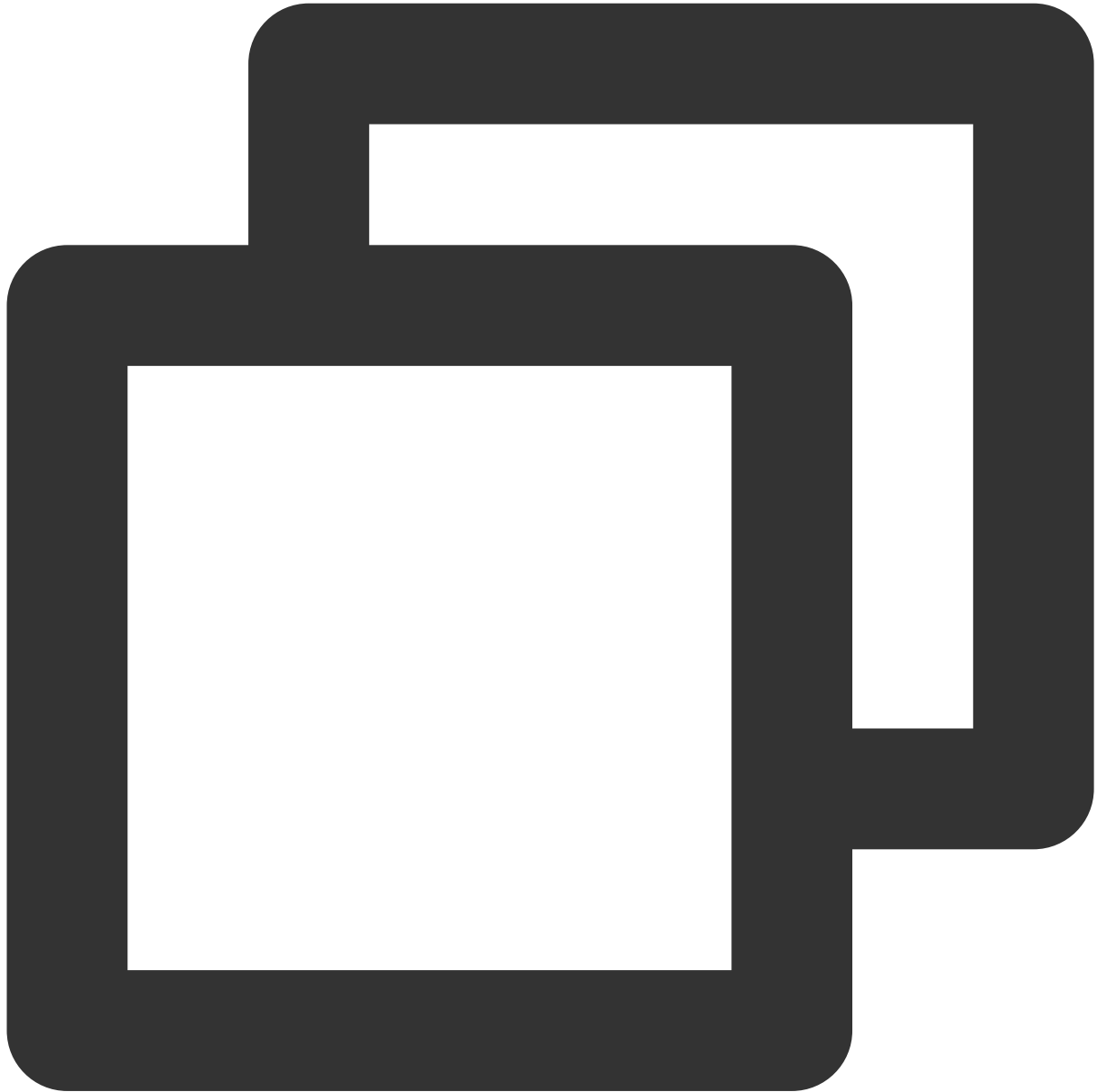
You have created a TDSQL for MySQL instance as instructed in [Creating Instances](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

You need to create a `__tencentdb__` database in advance in the source TDSQL for MySQL database.

You need to have the permissions of the source database.



```
CREATE USER 'migration account'@'%' IDENTIFIED BY 'migration password';
GRANT SELECT,RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES
// If the source database is a TDSQL for MySQL database, you need to submit a ticket
GRANT INSERT, UPDATE, DELETE, DROP, SELECT, INDEX, ALTER, CREATE ON `__tencentdb__`
```

Permissions required of the target database: ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT,

LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, and UPDATE.

Application Restrictions

Only basic tables can be migrated, while objects such as views, functions, triggers, and stored procedures cannot.

System tables and user information such as `information_schema` , `sysdb` , `test` , `sys` , `performance_schema` , `__tencentdb__` , and `mysql` cannot be migrated.

Only data with the InnoDB database engine can be migrated. Tables with other engines will be skipped during migration by default.

Correlated data objects must be migrated together; otherwise, migration will fail.

During incremental migration, if the source database has binlog statements in the `STATEMENT` format, the migration will fail.

Two-level partitioned tables as described in [Subpartitioning](#) cannot be migrated.

During migration from TDSQL for MySQL to MySQL/MariaDB, if a two-level partitioned table is encountered, the task will report an error.

During migration from TDSQL for MySQL to TDSQL for MySQL, if the migrated databases/tables contain two-level partitioned tables, such partitioned table will be skipped. If the entire database or instance is migrated, and a two-level partitioned table is encountered, the task will report an error and pause.

Scenarios that contain both DML and DDL statements in the same transaction are not supported. They will trigger errors during task execution.

Geometry data types are not supported. They will trigger errors during task execution.

Operation Restrictions

During migration, do not perform the following operations; otherwise, the migration task will fail:

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers.

Do not write binlog data in the `STATEMENT` format into the source database.

Do not clear binlogs in the source database.

Do not delete the system table `__tencentdb__` during incremental migration.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.

During incremental migration, you cannot add new shards or adjust the shard specification in the source database; otherwise, the migration task will not sync the data in the new shards or will report an error and pause. If you need to

maintain incremental sync for a long time and add or adjust shards in the source database, see [Sync from TDSQL for MySQL to TDSQL for MySQL](#).

Supported SQL Operations

Operation Type	Synchronizable SQL Operation
DML	INSERT, UPDATE, DELETE, REPLACE
DDL	TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE VIEW: CREATE VIEW, DROP VIEW INDEX: CREATE INDEX, DROP INDEX DATABASE: CREATE DATABASE, ALTER DATABASE, DROP DATABASE CREATE TABLE table name AS SELECT is not supported.

Environment Requirements

Note:

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment Requirements
Requirements for the source database	<p>The source and target databases can be connected.</p> <p>Requirements for instance parameters:</p> <p><code>table_row_format</code> cannot be set to <code>FIXED</code>.</p> <p>The values of the <code>lower_case_table_names</code> variable in both the source and target databases must be the same.</p> <p>The <code>max_allowed_packet</code> parameter in the target database must be at least 4 MB.</p> <p>The <code>connect_timeout</code> variable in the source database must be greater than or equal to 10.</p> <p>Requirements for binlog parameters:</p> <p>The <code>binlog_format</code> variable in the source database must be set to <code>ROW</code>.</p> <p>The <code>log_bin</code> variable in the source database must be set to <code>ON</code>.</p> <p>The <code>binlog_row_image</code> variable in the source database must be set to <code>FULL</code>.</p> <p>On v5.6 or later, if the <code>gtid_mode</code> variable in the source database is not <code>ON</code>, a warning will be triggered. We recommend you enable <code>gtid_mode</code>.</p> <p>It is not allowed to set <code>do_db</code> and <code>ignore_db</code>.</p>

	<p>If the source instance is a replica database, the <code>log_slave_updates</code> variable must be set to <code>ON</code> .</p> <p>We recommend that you retain the binlog of the source database for at least three days; otherwise, the task cannot be resumed from the checkpoint and will fail.</p> <p>Foreign key dependency:</p> <p>Foreign key dependency can be set to only one of the following two types: <code>NO ACTION</code> and <code>RESTRICT</code> .</p> <p>During partial table migration, tables with foreign key dependency must be migrated.</p> <p>The environment variable <code>innodb_stats_on_metadata</code> must be set to <code>OFF</code> .</p>
Requirements for the target database	<p>If the target database is a distributed database, we recommend you manually create a partitioned table and plan the shardkey in advance; otherwise, DTS will create a table in the target database based on the table style of the source database. If the source database is a standalone instance, the target database will be created as a single table.</p> <p>The target database version must be later than or equal to the source database version.</p> <p>The target database space must be at least 1.2 times the space of databases/tables to be migrated in the source database.</p> <p>The target database cannot have tables that conflict with the source database.</p>

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** to enter the **Create Migration Task** page.
2. On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Items	Description
Source Instance Type	<p>Select the source database type, which cannot be changed after purchase. Here, select MySQL.</p> <p>TDSQL for MySQL has three kernel versions (MySQL/MariaDB/Percona). No matter which kernel is used, you still need to select TDSQL for MySQL as the source database type.</p>
Source Instance Region	Select the source database region.
Target Instance Type	<p>Select the target database type, which cannot be changed after purchase. Here, select TDSQL for MySQL.</p>
Target Instance Region	Select the target database region.

Specification	Select the specification of the migration link based on your business conditions. For the performance and billing details of different specifications, see Billing Overview .
---------------	---

3. On the **Set source and target databases** page, configure the task, source database, and target database settings. After the source and target databases pass the connectivity test, click **Create**.

Note:

If the connectivity test fails, troubleshoot and fix the problem as prompted and as instructed in [Troubleshooting Guide](#) and try again.

Category	Configuration Items	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Tag	Tags are used to manage resources by category in different dimensions. If the existing tags do not meet your requirements, go to the console to create more.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	Select Database .
	Database Instance	Select the instance ID of the source database.
	Account	Account of the source TDSQL for MySQL database, which must have the required permissions.
	Password	Password of the source TDSQL for MySQL database account.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select Database .

	Database Instance	Select the ID of the target TDSQL for MySQL instance.
	Account	Account of the target TDSQL for MySQL database, which must have the required permissions.
	Password	Password of the target TDSQL for MySQL database account.

4. On the **Set migration options and select migration objects** page, configure the migration type and objects and click **Save**.

Note:

If you want to use a tool such as gh-ost and pt-osc to perform online DDL operations on a table during migration, you must select the entire database (or entire instance) where the table resides rather than only the table as the **migration object**; otherwise, the temporary table data generated by online DDL changes cannot be migrated to the target database.

If you want to rename a table (for example, rename table A table B) during migration, you must select the entire database (or entire instance) where table A resides rather than only table A as the **migration object**; otherwise, after the renaming operation, the data in table B will not be synced to the target database.

✓ Set source and target databases

2 Set migration options and select migration objects

3 Verify migration

Migration Type ⓘ *

Structural migration

Full migration

Full + Incremental migration

Migration Object ⓘ *

Entire instance

Specify object

Source Database Object

Search database name, supporting fuzzy match

1 database in total, with 1 displayed

More

db-dst

Selected Object ⓘ

Globally search for

db-dst

Refresh

Select all

Clear

Unfold all

Fold all

Save

Configuration Items	Description
Migration Type	<p>Select an option as needed.</p> <p>Structural migration: Structured data such as databases and tables in the database will be migrated.</p> <p>Full migration: The entire database will be migrated. The migrated data will only be existing content of the source database when the task is initiated but not include the incremental data written to the source database after the task is initiated.</p> <p>Full + Incremental migration: The migrated data will include the existing content of the source database when the task is initiated as well as the incremental data written to the source database after the task is initiated. If there are data writes to the source database during migration, and you want to smoothly migrate the data in a non-stop manner, select this option.</p>
Migration Object	<p>Entire instance: Migrate the entire database instance excluding the system databases such as <code>information_schema</code> , <code>mysql</code> , <code>performance_schema</code> , and <code>sys</code> .</p> <p>Specified objects: Migrate specified objects.</p>
Specify object	Select the objects to be migrated in Source Database Object and move them to the Selected Object box.

©2013-2022 Tencent Cloud. All rights reserved.

Page 80 of 341

5. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

6. Return to the data migration task list, and you can see that the task has entered the **Creating** status. After 1–2 minutes, the data migration task will be started.

Select **Structural migration** or **Full migration**: Once completed, the task will be stopped automatically.

Select **Full + Incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data sync.

Manually complete incremental data sync and business switchover at appropriate time.

Check whether the migration task is in the incremental sync stage without any lag. If so, stop writing data to the source database for a few minutes.

Manually complete incremental sync when the data gap between the target and the source databases is 0 MB and the time lag between them is 0 seconds.

7. (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

8. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Migration from MySQL/MariaDB/Percona/TDSQL MySQL to TDSQL for MySQL

Last updated : 2024-07-08 19:41:52

The steps of data migration from MySQL/MariaDB/Percona to TDSQL for MySQL are basically the same as those described in [Migration from TDSQL for MySQL to TDSQL for MySQL](#).

Migrating to TDSQL TDStore

Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)

Last updated : 2024-08-13 17:20:38

The steps of data migration from MySQL/MariaDB/Percona/TDSQL MySQL to TDSQL MySQL (TDStore) are basically the same as those described in [Migration from TDSQL for MySQL to TDSQL for MySQL](#).

Advanced Operations for MySQL Series

Migration

Migrating Advanced Object

Last updated : 2024-07-08 19:41:52

Overview

DTS migrates advanced objects like functions, triggers, procedures, and events on a one-time basis. It only migrates advanced objects already in the source database to the target database before the task starts, not those added to the source database after the task starts.

Note:

Currently, advanced objects can be migrated between MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Scope of application

Advanced objects can be migrated only with NewDTS.

Notes

We recommend you not rename tables when migrating advanced objects; otherwise, the migration may fail.

As the failure to migrate advanced objects does not affect the entire migration task, the success of the entire migration task does not necessarily mean that the advanced objects are also successfully migrated. Therefore, we recommend you check whether they are migrated on the **Migration Progress** page after the migration is completed.

When procedures and functions are migrated, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the migration account `user2`, and if not, DTS will change the `SQL SECURITY` attribute of `user1` in the target database from `DEFINER` to `INVOKER` (`[INVOKER = user1]`) after the migration, and set the `DEFINER` in the target database to the migration account `user2` (`[DEFINER = migration account user2]`).

Procedures and functions will be migrated during **source database export**. If there are no incremental migration tasks, triggers and events will be migrated when the task stops; otherwise, they will be migrated after you click **Done**, in which case the transition will take a longer time.

For cross-version migration, if the `sql_mode` set for the advanced objects in the source database is not supported by the target database, `sql_mode` will be changed to `NO_AUTO_VALUE_ON_ZERO` after the advanced objects

are migrated.

When migrating triggers and events, you need to grant the migration account the `TRIGGER` and `EVENT` permissions of the source database.

Directions

1. On the **Set migration options and select migration objects** page of the [data migration task](#), set the **Advanced Object** feature as needed, which is not enabled by default.

✓ Set source and target databases >

2 Set migration options and select migration objects >

3 Ver

Migration Type ⓘ *

Structural migration

Full migration

Full + Incremental migration

Data Consistency Check ⓘ *

Full check

No check

If the source instance is read-only, data consistency check will be skipped after data migration. For details, see [Migration](#)

Data Check

Content check

Migration Object ⓘ *

Entire instance

Specify object

Advanced Migration Object ⓘ

☒ Procedure

☒ Function

☒ Trigger

☒ Event

Advanced objects can only be copied once, which means you cannot copy new objects once the task is started.

Migrate Account ⓘ *

☐

Source Database Object

Search database name, supporting fuzzy match

1 database in total, with 1 displayed

More

☐ db-dst

☒ Tables

☐ Views

☒ Procedures (0 / 0)

☒ Functions

☒ Triggers

☒ Events

Refresh

Select all

Clear

Selected Object ⓘ

Globally search for o

☐ db-dst

☐ Table

☐ Proce

☐ Funct

☐ Trigg

☐ Event

Unfold all

Fold all

Sel

For migration notes, see [Migration FAQs](#)

Previous

Save

2. The advanced object check item will be added to the verification task. For more information, see [Advanced Object Check](#).

Creating Data Consistency Check Task (MySQL)

Last updated : 2024-07-08 20:02:27

Overview

During data consistency check, DTS compares the table data between the source and target databases and outputs the comparison result and inconsistency details for you to quickly process the inconsistent data. A data consistency check task is independent and does not affect the normal business in the source database or other DTS tasks.

Note:

Links currently supporting data consistency check are as follows:

MySQL/MariaDB/Percona/TDSQL for MySQL > MySQL

MySQL/MariaDB/Percona/TDSQL for MySQL > MariaDB

MySQL/MariaDB/Percona > TDSQL-C for MySQL

MySQL/MariaDB/Percona/TDSQL for MySQL > TDSQL for MySQL

MySQL/MariaDB/Percona/TDSQL for TDSQL > TDSQL for TDSQL

Notes

During data consistency check, only the database/table objects selected in the source database are compared with those migrated to the target database. The consistency is not checked for data written during migration, other advanced objects (such as procedures and events), and accounts.

A data consistency check task may increase the load in the source database instance. Therefore, you need to perform such tasks during off-peak hours.

A data consistency check task can be executed repeatedly, but one DTS instance can initiate only one such task at any time. **A task can be initiated when the last task is running but cannot after creation is completed.**

A table to be checked must have a primary key or unique key; otherwise, it will be skipped by DTS during the check. If you choose to **complete** or **terminate** a DTS task before a data consistency check task is completed, the check task will fail.

As data consistency check requires creating a new database `__tencentdb__` in the source database and writing the checksum table to the database, if the source database is read-only, data consistency check will be skipped.

Restrictions

Currently, check tasks are imperceptible to the DDL operations. If you perform DDL operations in the source database during migration, the check result will be inconsistent with the actual data, and you need to initiate another check task to get the accurate comparison result.

How It Works

DTS consistency check on MySQL databases is based on the row mode (`binlog_format=row`) that can correctly copy the source and replica data to ensure the data security.

1. Create the checksum database `__tencentdb__.Checksums` in the source database to store the data comparison information during the migration task.
2. Select the non-null unique key of the target table as the fixed check field.
3. Calculate the `crc1` checksum and row count `count1` of the source database and write them into the `__tencentdb__.Checksums` of the source database.

Similar to chunk check, during CRC calculation, select a fixed range (for example, data with primary keys ranging from `1` to `1000` in table A) based on the fixed check field, splice the data by row, calculate the CRC value (`crc`) for each chunk, and then calculate `crc1` for all the data in the source database.

4. DTS parses the binlog data in row mode, restores the SQL statement for writing the checksum into the source database, and replays the SQL statement in the target database.

In the target database, use the same variables as those in the source database to calculate the checksum and row count to get `crc2` and `count2` .

5. Compare the checksum and row count values of the source and target databases and display the comparison result.

Creating a Data Consistency Check Task

Automatic triggering

On the **Set migration options and select migration objects** page of a [data migration task](#), select **Full check** for **Data Consistency Check**. In this way, a data consistency check task will be triggered automatically when the migration task enters the **incremental sync** step.

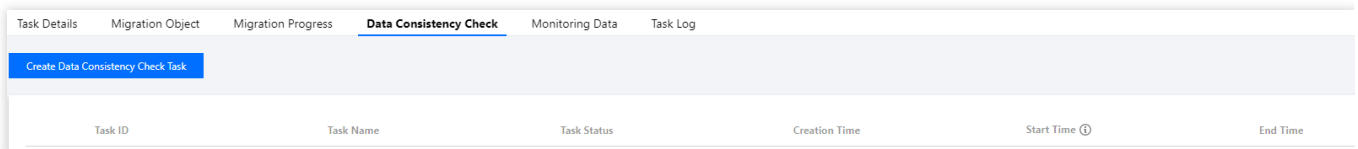
Manual creation

1. Log in to the [DTS console](#).
2. On the **Data Migration** page, select the target migration task and click **Create Data Consistency Check Task** in the **Operation** column.

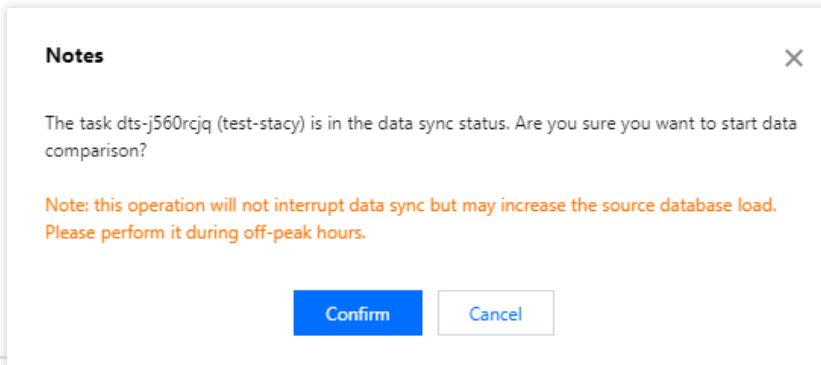
3. Click **Create Data Consistency Check Task**.

A data consistency check task can be created only when the corresponding migration task is in the **incremental sync** step. If the button is grayed out, the DTS task status does not meet the requirement; for example, the task has not entered the **incremental sync** step, has failed, or is terminated.

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



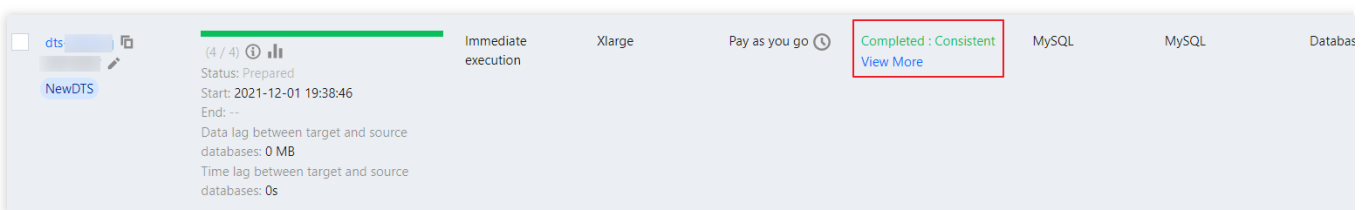
5. After configuring data consistency check parameters, click **Create and Start Consistency Check Task**.



Parameter	Description
Check Object	All migration objects: Check all the objects selected in the migration task. Custom: Check the selected migration objects.
Comparison Type	Full comparison: Check all the data for the selected objects. Sampling: Check the data of a certain proportion (10%, 20%, 30%, ..., 90%) for the selected objects. Row count comparison: Only compare the data row count for the selected check objects. In this case, even tables without the primary key can also be checked.
Thread Count	The value ranges from 1 to 5 and can be set as needed. Increasing the number of threads can speed up the consistency check, but will also increase the load of the source and target databases.

Viewing the Data Consistency Check Result

1. On the migration task homepage, view whether the check result is **Consistent** or **Inconsistent** in the **Last Check Result** column. Click **View More** to enter the **Verification Details** page.



2. Click **View** to view the check result.

Task Details	Migration Object	Migration Progress	Data Consistency Check	Monitoring Data	Task Log
Create Data Consistency Check Task					
Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	
dts-	dts-test	Completed	2021-12-16 14:31:30	2021-12-16 14:31:33	

3. If the data is consistent, the result is as follows:

You can view the numbers of estimated tables, checked tables, inconsistent tables, and inconsistent chunks. Here, the number of estimated tables is an estimation that may differ from the actual value, as providing an accurate value will compromise the overall check performance.

Possible causes for not checking a table: There is no primary key or non-null unique key, the table is empty, the engine type is not supported, or the table does not exist.

Overview

Status Completed

Compare Conclusion Consistent

Total Tables

1

Checked Tables

1

Inconsistent Tables

0

Details

Database	Data Table	Chunk ID	Index Name	Last Index Key	First Index Key	Source Data Value	Target Data Value
No data							

Total items: 0

Details of Unchecked Tables

Database	Data Table	Reason
No data yet		

Total items: 0

If the data is inconsistent, the result is as follows:

概要

分库分表类型: 垂直拆分
分库数: 已同步
分表数: 0

待同步总数

813个

已同步表数

632个

不一致表数

1个

不一致分库数

1个

不一致详情

多个关键字用逗号分隔, 1个分隔, 多个过滤条件用逗号分隔

数据库	数据表	分库号	索引名称	索引上边界	索引下边界	源库数据	目标库数据	源库时间 (毫秒)	完成时间
newsdb	news	1	PRIMARY	[*]	[*]	369F06F	97543042	177.04	2022-12-20 13:16:22

共 1 条

20条 / 页1 / 1页

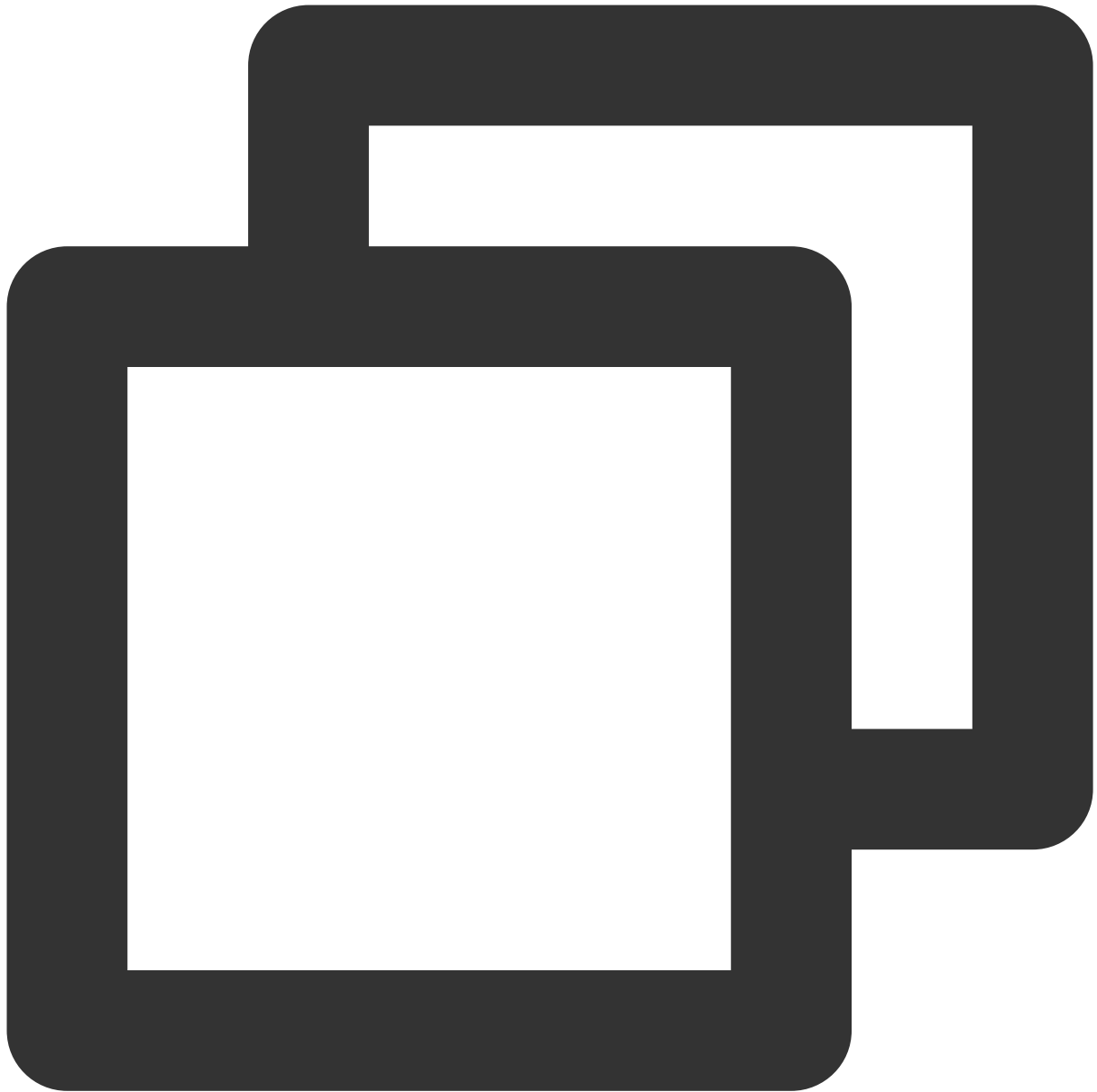
未检查表详情

多个关键字用逗号分隔, 1个分隔, 多个过滤条件用逗号分隔

数据库	数据表	未检查原因
mysqlbak	replication_asynchronous_connect	empty table
mysqlbak	replication_asynchronous_connect	empty table
mysqlbak	replication_asynchronous_connect	empty table
mysqlbak	replication_asynchronous_connect	empty table
mysqlbak	replication_asynchronous_connect	empty table

For inconsistent data, you need to manually confirm the corresponding data content of the source and target databases. Specifically, compare the values based on **Database**, **Data Table**, **Index Name**, **Last Index Key**, and **First Index Key** parameters displayed on the page.

The steps are as follows:



1. Log in to the source database and query the prompted index range.
`select * from table_name where col_index >=1 and <=5;`
2. Log in to the target database and query the prompted index range.
3. Compare the data of the source and target databases.

Account Migration

Last updated : 2024-07-08 19:41:52

Overview

This document describes how to migrate the user information from the source database to the target database.

Note:

Currently, the following data migration scenarios supports account migration: MySQL > MySQL and MySQL > TDSQL-C.

Notes

1. During account migration, DTS will check the account information of the source database and migrate accounts meeting the requirements. DTS won't migrate ineligible accounts or will migrate them with fewer permissions. Account check may have the following results:

Full migration: The account information fully meets the check requirements and can be migrated normally.

Migration with fewer permissions: If the check requirements are partially meet, only compliant permissions will be migrated.

DTS supports permission migration only at the database and global but not other levels such as table, stored procedure, or column. For example, if an account in the source database has permissions at both the database and table levels, only the permissions at the database level will be migrated to the target database.

If the account executing the migration task doesn't have the same permissions in the target database, such permissions cannot be migrated (for database and global levels only). For example, if it doesn't have the deletion permission in the target database, the deletion permission of the source database account cannot be migrated.

No migration: The account information doesn't meet the requirements and cannot be migrated. For more information, see [Limits](#).

2. If the source and target databases have the same account, DTS will overwrite the account information in the target database with that in the source database.

If an account in the source database has the same `user` and `host` as an account in the target database (except the account executing the migration task), but they have different passwords, after account migration, the password of the account in the target database will be overwritten by the password of the account in the source database.

If an account to be migrated in the source database has the same information as the account executing the migration task, the following attributes of the source database account will overwrite those in the target database and may affect the performance of the migration task: `MAX_QUERIES_PER_HOUR` , `MAX_UPDATES_PER_HOUR` ,

`MAX_CONNECTIONS_PER_HOUR` , `MAX_USER_CONNECTIONS` , `PASSWORD_EXPIRE` , and `ACCOUNT LOCK` .

If an account to be migrated in the source database has the same `user` and `host` as the account executing the migration task, but they have different passwords, an error will be reported during the verification task. In this case, you need to change the passwords to the same one before executing the migration task again.

3. Account migration is not affected by database/table mappings.

Limits

1. System accounts of Alibaba Cloud instances cannot be migrated, such as `replicator` , `aurora` , `aurora_proxy` , `root` , and `aliyun_root` .

2. When migrating accounts from an Alibaba Cloud instance, DTS will get the user data from the `mysql.user` table first. If the `mysql.user` table doesn't exist, DTS will get the data from `mysql.user_view` . If `mysql.user_view` doesn't have the password column, accounts cannot be migrated.

3. Special limits for MySQL 8.0:

If the source database is MySQL 8.0, accounts with `SYSTEM_USER` permissions cannot be migrated.

If the target database is MySQL 8.0 and has an account with stored procedures of the account to be migrated in the source database, as DTS currently cannot migrate stored procedures, an error will be reported for the `CREATE USER` statement. In this case, you need to delete the account in the target database first and execute the migration task again.

If the target database is MySQL 8.0, and the account to be migrated in the source database has the `SYSTEM_USER` permissions, but the account executing the migration task doesn't, the source database account cannot be migrated.

4. If the target database is MySQL 5.5 or 5.6, and the account executing the migration task doesn't have the `with grant option` permission, account migration is not supported.

5. Roles created in the source database will be migrated as users. However, users with role permissions won't be migrated.

6. System accounts such as `mysql` cannot be migrated.

7. Proxy users cannot be migrated.

8. If an account in the source database contains special symbols such as the hex `0x00` symbol, the account migration task may report an error.

Directions

1. On the **Set migration options and select migration objects** page of a [data migration task](#), select **Migrate Account**.

1 Set source and target databases > 2 Set migration options and select migration objects > 3 Verify task

Migration Type ⓘ * Structural migration Full migration Full + Incremental migration

Data Consistency Check ⓘ * Full check No check

If the source instance is read-only, data consistency check will be skipped after data migration. For details, see [Migration FAQs](#).

Data Check Content check

Migration Object ⓘ * Entire instance Specify object

Advanced Migration Object ⓘ ☐ Procedure ☐ Function ☐ Trigger ☐ Event

Advanced objects can only be copied once, which means you cannot copy new objects once the task is started.

Migrate Account ⓘ * ☒

Source Database Object

Search database name, supporting fuzzy match

1 database in total, with 1 displayed [More](#)

- ☐ db-dst
 - ☒ Tables
 - ☐ Views

[Refresh](#) [Select all](#) [Clear](#)

Selected Object ⓘ [Batch Rename ⓘ](#)

Globally search for original object names, with fuzzy match supported

- ☐ db-dst

[Unfold all](#) [Fold all](#) [Select all](#) [Clear](#) [Revert to Original Name](#)

ⓘ For migration notes, see [Migration FAQs](#)

[Previous](#) [Save](#)

2. The verification task will check the account information in the source database and will migrate eligible accounts.

3. Click **View Details** to view the detailed account migration result.

Full migration: The account meets the check requirements and can be migrated normally.

Migration with fewer permissions: If the check requirements are partially meet, only compliant permissions will be migrated.

No migration: The account doesn't meet the check requirements and cannot be migrated. Possible causes include:
The account is a system user of an Alibaba Cloud database, or the user account and password information cannot be obtained.

The source database is MySQL 8.0 and the account has `SYSTEM_USER` permissions, or the account has `SYSTEM_USER` permissions but the account executing the migration task doesn't.

The target database is MySQL 5.5 or 5.6, but the account executing the migration task doesn't have the `grant` permission.

The account is a system account such as `mysql` , `sqlserver` , or `orcal` .

The account is a proxy user.

A system problem such as DTS parsing failure occurs.

Database/Table Renaming

Last updated : 2024-07-08 19:41:52

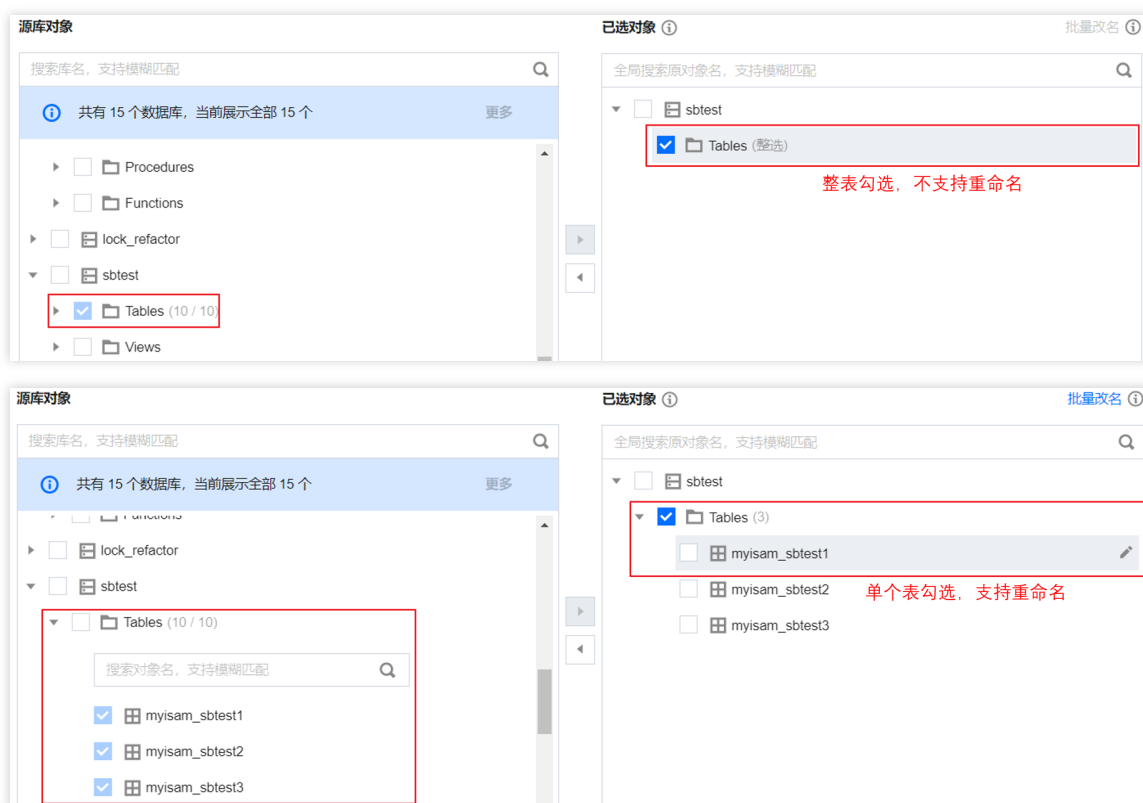
Overview

Database/table renaming is also known as database/table mapping. In data migration scenarios, you can rename the databases/tables to be synced in order to avoid name conflicts between the source and target databases.

You can either rename a single database/table or rename them in batches. In batch renaming scenarios, there are three renaming methods: adding prefix/suffix, renaming uniformly, and replacing keyword.

Note

1. If you select an entire database or table as the migration object, you cannot rename a specified table. If you need to do so, select the target tables one by one when selecting migration objects.



2. In batch renaming scenarios, if you select the uniform renaming method, the result of the data consistency check task will be "Inconsistent". You need to check data consistency on your own.

3. In many-to-one migration scenarios:

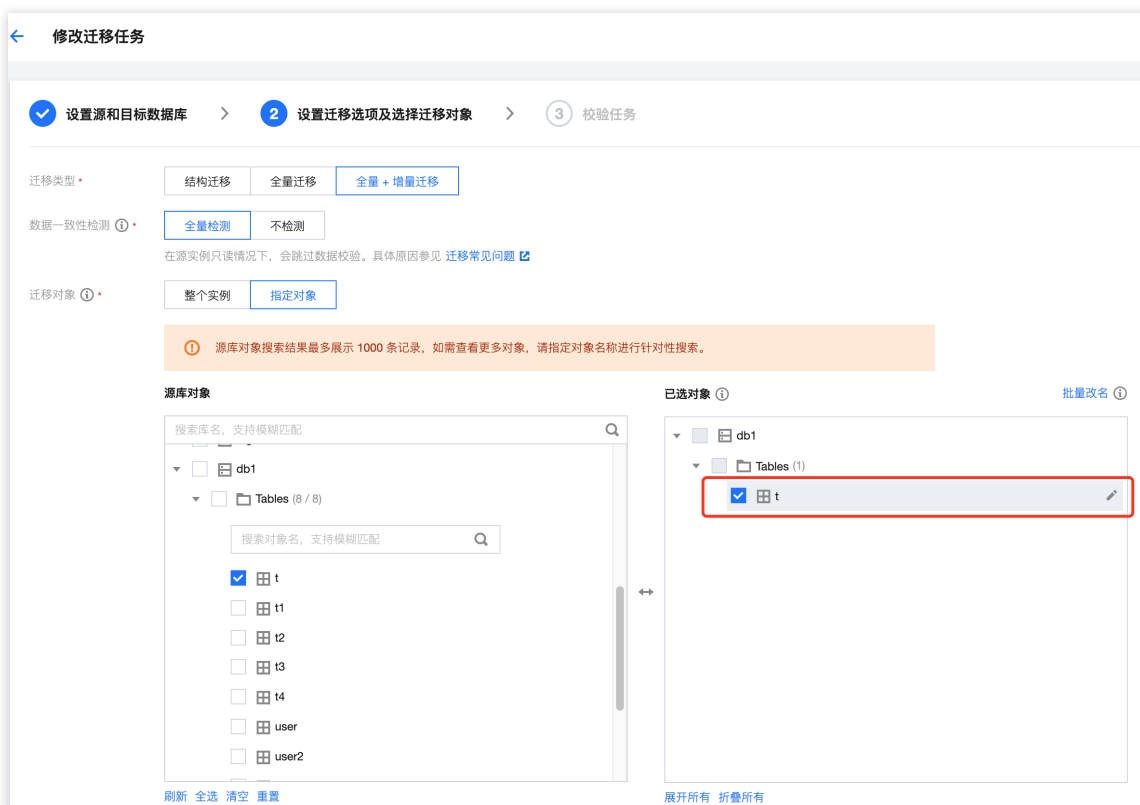
After you use the batch renaming feature, a structure initialization failure may be reported on the **Task Progress** tab in the console. The failure is reported because the target database already contains the structure information of this table. You can ignore this error because the data updates of this table will be migrated normally.

In the task check stage, the check item "Target instance content conflict check" will report an error, which can be ignored. To skip this check item, [submit a ticket](#).



Renaming a Specified Database/Table

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and create a migration task.
2. In the **Set migration options and select migration objects** step, in **Selected Object** on the right, hover over an object to be modified and click the displayed **Edit** icon.



3. In the pop-up window, rename the object and click **OK**.

Database name mapping: In the pop-up window, set the name of the database in the target instance.

编辑库映射

×

原库名 DB2

库名 *

确定取消

Table name mapping: In the pop-up window, set the name of the table in the target instance.

编辑表映射

×

原表名 TABLE_1

表名 *

确定取消

Batch Renaming

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and create a migration task.
2. In the **Set migration options and select migration objects** step, select the migration objects and click **Batch Rename** on the right of the **Selected Object** box.

1 设置源和目标数据库 > 2 设置迁移选项及选择迁移对象 > 3 校验任务

迁移类型 ① •

数据一致性检测 ① •

在源实例只读情况下，会跳过数据校验。具体原因参见 [迁移常见问题](#)

数据校验 内容校验

迁移对象 ① •

高级迁移对象 ① ☐ Procedure ☐ Function ☐ Trigger ☐ Event

高级对象的复制是一次性动作，在任务启动后，不会再复制新创建的高级对象。

是否迁移帐号 ① • ☐

源库对象

搜索库名，支持模糊匹配

① 共有 15 个数据库，当前展示全部 15 个 [更多](#)

DB2

Tables (4 / 4)

搜索对象名，支持模糊匹配

☒ TABLE_1

☒ t1

☐ t2

☐ test_datetime

Views

miratedb4dts

刷新 全选 清空

已选对象 ① [批量改名](#) ①

全局搜索原对象名，支持模糊匹配

☒ DB2

Tables (2)

展开所有 折叠所有 全选 清空 恢复原名称

是否同步 Online DDL 临时表 ① ☐

3. Set the batch renaming rules and click **OK**.

Renaming Scope

All selected databases and tables: All the selected databases and tables will be renamed in batches.

All selected databases: All the selected databases will be renamed in batches.

All selected tables: All the selected tables will be renamed in batches.

Renaming Rule

Add prefix/suffix

批量改名

×

改名范围

所选全部库表

所选全部库

所选全部表

改名规则

添加前缀/后缀 ▼

11

22

所选全部库表名称，添加 11 前缀，添加 22 后缀

i • 整选库或整选表时，无法对表进行重命名；如果要对库内的表重命名，请勾选库中具体的表

确定

取消

Rename uniformly: The selected objects will be renamed uniformly, which is suitable for multi-table merge scenarios. Make sure that there is no primary key conflict when migrating table data from multiple sources to the target database.

批量改名

×

改名范围

所选全部库表

所选全部库

所选全部表

改名规则

统一重命名 ▼

table_a

所选全部库表名称，统一重命名为 table_a

i • 统一重命名适用于多表合一的场景，请确保无主键冲突

• 统一重命名后，一致性校验结果为数据不一致，需要自行进行数据校验

• 整选库或整选表时，无法对表进行重命名；如果要对库内的表重命名，请勾选库中具体的表

确定

取消

Replace keyword: The keyword in the names of the selected objects will be replaced uniformly. If an object name contains multiple keywords, only the first one will be replaced.

批量改名

×

改名范围

所选全部库表

所选全部库

所选全部表

改名规则

替换关键词

关键词

替换内容

ⓘ

- 整选库或整选表时，无法对表进行重命名；如果要对库内的表重命名，请勾选库中具体的表

确定

取消

Migrating Online DDL Temp Table

Last updated : 2024-07-08 19:41:52

Overview

When performing an online DDL operation on tables in the source database with the gh-ost or pt-online-schema-change (pt-osc) tool, you need to migrate the temp tables generated by online DDL changes to the target database. DTS allows you to associate the temp table names of the object table in advance when selecting the migration objects, so that the object table can be migrated together with the temp tables subsequently generated by the source database.

When you perform an online DDL operation on the table `table name` with the gh-ost tool, DTS supports migrating temp tables `_table name_ghc` , `_table name_gho` , and `_table name_del` to the target database.

When you perform an online DDL operation on the table `table name` with the pt-osc tool, DTS supports migrating temp tables `_table name_new` and `_table name_old` to the target database.

Scope of application

Currently, temp tables can be migrated between MySQL, MariaDB, Percona, and TDSQL-C for MySQL.

Restrictions

The table mapping (table renaming) feature conflicts with the temp table migration feature (online DDL), and only one of them can be enabled at any time.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and create a migration task.
2. In the **Set migration options and select migration objects** step, select **Sync Online DDL Temp Table** and **Online DDL Tool**

If you select **gh-ost**, DTS will migrate the temp tables (`_table name_ghc` , `_table name_gho` , and `_table name_del`) generated by the gh-ost tool to the target database.

If you select **pt-osc**, DTS will migrate the temp tables (`_table name_new` and `_table name_old`) generated by the pt-osc tool to the target database.

In **Selected Object** on the right, you need to select the specified table object, and then the button **Sync Online DDL Temp Table** will be displayed below. If you select the databases or tables, the button cannot be displayed.

If the source database already contains tables with the same names as temp tables `_table name_new` and `_table name_old`, pt-osc will generate other temp tables with different names, which DTS cannot migrate. In this case, if you just choose the table itself as the **Migration Object**, data in temp tables created by online DDL updates can't be migrated to the target database. Instead, you must select the entire database or instance where the table to be migrated resides.

Processing for MySQL Series Pre-Check Failure

Database Connection Check

Last updated : 2024-07-08 19:41:52

Check Details

The source and target databases need to be normally connected, and if not, the error message "Failed to connect to the source database" will be displayed.

Causes

The network or server where the source database resides has a security group or firewall configured.

The source IP addresses are blocked in the source database.

The network port is closed.

The database account or password is incorrect.

Security Group or Firewall Configured in Network or Server of Source Database

Check method

A security group is similar to a firewall. It is a group of network security settings for databases in the cloud.

Check as follows based on the actual conditions:

Check whether the server where the source database resides is configured with firewall policies.

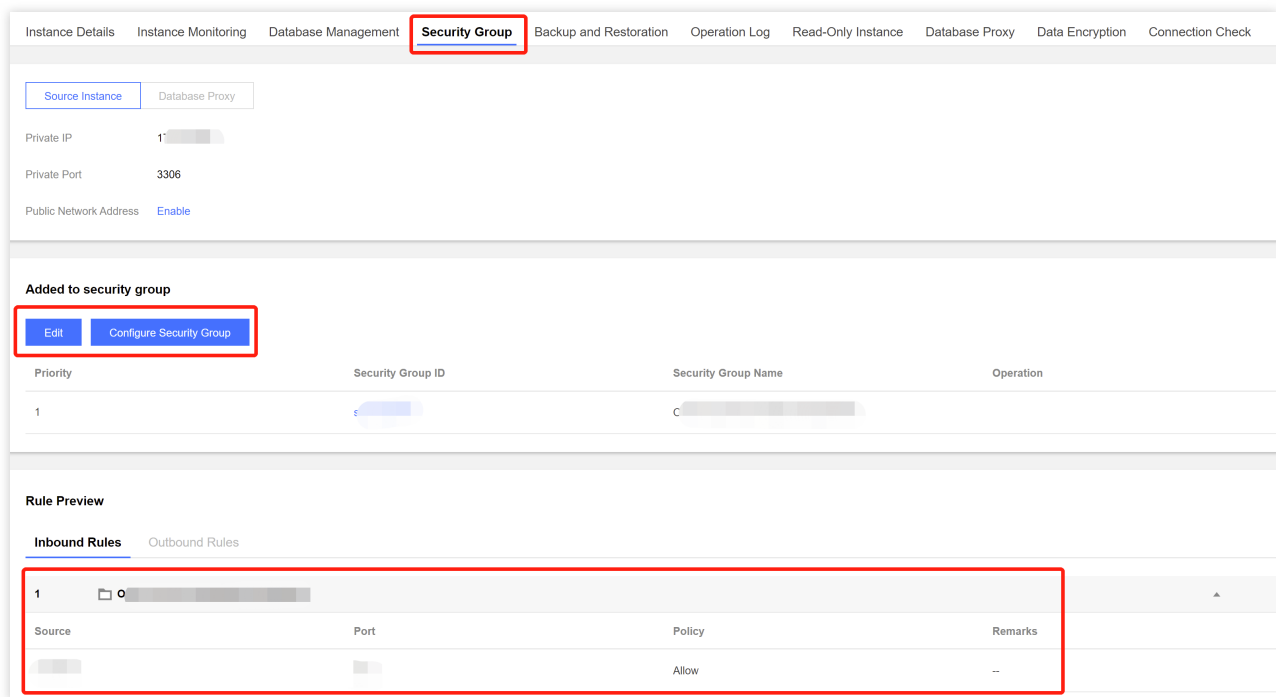
Windows: open Control Panel and find the Windows Defender Firewall and check whether firewall policies are configured.

Linux: run the `iptables -L` command to check whether the server is configured with firewall policies.

Check whether DTS IP range is blocked in the security group of the database.

1.1 Log in to the [corresponding database](#) and click an instance ID in the instance list to enter the instance management page.

1.2 On the instance management page, select the **Security Group** tab and check whether there are policies blocking the SNAT IP range of DTS.



Fix

Fix it as follows based on the actual conditions:

The firewall is enabled on the server:

1.1 Disable the server firewall, log in to DTS, and run the verification task again.

Note:

This method is applicable to both Windows and Linux.

1.2 Set the DTS IP range policy to **Allow**.

The SNAT IP range of DTS is blocked in the security group:

1.1 Click the corresponding security group ID on the **Security Group** tab.

2. Set the DTS IP range policy to **Allow**.

Source IP Addresses Blocked in Source Database

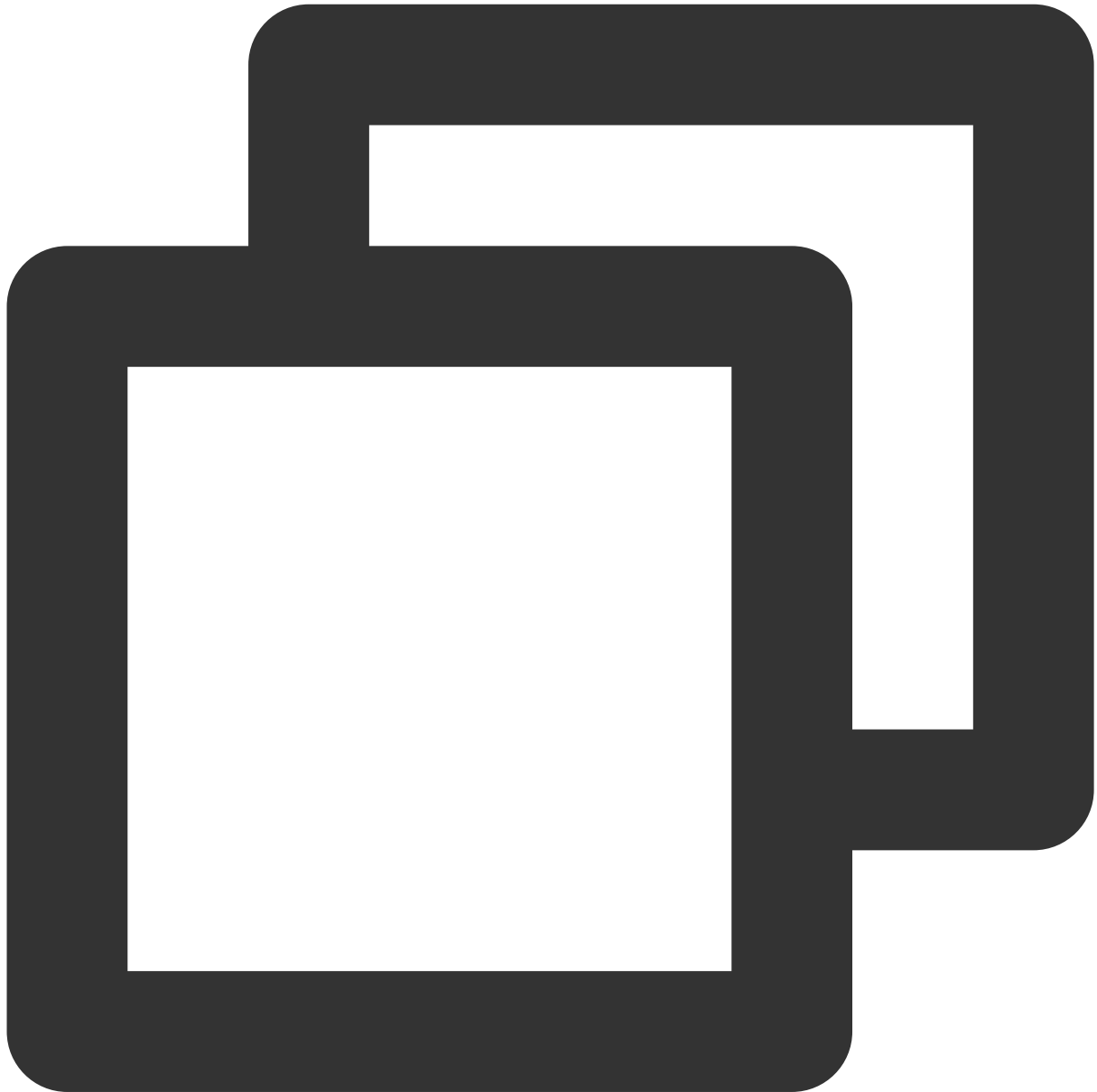
Check method

MySQL

On the server where the source database is deployed, use the database account and password entered in the data migration task to connect to the source database. If the database can be normally connected, the source IP address may be blocked in the source database.

For self-built database, you need to check the `bind-address` configuration in the database. If it is not `0.0.0.0`, the IP is blocked.

If the source database is MySQL, you can use the MySQL client to connect to it, run the following SQL statement, and check whether the list of authorized IP addresses contains the SNAT IP addresses of DTS in the output result. When granting database permissions to users, the authorized IPs must include the SNAT IPs; otherwise, they may be blocked; for example:



```
root@10.0.0.0/8 // Authorize users to access through `10.0.0.0/8`, and other IPs w
root@%          // Authorize users to access all IPs, which should include the SNA
```

You can verify as follows:



```
select host,user,authentication_string,password_expired,account_locked from
mysql.user WHERE user='[\\$Username]';    // `[\\$Username]` is the database account
```

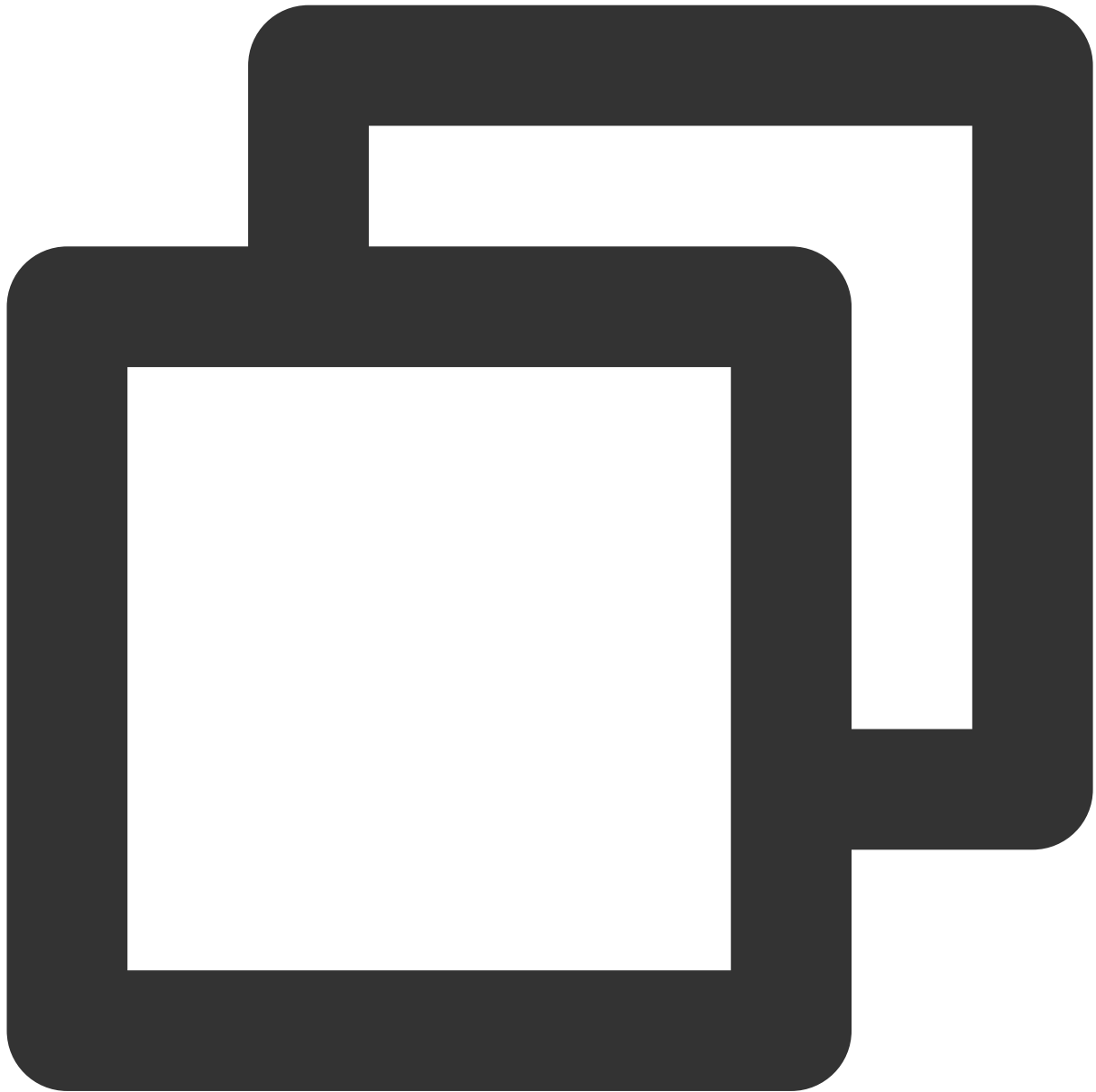
SQL Server

Check whether there is an endpoint or trigger that blocks the access source IP address in the source database.

PostgreSQL

If the source database is another database in the cloud, check whether the secure access policies in the source database have restrictions. Check as follows according to the specific cloud vendor:

If the source database is a self-built PostgreSQL database, enter the `data` directory in the `$PGDATA` directory, find the `pg_hba.conf` file, and check whether the file contains a `deny` policy or only allows access from certain IP addresses over the network.



```
# cat pg_hba.conf
local    replication    all                                     trust
host     replication    all             127.x.x.1/32          trust
host     replication    all             ::1/128              trust
host     all             all             0.0.0.0/0            md5
host     all             all             172.x.x.0/20         md5
```

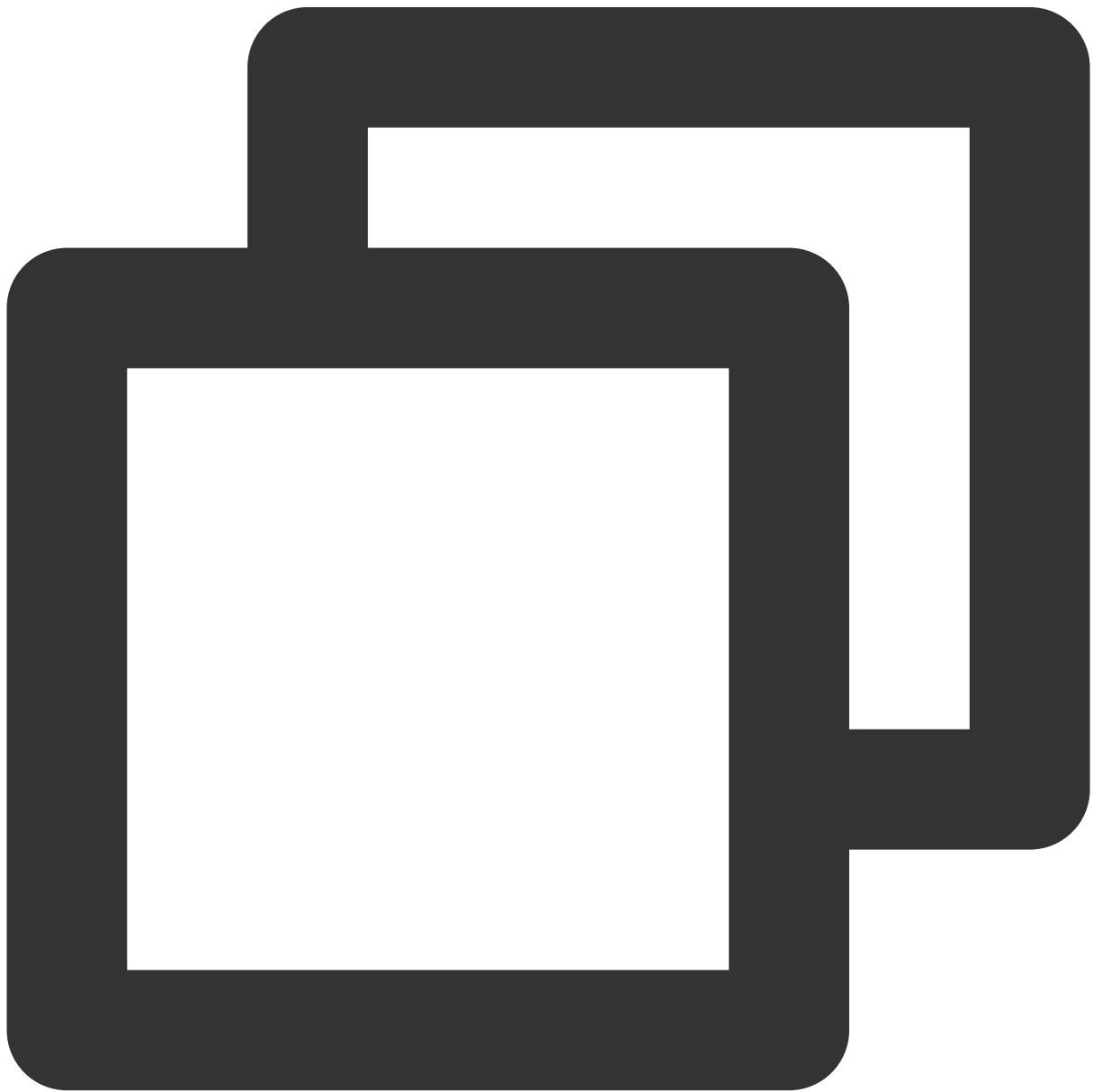
MongoDB

For self-built database, you need to check the `bind-address` configuration in the database. If it is not `0.0.0.0`, the IP is blocked.

Fix

MySQL

1. If the source database is MySQL, run the following SQL statement in it to authorize the user configured in the data migration task.



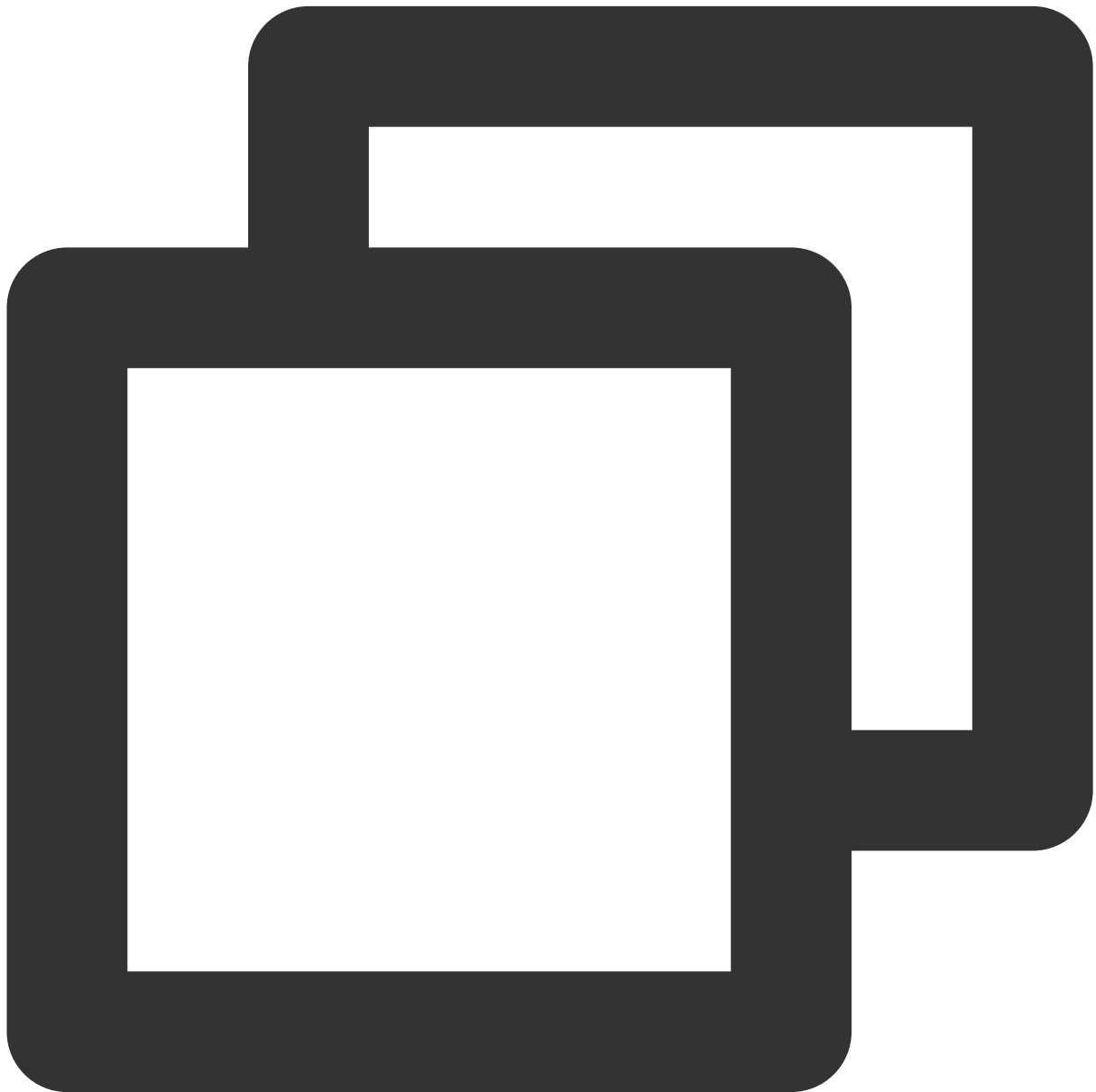
```
mysql> grant all privileges on . to '[\\$UserName]@'%; // `[\\$Username]` is the
mysql> flush privileges;
```

2. For a self-built database, if the `bind-address` configuration is incorrect, modify it as instructed below.

2.1. Add the following content to the `/etc/my.cnf` file:

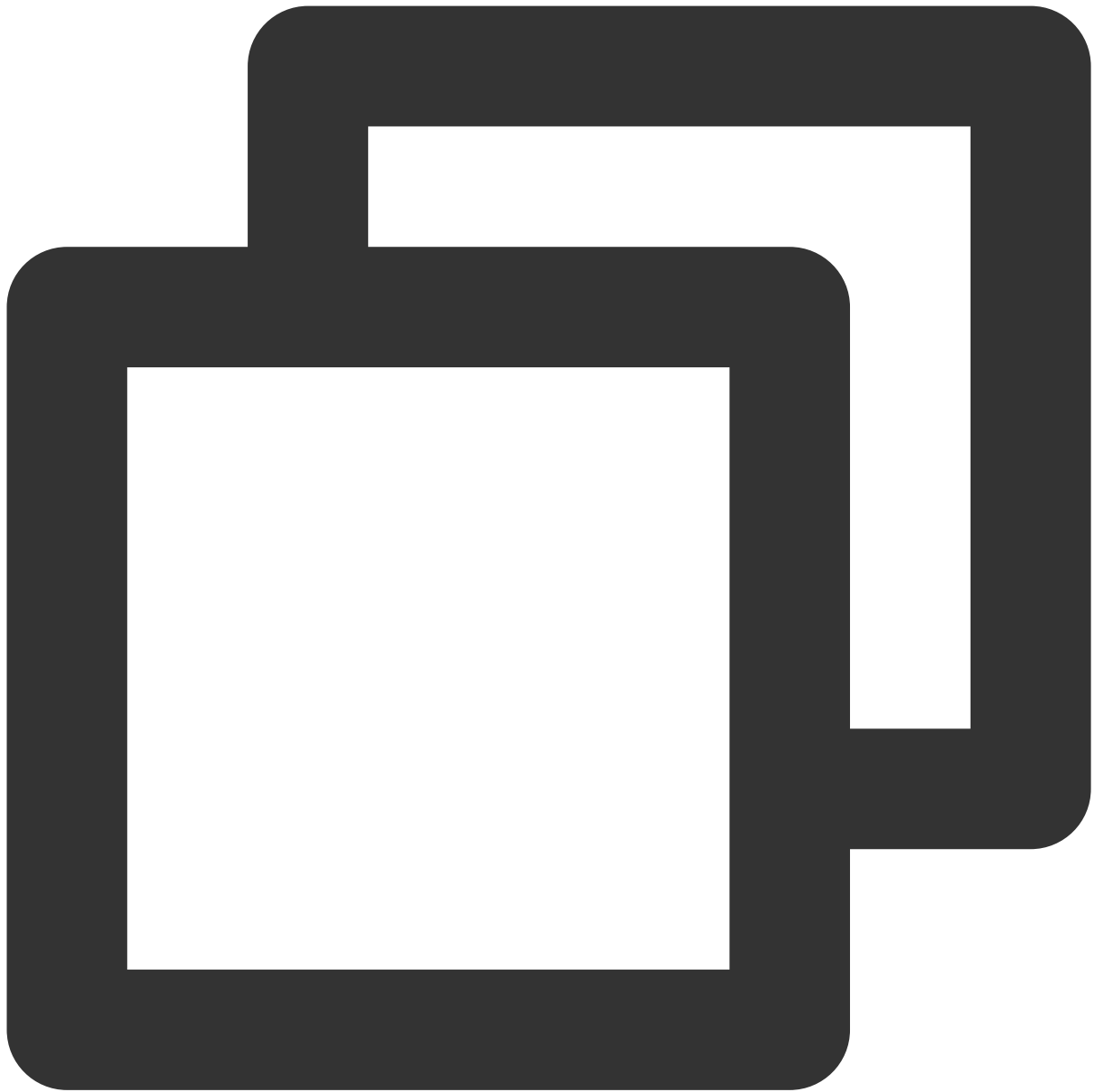
Note:

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.



```
bind-address=0.0.0.0 # All IP addresses or specified addresses
```

2.2. Restart the database.



```
service mysqld restart
```

2.3. Check whether the configuration takes effect.



```
netstat -tln
```

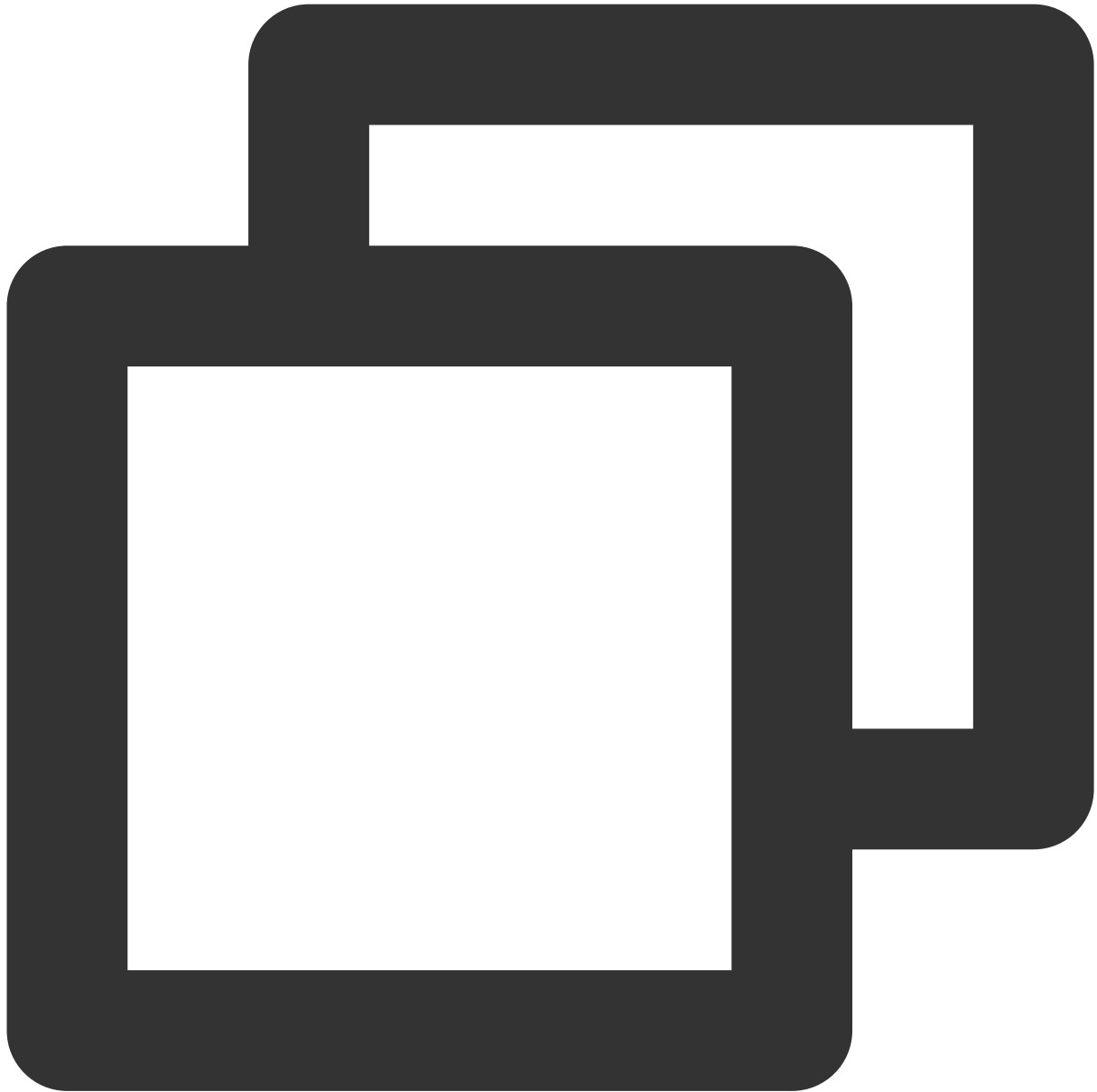
3. Run the verification task again.

SQL Server

Disable the firewall or trigger.

PostgreSQL

1. Add an access policy allowing the DTS IP range to the `pg_hba.conf` file or temporarily open all IP ranges in the access policy during migration. For example, add the following line to the `pg_hba.conf` file:



```
host      all             all             0.0.0.0/0      md5
```

2. After the modification is completed, you can restart the database to make the configuration take effect:



```
pg_ctl -D $PGDATA restart
```

3. Run the verification task again.

MongoDB

Configure `bind-address` as instructed in [MySQL](#).

Closed Network Port

Check method

Below are the default ports for common databases. You need to check whether they are opened, and if not, open them based on the actual conditions:

MySQL: 3306

SQL Server: 1433

PostgreSQL: 5432

MongoDB: 27017

Redis: 6379

Fix

Open the corresponding database port.

If the source database is SQL Server, you need to open the file sharing service port 445 at the same time.

Incorrect Database Account or Password

Check method

Log in to the source database to check whether the account and password are correct.

Fix

Modify the data migration task in the [DTS console](#), enter the correct database account and password, and run the verification task again.

Peripheral Check

Last updated : 2024-07-08 19:41:52

MySQL/TDSQL for MySQL/TDSQL-C check details

Check requirements: The `innodb_stats_on_metadata` environment variable in the source database must be set to `OFF` .

-- Check description:

If the `innodb_stats_on_metadata` parameter is enabled, every time tables in the `information_schema` metadatabase are queried, InnoDB will update the `information_schema.statistics` table, causing slower access. After this parameter is disabled, access to the schema table can be faster.

On MySQL versions earlier than 5.6.6, the default value of the `innodb_stats_on_metadata` parameter is `ON` , and you need to change it to `OFF` . On MySQL 5.6.6 or later, the default value is `OFF` , which has no problem.

Troubleshooting

1. Log in to the source database.
2. Change the value of `innodb_stats_on_metadata` to `OFF` .



```
set global innodb_stats_on_metadata = OFF;
```

3. Check whether the configuration takes effect.



```
show global variables like '%innodb_stats_on_metadata%';
```

The system should display a result similar to the following:



```
mysql> show global variables like '%innodb_stats_on_metadata%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| innodb_stats_on_metadata | OFF   |
+-----+-----+
1 row in set (0.00 sec)
```

4. Run the verification task again.

Version Check

Last updated : 2024-07-08 19:41:52

Check Details

Check requirements: The target database version must be later than or equal to the source database version, and all versions in migration and sync tasks must meet the version requirements.

Check description: Here, the versions are differentiated by the major version number; for example, v5.6.x supports migration or sync to v5.6.x, v5.7.x, and later versions. The last digit is the minor version number, which is not restricted; for example, v5.6.5 can be migrated or synced to v5.6.4, but there may be compatibility issues.

Troubleshooting

Check the source and target databases as instructed in [Databases Supported for Data Migration](#) and [Databases Supported for Data Sync](#). If the source or target database version is not supported, upgrade the target database version or use a database instance on a higher version.

Source Instance Permission Check

Last updated : 2024-07-08 19:41:52

Check Details

Check whether you have the operation permissions of the database by referring to the following:

[Permission requirements for data migration](#)

[Permission requirements for data sync](#)

[Permission requirements for data subscription](#)

Troubleshooting

If you don't have the operation permissions, get authorized based on the permission requirements in the check details, and run the verification task again.

Partial Database Parameter Check

Last updated : 2024-07-08 19:41:52

Check Details

`row_format` in the source database table cannot be `FIXED` .

The values of the `lower_case_table_names` variable in both the source and target databases must be the same.

The `max_allowed_packet` parameter of the target database must be set to 4 MB or above.

The `connect_timeout` variable of the source database must be above 10.

In migration from MySQL/TDSQL for MySQL/TDSQL-C to MySQL, if a time-consuming SQL statement is running on the source database, a warning will be reported, with the content being "A time-consuming SQL statement is running on the source database, which may cause table locks. Please try again later or process the SQL statement on the source database".

Troubleshooting

Modifying the `row_format` parameter in the source database

If the value of `row_format` in a database table is `FIXED` , an error will be reported when the storage length of each row of the table exceeds the limit. Therefore, you need to change the value of `row_format` to `DYNAMIC` so that the storage length of each row varies by the content length.

If a similar error occurs, fix it as follows:

1. Log in to the source database.
2. Set `row_format` to `DYNAMIC` .



```
alter table table_name row_format = DYNAMIC;
```

3. Check whether the configuration takes effect.



```
show table status like 'table_name'\\G;
```

The system will display a result similar to the following:



```
mysql> show table status like 'table_name'\\G;
***** 1. row *****
      Name: table_name
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 5
      .....
1 row in set (0.00 sec)
```

4. Run the verification task again.

Making `lower_case_table_names` have the same value in source and target databases

`lower_case_table_names` sets the letter case sensitivity in MySQL. It has the following valid values:

Windows and macOS environments are case-insensitive, but Linux environments are case-sensitive. To ensure the compatibility between different operating systems, you need to use the same letter case sensitivity rule.

- 0 : The name of a stored table is in the specified letter case and is case-sensitive during comparison.
- 1 : The name of a stored table is in lowercase on the disk and is case-insensitive during comparison.
- 2 : The name of a stored table is in the specified letter case and is in lowercase during comparison.

If a similar error occurs, set the parameter in the source and target databases to the same value as follows:

1. Log in to the source database.
2. Check the values of `lower_case_table_names` in the source and target databases.



```
show variables like '%lower_case_table_names%';
```

3. Modify the `my.cnf` configuration file of the source database as follows:

Note

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.



```
lower_case_table_names = 1
```

4. Run the following command to restart the database:



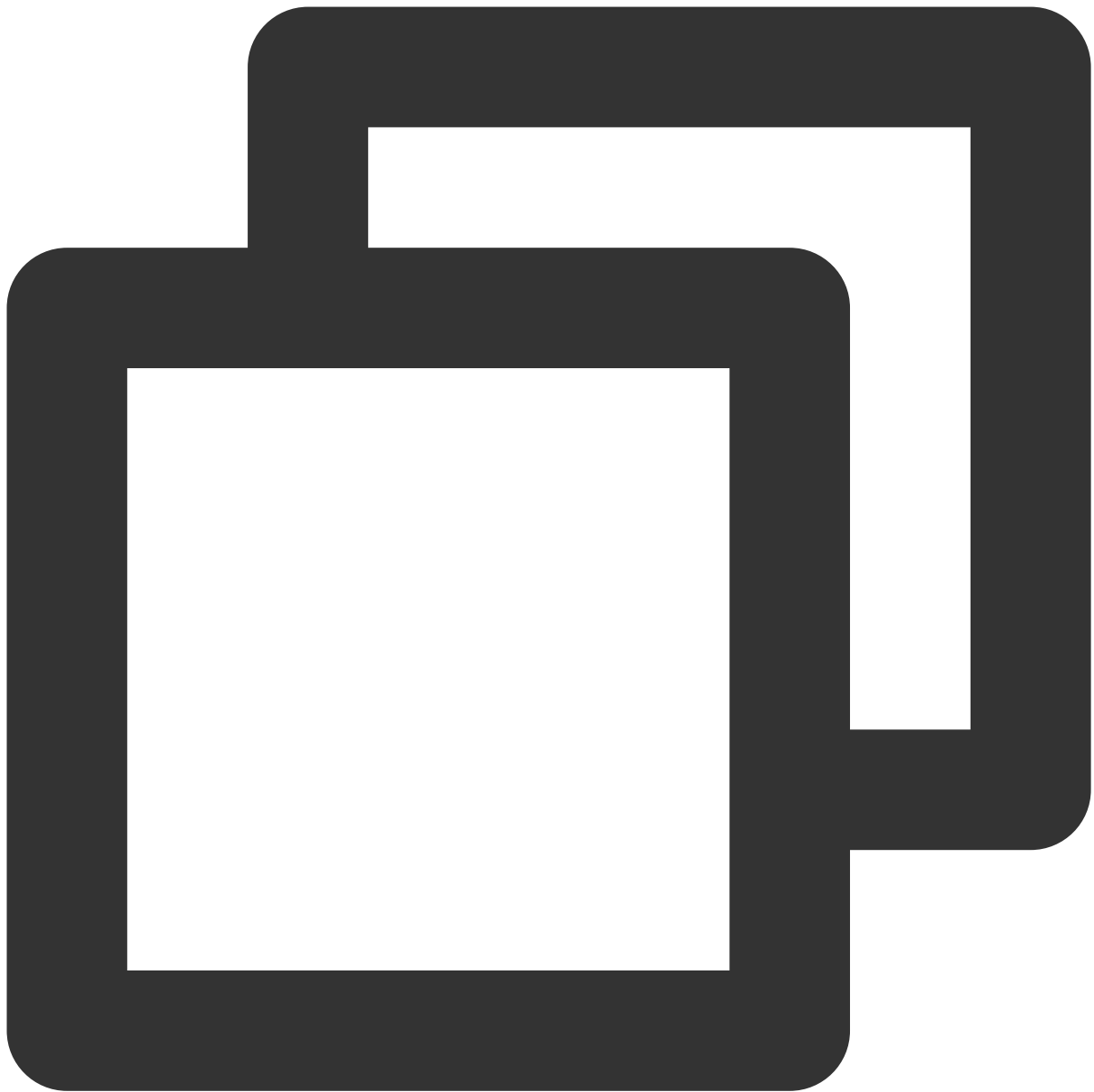
```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown  
[$Mysql_Dir]/bin/safe_mysqld &
```

5. Check whether the configuration takes effect.



```
show variables like '%lower_case_table_names%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%lower_case_table_names%';
+-----+
| Variable_name          | Value |
+-----+
| lower_case_table_names | 1     |
+-----+
1 row in set (0.00 sec)
```

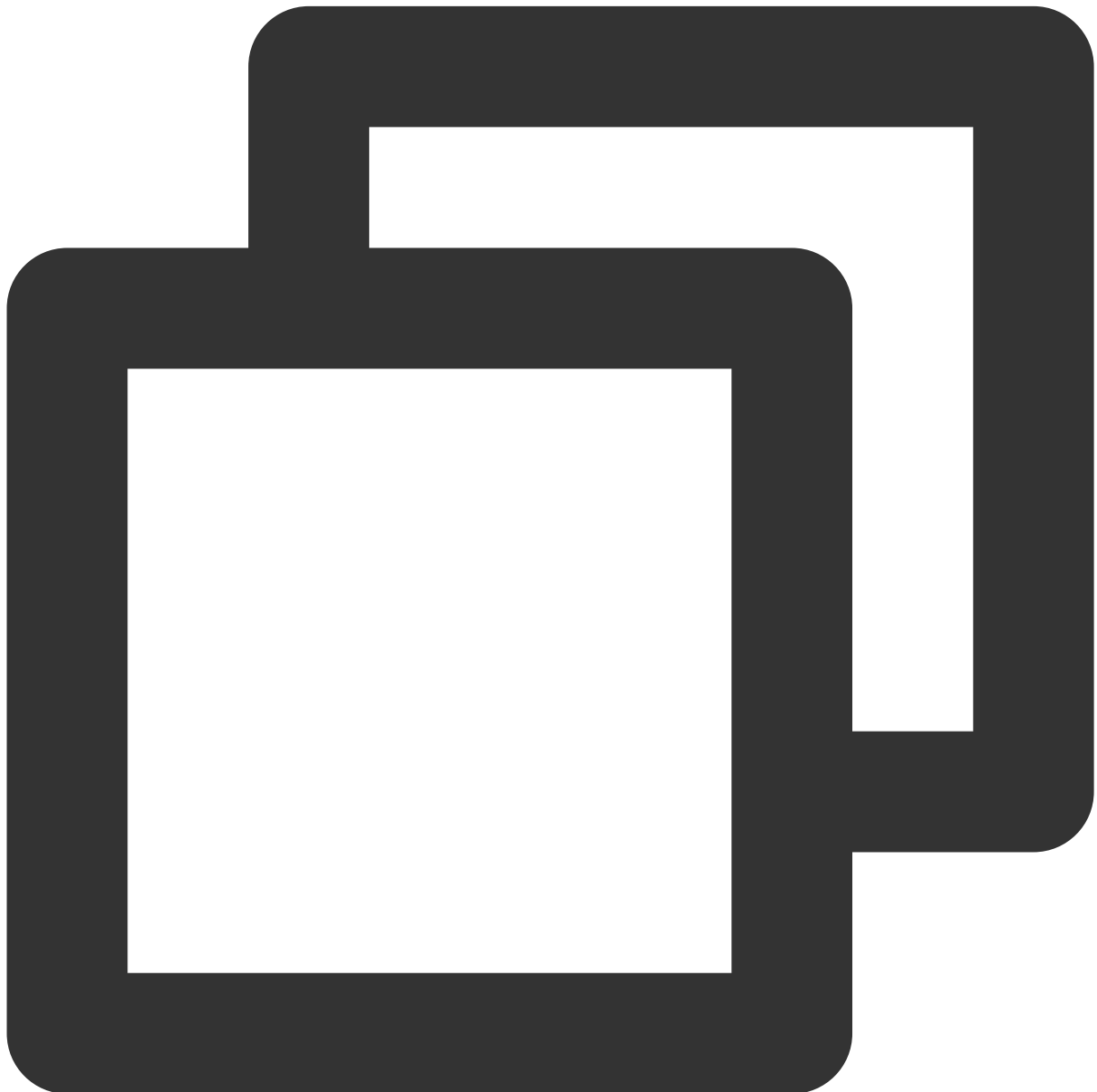
6. Run the verification task again.

Modifying the `max_allowed_packet` parameter in the target database

`max_allowed_packet` is the maximum size of a packet that can be transferred. If its value is too large, more memory will be used, causing packet losses and the inability to capture the SQL statements of large exception event packets. If its value is too small, program errors may occur, causing backup failures and frequent sending/receiving of network packets, which compromises the system performance.

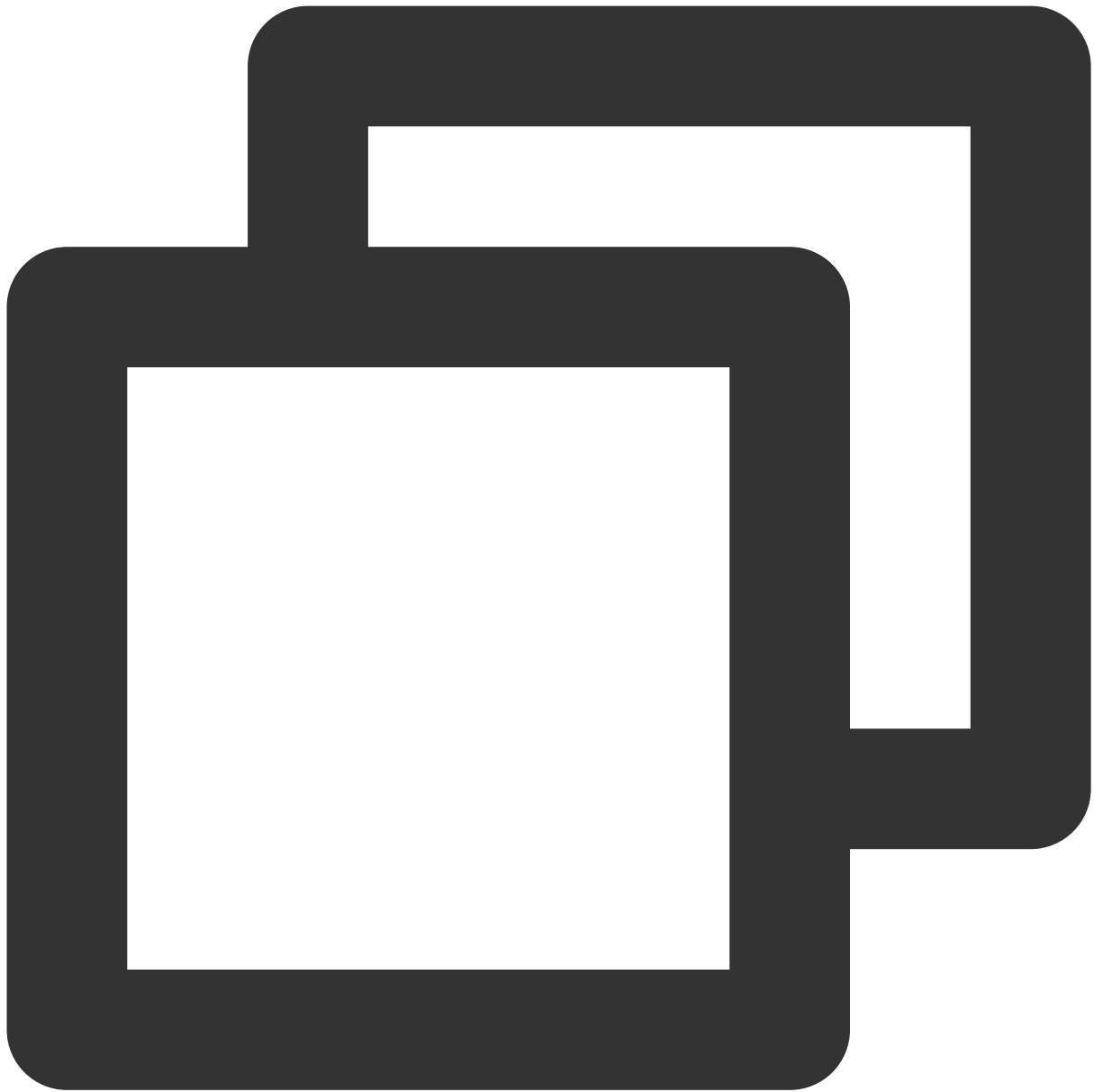
If a similar error occurs, fix it as follows:

1. Log in to the target database.
2. Modify the `max_allowed_packet` parameter.



```
set global max_allowed_packet = 4*1024*1024;
```

3. Check whether the configuration takes effect.



```
show global variables like '%max_allowed_packet%';
```

The system will display a result similar to the following:



```
mysql> show global variables like '%max_allowed_packet%';
+-----+
| Variable_name      | Value      |
+-----+
| max_allowed_packet | 4194304    |
+-----+
1 row in set (0.00 sec)
```

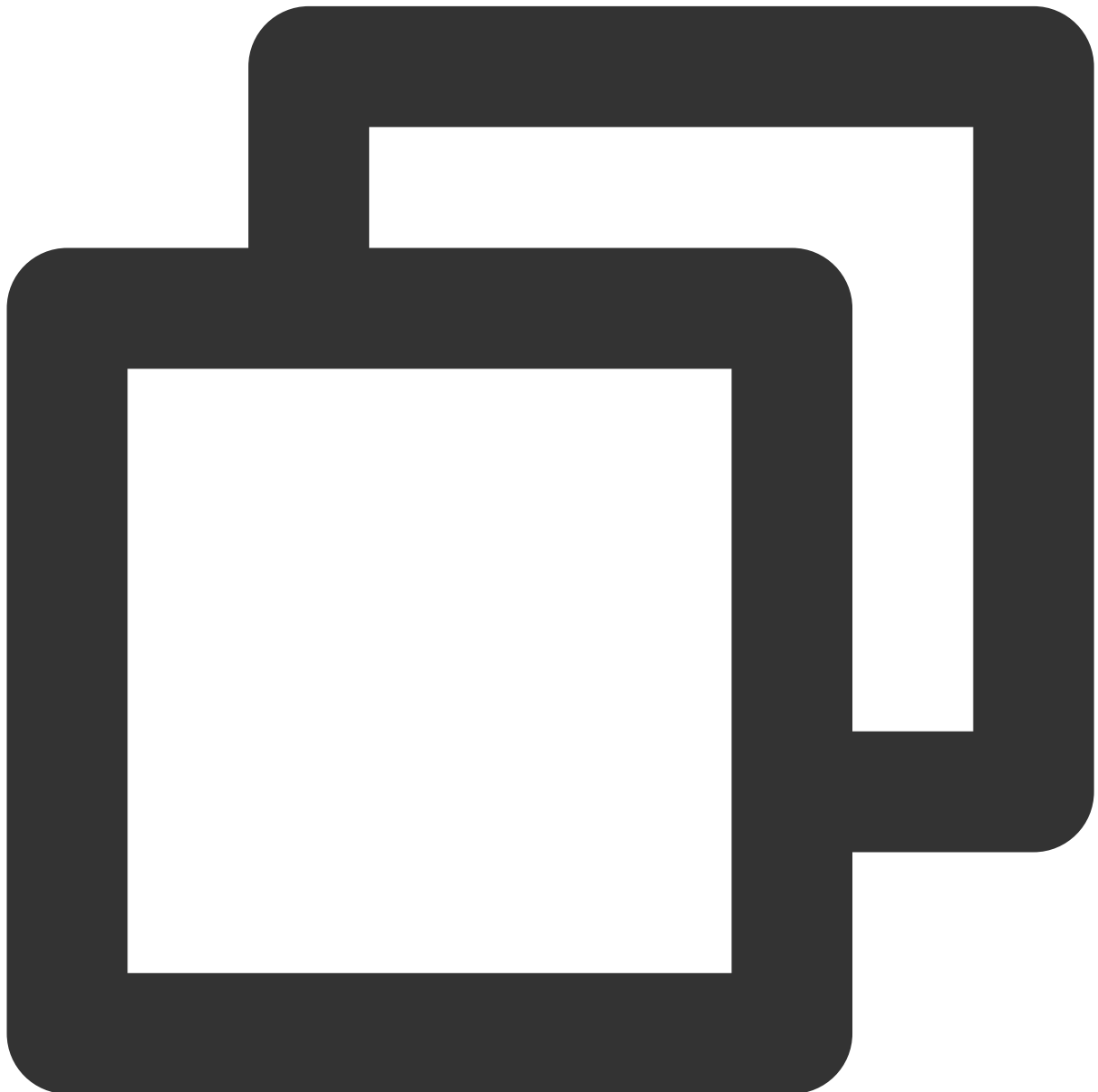
4. Run the verification task again.

Modifying the `connect_timeout` variable in the source database

`connect_timeout` is the database connection timeout, and a connection request will be denied if the connection duration is greater than the value of `connect_timeout`. If the value of `connect_timeout` is too small, the database will be frequently disconnected, which will impact the database processing efficiency. Therefore, we recommend that you set a value greater than 10 for this parameter.

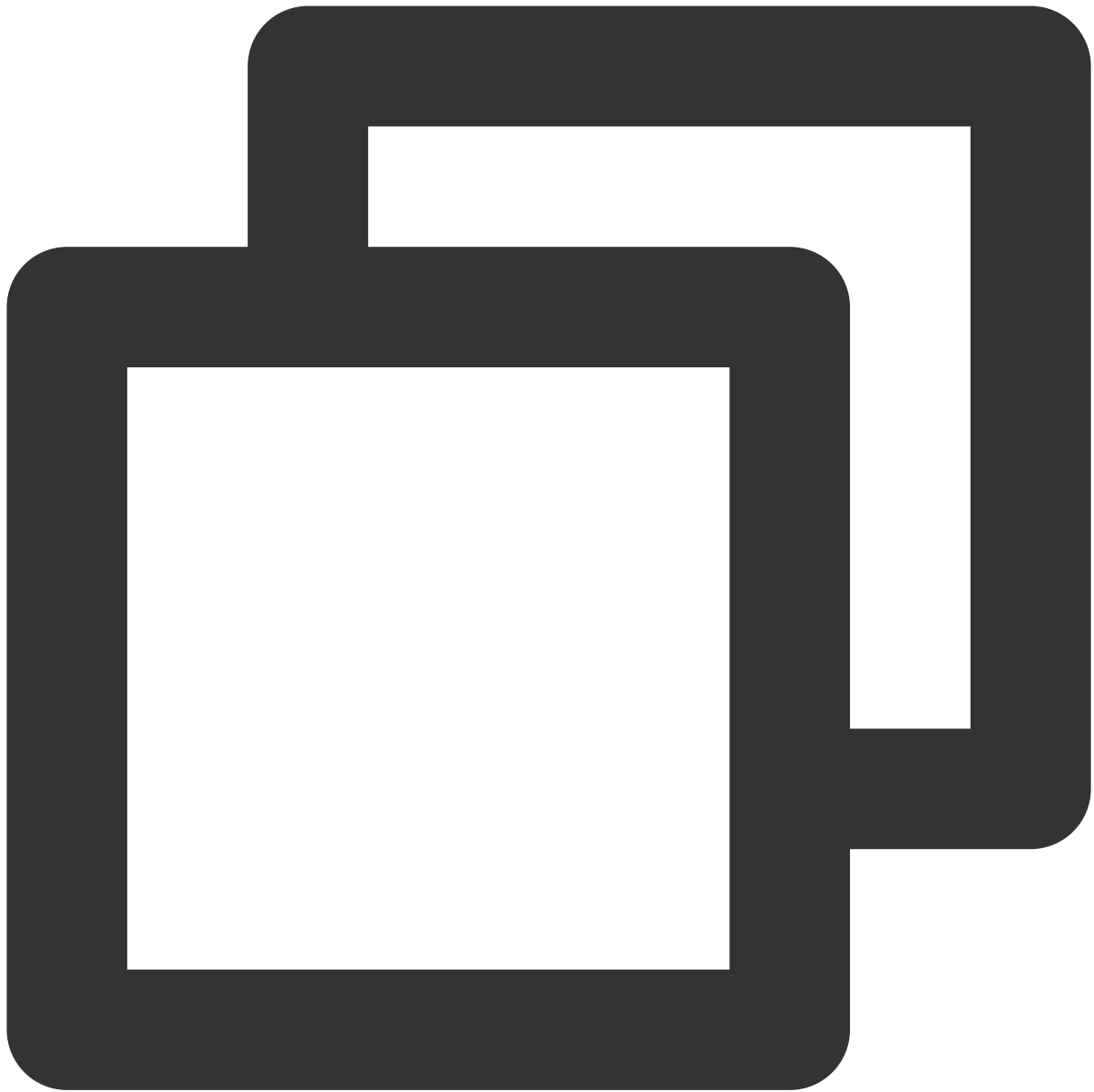
If a similar error occurs, fix it as follows:

1. Log in to the source database.
2. Modify the `connect_timeout` parameter.



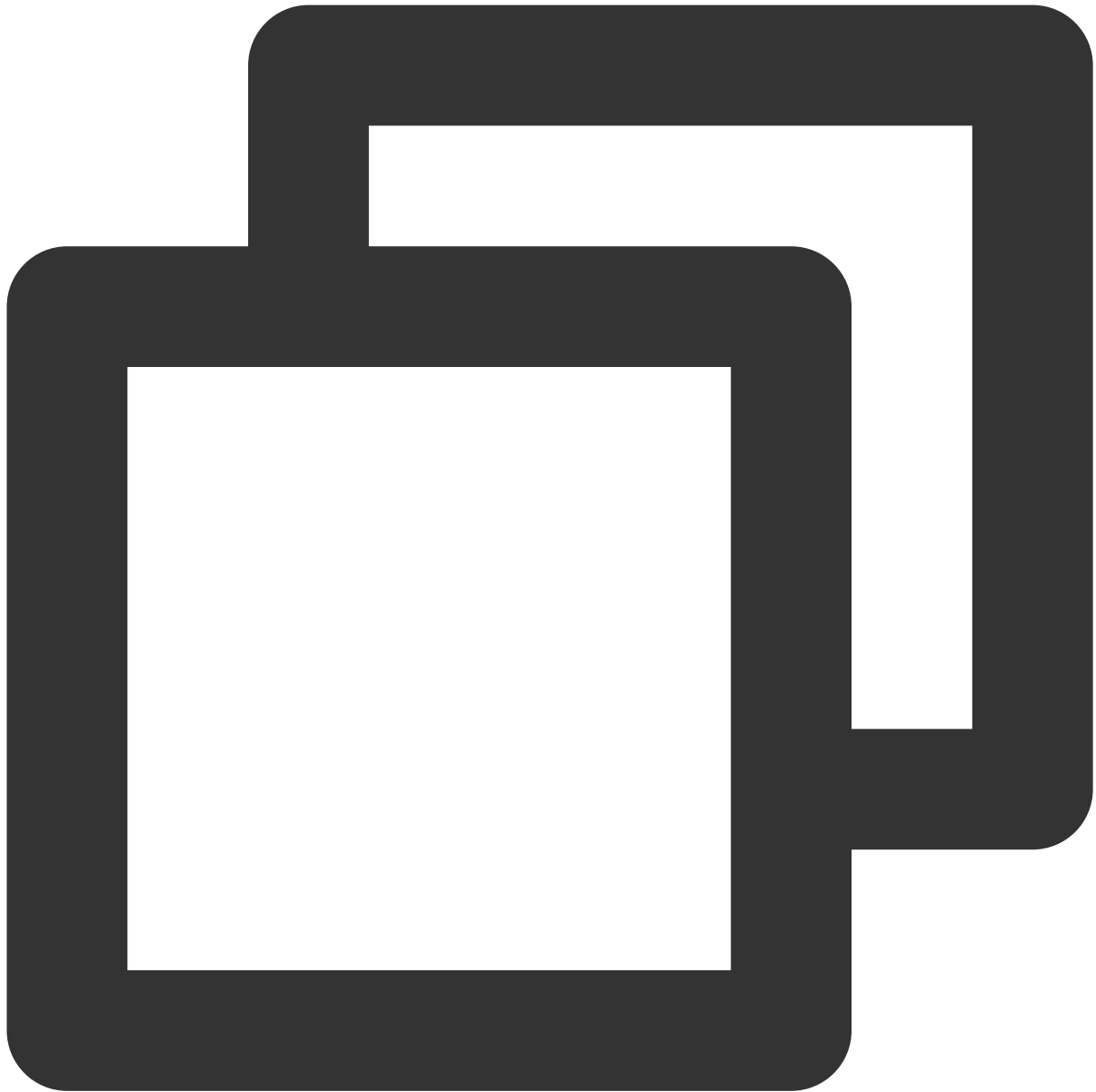
```
set global connect_timeout = 10;
```

3. Check whether the configuration takes effect.



```
show global variables like '%connect_timeout%';
```

The system will display a result similar to the following:



```
mysql> show global variables like '%connect_timeout%';
```

```
+-----+
| Variable_name      | Value  |
+-----+
| connect_timeout    | 10     |
+-----+
1 row in set (0.00 sec)
```

4. Run the verification task again.

Target Instance Permission Check

Last updated : 2024-07-11 15:03:48

Check Details

1. Check whether the user has permission to operate the database. In migration/synchronization/subscription processes, each database type has detailed permission requirements. See corresponding guidance.
2. For the target end being MySQL versions 8.0.26 and later, DTS will check the `innodb_strict_mode` parameter. It is recommended that `innodb_strict_mode` be set to OFF during migration/synchronization tasks. If it is not set to OFF, DTS will give a prompt during the pre-verification. For MySQL versions earlier than 8.0.26 or other database types, DTS will modify this parameter at the session level to OFF.

Parameter Description: `innodb_strict_mode` controls whether the InnoDB storage engine enables strict mode for improper data.

`innodb_strict_mode=ON`: The InnoDB storage engine will perform strict data validation. If there are syntax errors during operations like CREATE TABLE, ALTER TABLE, and CREATE INDEX, it will throw an error directly.

`innodb_strict_mode=OFF`: The InnoDB storage engine relaxes data validation. The above syntax errors will not trigger an error, and default syntax will be used instead.

Setting recommendation: If the source database contains improper data, it is recommended that the `innodb_strict_mode` parameter be set to OFF during migration/synchronization tasks. This will allow data to be synchronized normally to the target.

Fixing Solution

If the user does not have the required permissions, grant the corresponding permissions as specified in the check requirements, then re-run the verification task.

Target Database Content Conflict Check

Last updated : 2024-07-08 19:41:52

Check Details

The target instance cannot contain objects with the same name as those in the source database. If a conflict causes an error, troubleshoot by using any of the following methods.

Option 1: [Use database/table mapping](#).

Option 2: [Rename or delete objects with the same name in the target database](#).

Option 3: [Remove objects with the same name from the migration objects](#).

If an entire instance is migrated, the target instance must be empty. If a conflict causes an error, you need to delete the instance content.

If advanced objects are selected, the target database cannot contain conflicted advanced objects. If a conflict causes an error, you need to delete the conflicted objects.

Troubleshooting

Using database/table mapping for MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL only)

You can use the DTS table mapping feature to map the names of the objects to be migrated with the same name to other names in the target database.

1. Log in to the [DTS console](#), select the migration task, and click **More > Modify** in the **Operation** column.
2. In **Selected Object** on the right, hover over an object to be modified, click the displayed **Edit** icon, and rename the object.
3. Run the verification task again.

Modifying objects with the same name in target database

Log in to the target database and rename or delete the objects with the same name as the migration objects.

Removing objects with the same name from migration objects

Modify the migration task configuration to remove the objects with the same name from the migration objects. The removed objects cannot be migrated.

1. Log in to the [DTS console](#), select the migration task, and click **More > Modify** in the **Operation** column.
2. Remove the objects with the same name from the migration objects.
3. Run the verification task again.

Deleting target database content

Log in to the target database, delete the objects with the same name as those in the source database or the entire database, and run the verification task again.

Target Database Space Check

Last updated : 2024-07-08 19:41:52

Check Details

The storage space of the target database must be at least 1.2 times the size of the tables to be migrated in the source database.

Full data migration will execute INSERT operations concurrently, generating data fragments in some tables of the target database. Therefore, after full migration is completed, the storage space of the tables in the target database may be larger than that in the source instance.

Troubleshooting

Delete some data from the target database to free up sufficient space.

Upgrade the storage specification of the target database to use an instance with a larger capacity for migration. For more information, see [Adjusting Database Instance Specifications](#).

Binlog Parameter Check

Last updated : 2024-07-08 19:41:52

Check Details

You need to configure the source database's binlog parameters in compliance with the following requirements. If the verification fails, fix it as instructed in this document.

The `log_bin` variable must be set to `ON`.

The `binlog_format` variable must be set to `ROW`.

`binlog_row_image` must be set to `FULL`.

If the source database is MySQL 5.6 or later, `gtid_mode` can only be set to `ON` or `OFF`. We recommend that you set it to `ON`, because if it is set to `OFF`, an alarm will be triggered, and if it is set to `ON_PERMISSIVE` or `OFF_PERMISSIVE`, an error will be reported.

The `server_id` parameter must be set manually and cannot be `0`.

It is not allowed to set `do_db` and `ignore_db`.

If the source database is a replica database, the `log_slave_updates` variable must be set to `ON`.

We recommend that you retain the binlog of the source database for at least three days; otherwise, the task cannot be resumed from the checkpoint and will fail.

Troubleshooting

Enabling binlog

`log_bin` controls the binlog switch. You need to enable binlog to log all database table structure and data changes.

If a similar error occurs, fix it as follows:

1. Log in to the source database.
2. Modify the `my.cnf` configuration file of the source database as follows:

The modification of the `log_bin` parameter only takes effect after the database is restarted. Therefore, if errors are also reported for the `binlog_format`, `server_id`, `binlog_row_image` and `expire_logs_days` parameters in the verification stage, we recommend that you modify all these parameters before restarting the database so that all the modifications can take effect.

Note

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.



```
log_bin = MYSQL_BIN
binlog_format = ROW
server_id = 2      //We recommend that you set it to an integer above 1. The value h
binlog_row_image = FULL
expire_logs_days = 3    //Modify the binlog retention period (at least 3 days pref
```

3. Run the following command to restart the source database:

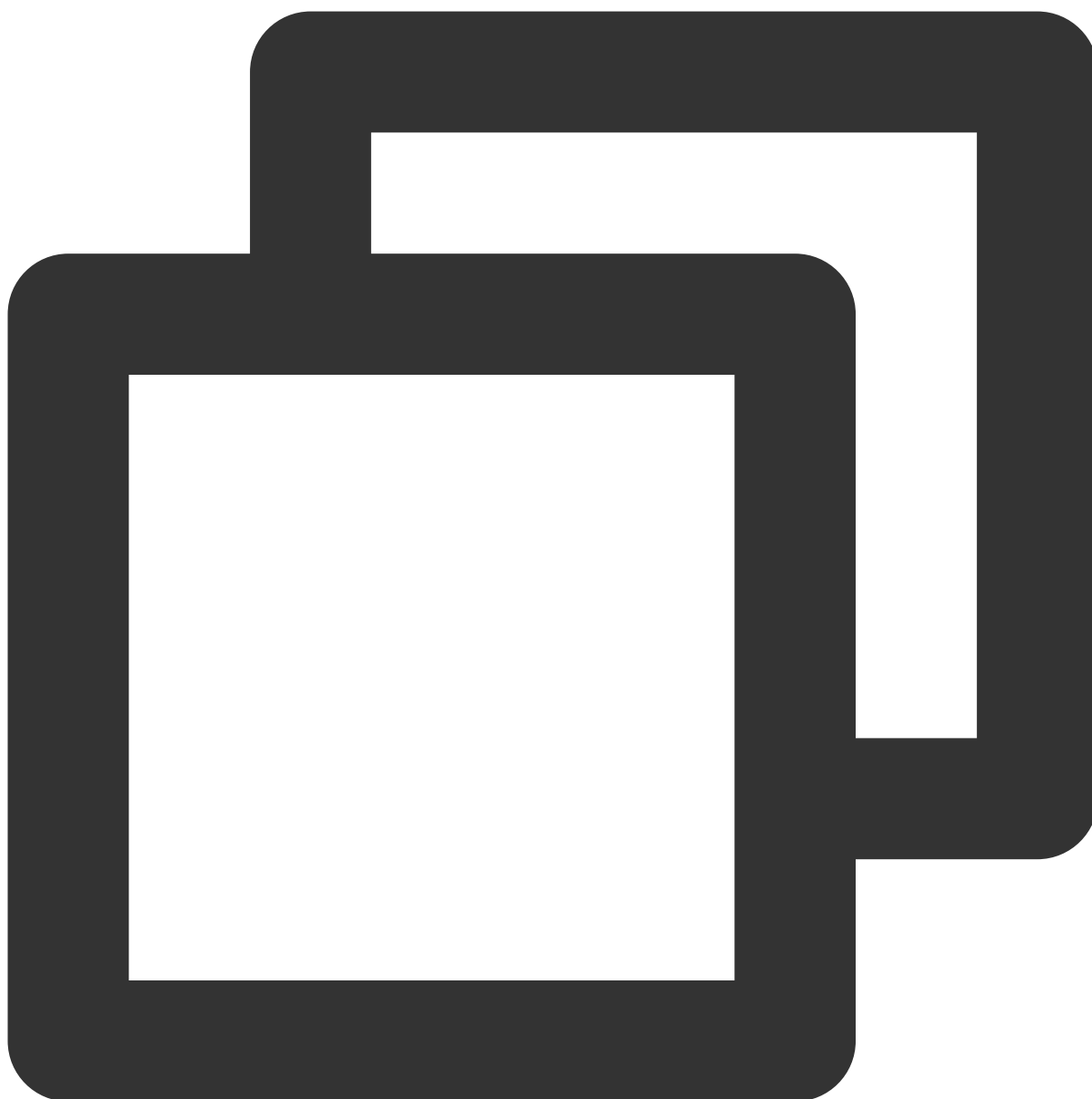


```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown  
[$Mysql_Dir]/bin/safe_mysqld &
```

Note

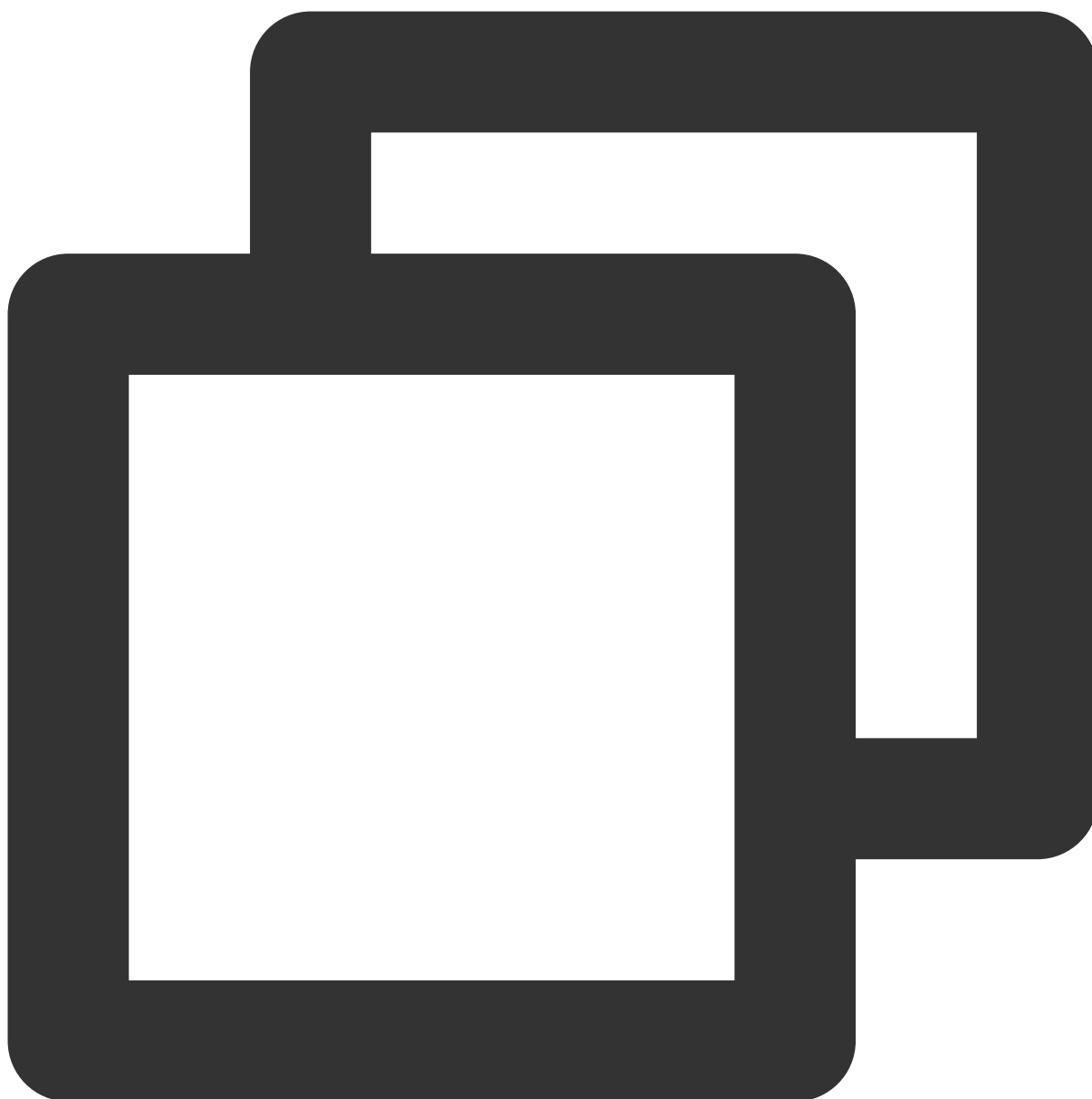
[\$Mysql_Dir] is the installation path of the source database. Replace it with the actual path.

4. Check whether the binlog feature has been enabled.



```
show variables like '%log_bin%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%log_bin%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON    |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
| binlog_row_image | FULL |
+-----+-----+
1 row in set (0.00 sec)
```

5. Run the verification task again.

Modifying `binlog_format` parameter

`binlog_format` specifies one of the following three binlog formats:

STATEMENT : Each SQL statement that modifies the data will be logged into the binlog of the source/primary database. When replicating data, the replica/secondary database will run the same SQL statements as those in the source/primary database. This format can reduce the binlog size. However, the replica/secondary database may not be able to properly replicate certain functions.

ROW : The binlog will log the modification of each data row, and the replica/secondary database will modify the same data. This format guarantees the correct source-replica or primary-secondary replication, but the binlog size will increase.

MIXED : It is a combination of the above two formats. MySQL will automatically select **STATEMENT** or **ROW** format to log each executed SQL statement.

Therefore, to ensure the correct source-replica or primary-secondary replication, the `binlog_format` parameter should be set to **ROW** . If a similar error occurs, fix it as follows:

Note

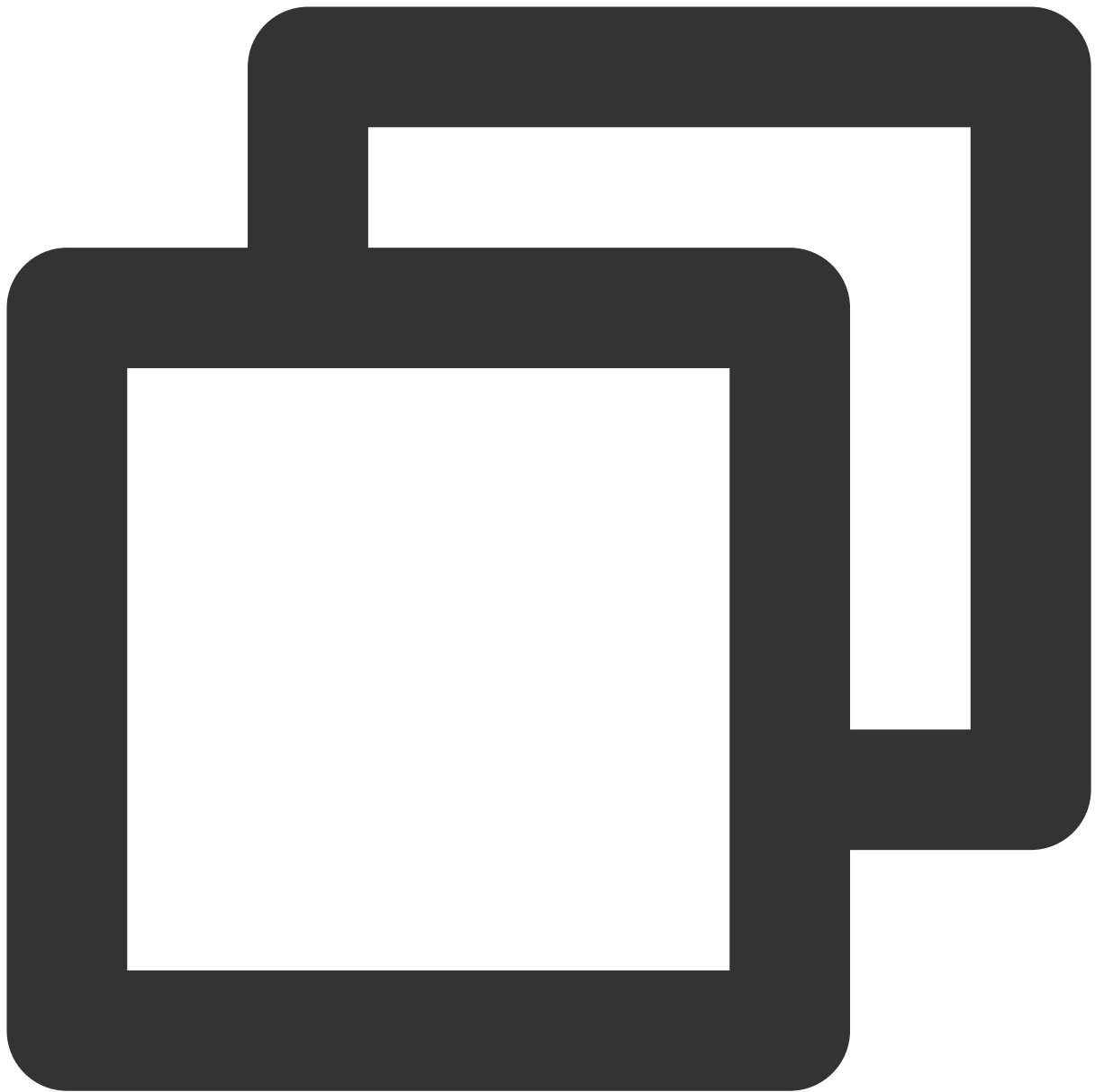
Changes to this parameter can only take effect after all connections to the database are reset. If the source database is a replica/secondary database, you also need to restart the source-replica or primary-replica sync SQL thread to prevent current business connections from continuing writing data in the mode before modification.

1. Log in to the source database.
2. Run the following command to modify `binlog_format` .



```
set global binlog_format = ROW;
```

3. Restart the thread for the configuration to take effect. Then, run the following command to check whether the parameter modification takes effect:



```
show variables like '%binlog_format%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%binlog_format%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
1 row in set (0.00 sec)
```

4. Run the verification task again.

Modifying `binlog_row_image` parameter

The `binlog_row_image` parameter determines how the binlog logs the pre-image (content before modification) and post-image (content after modification), which directly affects features such as data flashback and source-replica or primary-replica replication.

The `binlog_row_image` parameter takes effect only if `binlog_format` is set to `ROW`. The following describes the effects of specific values:

`FULL` : In `ROW` format, binlog will log all the pre-image and post-image column data information.

`MINIMAL` : In `ROW` format, if a table has no primary key or unique key, the pre-image will log all columns, and the post-image will log the modified columns. If it has a primary key or unique key, both the pre-image and post-image will only log the affected columns.

Therefore, you need to set `binlog_row_image` to `FULL` to make the source database binlog log the full image. If an error occurs, troubleshoot as follows:

Note

Changes to this parameter can only take effect after all connections to the database are reset. If the source database is a replica/secondary database, you also need to restart the source-replica or primary-replica sync SQL thread to prevent current business connections from continuing writing data in the mode before modification.

1. Log in to the source database.
2. Run the following command to modify `binlog_row_image` :



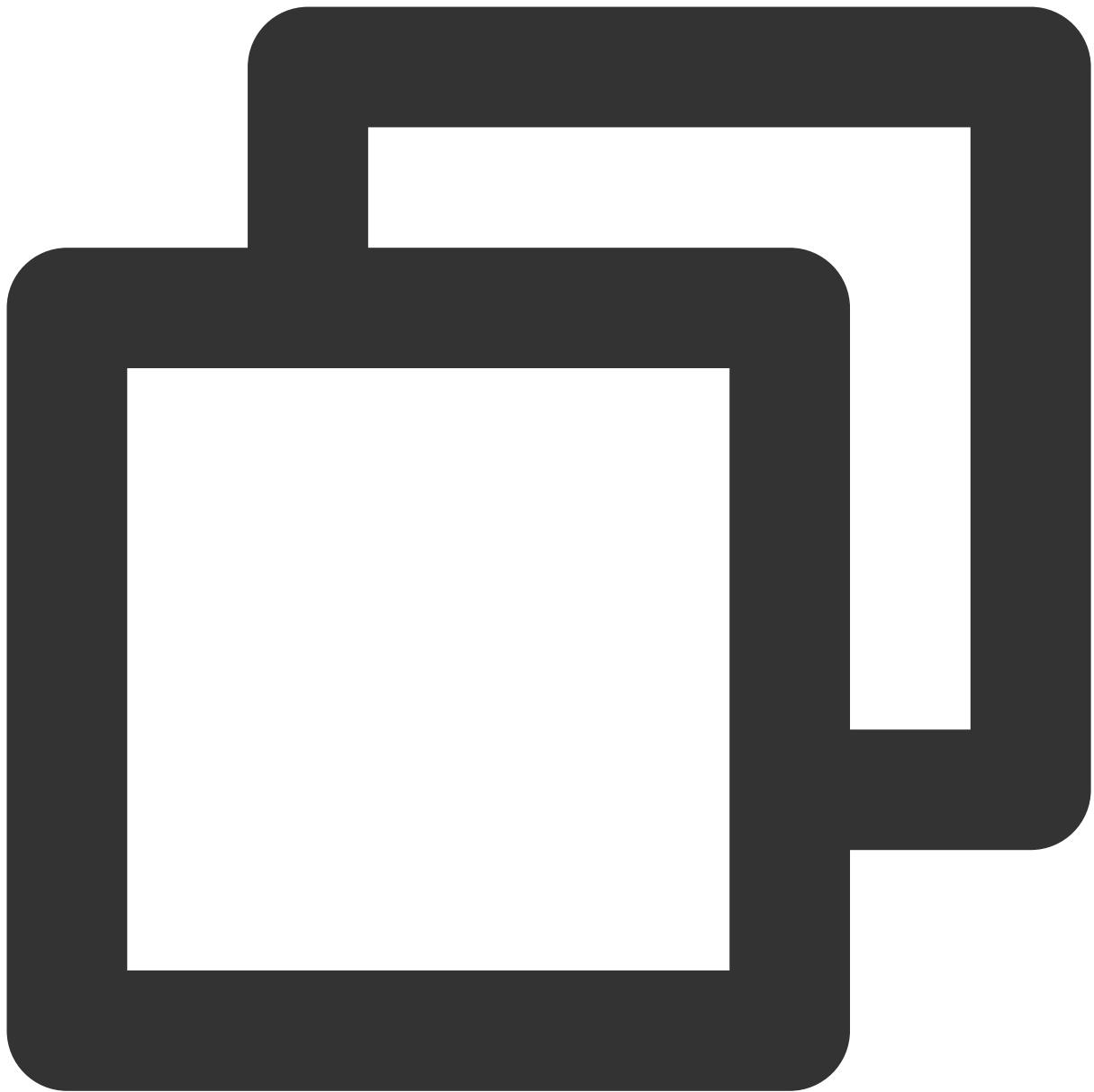
```
set global binlog_row_image = FULL;
```

3. Restart the thread for the configuration to take effect. Then, run the following command to check whether the parameter modification takes effect:



```
show variables like '%binlog_row_image%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%binlog_row_image%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_row_image | FULL |
+-----+-----+
1 row in set (0.00 sec)
```

4. Run the verification task again.

Modifying `gtid_mode` parameter

A global transaction identifier (GTID) uniquely identifies a transaction in the binlog. Using GTIDs can prevent disordered data or source-replica or primary-replica inconsistency due to repeated transaction executions. GTID is a new feature on MySQL 5.6. Therefore, this problem may only occur on MySQL 5.6 or later versions. DTS only allows you to set `gtid_mode` to `ON` or `OFF`. We recommend that you set it to `ON`; otherwise, an alarm will be triggered during verification.

The alarm does not affect the migration or sync task but affects the business. After GTID is set, if HA switch occurs in the source database during incremental data sync, DTS will be switched and restarted, which is almost imperceptible to the task; if GTID is not set, the task will fail after disconnection and cannot be resumed.

Below are the valid values of `gtid_mode`. When modifying the value, you can only do so in the specified sequence step by step; for example, if you want to change `OFF` to `ON`, you should modify the `gtid_mode` value in the following sequence: `OFF` <-> `OFF_PERMISSIVE` <-> `ON_PERMISSIVE` <-> `ON`.

`OFF`: All new transactions in the source/primary database and all transactions in the replica/secondary database must be anonymous.

`OFF_PERMISSIVE`: All new transactions in the source/primary database must be anonymous. Transactions in the replica/secondary database can be anonymous or GTID transactions but cannot only be GTID transactions.

`ON_PERMISSIVE`: All new transactions in the source/primary database must be GTID transactions, and transactions in the replica/secondary database can be anonymous or GTID transactions.

`ON`: All new transactions in the source/primary database and all transactions in the replica/secondary database must be GTID transactions.

If a similar alarm is triggered, fix it as follows:

1. Log in to the source database.
2. Set `gtid_mode = OFF_PERMISSIVE` on the source/primary and replica databases.

On MySQL versions earlier than v5.7.6, you need to modify the parameter in the `my.cnf` configuration file and restart the database to make the change take effect. On v5.7.6 and later, you can modify the parameter through global naming without restarting the database, but you must reset all business connections.



```
set global gtid_mode = OFF_PERMISSIVE;
```

3. Set `gtid_mode = ON_PERMISSIVE` on the source/primary and replica databases.



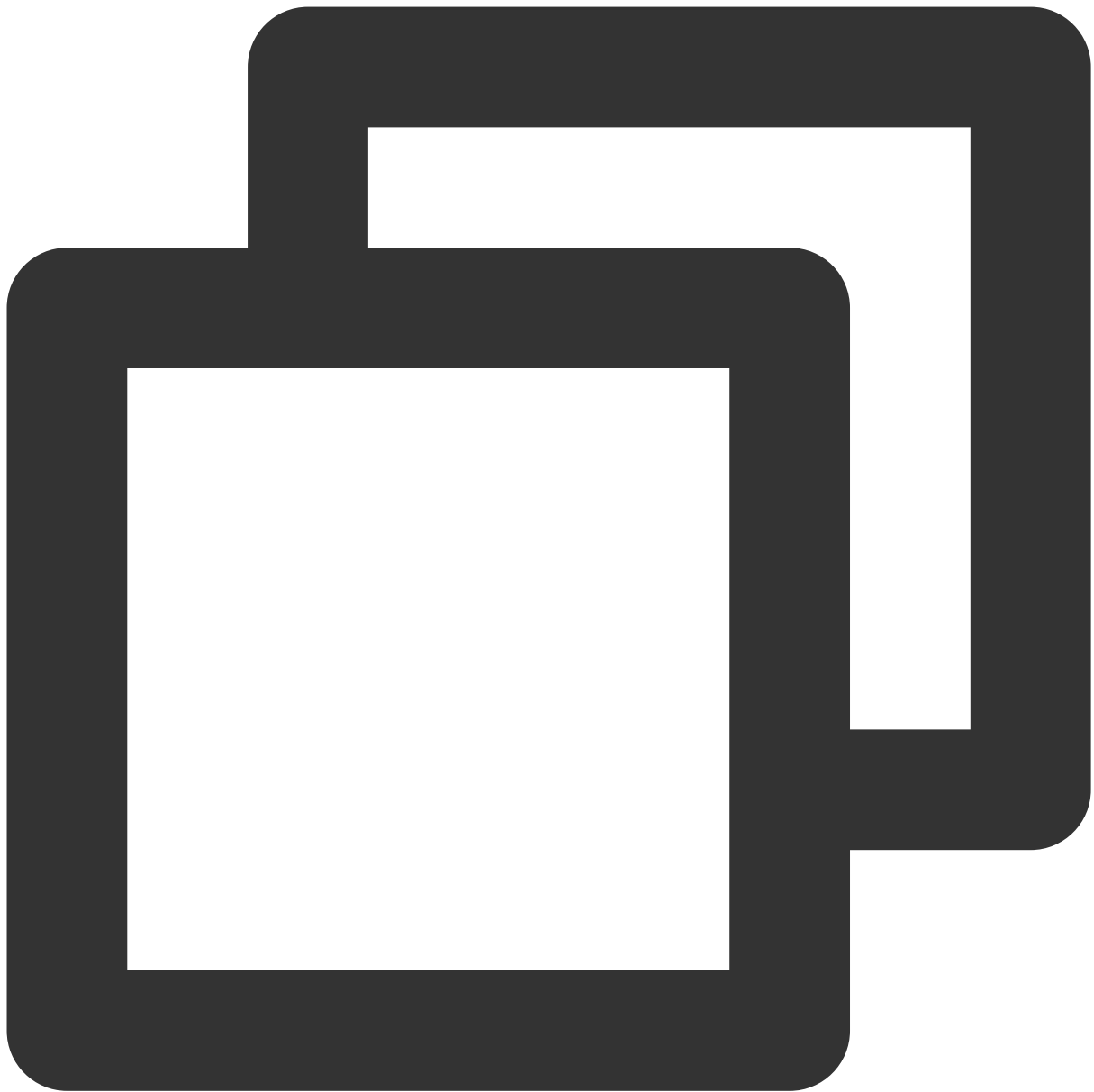
```
set global gtid_mode = ON_PERMISSIVE;
```

4. Run the following command on each instance node to check whether consumption of anonymous transactions is completed. If the parameter value is `0`, the consumption is completed.



```
show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
```

```
+-----+-----+
| Variable_name                | Value |
+-----+-----+
| Ongoing_anonymous_transaction_count | 0      |
+-----+-----+
1 row in set (0.00 sec)
```

5. Set `gtid_mode = ON` on the source/primary and replica databases.



```
set global gtid_mode = ON;
```

6. Add the following content to the `my.cnf` file and restart the database to make the initial values take effect.

Note

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.



```
gtid_mode = on
enforce_gtid_consistency = on
```

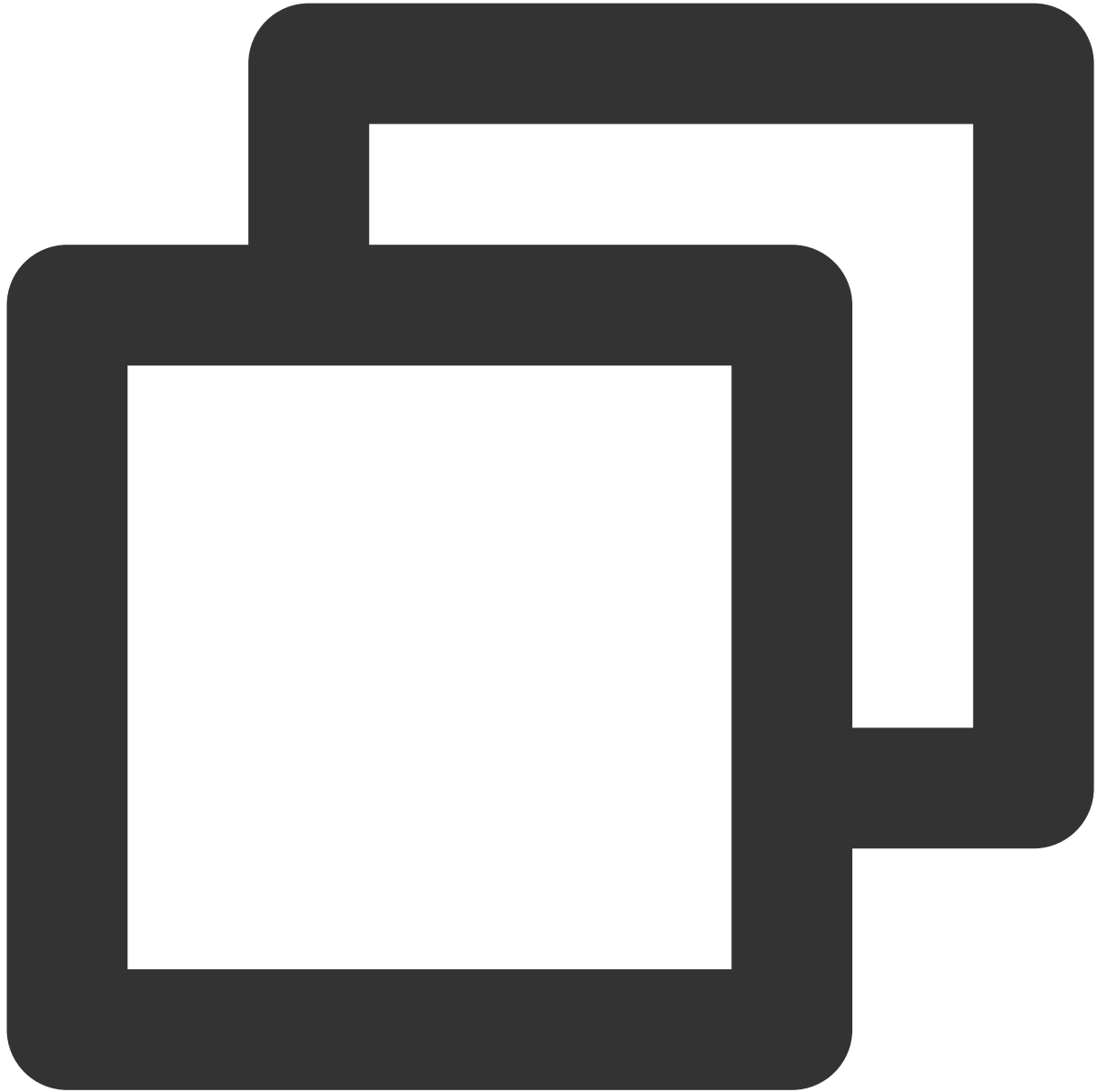
7. Run the verification task again.

Modifying `server_id` parameter

The `server_id` parameter must be set manually and cannot be `0`. The default value of this parameter is `1`, but the configuration may not be correct even if the queried parameter value is `1`. You still need to set it manually.

1. Log in to the source database.

2. Run the following command to modify `server_id` :



```
set global server_id = 2;  // We recommend that you set it to an integer above 1. T
```

3. Run the following command to check whether the parameter modification takes effect:



```
show global variables like '%server_id%';
```

The system will display a result similar to the following:



```
mysql> show global variables like '%server_id%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id     | 2     |
+-----+-----+
1 row in set (0.00 sec)
```

4. Run the verification task again.

Deleting `do_db` and `ignore_db` settings

The binlog logs all executed DDL and DML statements in the database, while `do_db` and `ignore_db` are used to set the filter conditions for binlog.

`binlog_do_db` : Only the specified databases will be logged in the binlog (all databases will be logged by default).

`binlog_ignore_db` : The specified databases will not be logged in the binlog.

After `do_db` and `ignore_db` are set, some cross-database operations will not be logged in the binlog, and source-replica or primary-replica replication will be abnormal; therefore, this setting is not recommended. If a similar error occurs, fix it as follows:

1. Log in to the source database.
2. Modify the `my.cnf` configuration file in the source database to delete `do_db` and `ignore_db` settings.

Note

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.

3. Run the following command to restart the source database:



```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown  
[$Mysql_Dir]/bin/safe_mysqld &
```

Note

[\$Mysql_Dir] is the installation path of the source database. Replace it with the actual path.

4. Check whether the parameter modification takes effect.



```
show master status;
```

The system will display a result similar to the following:



```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
binlog.000011	154			

5. Run the verification task again.

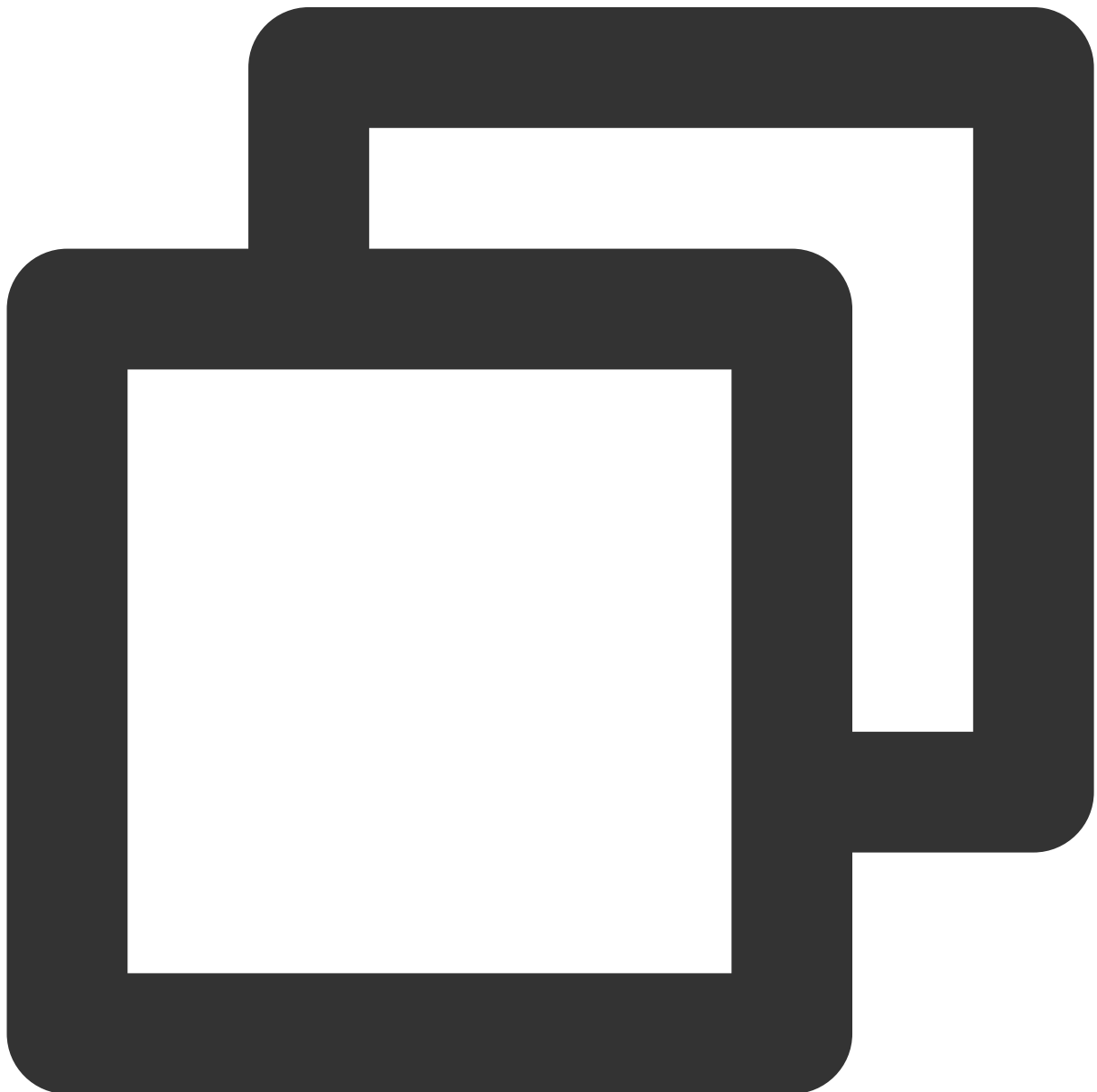
Modifying `log_slave_updates` **parameter**

In the source-replica or primary-replica reuse structure, if the `log-bin` parameter is enabled in the replica/secondary database, data operations directly performed in this database can be logged in the binlog, but data replications from the source/primary database to the replica/secondary database cannot be logged. Therefore, if the replica/secondary database is to be used as the source/primary database of another database, the `log_slave_updates` parameter needs to be enabled.

1. Log in to the source database.
2. Add the following content to the `my.cnf` configuration file of the source database.

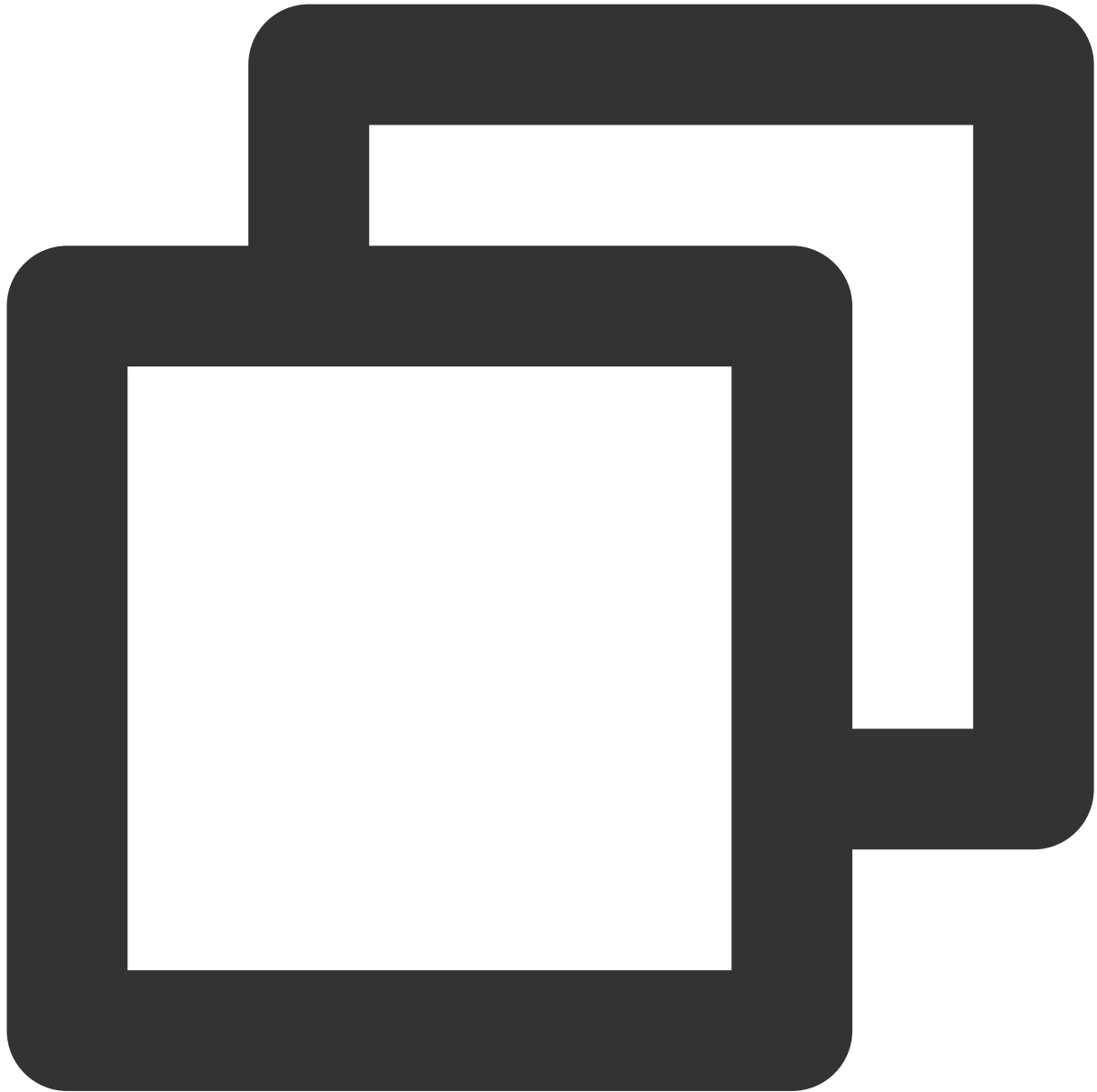
Note

The default path of the `my.cnf` configuration file is `/etc/my.cnf`, subject to the actual conditions.



```
log_slave_updates = ON
```

3. Run the following command to restart the source database:



```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown  
[$Mysql_Dir]/bin/safe_mysqld &
```

Note

[\$Mysql_Dir] is the installation path of the source database. Replace it with the actual path.

4. Check whether the configuration takes effect.



```
show variables like '%log_slave_updates%';
```

The system will display a result similar to the following:



```
mysql> show variables like '%log_slave_updates%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_slave_updates | ON    |
+-----+-----+
1 row in set (0.00 sec)
```

5. Run the verification task again.

Foreign Key Dependency Check

Last updated : 2024-07-08 19:41:52

Check Details

Data migration among MySQL, MariaDB, Percona, and TDSQL-C for MySQL: Foreign key dependency check can be set to `CASCADE` , `SET NULL` , `RESTRICT` , `NO ACTION` , or `SET DEFAULT` . You need to enable this check item first.

Data sync among MySQL, MariaDB, Percona, and TDSQL-C for MySQL: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT` ; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task.

TDSQL for MySQL data migration and sync: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT` ; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task.

TDSQL for TDStore data migration: Foreign key-dependent data is not supported. If the source database has such data, the task verification will report an error.

During partial table migration, tables with foreign key dependency must be migrated.

Foreign Key Dependency Parameters

When you set a foreign key in MySQL, there are four values that can be selected for the `ON DELETE` and `ON UPDATE` columns:

`CASCADE` : When a record is deleted or updated in the parent table, its associated records will also be deleted or updated in the child table.

`SET NULL` : When a record is deleted or updated in the parent table, the column of the foreign key field of its associated records will be set to `null` in the child table (child table foreign keys cannot be set to `not null`).

`RESTRICT` : When a record is deleted or updated in the parent table, if it is associated with records in the child table, the deletion request in the parent table will be denied.

`NO ACTION` : Similar to `RESTRICT` , the foreign key will be checked first.

`SET DEFAULT` : When a record is deleted or updated in the parent table, the foreign key column in the child table will be set to the default value, but the InnoDB engine cannot recognize it.

Ignoring Errors or Enabling Foreign Key Dependency Migration

For data migration among MySQL, MariaDB, Percona, and TDSQL-C for MySQL, foreign key dependency check can be set to `CASCADE` , `SET NULL` , `RESTRICT` , `NO ACTION` , or `SET DEFAULT` .

DTS supports `RESTRICT` and `NO ACTION` by default. If the data migrated from the source database has other types of foreign key dependency configuration, the check system will report an error, and you need to enable foreign key dependency check as a check item as follows:

1.1 On the **Verify task** page during data migration, the source database has foreign key dependency rules `CASCADE` , `SET NULL` , and `SET DEFAULT` , and an error is reported for the check item. You can view the details of the check item.



Check Item	Result
创建校验任务	
查询校验结果	
连接DB检查	通过
库边检查	通过
版本检查	通过
源实例权限检查	通过
部分实例参数检查	通过
目标实例权限检查	通过
目标实例内容冲突检查	通过
目标实例空间检查	通过
binlog参数检查	通过
外键依赖检查	失败 查看详情
外键部分库表依赖检查	通过
视图检查	通过
警告项检查	警告 查看详情
高级对象检查	通过

2. Choose to ignore the error or migrate the foreign key dependency. After confirming your selection, indicate your consent and click **OK**.

Option	Description
Ignore foreign key dependency check error	<p>After this option is selected, the system will ignore existing check errors and proceed with the migration task.</p> <p>If the source database has foreign key rules <code>CASCADE</code> , <code>SET NULL</code> , or <code>SET DEFAULT</code> , the data can still be migrated, but the data results between the source and target databases may be inconsistent if data in the parent table is deleted or updated after errors are ignored.</p>
Migrate foreign key dependency	<p>After this option is selected, <code>CASCADE</code> , <code>SET NULL</code> , and <code>SET DEFAULT</code> foreign key dependencies can be migrated, but DTS will check the foreign key dependencies only when a task is initiated to guarantee the consistency of the existing data migrated to the target database.</p> <p>1. During migration, do not modify the foreign key dependency check rule in the source database; otherwise, the data in the source and target databases will become inconsistent.</p> <p>If you change the foreign key dependency check rule in the source database from <code>CASCADE</code> to <code>NO ACTION</code> during migration, updates and deletions performed on the parent table in the target database will still affect child tables.</p>

2. If the foreign key dependency check rule is `CASCADE` or `SET NULL`, after you select **Migrate foreign key dependency**, tables in the source database will be locked for about 5–10 seconds during full migration.
3. To guarantee the data consistency, during incremental sync, the original row-level sync of data in foreign key-related tables will be downgraded to table-level, and the performance will decrease.

校验详情

校验结果

失败

错误详情

只支持外键规则: `RESTRICT`, `NO ACTION`。subs_test.rooms 依赖 subs_test.buildings 的 update_rule: `NO ACTION`, delete_rule: `CASCADE` [帮助文档](#)

风险警告

打开外键依赖或屏蔽外键依赖迁移可能带来的风险

建议

方式一：屏蔽外键依赖检查报错

风险提示：屏蔽失败可能会因为父表进行删除或者更新时导致最终数据不一致

方式二：打开外键依赖关系迁移

风险提示：

1. DTS 仅在任务发起时检查外键依赖关系，若在迁移过程中修改外键依赖关系，将导致目标库数据与源库不一致
2. 在外键规则为 `CASCADE` 或 `SET NULL` 时，开启外键依赖后，在全量迁移阶段会对源库短暂加表锁，影响时间约 5-10 秒
3. 为了保证数据的一致性，在增量同步阶段，对外键相关表的数据传输将从行级同步降级为表级同步，性能会有所下降

外键依赖

屏蔽外键依赖检查报错

迁移外键依赖关系

DTS 默认支持 `RESTRICT`, `NO ACTION` 外键规则。选择迁移外键依赖关系后可

☒ 我已知晓并同意：打开外键依赖关系迁移可能会带来的风险

关闭

确认

3. Run the verification task again.

Fixing an Error

Data sync among MySQL, MariaDB, Percona, and TDSQL-C for MySQL: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT`; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task. You can fix the error as follows:

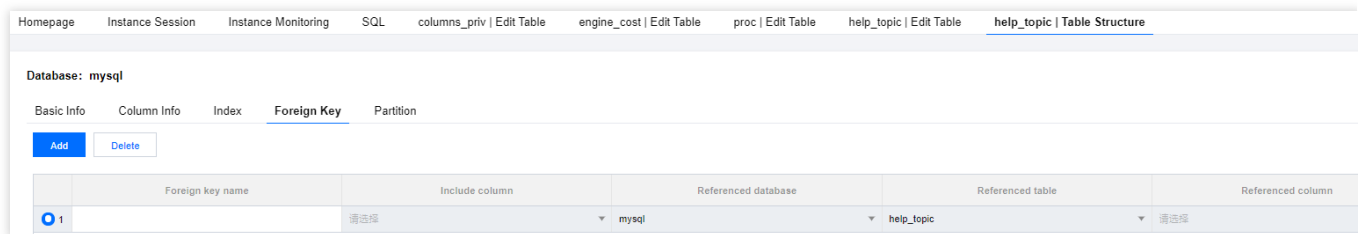
TDSQL for MySQL data migration and sync: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT`; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task. You can fix the error as follows:

TDSQL for MySQL (TDStore Edition): Foreign key-dependent data is not supported. You can proceed with the task only after deleting the corresponding foreign key parameter.

Modifying a foreign key rule

Windows

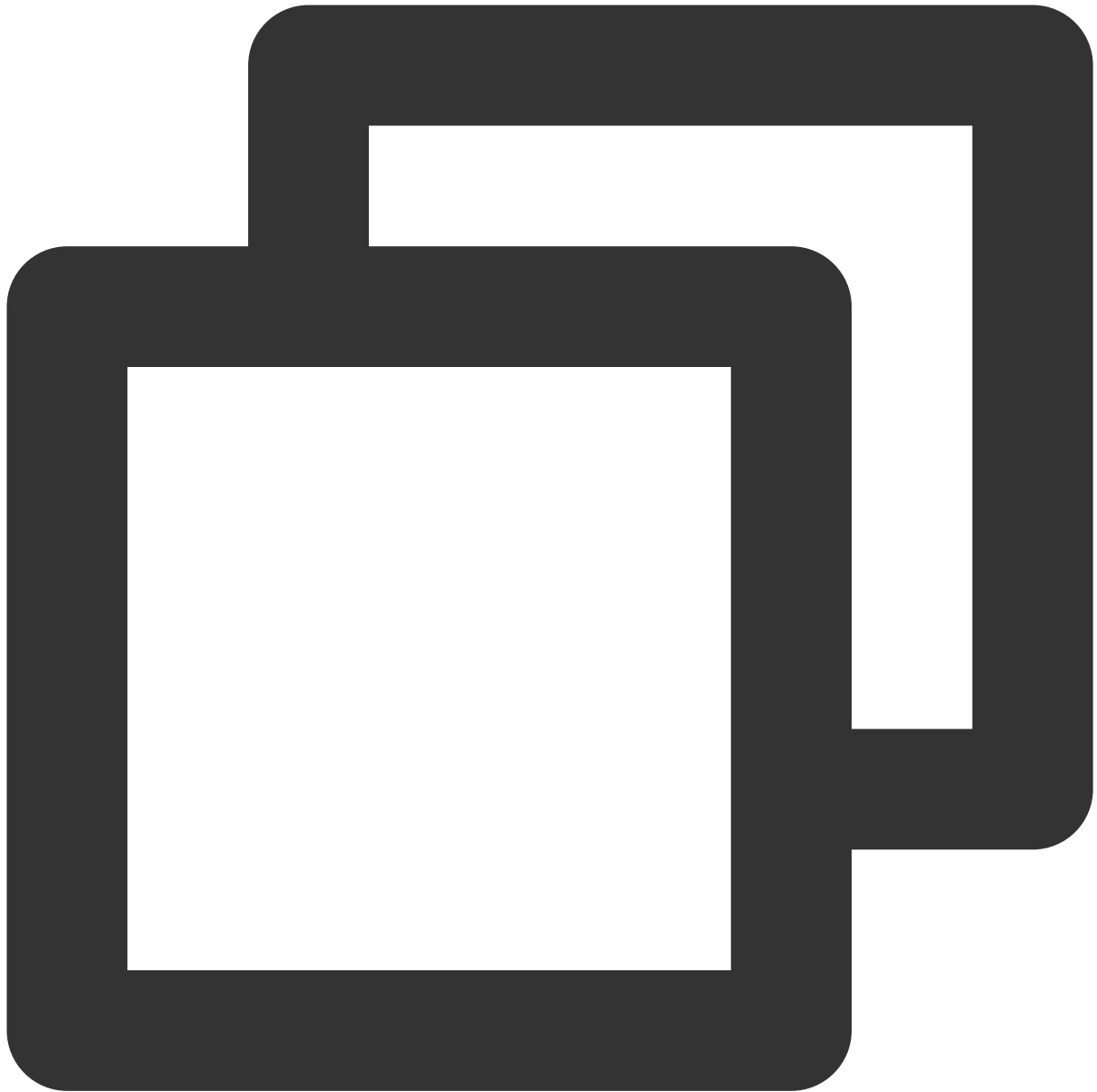
1. Log in to DMC in the source database as instructed in [DMC Management](#).
2. Select the table to be modified in the target tree on the left and click the **Foreign Key** tab on the opened table editing page to modify the foreign key parameter.



3. After completing the modification, click **Save**.
4. Run the verification task again.

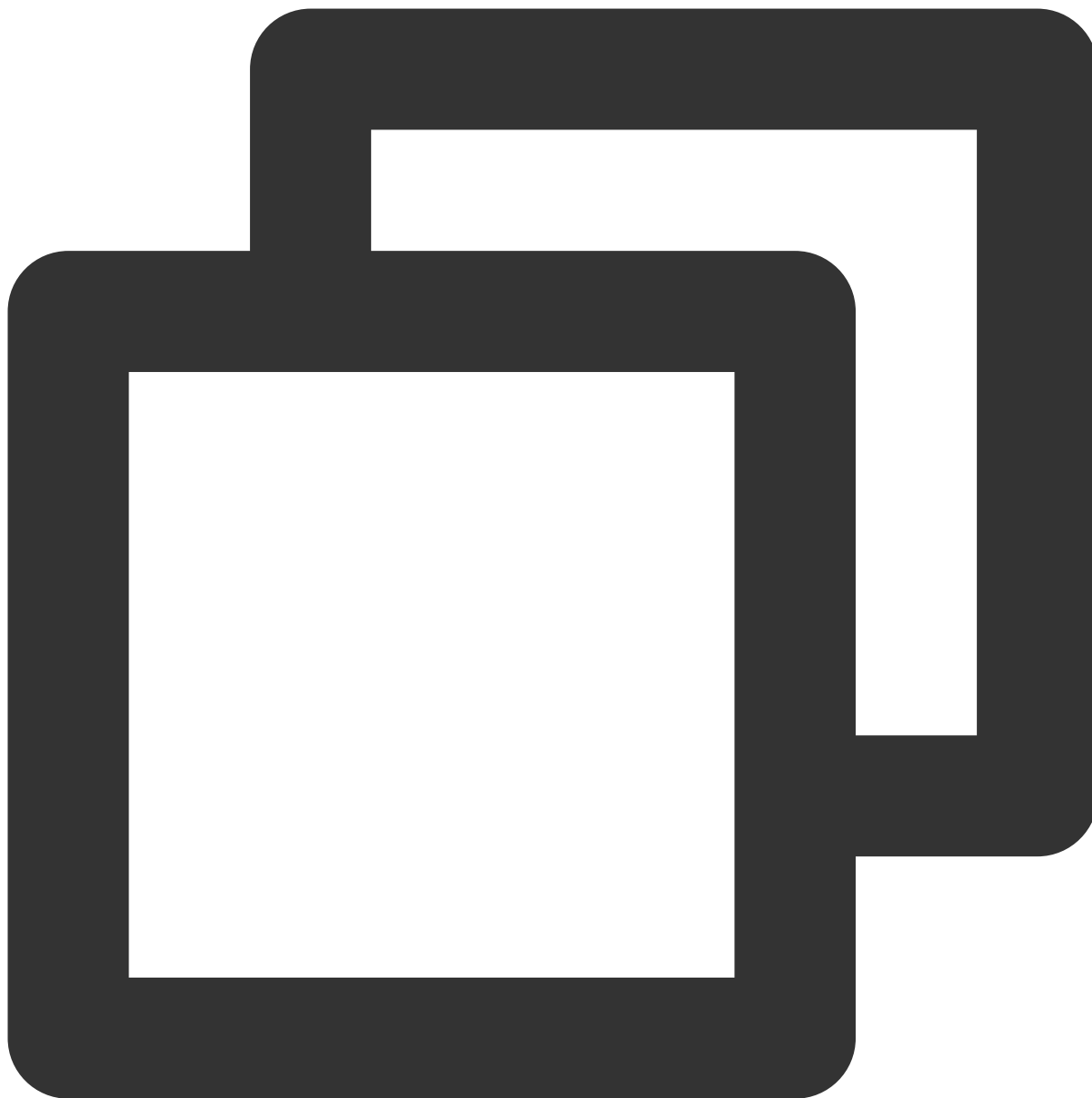
Linux

1. Log in to the source database as instructed in [Connecting to MySQL Instance](#).
2. Delete the original foreign key settings.



```
alter table `table name 1` drop foreign key `foreign key name 1`;
```

3. Add the foreign key settings again.



```
alter table `table name 1` add constraint `foreign key name 2` foreign key `table n  
on update no action on delete no action;
```

4. Run the verification task again.

Completing migration objects

When modifying the migration task configuration, include objects with associations in **Migration Object**.

1. Log in to the [DTS console](#), select the target migration task, and click **More > Modify** in the **Operation** column.
2. Select the objects with associations in **Migration Object**.

3. Run the verification task again.

View Check

Last updated : 2024-07-08 19:41:52

MySQL/TDSQL-C Check Details

Check requirements: when exporting a view structure, DTS will check whether `user1` corresponding to `DEFINER ([DEFINER = user1])` in the source database is the same as `user2` in the migration target.

If they are the same, do not modify the settings after migration.

If they are different, change the `SQL SECURITY` attribute of `user1` in the target database after migration from `DEFINER` to `INVOKER ([INVOKER = user1])`, and set the `DEFINER` in the target database to `user2` of the migration target (`[DEFINER = migration target user2]`).

Check description: the `SQL SECURITY` parameter indicates according to whose permissions the system runs the command when a user accesses the specified view.

`DEFINER` : only the definer can run the command.

`INVOKER` : only invokers with the invocation permissions can run the command.

By default, `DEFINER` is specified by the system.

TDSQL for MySQL Check Details

Only a definer that is the same as the migration target's `user@host` is allowed; that is, when a view structure is exported, DTS will check whether the `user1` corresponding to the definer in the source database (`[DEFINER = user1]`) is the same with the `user2` in the migration target's `user@host` , and if yes, the view can be migrated; otherwise, it cannot.

For a definer different from that of the migration target's `user@host` , if you want to migrate it, you need to modify the definer in the source database view to the migration target's user, or do not select it during the migration/sync task and then manually sync the view after the task is completed.

Advanced Object Check

Last updated : 2024-07-08 19:41:52

MySQL/MariaDB/Percona check details

If you select to migrate/sync advanced objects, DTS will verify the following content. You must fix errors to continue the task. For warnings, you can ignore them based on business risk assessment and continue the task.

Error: The target instance parameter `log_bin_trust_function_creators` must be `ON` .

Warnings:

The advanced object migration/sync feature conflicts with the database/table renaming feature. After selecting advanced objects, you must cancel database/table renaming.

If you select functions or stored procedures as advanced objects, DTS will check whether `user1` corresponding to `DEFINER ([DEFINER = user1])` in the source database is the same as the task execution account `user2` .

If they are the same, do not modify the settings after migration/sync.

If they are different, change the `SQL SECURITY` attribute of `user1` in the target database after migration/sync from `DEFINER` to `INVOKER ([INVOKER = user1])`, and set the `DEFINER` in the target database to the task execution account `user2 ([DEFINER = task execution account user2])`.

Migration/Sync time of advanced objects:

Stored procedures and functions: They will be migrated/synced during **source database export**.

Triggers and events: If there are no incremental migration/sync tasks, they will be migrated/synced when the task stops; otherwise, they will be migrated/synced after you click **Done**, in which case the transition will take a longer time.

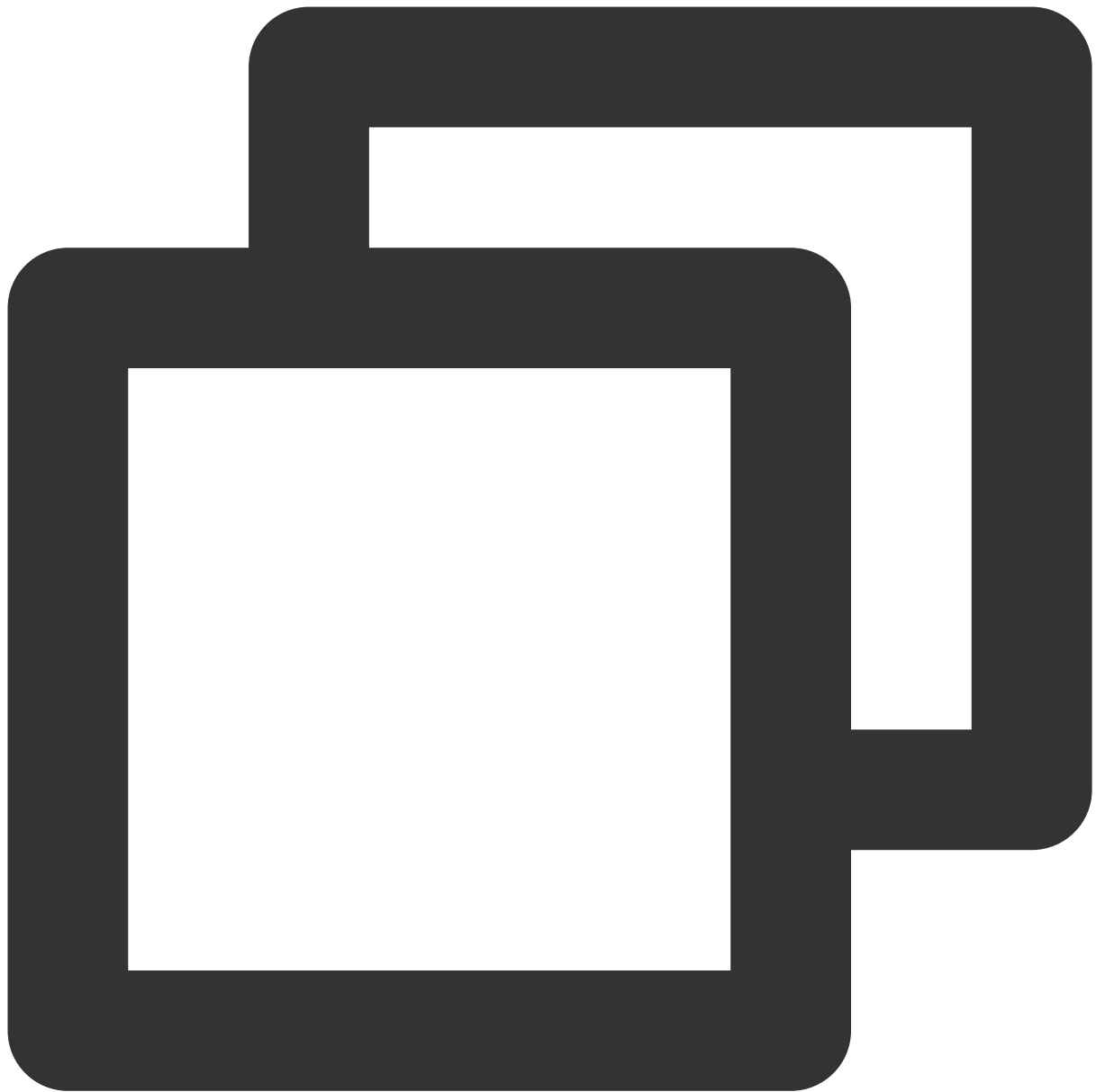
Troubleshooting

Modify the `log_bin_trust_function_creators` parameter.

`log_bin_trust_function_creators` controls whether to allow users to write stored functions to the binlog. If it is set to `OFF` , only users with `SUPER` permissions can write stored function creation operations to the binlog. If it is set to `ON` , users with no `SUPER` permissions can also do this.

If an error occurs, troubleshoot as follows:

1. Log in to the source database.
2. Run the following command to modify the `log_bin_trust_function_creators` parameter:



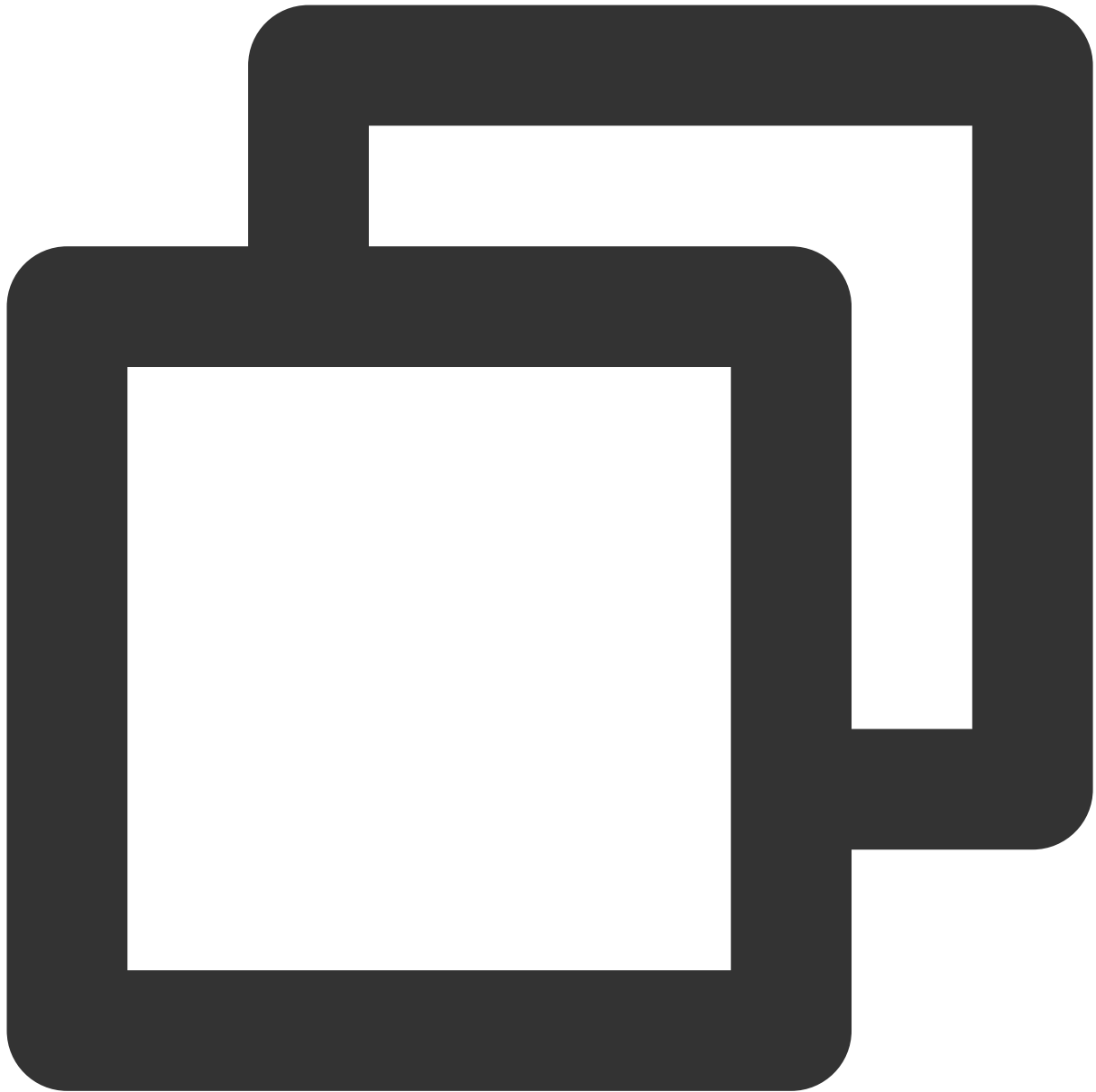
```
set global log_bin_trust_function_creators = ON;
```

3. Run the following command to check whether the parameter modification takes effect:



```
show variables like '%log_bin_trust_function_creators%';
```

The system should display a result similar to the following:



```
mysql> show variables like '%log_bin_trust_function_creators%';
```

+-----+-----+	
Variable_name	Value
+-----+-----+	
log_bin_trust_function_creators	ON

4. Run the verification task again.

Warning Item Check

Last updated : 2024-07-08 19:41:52

MySQL/TDSQL-C/MariaDB/Percona/TDSQL for MySQL Check Details

You need to configure the following parameter as required; otherwise, the system will report a warning during verification. The warning will not affect the migration task progress but will affect the business. You need to assess and determine whether to modify the parameters.

We recommend that you set `max_allowed_packet` in the target database to a value greater than that in the source database.

Impact on the business: If the value of `max_allowed_packet` in the target database is smaller than that in the source database, data cannot be written to the target database, leading to full migration failures.

Fix: Change the value of `max_allowed_packet` in the target database to a value greater than that in the source database.

We recommend that you set `max_allowed_packet` in the target database to a value greater than 1 GB.

Impact on the business: If the value of `max_allowed_packet` is too large, more memory will be used, causing packet losses and inability to capture the SQL statements of large exception transaction packets. If the value is too small, program errors may occur, causing backup failures and frequent sending/receiving of network packets, which compromises the system performance.

Fix: Run the following command to modify the `max_allowed_packet` parameter:



```
set global max_allowed_packet = 1GB
```

We recommend that you use the same character set for the source and target databases.

Impact on the business: If the character sets of the source and target databases are different, there may be garbled characters.

Fix: Run the following command to change the character sets of the source and target databases to the same one:



```
set character_set_server = 'utf8';
```

We recommend that you use an instance with 2-core CPU and 4 GB memory or higher specifications.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, we recommend that you select full + incremental data migration to ensure data consistency in real time.

For lock-involved data export, you need to use the `FLUSH TABLES WITH READ LOCK` command to lock tables in the source instance temporarily, but the MyISAM tables will be locked until all the data is exported. The lock wait timeout period is 60s, and if locks cannot be obtained before the timeout elapses, the task will fail.

For lock-free data export, only tables without a primary key are locked.

To avoid duplicate data, make sure that the tables to be migrated have a primary key or non-null unique key.

If the source database instance is a distributed database, such as TDSQL for MySQL, you need to create sharded tables in the target database in advance; otherwise, the source database tables will become non-sharded ones after being migrated.

If the target database is MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for TDStore, you need to check the `explicit_defaults_for_timestamp` parameter in the source and target databases. If it is set to `OFF` in the source database or if it is set to `ON` in both the source and target databases, the task will report a warning to remind you of not modifying this parameter when the task is running.

You need to check the `COLUMN_DEFAULT` and `IS_NULLABLE` attributes of tables in the full database/table structure export stage. If `COLUMN_DEFAULT` is set to `NULL` and `IS_NULLABLE` is set to `NOT NULL` for tables in the source database, the table structure will not be migrated or synced, because otherwise, the MySQL system will automatically add the default `CURRENT_TIMESTAMP` parameter for the migrated or synced data of the `TIMESTAMP` type.

TDSQL Proxy Check

Last updated : 2024-07-08 19:41:52

Check Details

In scenes where the Proxy method is used for the target database TDSQL for MySQL connection, DTS executes SQL by combining multiple statements into one SQL. You need to set the parameter `gateway.mode.multi_query.open` to 1 to support the merging of multiple SQL. If the parameter is not enabled, the pre-verification task reports an error.

Fixing Solution

Users cannot modify it by themselves. [Submit a ticket](#) for modification.

TDSQL Kernel Check

Last updated : 2024-07-11 15:08:15

Check Details

When the source is TDSQL for MySQL, check the kernel version of the source. Mariadb kernel is not supported. If it is a Mariadb kernel, the pre-verification task reports an error.

Fixing Solution

Unable to synchronize TDSQL for MySQL (Mariadb kernel). Please select another version for synchronization.

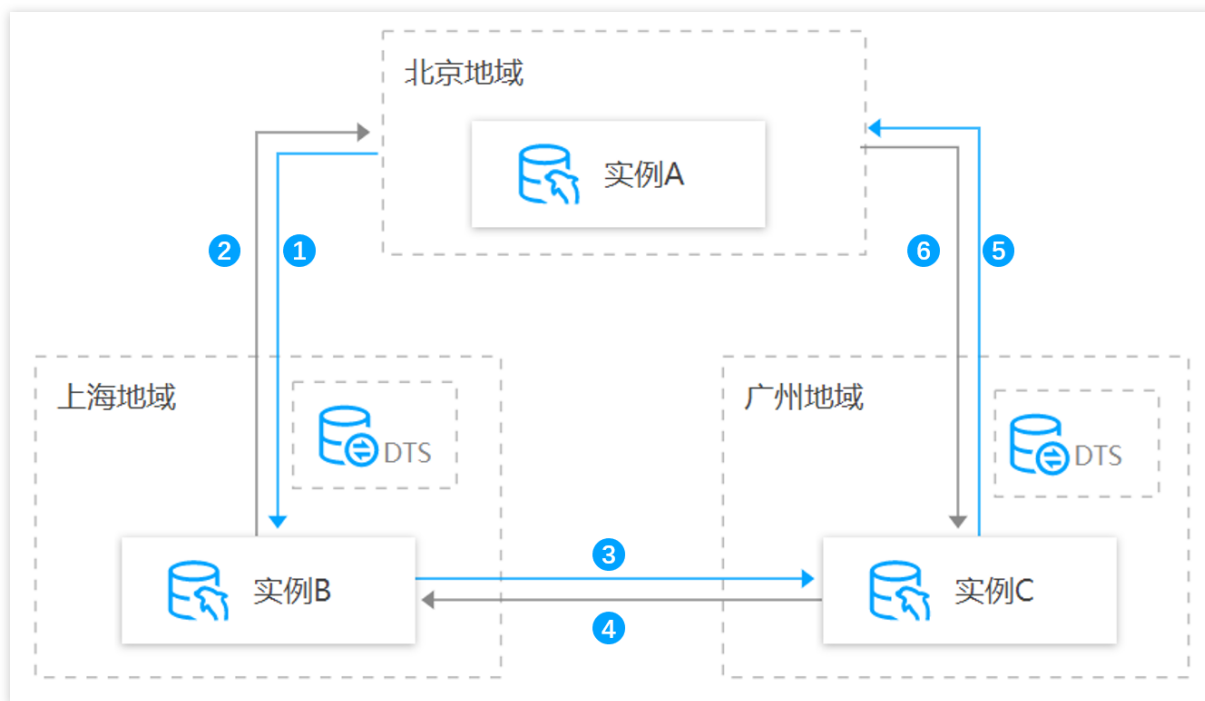
DDL Ring Linkage Synchronization Detection for Same Database and Table Objects

Last updated : 2024-07-11 15:09:18

Check Details

In scenes requiring configuring multiple synchronization tasks such as bidirectional synchronization, many-to-one synchronization, and active-active, DDL configurations can not form a ring linkage; otherwise, DDL statements may loop in the system, causing errors.

Example: In the following figure, you can select DDL in up to two of the three synchronization tasks (1, 3, and 5) marked by blue lines. Selecting DDL in three tasks would form a ring linkage.



Fixing Solution

To modify synchronization task configurations, in **Sync Options and Synchronization Objects > Data Synchronization Options > Synchronization Operation Type**, adjust DDL parameter configurations to avoid forming a ring linkage.

DDL Synchronization Conflict Detection for the Same Target and Same Database and Table Object

Last updated : 2024-07-11 15:10:55

Check Details

In scenes requiring configuring multiple synchronization tasks, such as many-to-one synchronization and multi-active, the same database and table object cannot receive DDL synchronization from multiple data centers; otherwise, such DDL may conflict with each other in the target, causing errors.

Fixing Solution

To modify the synchronization task configuration, adjust the DDL parameter configuration in **Sync Options and Synchronization Objects > Data Synchronization Options > Synchronization Operation Type** to avoid the same database and table object receiving DDL synchronization from multiple data centers.

Offline Migration of MySQL Data

Last updated : 2024-07-08 19:41:52

This document describes how to migrate data through the console and command line tool.

Data Migration Through Console

There are two modes for migrating data through the console: physical backup and logical backup. For more information, see:

[Restoring Database from Physical Backup](#)

[Restoring Database from Logical Backup](#)

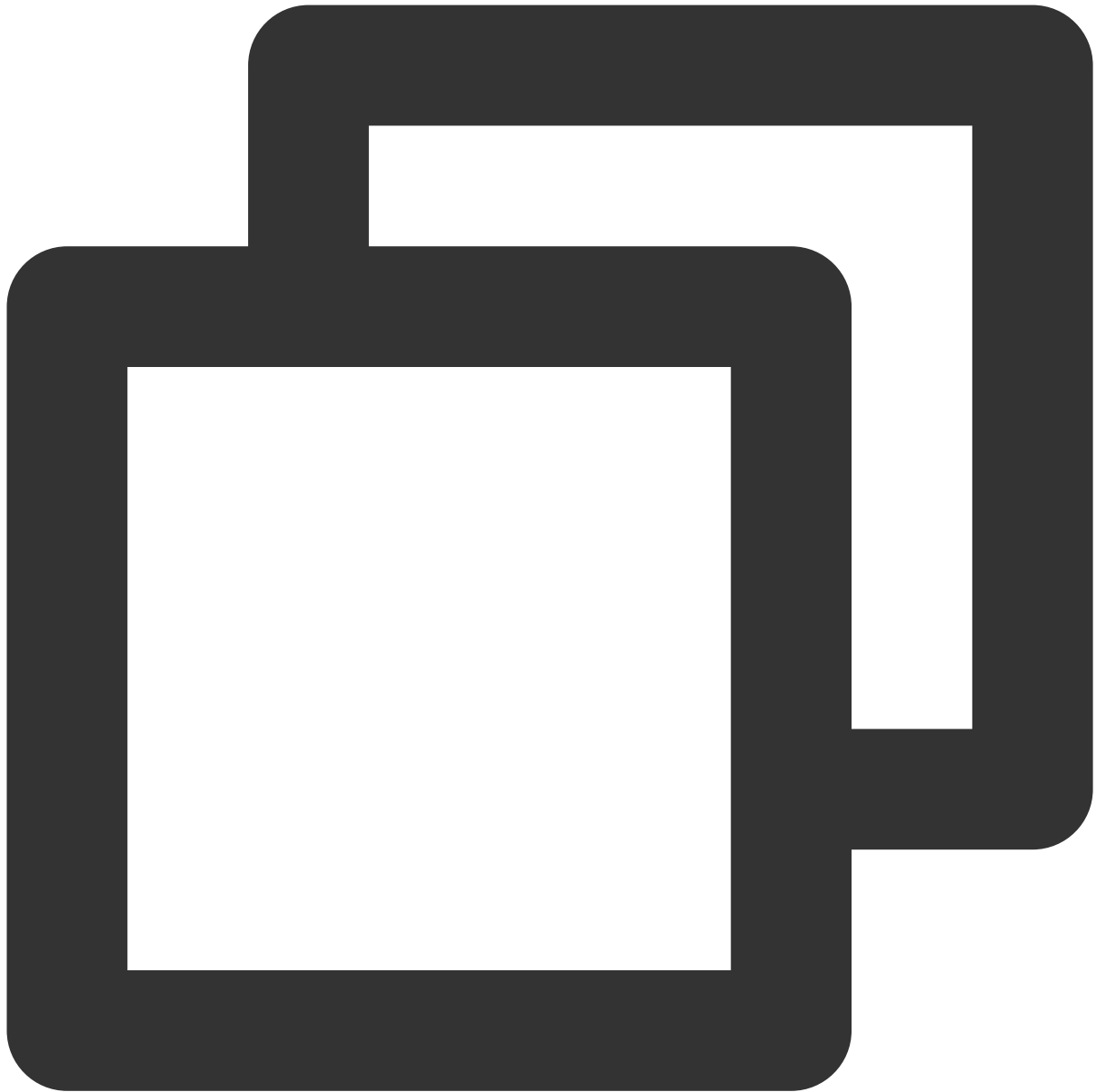
Data Migration with Command Line Tool

1. Generate the SQL file to be imported with the MySQL command line tool "mysqldump" in the following way:

Note:

The data files exported using mysqldump must be compatible with the SQL specification of your purchased TencentDB for MySQL version. 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".

Make sure that the same source and target database versions, character sets, and mysqldump tool versions are used. You can specify the character set with the `--default-character-set` parameter.

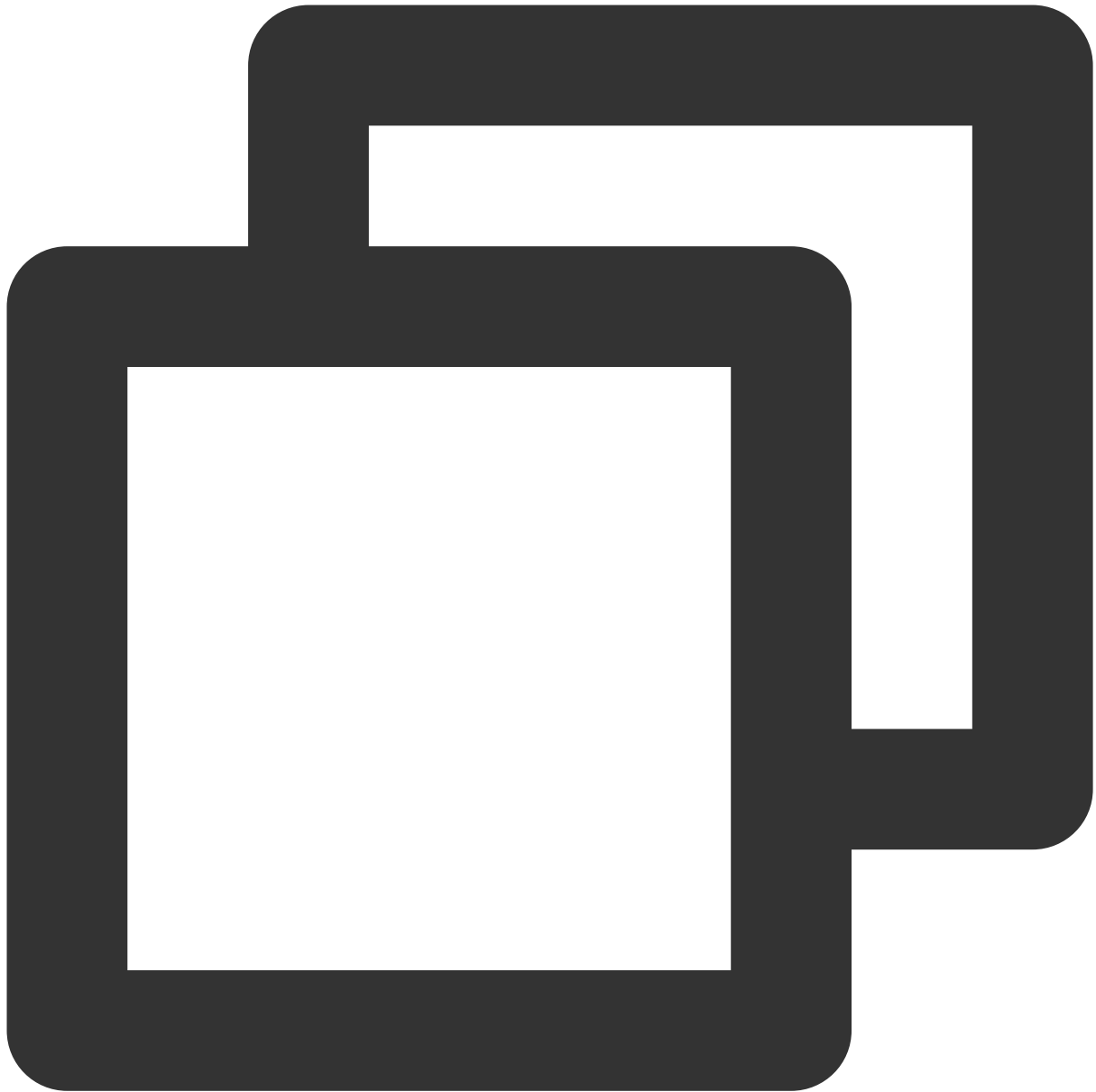


```
shell > mysqldump [options] db_name [tbl_name ...] > bak_pathname
```

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. Import data to the target database with the MySQL command line tool as follows:



```
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 (Windows)

1. Use the Windows version of mysqldump to generate the SQL file to be imported. For more information, see the description in [Data Migration with Command Line Tool](#).
2. Enter the command prompt and import the data into the target database with the MySQL command line tool.


```
>mysql -h -p -u -p <  
Enter password: *****
```

3. [Log in to the target MySQL database](#), run the `show databases;` command, and you can see that the backup database has been imported into the target database.

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input s  
  
mysql> show databases;  
+-----+  
| Database  
+-----+  
| information_schema  
| db_blog  
| la  
| lailailai  
| lalalalalalaa  
| mysql  
| performance_schema  
| sakila  
| test  
| world  
+-----+  
10 rows in set (0.01 sec)
```

Migrating data (Linux)

This document uses a Linux CVM instance as an example. For more information on how to access a database from a CVM instance, see [Accessing MySQL Database](#)

1. [Log in to the CVM instance](#) and generate the SQL file to be imported with the MySQL command line tool "mysqldump". Take the `db_blog` database in TencentDB as an example:

```
Send CtrlAltDel ▼
[root@UM_74_55_centos home]# mysqldump -h [REDACTED] -u root -p db_blog
Enter password:
[root@UM_74_55_centos home]# ls /home
db_back db_blog.bak
[root@UM_74_55_centos home]#
```

2. Use the MySQL command line tool to restore the data to the target database.
3. Log in to the target MySQL database, run the `show databases;` command, and you can see that the backup database has been imported into the target database.

```
[root@UM_74_55_centos lib]# mysql -h localhost -u root -p db_blog < /home/db_blog
Enter password:
[root@UM_74_55_centos lib]# mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 5.5.52-MariaDB MariaDB Server

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

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

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| db_blog |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.00 sec)

MariaDB [(none)]> _
```

Issues with Character Set of Imported Data Files

1. If no character set is specified during data file import into TencentDB, 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](#).

Migrating to PostgreSQL

Migration from PostgreSQL to TencentDB for PostgreSQL

Last updated : 2024-07-08 19:56:53

This document describes how to use the data migration feature of DTS to migrate data from PostgreSQL to TencentDB for PostgreSQL.

The steps of data migration from TDSQL-C for PostgreSQL to TencentDB for PostgreSQL are basically the same as those in this scenario.

Notes

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

When you migrate an instance over the public network, make sure that the source database service is accessible over the public network and keep the public network connection stable. If the network fluctuates or fails, migration will fail, and you need to restart the migration task.

If the source PostgreSQL database is not TencentDB for PostgreSQL, it must have the replication permission; otherwise, the precheck will fail.

During migration, the migration rate can be affected by factors such as the read performance of the source, the network bandwidth between the source and the target instances, and the specification of the target instance. Migration concurrency is determined by number of CPU cores of target instance, for instance, if the target instance has 2 cores, concurrency will be 2.

When you configure the migration task, the source database address must be an IP and port that can directly connect to the PostgreSQL instance instead of IPs and ports of proxies such as Pgpool.

Prerequisites

You have created a TencentDB for PostgreSQL instance as instructed in [Creating TencentDB for PostgreSQL Instance](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

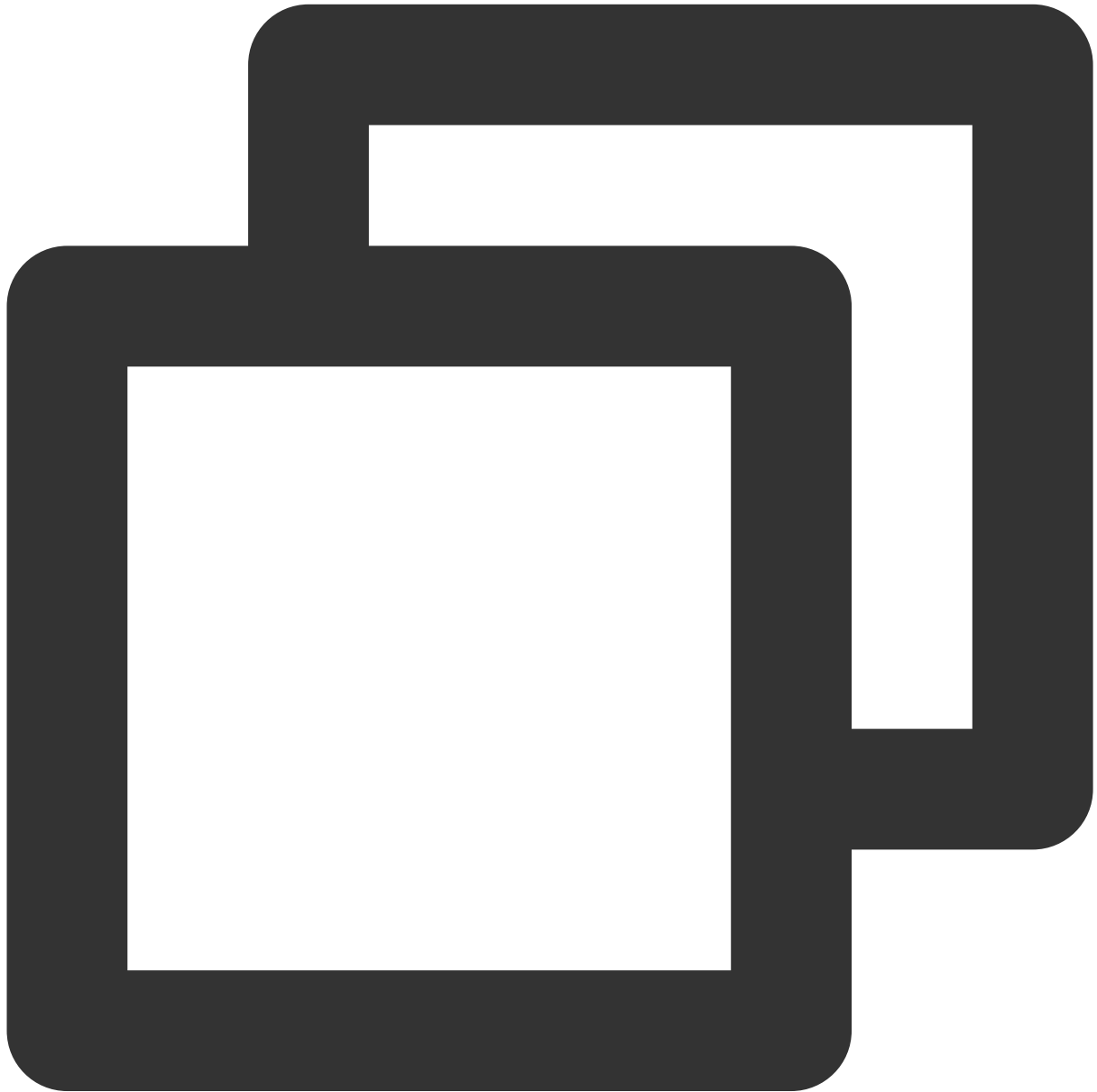
You have completed all the preparations as instructed in [Overview](#).

Permissions required for the source database:

If the source PostgreSQL database is not TencentDB for PostgreSQL, it must have the replication permission.

If the source database is TencentDB for PostgreSQL, the source database account must be the initial user selected when the TencentDB instance is created.

If some tables or objects have no permissions, you can use a high-privileged user to run the following sample statements to grant them permissions:

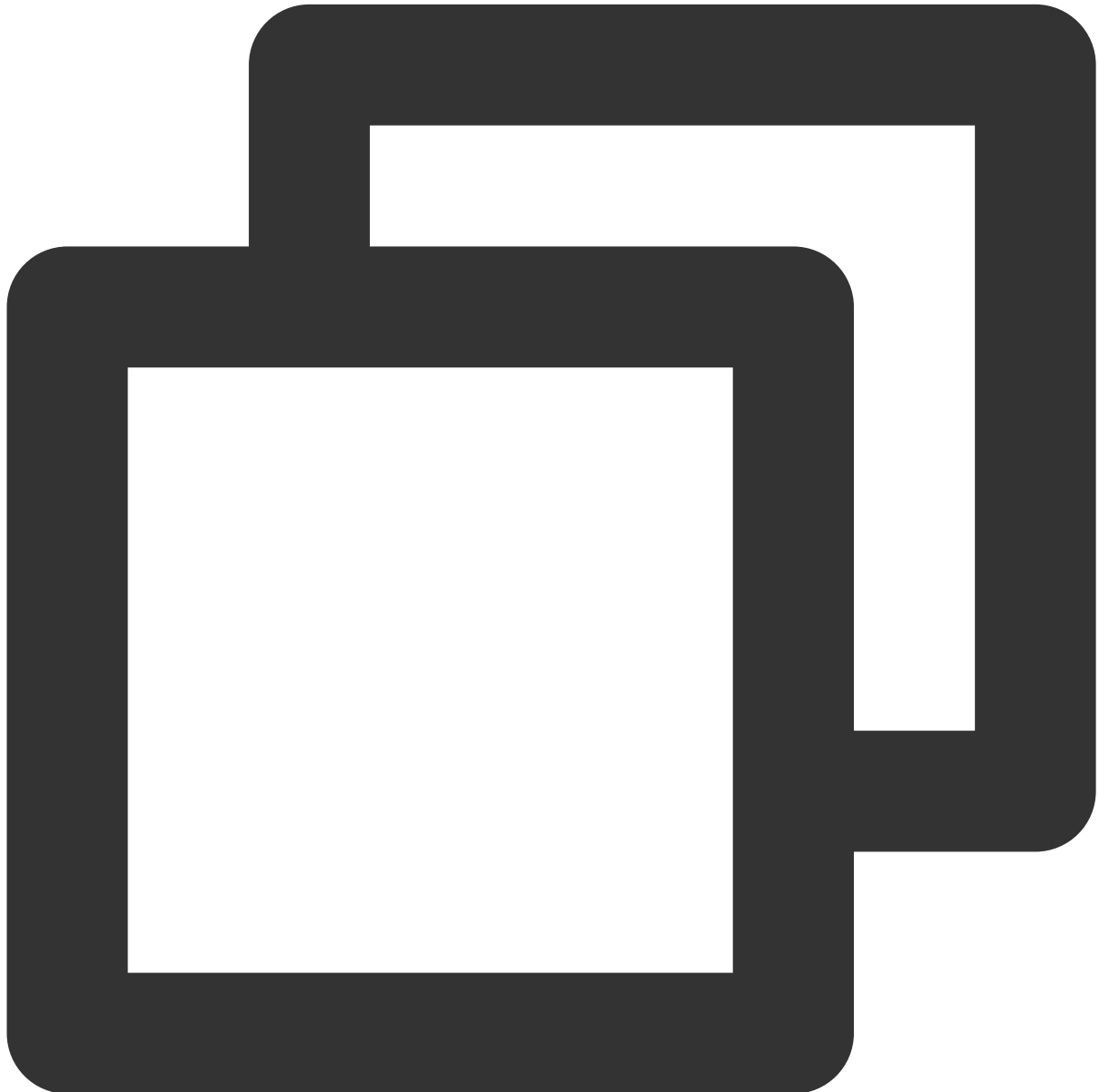


```
grant select on table `table name` to `username`;  
grant select on SEQUENCE `sequence name` to `username`;  
grant connect on database `database name` to `username`;
```

```
grant select on large object `large object name` oid to `username`;  
GRANT USAGE ON SCHEMA `schema name` to `username`;
```

The target database account must be the initial user selected when the TencentDB instance is created.

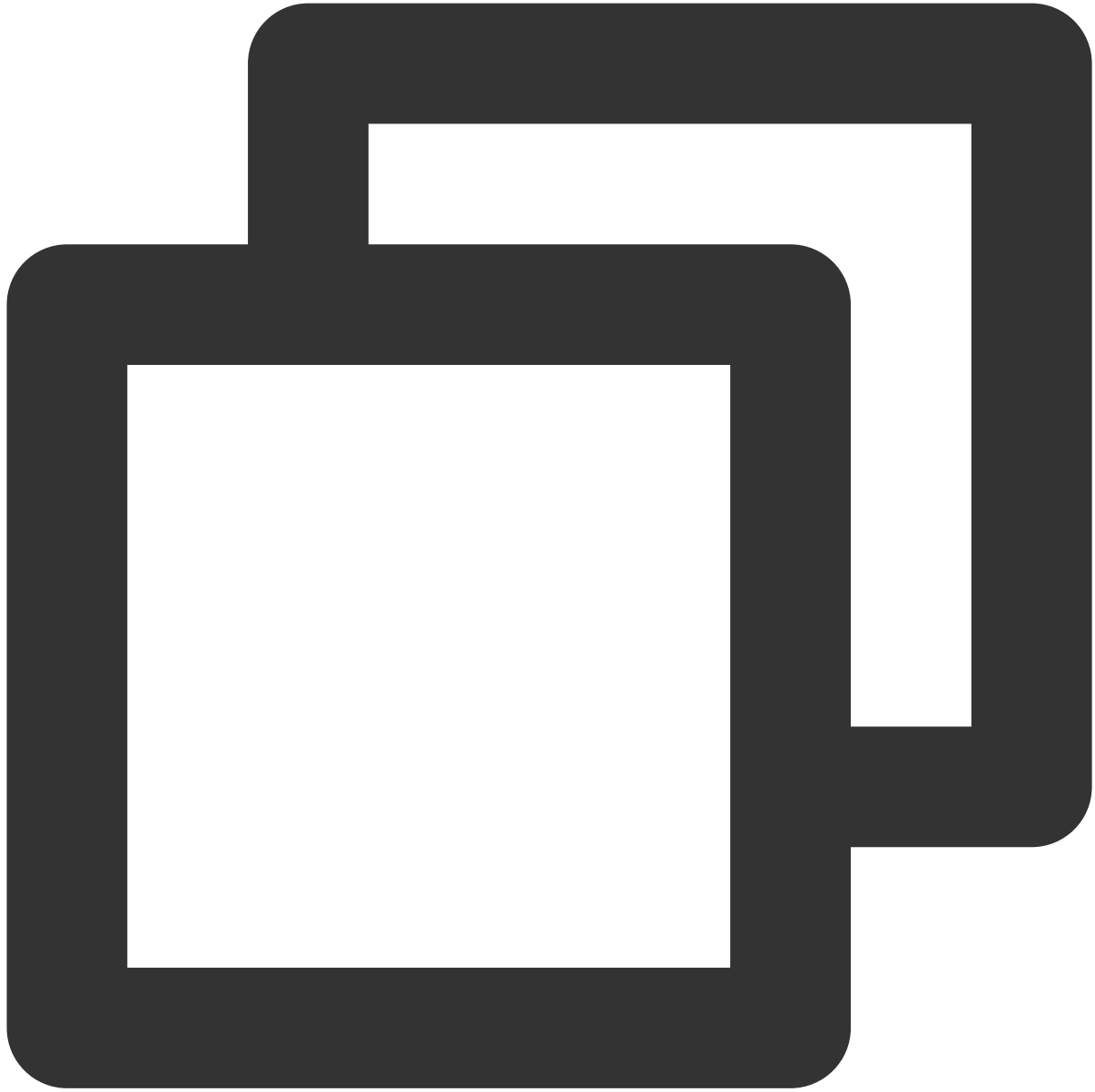
If the target database instance contains the database to be migrated and the database owner is not the current migration user, run the following statement to grant the database permissions to the migration user:



```
alter database `database name` owner to `migration user`;
```

If the migration user (the account executing the migration task) is not a user with the `pg_tencentdb_superuser` role, the system will prompt "Failed to verify the target instance permissions and unable to get the schema list" during

verification. In this case, run the following statement to grant the initial user's permissions to the migration user:



```
grant pg_tencentdb_superuser to migration user;
```

Application restrictions

Correlated data objects must be migrated together; otherwise, migration will fail. Common correlations include table referenced by views, view referenced by views, view/table referenced by procedures/functions/triggers, and tables

correlated through primary/foreign keys.

To ensure the migration efficiency, the data of CVM-based self-built instances cannot be migrated across regions over the private network. If you need to migrate data across regions, you can do so over the public network.

To migrate the entire instance, there cannot be users and roles in the target database with the same name as those in the source database.

If you select **Full + Incremental migration**, tables in the source database must have a primary key; otherwise, data inconsistency will occur in the source and target database. We recommend you select **Full migration** for tables without the primary key.

Operation restrictions

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers during migration.

Do not perform DDL or large object operations during structural, full, and incremental migration; otherwise, the migrated data will be inconsistent.

If you only perform full data migration, only data before the migration start time will be migrated. If you write new data into the source instance during migration, there will be data inconsistency between the source and target databases. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.

Environment requirements

Note:

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment requirements
Requirements for the source database	<p>The source and target databases can be connected.</p> <p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p> <p>Requirements for instance parameters:</p> <p>The <code>wal_level</code> parameter in the source database must be set to <code>logical</code> during incremental migration.</p> <p>The value of <code>max_replication_slots</code> in the source database must be greater than the number of databases to be migrated during incremental migration.</p> <p>The value of <code>max_wal_senders</code> in the source database must be greater than the number of databases to be migrated during incremental migration.</p>

Requirements for the target database

In full migration, the target database version must be greater than or equal to the source database version; in incremental migration, migration between versions later than 10.x is supported.

The available size of the target database space must be at least 1.2 times that of the instances to be migrated in the source database. Incremental data migration will execute UPDATE and DELETE operations, causing some tables in the database to generate data fragments.

Therefore, after migration is completed, the size of the tables in the target database may be larger than that in the source database. This is mainly because that the `autovacuum` trigger conditions of the source and target databases are different.

The target database cannot have migration objects such as usernames and tables with the same name as those in the source database.

The value of `max_worker_processes` of the target database during incremental migration must be greater than that of `max_logical_replication_workers`.

Directions

1. (Optional) When PostgreSQL 9.4, 9.5, and 9.6 are used as the source database for "full + incremental migration", you can install the `tencent_decoding` extension as instructed below. For other scenarios, skip this step.

1.1 Download extension based on architecture of server where source database resides.

Only "x86_64" and "aarch64" system architectures are supported.

The version of the extension version need to be the same as that of PostgreSQL.

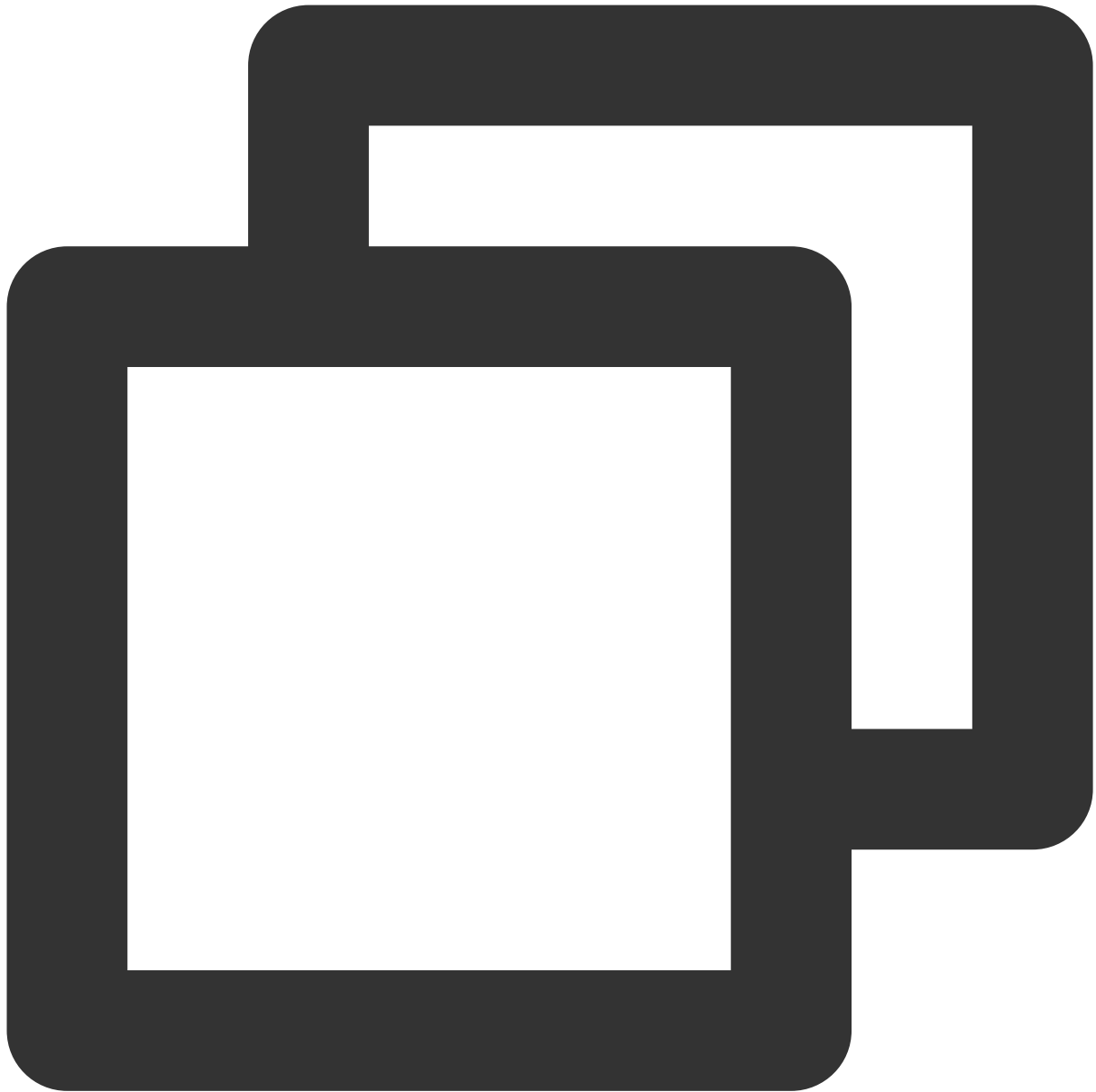
Requirements for the Glibc version: x86_64 system should be v2.17 - 323 or later; aarch64 system should be v2.17 - 260 or later.

View the Glibc version on Linux:



```
RHEL/CentOS: rpm -q glibc
```

View the Glibc version on other operation systems (Debian/Ubuntu/SUSE):



```
ldd --version | grep -i libc
```

Download address: [x86_64 9.4](#), [x86_64 9.5](#), [x86_64 9.6](#), [aarch64 9.4](#), [aarch64 9.5](#), [aarch64 9.6](#).

1.2 Place the downloaded tencent_decoding.so file in the lib folder of the Postgres process directory without restarting the instance.

2. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** to enter the **Create Migration Task** page.

3. On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Item	Description
Source Instance Type	Select the source database type, which cannot be changed after purchase. In this scenario, select PostgreSQL .
Source Instance Region	Select the source database region. If the source database is a self-built one, select a region nearest to it.
Target Instance Type	Select the target database type, which cannot be changed after purchase. In this scenario, select PostgreSQL .
Target Instance Region	Select the target database region.
Specification	Select the specification of the migration link based on your business conditions. For the performance and billing details of different specifications, see Billing Overview .

4. On the **Set source and target databases** page, configure the task, source database, and target database settings. After the source and target databases pass the connectivity test, click **Create**.

Note:

If the connectivity test fails, troubleshoot as prompted or as instructed in [Database Connection Check](#) and try again.

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Tag	Tags are used to manage resources by category in different dimensions. If the existing tags do not meet your requirements, go to the console to create more.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be

		changed.
	Access Type	<p>Select a type based on your scenario. In this scenario, select Database. For the preparations for different access types, see Overview.</p> <p>To ensure the migration efficiency, the data of CVM-based self-built instances cannot be migrated across regions over the private network. If you need to migrate data across regions, you can do so over the public network.</p> <p>Public Network: The source database can be accessed through a public IP.</p> <p>Self-Build on CVM: The source database is deployed in a CVM instance.</p> <p>Direct Connect: The source database can be interconnected with VPCs through Direct Connect.</p> <p>VPN Access: The source database can be interconnected with VPCs through VPN Connections.</p> <p>Database: The source database is a TencentDB instance.</p> <p>CCN: The source database can be interconnected with VPCs through CCN.</p>
	Cross-/Intra-Account	<p>Intra-account: The source and target database instances belong to the same Tencent Cloud root account.</p> <p>Cross-account: The source and target database instances belong to different Tencent Cloud root accounts. The following uses migration under the same account as an example.</p>
	Database Instance	Select the instance ID of source PostgreSQL database.
	Account	Account of the source PostgreSQL database, which must have the required permissions.
	Password	Password of the source PostgreSQL database account.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database.

5. On the **Set migration options and select migration objects** page, configure the migration type and objects and click **Save**.

Configuration Item	Description
Migration Type	<p>Select an option based on your scenario.</p> <p>Structural migration: Structured data such as databases and tables in the database will be migrated.</p> <p>Full migration: The entire database will be migrated. The migrated data will only be existing content of the source database when the task is initiated but not include the incremental data written to the source database after the task is initiated.</p> <p>Full + Incremental migration: The migrated data will include the existing content of the source database when the task is initiated as well as the incremental data written to the source database after the task is initiated. If there are data writes to the source database during migration, and you want to smoothly migrate the data in a non-stop manner, select this option.</p>
Migration Object	<p>Entire instance: Migrate the entire database instance, including the metadata definitions of roles and users but excluding system databases such as system objects in PostgreSQL.</p> <p>Specified objects: Migrate specified objects.</p>
Specify object	Select the objects to be migrated in Source Database Object and move them to the Selected Object box.

6. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

7. Return to the data migration task list, and you can see that the task has entered the **Preparing** status. After 1–2 minutes, the data migration task will be started.

Select **Structural migration** or **Full migration**: Once completed, the task will be stopped automatically.

Select **Full + Incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data sync. Then, the task will enter the **Completed** status. At this point, do not make any changes to the source and target databases. The backend will automatically align some objects with the source.

Manually complete incremental data sync and business switchover at appropriate time.

Check whether the migration task is in the incremental sync stage without any lag. If so, stop writing data to the source database for a few minutes.

Manually complete incremental sync when the data gap between the target and the source databases is 0 MB and the time lag between them is 0 seconds.

8. (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

9. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Migration from TDSQL-C PostgreSQL to TencentDB for PostgreSQL

Last updated : 2024-07-08 19:56:53

TDSQL-C PostgreSQL migration to TencentDB for PostgreSQL requirements and guidance, is largely consistent with [PostgreSQL migration to PostgreSQL](#). Please refer to the related content for operations.

PostgreSQL Physical Migration

Last updated : 2024-07-11 16:02:21

Physical migration uses PostgreSQL physical streaming replication, which is faster and more stable. For more details, see [Configuring Physical Migration Task](#) in the PostgreSQL product documentation.

Processing for PostgreSQL Series Pre-Check Failure

Database Connection Check

Last updated : 2024-07-08 19:56:53

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see [Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database](#).

The source IP addresses are blocked by the source database. For more information, see [Failed Connectivity Test > Source IP Addresses Blocked in Source Database](#).

The network port is closed. For more information, see [Failed Connectivity Test > Closed Network Port](#).

The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Object Dependency Check

Last updated : 2024-07-08 19:56:53

Version Check

Last updated : 2024-07-08 19:56:53

Check Details

Migration scenario

Source database versions earlier than PostgreSQL 10.x (such as 9.x) do not support full + incremental migration. If an incremental migration task is configured for the source database, the verification will fail.

If the source and target database versions are inconsistent, there may be some compatibility issues and the task will report a warning. You need to read the compatibility report of each version to check whether your business has used incompatible features.

Sync scenario

Source database versions earlier than PostgreSQL 10 do not support data sync. A data sync task requires that the target instance version be later than or the same as the source instance version.

If the source and target database versions are inconsistent, there may be some compatibility issues and the task will report a warning. You need to read the compatibility report of each version to check whether your business has used incompatible features.

Troubleshooting

Check the source and target databases as instructed in [Databases Supported for Data Migration](#) and [Databases Supported for Data Sync](#). If the source or target database version is not supported, upgrade the target database version or use a database instance on a higher version.

Source or Target Instance Permission Check

Last updated : 2024-07-08 19:56:53

Check Details

Check whether you have the operation permissions of the database.

Troubleshooting

If you don't have the operation permissions, get authorized based on the permission requirements in the check details, and run the verification task again.

Incremental Migration Pre Check

Last updated : 2024-07-08 19:56:53

Check Details

If you select incremental migration as the migration type, you need to check the following conditions; otherwise, the verification will fail.

The major version of the source and target databases need to be below PostgreSQL 10.x.

`wal_level` in the source database must be set to `logical` .

The `max_replication_slots` and `max_wal_senders` values in the target database must be greater than the total number of databases to be migrated.

The `max_worker_processes` value in the target database must be greater than the `max_logical_replication_workers` value.

The tables to be migrated should not include unlogged tables; otherwise, they cannot be migrated.

Fix

If the version does not meet the requirements, you need to upgrade it. You can change the values of the

`wal_level` , `max_replication_slots` , `max_worker_processes` , and `max_wal_senders` as follows:

1. Log in to the source database.

Note:

If the source database is self-built, you need to log in to the server where the database runs and enter the main data directory of the database, which is usually `$PGDATA` .

If the source database is in another cloud, modify the parameters as requested by the corresponding cloud vendor.

If you need to modify the parameters in the target database, [submit a ticket](#) for assistance.

2. Open the `postgresql.conf` file and modify `wal_level` .



```
wal_level = logical
```

3. After the modification is completed, restart the database.
4. Log in to the database and run the following command to check whether the parameters are correctly set:



```
postgres=> select name,setting from pg_settings where name='wal_level';
name      | setting
-----+-----
 wal_level | logical
(1 row)
postgres=> select name,setting from pg_settings where name='max_replication_slots';
          name          | setting
-----+-----
 max_replication_slots | 10
(1 row)
postgres=> select name,setting from pg_settings where name='max_wal_senders';
```



```
name          | setting
-----+-----
max_wal_senders | 10
(1 row)
```

5. Run the verification task again.

Structural Compatibility Check

Last updated : 2024-07-08 19:56:53

Check Details

If the target database version is 12 or later, and the table to be migrated contains the data types `abstime` , `reltime` , and `tinterval` , the verification task reports an error.

The data types `abstime` , `reltime` , and `tinterval` are older time and date types that have been deprecated in the new PostgreSQL versions. It is recommended to use the following types as replacements:

`abstime` : You can use `timestamp` or `timestamp with time zone` as replacements. These types provide a broader range of date and time representation capabilities.

`reltime` : You can use the `interval` type as a replacement. The `interval` type is used to represent time intervals, which can include units such as years, months, days, hours, minutes, and seconds.

`tinterval` : You can use `tsrange` or `tstzrange` as replacements. These types are used to represent time ranges, which can include the origin and end times.

Fixing Solution

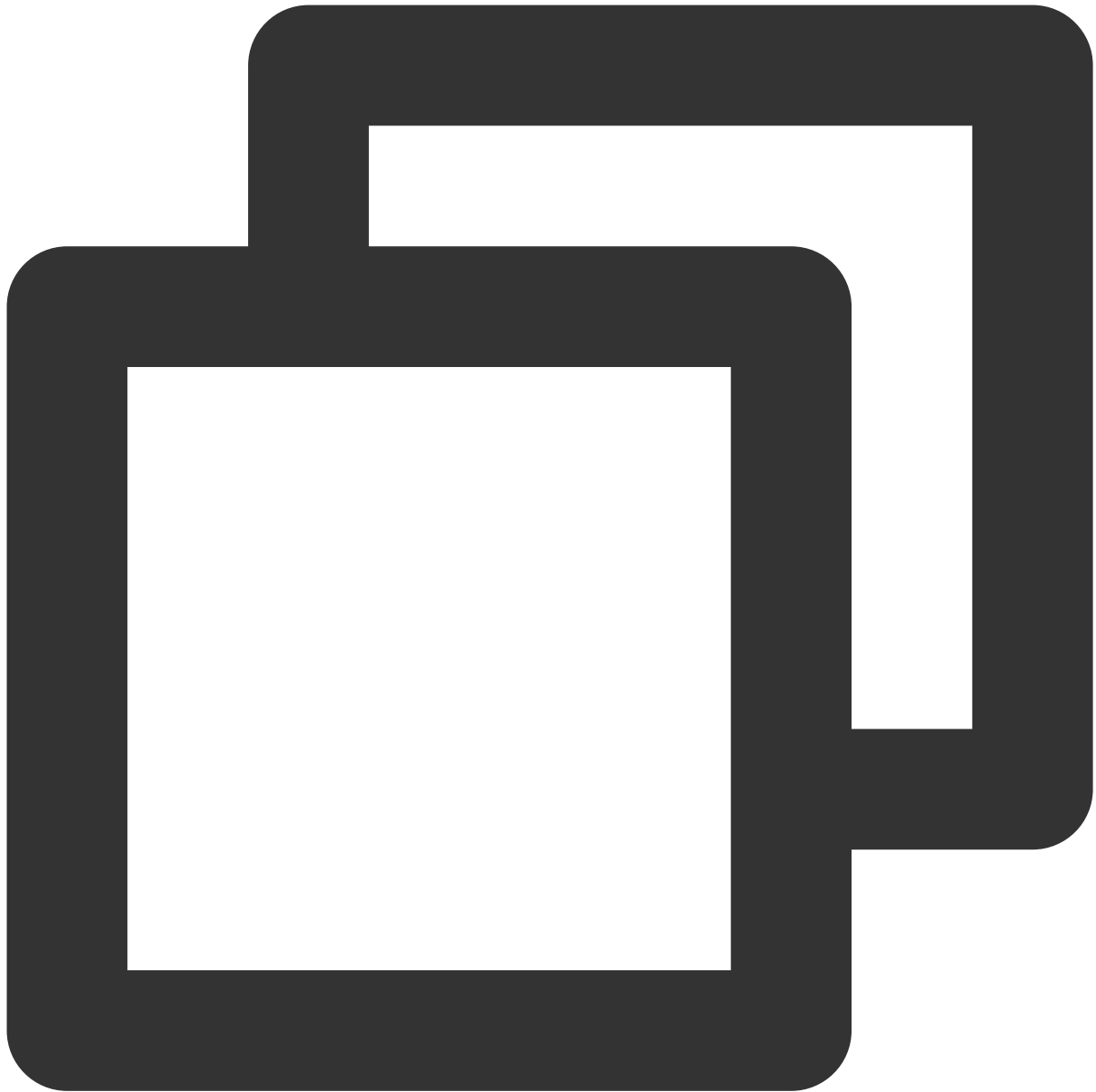
The methods for modifying the data types `abstime` , `reltime` , and `tinterval` are similar. First, create a column, convert the old data format to the new one, and store it in the new one. After the new column's data is verified that it is correct, delete the old column. Finally, if necessary, rename the new column to the name of the old column. Below is an introduction to modifying `abstime` to `timestamp` as an example.

1. In the table where the verification task prompts an error, add a new column of the timestamp type.



```
ALTER TABLE your_table ADD COLUMN new_column TIMESTAMP;
```

2. Convert the data in the abstime column to the timestamp type and store the result in the new column.



```
UPDATE your_table SET new_column = your_abstime_column::TIMESTAMP;
```

3. Verify whether the data in the new column is correct.



```
SELECT * FROM your_table;
```

4. If the data in the new column is correct, you can delete the old abstime column.



```
ALTER TABLE your_table DROP COLUMN your_abstime_column;
```

5. If necessary, you can rename the new column to the name of the old column.



```
ALTER TABLE your_table RENAME COLUMN new_column TO your_abstime_column;
```

Structural Conflict Check

Last updated : 2024-07-08 19:56:53

Check Details

In PostgreSQL data migration scenarios, the target instance cannot contain objects with the same name as those in the source database.

If an entire PostgreSQL instance is migrated, the target instance must be empty. If a conflict causes an error, you need to delete the instance content.

Troubleshooting

If a conflict is detected, you need to delete the conflicted content and verify again.

Account Conflict Check

Last updated : 2024-07-08 19:56:53

Check Details

Check whether the target database user conflicts with the source database user in a data migration scenario.

Troubleshooting

In full database migration, if the target database has the same account as that in the source database, you need to delete it.

If the account in the target database is the initial account, use it to log in to the database and run the following statements:



```
CREATE USER new user WITH PASSWORD 'password';  
GRANT pg_tencentdb_superuser TO new username;  
ALTER USER new user WITH CREATEDB;  
ALTER USER new user WITH CREATEROLE;
```

If the account in the target database is a new user, use it to log in to the database and delete the conflicting user.



```
DROP USER conflicting user;  
# If the conflicting user has resource dependencies, run the following statement to  
ALTER TABLE table name OWNER TO new user;
```

After the conflicting user is deleted, run the verification task again.

Parameter Configuration Conflict Check

Last updated : 2024-07-08 19:56:53

Check Details

Check requirements: check the following parameter values of the source and target databases. If a parameter has different values in the source database and target database, a verification warning will be reported, which will not block the migration task but will affect the business. You need to assess and determine whether to modify the parameter.

TimeZone, lc_monetary, lc_numeric, array_nulls, server_encoding, DateStyle, extra_float_digits, gin_fuzzy_search_limit, xmlbinary, and constraint_exclusion.

Impact on the business: if the parameter values are different, data inconsistency between the source and target databases may be caused. Below is the specific impact:

TimeZone: sets the instance time zone. If this parameter has different values, data may be incorrect after migration.

lc_monetary: sets the instance currency mode. If this parameter has different values, currency numbers may be incorrect after migration.

lc_numeric: sets the instance numeric mode. If this parameter has different values, data may be incorrect after migration.

array_nulls: sets whether arrays can be empty. If this parameter has different values, data inconsistency may occur, and certain data may fail to be migrated.

server_encoding: sets the instance character set. If this parameter has different values, garbled characters may be present in the stored data.

DateStyle: sets the date format. If this parameter has different values, data migration may fail.

extra_float_digits: sets the floating-point value output precision. If this parameter has values, data precision will be affected. In high-precision database use cases, data inconsistency will occur after migration.

gin_fuzzy_search_limit: sets the upper limit of the size of the set returned by the GIN index. If this parameter has different values, data display inconsistency will occur after migration.

xmlbinary: sets the XML function conversion result. If this parameter has different values, function execution in the target database may be different from that in the source database.

constraint_exclusion: sets whether the restraints take effect. If this parameter has different values, data inconsistency may occur after migration.

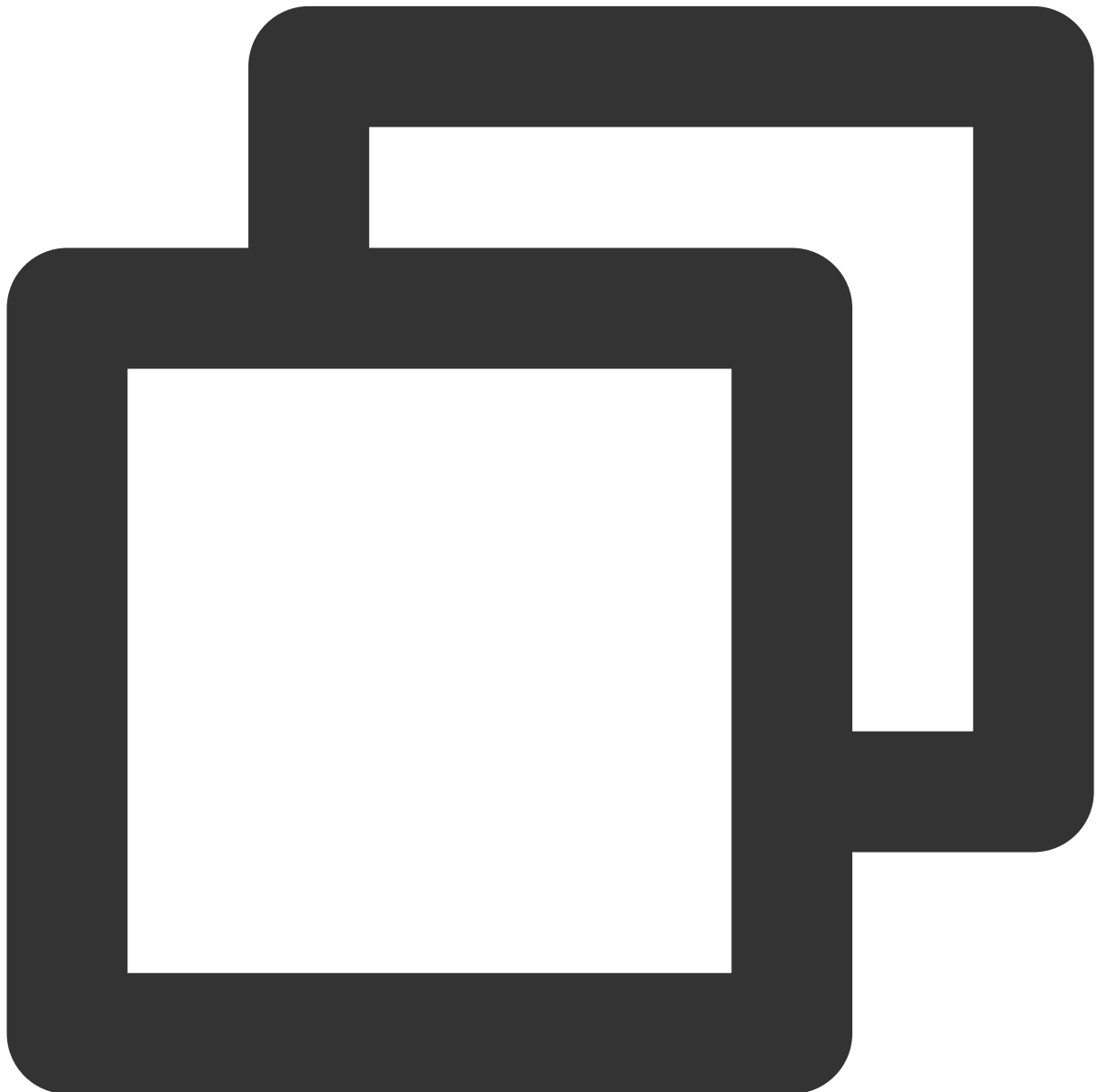
Fix

1. Log in to the source database with the `superuser` account.

2. Run the following sample statements to modify the corresponding parameters:

You can choose to first modify the parameters in the source database. If such parameters cannot be modified, you need to modify them in the target database by [submitting a ticket](#).

The `server_encoding` parameter cannot be modified in the source database. If it is exceptional, check whether it has been created in the target database, and if so and it is different from that in the source database, you need to apply for a new instance; and if not, modify it as follows (currently, TencentDB instances only support two character sets: UTF-8 and LATIN):



```
alter system set timezone='parameter value';  
alter system set lc_monetary='parameter value';
```

```
alter system set lc_numeric='parameter value';
```

Plugin Compatibility Check

Last updated : 2024-07-08 19:56:53

Check Details

Check requirements: check whether extensions/plugins in the source database also exist in the target database. Before migration, you don't need to create extensions/plugins in the target database since DTS will create them for you. If an extension/plugin cannot be created in it, or the extension/plugin version in it differs from that in the source database, a verification warning will be reported. The warning will not block the migration task but will affect the business.

Impact on the business: PostgreSQL has many extensions. Most extension compatibility issues don't affect data migration, but those related to the storage engine (such as TimescaleDB, PipelineDB, and PostGIS) will cause the migration task to fail.

Fix

For extension/plugin compatibility issues not related to the engine (such as `pg_hint_plan`, `pg_prewarm`, `tsearch2`, `hll`, `rum`, and `zhparser`), you generally can ignore them and fix them by yourself based on your business conditions. For extension compatibility issues related to the engine (such as TimescaleDB, PipelineDB, and PostGIS), we recommend you [submit a ticket](#) for assistance.

Multi Task Conflict Detection

Last updated : 2024-07-08 19:56:53

Check Details

PostgreSQL does not support rings in data sync. A ring will cause the task verification to report an error. If there is a ring in a data sync task, in which case the sync objects in the source database have been involved in multiple tasks, a warning will be reported. You need to check your task configurations.

Troubleshooting

Cancel the tasks with rings or duplicate sync configurations as prompted and execute the verification task again.

Long Transaction Check

Last updated : 2024-07-08 19:56:53

Check Details

If there are long-running transactions in the source database, the verification task raises an warning. Users can ignore the warning and continue with the task, but it is recommended that long transactions be handled before they initiate the DTS task.

Fixing Solution

Handle the long transactions in the source database and initiate the DTS task after the transactions have ended.

Migrating to MongoDB

Migration from MongoDB to TencentDB for MongoDB

Last updated : 2024-07-08 20:23:27

This document describes how to use the data migration feature of DTS to migrate data from MongoDB to TencentDB for MongoDB.

MongoDB supports heterogeneous migration between replica sets and sharded clusters, that is, it supports four source-target architecture scenarios: replica set – replica set, replica set – sharded cluster, sharded cluster – replica set, sharded cluster – sharded cluster.

Note

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

To migrate instances over the public network, make sure that the source instance is accessible from the public network.

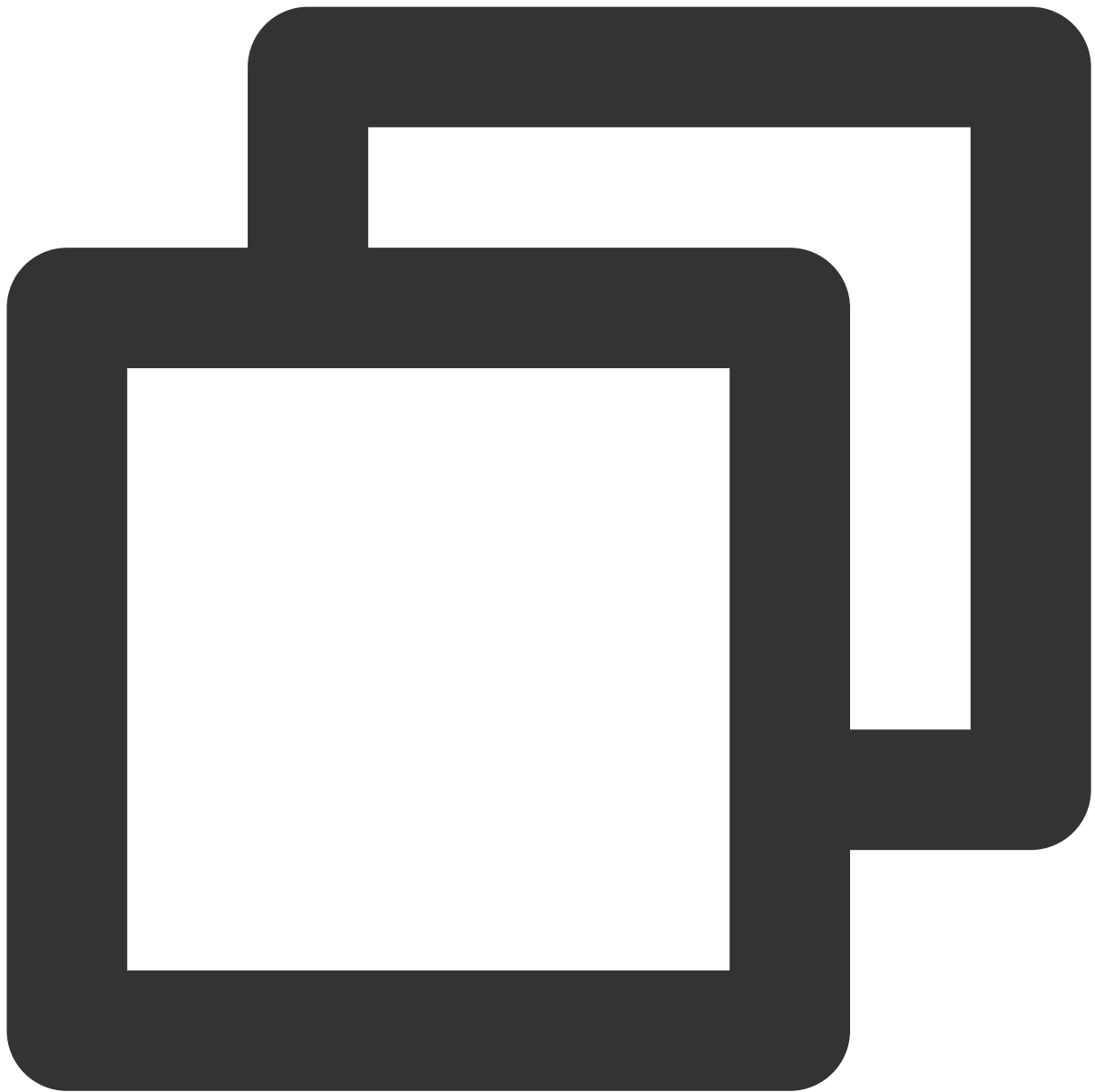
Prerequisites

You have created a TencentDB for MongoDB instance as instructed in [Creating TencentDB for MongoDB Instance](#).

The source and target databases must meet the requirements for the migration feature and version. For more information, see [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

You need to create a read-only account in the source instance for the migration; otherwise, the precheck will fail.



```
use admin
db.createUser({
  user: "username",
  pwd: "password",
  roles:
    [
      {role: "readAnyDatabase", db: "admin"},
      {role: "read", db: "local"}
    ]
})
```

Constraint

Incremental migration is not supported for self-built single-node instances as they have no oplogs.

Operation Restrictions

Below are the operation restrictions which will impact the success of the task:

Do not modify or delete user information (including username, password, and permissions) and port numbers in the source and target databases.

Do not clear oplogs in the source database.

Do not delete the target database `TencentDTSData` during data migration.

Manipulate data in the target database with caution during data migration; otherwise, data inconsistency may occur.

As DTS will filter out the DDL operations of the sharded cluster, do not perform DDL operations other than transactions on the source database during shard migration; otherwise, data inconsistency may occur.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, we recommend that you select full + incremental data migration to ensure data consistency in real time.

Supported SQL Operations

Note

Only replica set migration supports DDL operations, while shard migration will filter out DDL operations (except transactions).

Operation Type	Supported SQL Operations
DML	INSERT, UPDATE, DELETE
DDL	INDEX: createIndexes, createIndex, dropIndex, dropIndexes COLLECTION: createCollection, drop, collMod, renameCollection, convertToCapped DATABASE: dropDatabase, copyDatabase

Environment Requirements

Type	Environment Requirements

Requirements for the source database	<p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p> <p>The user provided by the source database must have the permission to read the database.</p> <p>The source database cannot have a database named <code>TencetDTSDData</code>.</p> <p>If the source database is in cluster mode, the balancer needs to be disabled before incremental sync.</p> <p>Oplogs can be obtained from the source database during full + incremental migration.</p>
Requirements for the target database	<p>The target database space must be at least 1.3 times the space of the collections to be migrated in the source database.</p> <p>The user provided by the target database must have the root permission.</p> <p>The target database cannot have a database named <code>TencetDTSDData</code>.</p> <p>The target database cannot have collections with the same name as those in the source database.</p> <p>If the source database is a sharded database, you need to correctly enter the mongos, config server, and mongod node information.</p> <p>The target database cannot have active businesses; otherwise, an alarm will be reported.</p> <p>The shard keys of the source and target databases must be the same; otherwise, an alarm will be reported.</p>

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task**.
2. On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Item	Description
Source Instance Type	Select the source database type, which cannot be changed after purchase. In this scenario, select MongoDB .
Source Instance Region	Select the region where the source database resides. If the source database is a self-built one, select a region nearest to it.
Target Instance Type	Select the target database type, which cannot be changed after purchase. In this scenario, select MongoDB .
Target Instance Region	Select the region where the target database resides.
Specification	Select the specification of the migration link based on your business needs. For the performance and billing details of different specifications, see Billing Overview .

3. On the **Set source and target databases** page, configure the task, source database, and target database.

Note

You need to create a read-only account in the source instance for the migration; otherwise, the precheck will fail.

Because there are many cross-scenario deployment modes and access types for the source database, and the migration steps for different scenarios are similar, the following only provides configuration examples for typical scenarios. For other scenarios, complete the configurations based on your needs.

Example 1: Migrate a local self-built MongoDB instance (sharded cluster) to a TencentDB instance over Direct Connect

Source Database Settings

Source Database Type *

MongoDB

Access Type *

Public Network

Self-Build on CVM

Direct Connect

VPN Access

Please add the DTS IP addresses to the security group allowlist in advance so that the connection is successful.

Architecture *

Replica Set

Cluster Migration

Single-Node

Region

South China(Guangzhou)

VPN Type

Cloud VPN Service

VPN Gateway *

Please select

VPC *

Please select VPC

Please select a subnet

Node - mongod *

Please enter the node info (IP:port) in the text box (separated by colon), and enter a new line such as 186.3.55.77:6379

Node - mongos *

IP

Enter the IP of mongos

Port

Enter the port of mongos

Node - Config Server *

IP

Enter the IP of Config Server

Port

Enter the port of Config Server

Authentication Required *

Yes

No

Authentication Database *

Please enter authentication database

Authentication Mechanism *

SCRAM-SHA-1

Account & Password
Selection *

Same account & password

Different accounts & passwords

Mongod, mongos, and config server roles use different accounts and passwords

Node - mongod *

Account

Enter the mongod account

Password

Enter the mongod password

Node - mongos *

Account

Enter the mongos account

Password

Enter the mongos password

Node - Config Server *

Account

Enter the config server account

Password

Enter the config server password

Test Connectivity

Setting Type	Configuration Item	Description
Task Settings	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: The task will be automatically started at the execution time you specify.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. This scenario takes Direct Connect as an example. For the preparations for different access types, see Overview . Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed on a CVM instance. For more information on CVM, see Cloud Virtual Machine Documentation . Direct Connect: The source database can be interconnected with VPCs through Direct Connect. For more information on Direct Connect, see Direct Connect Documentation . VPN Access: The source database can be interconnected with VPCs through VPN Connections. For more information on VPN Connections, see VPN Connections Documentation . Database: The source database is a TencentDB database. CCN: The source database can be interconnected with VPCs through CCN. For more information on CCN, see Cloud Connect Network Documentation .

		VPC: The source and target databases are both deployed in Tencent Cloud VPCs. For more information on VPC, see Virtual Private Cloud Documentation .
	Architecture	Select the architecture of the source database. This scenario takes Cluster migration as an example. The selected architecture type cannot be modified after the connectivity test is performed; otherwise, an error will be reported.
	VPC-Based Direct Connect Gateway	Only VPC-based Direct Connect gateway is supported when the access type is Direct Connect. Confirm the type of the network associated with the gateway.
	VPC	Select a VPC and subnet associated with the VPC-based Direct Connect gateway.
	Node – mongod	Enter the IP/domain name and port of the mongod node, such as 1xx.3.xx.77:6xx9. For multiple nodes, separate their IPs/domain names and ports by carriage return.
	Node – mongos	Enter the IP/domain name and port of the mongos node.
	Node – Config Server	Enter the IP/domain name and port of the config server
	Authentication Required	Select whether to check the security of username and password in the source database.
	Authentication Database	Enter the name of the database to be authenticated, that is, name of the database of the account executing the migration task, such as <code>admin</code> .
	Authentication Mechanism	SCRAM-SHA-1 is supported currently
	Account & Password Selection	If the three nodes of mongod, mongos, and config server in the source database have the same account and password, select Same account & password ; otherwise, select Different accounts & passwords and enter their respective accounts and passwords.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	In this scenario, select Database .

	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database account

Example 2: Migrate between two TencentDB instances

Source Database Settings

Source Database Type *

MongoDB

Access Type *

Public Network

Self-Build on CVM

Direct Connect

VPN Access

Please add the DTS IP addresses to the security group allowlist in advance so that the connect

Cross-/Intra-Account *

Intra-account

Cross-account

Region

South China(Guangzhou)

Database Instance *

Please select



Account *

Please enter account

Password *

Please enter password

Test Connectivity

Target Database Settings

Target Database Type *

MongoDB

Region

South China (Guangzhou)

Access Type *

Database

Database Instance *

Please select



Account *

Please enter account

Password *

Please enter password

Test Connectivity

Setting Type	Configuration Item	Description
Task Settings	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task

		<p>verification is passed.</p> <p>Scheduled execution: The task will be automatically started at the execution time you specify.</p>
	Tag	Tags are used to manage resources by category in different dimensions. If the existing tags do not meet your requirements, go to the console to create more.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	<p>If the source database is a TencentDB instance, select Database. For the preparations for different access types, see Overview.</p> <p>Public Network: The source database can be accessed through a public IP.</p> <p>Self-Build on CVM: The source database is deployed on a CVM instance. For more information on CVM, see Cloud Virtual Machine Documentation.</p> <p>Direct Connect: The source database can be interconnected with VPCs through Direct Connect. For more information on Direct Connect, see Direct Connect Documentation.</p> <p>VPN Access: The source database can be interconnected with VPCs through VPN Connections. For more information on VPN Connections, see VPN Connections Documentation.</p> <p>Database: The source database is a TencentDB database.</p> <p>CCN: The source database can be interconnected with VPCs through CCN. For more information on CCN, see Cloud Connect Network Documentation.</p> <p>VPC: The source and target databases are both deployed in Tencent Cloud VPCs. For more information on VPC, see Virtual Private Cloud Documentation.</p>
	Cross-/Intra-Account	<p>You need to configure this option when the access type is "Database".</p> <p>Intra-account: The source and target database instances belong to the same Tencent Cloud root account.</p> <p>Cross-account: The source and target database instances belong to different Tencent Cloud root accounts. For detailed directions, see Cross-Account TencentDB Instance Migration.</p>
	Database Instance	Select the instance ID of source database.
	Account	Account of the source TencentDB for MongoDB database, which must have the required permissions.

	Password	Password of the source TencentDB for MongoDB database.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database account

Example 3: Migrate a MongoDB instance (sharded cluster) of other cloud vendors to a TencentDB instance

Setting Type	Configuration Item	Description
Task Settings	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: The task will be automatically started at the execution time you specify.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	For a third-party cloud database, you can select Public Network generally or select VPN Access , Direct Connect , or CCN based on your actual network conditions. This scenario takes Public Network as an example. For the preparations for different access types, see Overview . Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed on a CVM instance. For more information on CVM, see Cloud Virtual Machine Documentation . Direct Connect: The source database can be interconnected with VPCs through Direct Connect. For more information on Direct Connect, see

		Direct Connect Documentation . VPN Access: The source database can be interconnected with VPCs through VPN Connections. For more information on VPN Connections, see VPN Connections Documentation . Database: The source database is a TencentDB database. CCN: The source database can be interconnected with VPCs through CCN. For more information on CCN, see Cloud Connect Network Documentation . VPC: The source and target databases are both deployed in Tencent Cloud VPCs. For more information on VPC, see Virtual Private Cloud Documentation .
	Architecture	Select the architecture of the source database. This scenario takes Cluster migration as an example. The selected architecture type cannot be modified after the connectivity test is performed; otherwise, an error will be reported.
	Node – mongod	Enter the IP/domain name and port of the mongod node, such as 1xx.xx.55.77:6xx9. For multiple nodes, separate their IPs/domain names and ports by carriage return.
	Node – mongos	Enter the IP/domain name and port of the mongos node.
	Node – Config Server	Enter the IP/domain name and port of the config server
	Authentication Required	Select whether to check the security of username and password in the source database.
	Authentication Database	Enter the name of the database to be authenticated, that is, name of the database of the account executing the migration task, such as <code>admin</code> .
	Authentication Mechanism	SCRAM-SHA-1 is supported currently
	Account & Password Selection	If the three nodes of mongod, mongos, and config server in the source database have the same account and password, select Same account & password ; otherwise, select Different accounts & passwords and enter their respective accounts and passwords.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.

	Access Type	In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database account

4. Test the connectivity between the source and target databases.

If the connectivity test fails, fix the problem as instructed in [Failed Connectivity Test](#).

5. On the **Set migration options and select migration objects** page, set the migration options and migration objects. You can select specified databases and collections.

Configuration Item	Description
Migration Type	<p>Select an option as needed.</p> <p>Full migration: The entire database will be migrated. The data to be migrated only includes the existing data of the source database. The incremental data written to the source database after the task is initiated won't be migrated.</p> <p>Full + incremental migration: The data to be migrated includes the existing data of the source database and the incremental data written to the source database after the task is initiated. If data needs to be written to the source database during migration and you want to smoothly migrate the data in a non-stop manner, select this option.</p>
Consistency Check	<p>If Full + incremental migration is selected, you can perform a data consistency check to carefully compare the data in the source and target databases after migration and determine the cutover time.</p> <p>If Full check is selected, DTS will automatically trigger a data consistency check task when the migration task enters the "incremental sync" stage, in which the time lag between the target and the source databases is 0 seconds and the data gap between them is 0 MB.</p> <p>If Full check is not selected, you can manually trigger a check task when the task enters the incremental sync stage. For more information, see Creating Data Consistency Check Task.</p>
Migration Object	<p>Entire instance: Migrate the entire database instance, including the metadata definitions of roles and users but excluding system databases such as system objects in PostgreSQL.</p> <p>Specify object: Migrate specified objects.</p>
Specify object	Select the objects to be migrated in the Source Database Object box and move them to the Selected Object box.

6. On the **Verify task** page, complete the precheck and click **Start Task**.

If the verification fails, troubleshoot as instructed in [Database Connection Check](#)

(<https://intl.cloud.tencent.com/document/product/571/42552173c84487192085ccd5cece97b25dbfcf>) and initiate the verification again.

Failed: It indicates that a check item fails and the task is interrupted. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

7. Return to the migration task list. After the incremental sync is 100% complete, click **Complete** in the **Operation** column to complete the migration task.

Select **Full migration**: The task will stop automatically after it is completed.

Select **Full + incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, and the incremental sync will not stop automatically. You need to click **Complete** to manually stop it. Click **Complete** when the time lag between the source and target databases is 0 seconds.



8. (Optional) If you want to view/delete a task or perform other operations, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

9. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Processing for MongoDB Series Pre-Check Failure

MongoDB Connection Check

Last updated : 2024-07-08 20:23:27

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see [Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database](#).

The source IP addresses are blocked by the source database. For more information, see [Failed Connectivity Test > Source IP Addresses Blocked in Source Database](#).

The network port is closed. For more information, see [Failed Connectivity Test > Closed Network Port](#).

The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Database/Table Content Conflict Check

Last updated : 2024-07-08 20:23:27

Check Details

In MongoDB data migration scenarios, the target database can contain collections with the same name as those in the source database, but the collections must be empty.

Troubleshooting

If a conflict causes an error, you can delete collections with the same name in the target database or clear their data.

Source Database Node Role Check

Last updated : 2024-07-08 20:23:27

Check Details

Check requirements: in a MongoDB migration task, if the source database is a sharded database, you need to enter the `mongos` , `config server` , and `mongod` node information.

Check description: the information of the `mongos` , `config server` , and `mongod` nodes cannot be disordered; otherwise, data migration will also be disordered; for example, the `mongos` node information should not be entered in the box for the `mongod` node. Note that you only need to enter the information of one `mongod` node for each shard.

Fix

Enter the correct node information in the DTS task configuration items.

Enter only one `mongod` for each shard.

Oplog Check

Last updated : 2024-07-08 20:23:27

Check Details

Check requirements: oplogs can be obtained from the source database during full + incremental migration.

Check description: incremental migration requires oplogs for replay. If the `oplog.rs` or `oplog.$main` table does not exist in the source `local` database, oplogs cannot be obtained.

Fix

Start the source database as a replica set or in primary/secondary mode to ensure that oplogs can be generated for operations and recorded in the source `local` database.

Source/Target Database Account Permission Check

Last updated : 2024-07-08 20:23:27

Check Details

Check whether you have the operation permissions of the database as described below:

Permission requirements for data migration: [Migration from MongoDB to TencentDB for MongoDB](#).

Troubleshooting

If you don't have the operation permissions, get authorized based on the permission requirements in the check details, and run the verification task again.

Database Version Check

Last updated : 2024-07-08 20:23:27

The source and target database versions must be supported by MongoDB.

Database Capacity Check

Last updated : 2024-07-08 20:23:27

Check Details

In MongoDB data migration scenarios, the storage space of the target database needs to be at least 1.3 times the size of the collections to be migrated in the source database.

Troubleshooting

Delete some data from the target database to free up space.

Upgrade the storage specification of the target database to use an instance with a larger capacity for migration.

Target Database Load Check

Last updated : 2024-07-08 20:23:27

Check Details

Check requirements: DTS migration will increase the load in the target database. If there is a business running in the target database during migration, a verification warning will be triggered. It will not block the task but will affect the business. You need to assess and determine whether to ignore the warning.

Impact on business: MongoDB DTS uses logical sync for data migration, which will cause certain pressure on the CPU load of the target database. If there is a business running in the target database, you need to assess and initiate the migration task with caution.

Fix

Stop any business running in the target database and run the verification task again.

ShardKey Check

Last updated : 2024-07-08 20:23:27

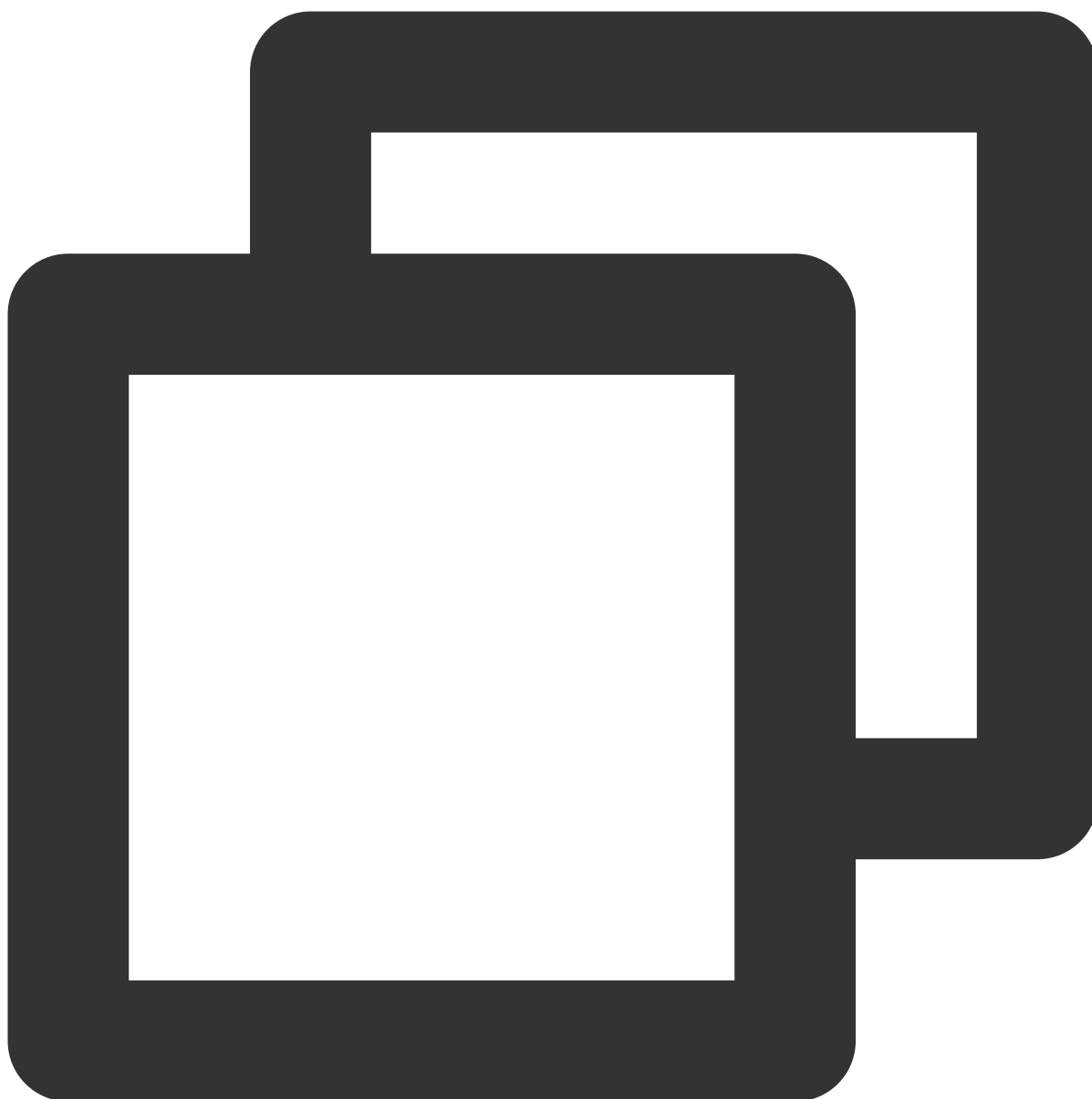
Check Details

Check requirement: if the target database is a sharded instance, you can preset the shardkey in it. If the table shardkeys in the source and target databases are different, an warning will be triggered. It will not block the task but will affect the business. You need to assess and determine whether to ignore the warning.

Impact on the business: if some shardkeys are different, the migration or sync task will fail.

Fix

If the target database has preset shardkeys, run the following command to shard the source database:



```
sh.shardCollection("<database>.<collection>", { <shard key> : "hashed" } , false, {
```

Run the verification task again.

Source Database Balancer Check

Last updated : 2024-07-08 20:23:27

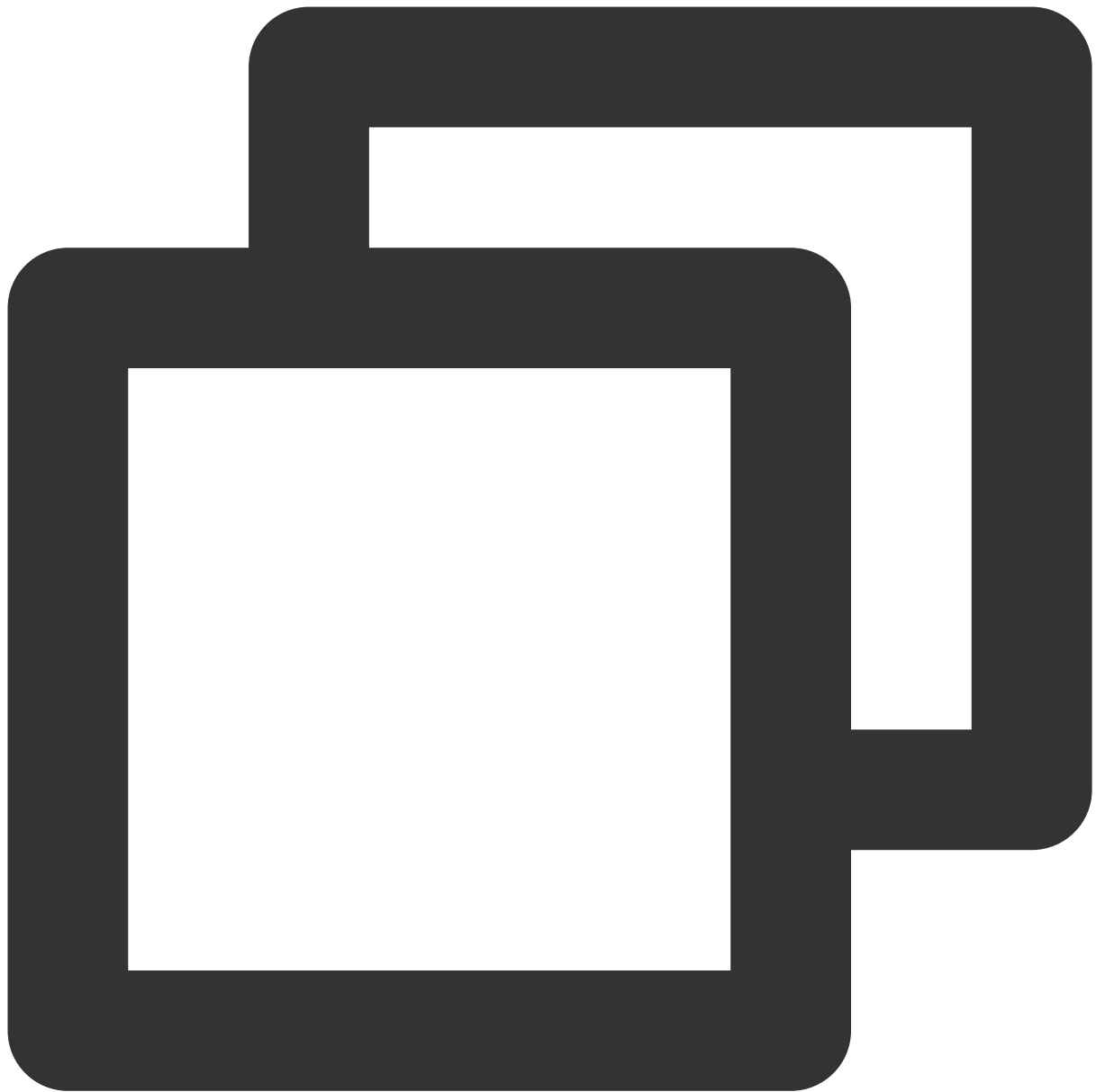
Check Details

Check requirements: if the source database is a sharded instance, you need to disable the balancer in it before you can start migration.

Check description: an incremental migration task will get the oplog. If the balancer is enabled, `moveChunk` in the source database may cause data inconsistency in the target database.

Fix

1. Log in to the source database.
2. Run the following command to disable the source database balancer:



```
sh.stopBalancer()  
sh.getBalancerState()
```

3. Run the verification task again.

Time Series Collection Verification

Last updated : 2024-07-11 15:14:32

Verification Details

MongoDB 5.0 and later versions support time series collection. When users migrate from version 5.0 and later to an earlier version, if the source database has time series collection, this verification item fails.

Fixing Solution

In scenes of migrating from version 5.0 and later to earlier versions, when users configure the task and select migration objects, they only choose the non-time series collection.

Compression Algorithm Verification

Last updated : 2024-07-08 20:23:27

Verification Details

Verify whether the compression algorithms used by the source and target databases are the same. If they are different, a warning is generated. The warning will not block the migration, and the user can ignore the warning to proceed with the task.

Note that when the compression algorithm used by the target database is verified, a random system table will be selected for verification. Since the compression algorithm of the system table does not change after modification, the warning might be inaccurate. If you are sure that the new compression algorithm is used in the target database, you can ignore the warning.

Fixing Solution

The disk size occupied by the same data varies under different compression algorithms. If the user wants the target database to use the same compression algorithm as the source database, modify the compression algorithm of the target database.

MongoDB Data Consistency Verification

Description of Consistency Verification Function

Last updated : 2024-07-08 20:23:27

Overview

During data consistency check, DTS compares the collection data between the source and target databases and outputs the comparison result and inconsistency details for you to perform a business cutover stably and reliably.

Notes

1. Data consistency check compares only the objects selected in the source database and objects migrated to the target database. If you write data into the target database during migration, then the written data will not be included in the consistency check.
2. A data consistency check task may increase the load in the source database instance. Therefore, you need to perform such tasks during off-peak hours.
3. A data consistency check task can be executed repeatedly, but one DTS instance can initiate only one such task at any time.
4. If you choose to **complete** or **terminate** a DTS task before a data consistency check task is completed, the check task will fail.
5. When creating a consistency check, the system will automatically create the `dts_verify_result` library on the target end to record content related to the consistency check. The table styles created under the `dts_verify_result` library are as follows:
 - `diff_5xxxxxxxx4231`: Saves inconsistent data detected
 - `diff_meta_5xxxxxxxx4231`: Saves inconsistent metadata detected
 - `result_5xxxxxxxx4231`: Records the results after phase validation
 - `status_5xxxxxxxx4231`: Records validation progress

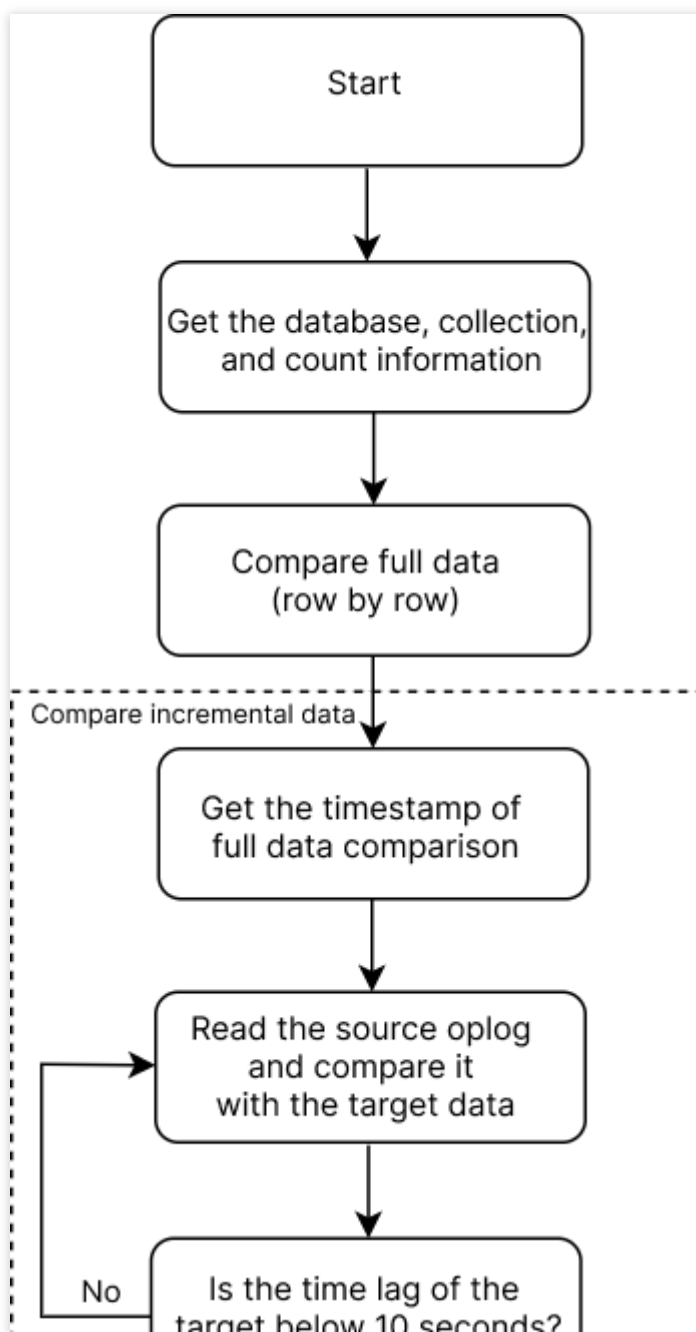
Restrictions

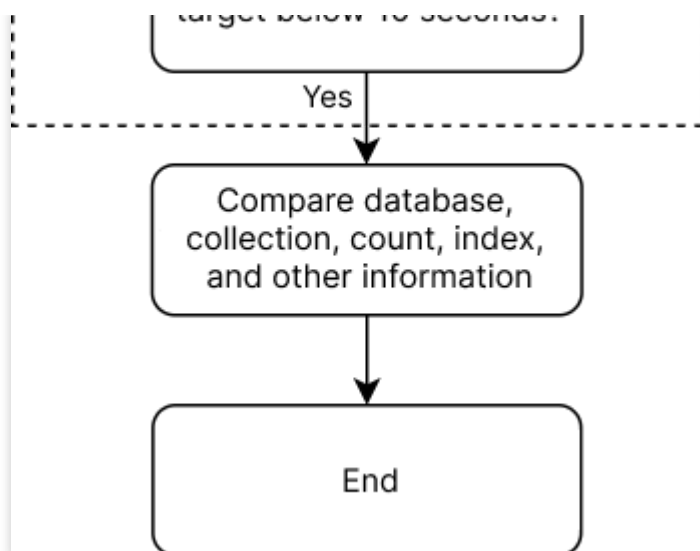
Currently, check tasks are imperceptible to the DDL operations. If you perform DDL operations in the source database during migration, the check result will be inconsistent with the actual data, and you need to initiate another check task

to get the accurate comparison result.

Check Scheme

DTS checks and compares all the data migrated during full migration and incremental migration from the source database. A full data check compares the data in the source and target databases row by row. Once the thread of the incremental data check finds that the full data comparison is completed, it immediately starts the incremental data check to get the start timestamp of the full data check, get the incremental oplog in the source database in a loop, and compare the differences between the source and target databases. When the time lag of data in the source and target databases is below 10 seconds, the comparison ends, and the check result is output.





Creating Data Consistency Check Task

Last updated : 2024-07-08 20:23:27

Overview

During data consistency check, DTS compares the table data between the source and target databases and outputs the comparison result and inconsistency details for you to determine the business cutover time. A data consistency check task is independent of the normal business in the source database or other DTS tasks.

Data consistency check tasks can be triggered automatically or created manually.

Automatic triggering: During migration task configuration, if **Full check** is selected for **Data Consistency Check**, a data consistency check task will be triggered automatically when the migration task enters the **incremental sync** step.

Manual creation: When the DTS task enters the **incremental sync** step, you can manually create one or multiple data consistency check tasks.

Triggering a data consistency check task automatically

On the **Set migration options and select migration objects** page of a [data migration task](#), select **Full check** for **Data Consistency Check**. In this way, a data consistency check task will be triggered automatically when the migration task enters the **incremental sync** step.

Note:

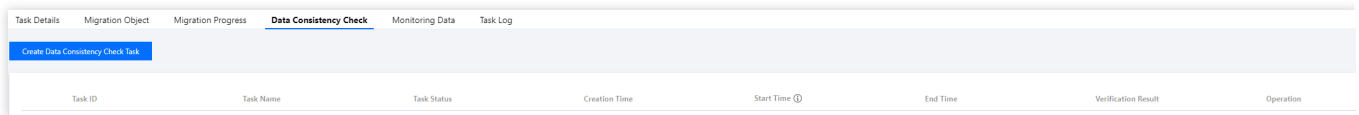
In this case, the full data and all the database information will be checked by default. If you need to filter check objects, create a data consistency check task manually.

Creating a data consistency check task manually

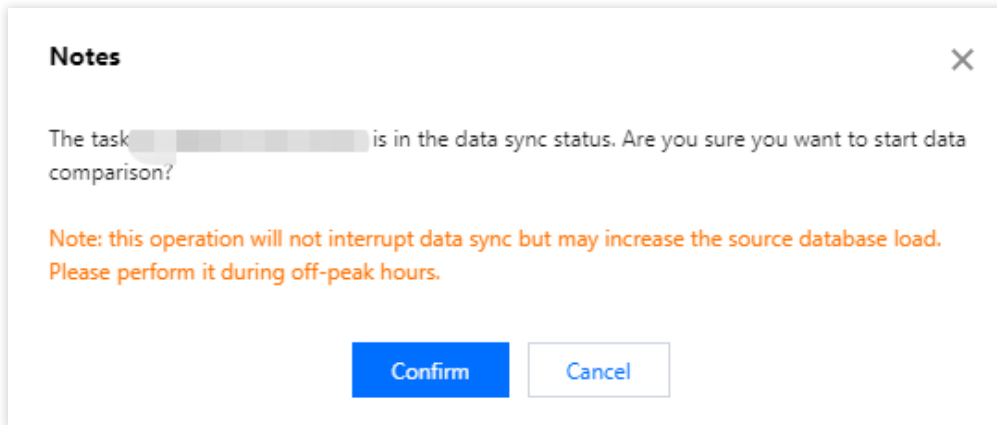
1. Log in to the [DTS console](#).
2. On the **Data Migration** page, select the target migration task and click **More > Create Data Consistency Check Task** in the **Operation** column.
3. Click **Create Data Consistency Check Task**.

Note:

A data consistency check task can be created only when the corresponding DTS task is in the **incremental sync** step. If the button is grayed out, the DTS task status does not meet the requirement; for example, the task has not entered the **incremental sync** step, has failed, or is terminated.



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



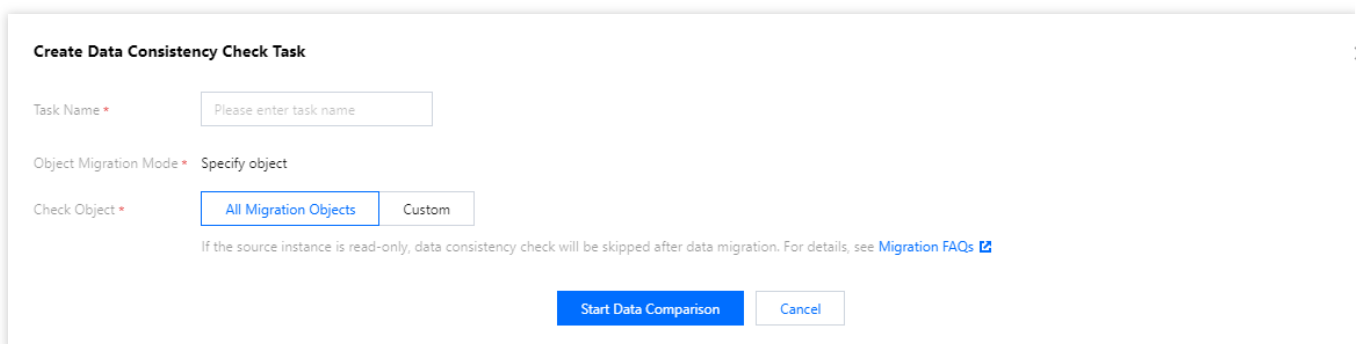
5. After configuring the data consistency check parameters, click **Start Data Comparison**.

Check Object: Select **All Migration Objects** or **Custom**.

Database Information: Select **Index**, **Shard key** (if both the source and target databases are sharded clusters), or **Database and table** for check.

Data Check: The **Row count check** option compares the number of data rows in the source and target databases. The **Content check** option compares the data content of the source and target databases.

Sampling: In scenarios with a high data volume, extracting all the data for check may increase the load of the source database. If you select **Content check**, you can set an appropriate percentage based on your business conditions to extract a certain proportion of data for comparison.



Viewing the data consistency check result

1. On the migration task homepage, view whether the check result is **Consistent** or **Inconsistent** in the **Last Check Result** column. Click **View More** to enter the **Verification Details** page.

2. Click **View** to view the check result.

If the data is consistent, the result will be like:

Inconsistency check result:

For inconsistent data, you need to manually confirm the corresponding data content of the source and target databases as prompted. For more details, please refer to [Common Consistency Check Issues](#).

Common Consistency Verification Issues

Last updated : 2024-07-08 20:23:27

Overview

During data consistency check, DTS compares the table data between the source and target databases and outputs the comparison result and inconsistency details for you to determine the business cutover time. A data consistency check task is independent of the normal business in the source database or other DTS tasks.

Data consistency check tasks can be triggered automatically or created manually.

Automatic triggering: During migration task configuration, if **Full check** is selected for **Data Consistency Check**, a data consistency check task will be triggered automatically when the migration task enters the **incremental sync** step.

Manual creation: When the DTS task enters the **incremental sync** step, you can manually create one or multiple data consistency check tasks.

Triggering a data consistency check task automatically

On the **Set migration options and select migration objects** page of a [data migration task](#), select **Full check** for **Data Consistency Check**. In this way, a data consistency check task will be triggered automatically when the migration task enters the **incremental sync** step.

Note:

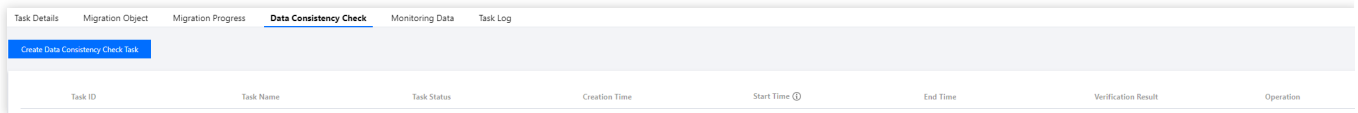
In this case, the full data and all the database information will be checked by default. If you need to filter check objects, create a data consistency check task manually.

Creating a data consistency check task manually

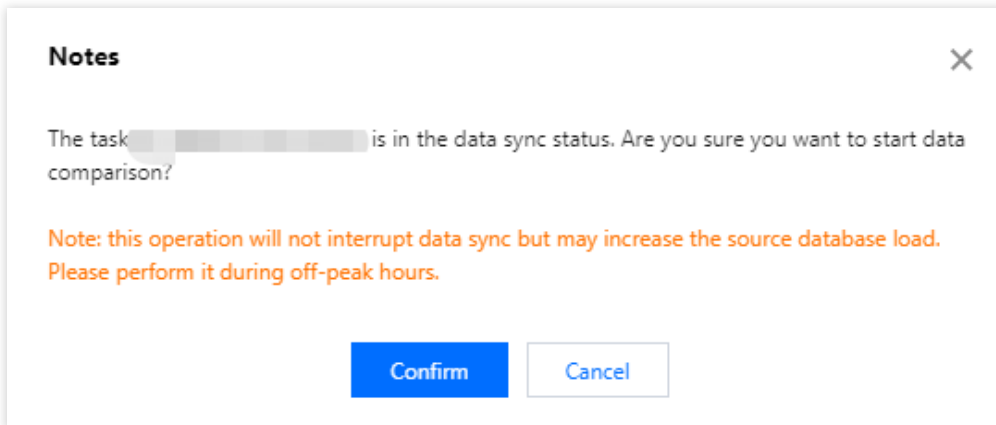
1. Log in to the [DTS console](#).
2. On the **Data Migration** page, select the target migration task and click **More > Create Data Consistency Check Task** in the **Operation** column.
3. Click **Create Data Consistency Check Task**.

Note:

A data consistency check task can be created only when the corresponding DTS task is in the **incremental sync** step. If the button is grayed out, the DTS task status does not meet the requirement; for example, the task has not entered the **incremental sync** step, has failed, or is terminated.



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



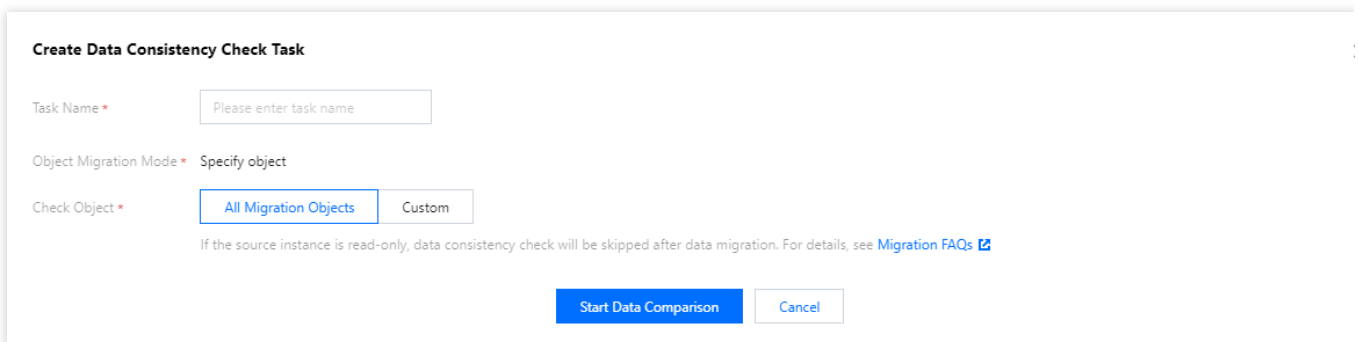
5. After configuring the data consistency check parameters, click **Start Data Comparison**.

Check Object: Select **All Migration Objects** or **Custom**.

Database Information: Select **Index**, **Shard key** (if both the source and target databases are sharded clusters), or **Database and table** for check.

Data Check: The **Row count check** option compares the number of data rows in the source and target databases. The **Content check** option compares the data content of the source and target databases.

Sampling: In scenarios with a high data volume, extracting all the data for check may increase the load of the source database. If you select **Content check**, you can set an appropriate percentage based on your business conditions to extract a certain proportion of data for comparison.



Viewing the data consistency check result

1. On the migration task homepage, view whether the check result is **Consistent** or **Inconsistent** in the **Last Check Result** column. Click **View More** to enter the **Verification Details** page.

dts

14 / 41

Immediate execution

Xlarge

Pay as you go

Completed : Consistent
View More

Database

Source: cdb-
Target: cdb-

2021-12-01 19:29:53

Done
Creat
More

NewDTS

Status: Prepared

Start: 2021-12-01 19:38:46

End:

Data lag between target and source databases: 0 MB

Time lag between target and source databases: 0s

2. Click **View** to view the check result.

Task Details

Migration Object

Migration Progress

Data Consistency Check

Monitoring Data

Task Log

Create Data Consistency Check Task

Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	End Time	Verification Result	Operations
dts- <div></div>	dts-test	Completed	2021-12-16 14:31:30	2021-12-16 14:31:33	2021-12-16 14:32:32	Consistent	View

If the data is consistent, the result will be like:

Overview

Status Completed

Compare Conclusion Consistent

Total Tables

1

Checked Tables

1

Inconsistent Tables

0

Inconsistent Chunks

0

Details

Database

Data Table

Chunk ID

Index Name

Last Index Key

First Index Key

Source Data Value

Target Data Value

Time Consumed (ms)

Completion Time

No data

Total items: 0

20 / page

1

1

Details of Unchecked Tables

Database

Data Table

Reason

No data yet

Total items: 0

20 / page

1

2

Inconsistency check result:

Note :

For inconsistent data, you need to manually confirm the corresponding data content of the source and target databases as prompted. For more details, please refer to [Common Consistency Check Issues](#).

Migrating to SQL Server

Migration from SQL Server to TencentDB for SQL Server

Last updated : 2024-07-08 20:25:43

This document describes how to use the data migration feature of DTS to migrate data from SQL Server to TencentDB for SQL Server.

Notes

When DTS performs full data migration, it will occupy some resources in the source instance, which may increase the load of the source instance and the database pressure. If your database has low configurations, we recommend that you migrate the data during off-peak hours.

Full migration is implemented with tables locked, during which write operations will be blocked for seconds.

Prerequisites

You have created a TencentDB for SQL Server instance as instructed in [Creating TencentDB for SQL Server Instance](#).

The source and target databases must meet the requirements for the migration feature and version as instructed in [Databases Supported by Data Migration](#).

You have completed all the preparations as instructed in [Overview](#).

The service where the source database is located must open the file sharing port 445.

The source database must be set to "full recovery mode", and we recommend you make a full backup before migration.

The local disk space of the source database must be large enough, so that the remaining free space can fit the size of the database to be migrated.

If the source instance is not a TencentDB for SQL Server or TencentDB for SQL Server Basic Edition instance (such as a public network/CVM-based self-built instance or instance on another cloud), an account with the `sysadmin` permission needs to be used in the target database for migration, and the `xp_cmdshell` stored procedure must be able to run. If the source instance is a TencentDB for SQL Server High-Availability Edition or Cluster Edition instance, there is no permission restriction on the target database account.

You need to change the account to **Local System** for SQL service startup in the source database. The source database account is unrestricted but needs to have the sysadmin permissions.

Start the SQL service in the source database, select **Built-in account** in the **Log on as** section, and select **Local System**.

Note:

After modifying the account, you need to restart the SQL Server service.

Application restrictions

Only one migration task can be initiated at any time for the same source instance.

Currently, cross-region migration is supported between the Chinese mainland and Hong Kong (China) but not between other regions.

Only database-level migration is supported (i.e., all objects in the database must be migrated together), while single-table migration is not supported.

Logins, job agents, triggers, and database links (link server) at the instance level cannot be migrated.

Operation restrictions

Do not modify or delete user information (including username, password, and permissions) in the source and target databases and port numbers during migration; otherwise, the migration task will fail.

Do not perform transaction log backup during incremental sync; otherwise, the transaction log will be truncated and become discontinuous.

If you only perform full data migration, do not write new data into the source instance during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.

For full + incremental data migration, after you click **Complete** and the task status becomes **Completed**, do not write new data to the source database. We recommend you stop writing for two minutes; otherwise, the data in the source and target databases may be inconsistent.

Supported SQL operations

Operation Type	Supported SQL Operations
DML	INSERT, UPDATE, DELETE, REPLACE
DDL	TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE VIEW: CREATE VIEW, ALTER VIEW, DROP VIEW INDEX: CREATE INDEX, DROP INDEX

DATABASE: CREATE DATABASE, ALTER DATABASE, DROP DATABASE

Environment requirements

Note:

The system will automatically check the following environment requirements before starting a migration task and report an error if a requirement is not met. If you can identify the failed check item, fix it as instructed in [Check Item Overview](#); otherwise, wait for the system verification to complete and fix the problem according to the error message.

Type	Environment Requirements
Requirements for the source database	<p>The service where the source instance is located must open the file sharing port 445.</p> <p>The source and target databases can be connected.</p> <p>The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected.</p>
Requirements for the target database	<p>Only migration from Basic Edition to High Availability Edition (including Dual-Server High Availability Edition and Cluster Edition) is supported, and the version number of the target instance must be later than of the source instance.</p> <p>The target database cannot have the same name as the source database.</p> <p>The disk space of the target database must be at least 1.5 times the size of the source database.</p> <p>The target database cannot have access requests or active businesses; otherwise, the migration will fail.</p>

Directions

- Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** to enter the **Create Migration Task** page.
- On the **Create Migration Task** page, select the types, regions, and specifications of the source and target instances and click **Buy Now**.

Configuration Item	Description
Source Instance Type	Select the source database type, which cannot be changed after purchase. In this document, select SQL Server .
Source Instance Region	Select the source database region. If the source database is a self-built one, select a region nearest to it.

Target Instance Type	Select the target database type, which cannot be changed after purchase. In this document, select SQL Server .
Target Instance Region	Select the target database region.
Specification	Select the specification of the migration link based on your business conditions. For the performance and billing details of different specifications, see Billing Overview .

3. On the **Set source and target databases** page, configure the task, source database, and target database settings. After the source and target databases pass the connectivity test, click **Create**.

Note:

If the connectivity test fails, troubleshoot as prompted or as instructed in [Database Connection Check](#) and try again.

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	Immediate execution: The task will be started immediately after the task verification is passed. Scheduled execution: You need to configure a task execution time and the task will be started automatically then.
	Tag	Tags are used to manage resources by category in different dimensions. If the existing tags do not meet your requirements, go to the console to create more.
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The source database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database . For the preparations for different access types, see Overview . Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance. Direct Connect: The source database can be interconnected with VPCs through Direct Connect . VPN Access: The source database can be interconnected with VPCs through VPN Connections . Database: The source database is a TencentDB instance.

		CCN: The source database can be interconnected with VPCs through CCN . For a third-party cloud database, you can select Public Network generally or select VPN Access , Direct Connect , or CCN based on your actual network conditions.
	Cross-/Intra-Account	Intra-account: The source and target database instances belong to the same Tencent Cloud root account. Cross-account: The source and target database instances belong to different Tencent Cloud root accounts. The following uses migration under the same account as an example.
	Database Instance	Select the instance ID of source database.
	Account	Account of the source SQL Server database, which must have the required permissions.
	Password	Password of the source SQL Server database account.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Account	Account of the target database, which must have the required permissions.
	Password	Password of the target database.

4. On the **Set migration options and select migration objects** page, configure the migration type and objects and click **Save**.

Configuration Item	Description
Migration Type	Select an option based on your scenario. Full migration: The entire database will be migrated. The migrated data will only be existing content of the source database when the task is initiated but not include the incremental data written to the source database after the task is initiated. Full + Incremental migration: The migrated data will include the existing content of the source database when the task is initiated as well as the incremental data written to the source

	database after the task is initiated. If there are data writes to the source database during migration, and you want to smoothly migrate the data in a non-stop manner, select this option.
Specify object	Only database-level migration is supported; that is, all objects in the specified database must be migrated together. Select the database to be migrated in Source Database Object and move it to the Selected Object box.

5. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in [Check Item Overview](#) and initiate the verification again.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

6. Return to the data migration task list, and you can see that the task has entered the **Preparing** status. After 1–2 minutes, the data migration task will be started.

Select **Full migration**: Once completed, the task will be stopped automatically.

Select **Full + Incremental migration**: After full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data sync.

Manually complete incremental data sync and business switchover at appropriate time.

After the migration stage status becomes **Prepared**, stop writing data to the source database for a few minutes and then manually complete incremental sync.

7. (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

8. After the migration task status becomes **Task successful**, you can formally cut over the business. For more information, see [Cutover Description](#).

Processing for SQL Server Pre-Check Failure Migration Parameter Check

Last updated : 2024-07-08 20:25:43

Check Details

The target instance cannot have databases with the same name as those in the source database; otherwise, an error will be reported.

Troubleshooting

If an error is reported, you need to delete the database with the same name from the target instance or rename it.

Migration Network Check

Last updated : 2024-07-08 20:25:43

Check Details

Check whether the internal migration network is connected.

Fix

If an error is reported during the migration network check, [submit a ticket](#) for assistance.

Source Library Connectivity Check

Last updated : 2024-07-08 20:25:43

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see [Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database](#)

The source IP addresses are blocked by the source database. For more information, see [Failed Connectivity Test > Source IP Addresses Blocked in Source Database](#).

The network port is closed. For more information, see [Failed Connectivity Test > Closed Network Port](#).

The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Local Disk Space Check

Last updated : 2024-07-08 20:25:43

Check Details

Check whether the disk space on the server of the source database is sufficient. During migration of a self-built source database, at least 50 GB disk space should be reserved on the server of the source database. We recommend you reserve a space at least 1.5 times the size of the migration data to avoid errors during migration.

Fix

Delete some data from the server of the source database to free up sufficient space.

Database Version Check

Last updated : 2024-07-08 20:25:43

Only migration from Basic Edition to High Availability Edition (including Dual-Server High Availability Edition and Cluster Edition) is supported, and the version number of the target database must be later than that of the source database. Cross-major version migration is not supported.

User Permission Check

Last updated : 2024-07-08 20:25:43

Check Details

Check whether you have the operation permissions of the database as instructed in [Migration from SQL Server to TencentDB for SQL Server](#).

Troubleshooting

If you don't have the operation permissions, get authorized based on the permission requirements in the check details and run the verification task again.

Source Database Existence Check

Last updated : 2024-07-08 20:25:43

Check Details

Check whether the source database to be migrated exists, and if not, an error will be reported.

Fix

Do not delete the source database during migration; otherwise, you need to create a migration task again.

Target Database Existence Check

Last updated : 2024-07-08 20:25:43

Check Details

Check whether the target database exists, and if not, an error will be reported.

Fix

Do not delete the target database during migration; otherwise, you need to create a migration task again.

Migrating to TencentDB for Redis

Migration from Redis to TencentDB for Redis

Last updated : 2024-07-08 20:28:47

Background Overview

DTS is a data transmission service that integrates such features as data migration, sync, and subscription. It helps you migrate your databases to the cloud without interrupting your business and build a high-availability database disaster recovery architecture through real-time sync channels. Its data subscription feature meets your requirements for commercial data mining and async business decoupling.

DTS for Redis currently supports the data migration feature for you to migrate data to TencentDB in a non-stop manner at a time. In addition, in its full + incremental data migration mode, historical data in the source database written before migration and incremental data written during migration can be migrated together.

Use Cases

DTS for Redis data migration supports source and target databases in the following deployment modes:

Note:

To try out the Redis to TencentDB for Redis/KeeWiDB migration feature of NewDTS, [submit a ticket](#) for application.

Source	Target	Description
Self-built Redis database in IDC and CVM	TencentDB for Redis and TencentDB for KeeWiDB	-
Third-party Redis database	TencentDB for Redis and TencentDB for KeeWiDB	The third-party cloud vendor must grant the <code>SYNC</code> or <code>PSYNC</code> command permission.
TencentDB for Redis	TencentDB for Redis and TencentDB for KeeWiDB	Data migration, version upgrade, and cross-region migration are supported between database instances under the same Tencent Cloud account.

Migration Support Description

Note:

For compatibility issues with migration from Standalone Edition to Memory Edition (Cluster Architecture), see [Check on Migration from Standard Architecture to Cluster Architecture](#).

Supported versions

Redis to TencentDB for Redis migration is supported for source Redis 2.8, 3.0, 3.2, 4.0, 5.0, and 6.0. To use it for v6.0, [submit a ticket](#) for application. We recommend that you migrate from an earlier version to a later version; otherwise, compatibility problems may occur.

Redis to TencentDB for KeeWiDB migration is supported for source Redis 4.0. To migrate to other versions, [submit a ticket](#) for application.

Standard architecture and cluster architecture instances can be migrated to each other; however, such heterogeneous migration may have compatibility problems.

Supported architectures include single-node, Redis cluster, Codis, and twemproxy.

Migration permission requirements: To migrate data through DTS, the source instance must support `SYNC` or `PSYNC` commands.

Supported networks

DTS supports data migration based on the public network, CVM instances, Direct Connect gateways, VPN, and CCN.

Public Network: The source database can be accessed through a public IP.

Self-Build on CVM: The source database is deployed in a [CVM](#) instance.

Direct Connect: The source database can be interconnected with VPCs through [Direct Connect](#).

VPN Access: The source database can be interconnected with VPCs through [VPN Connections](#).

CCN: The source database can be interconnected with VPCs through [CCN](#).

Limitations

To ensure the migration efficiency, cross-region migration of CVM-based self-built instances is not supported.

To migrate instances over the public network, make sure that the source instance can be accessed over the public network.

Only instances that are running normally can be migrated, while instances with no password initialized or with ongoing tasks cannot.

The target instance must be empty. During the migration process, the target instance will be read-only and cannot be written to.

After the successfully migrated data is verified by your business, you can disconnect the source instance and connect to the target instance. This document describes how to migrate data from Redis to TencentDB for Redis with DTS.

Environment Requirements

System check

Note:

The DTS system will check the following environment requirements before starting a migration task and report an error if a requirement is not met. You can also check them in advance. For more information on error handling, see [Check Item Overview](#).

Type	Environment Requirements
Requirements for the source database	<p>The source and target databases can be connected.</p> <p>The number of databases in the source database must be less than or equal to that in the target database.</p> <p>Redis to TencentDB for Redis migration is supported for source Redis 2.2.6 or later. Redis to TencentDB for KeeWiDB migration is supported for source Redis 4.0; to migrate to other versions, submit a ticket for application.</p> <p>The source database must be a replica node; otherwise, an alarm will be triggered.</p>
Requirements for the target database	<p>The target database version must be later than or equal to the source instance version; otherwise, an alarm will be triggered for compatibility problems during verification.</p> <p>The target database must have the latest proxy.</p> <p>For migration from Redis to TencentDB for Redis or TencentDB for KeeWiDB, the space of the target database must be at least 1.5 or 2 times the volume of the data to be migrated in the source database, respectively.</p> <p>The target database must be empty.</p>

Manual check

You should check and make sure that the following items are passed before the migration; otherwise, the migration may fail.

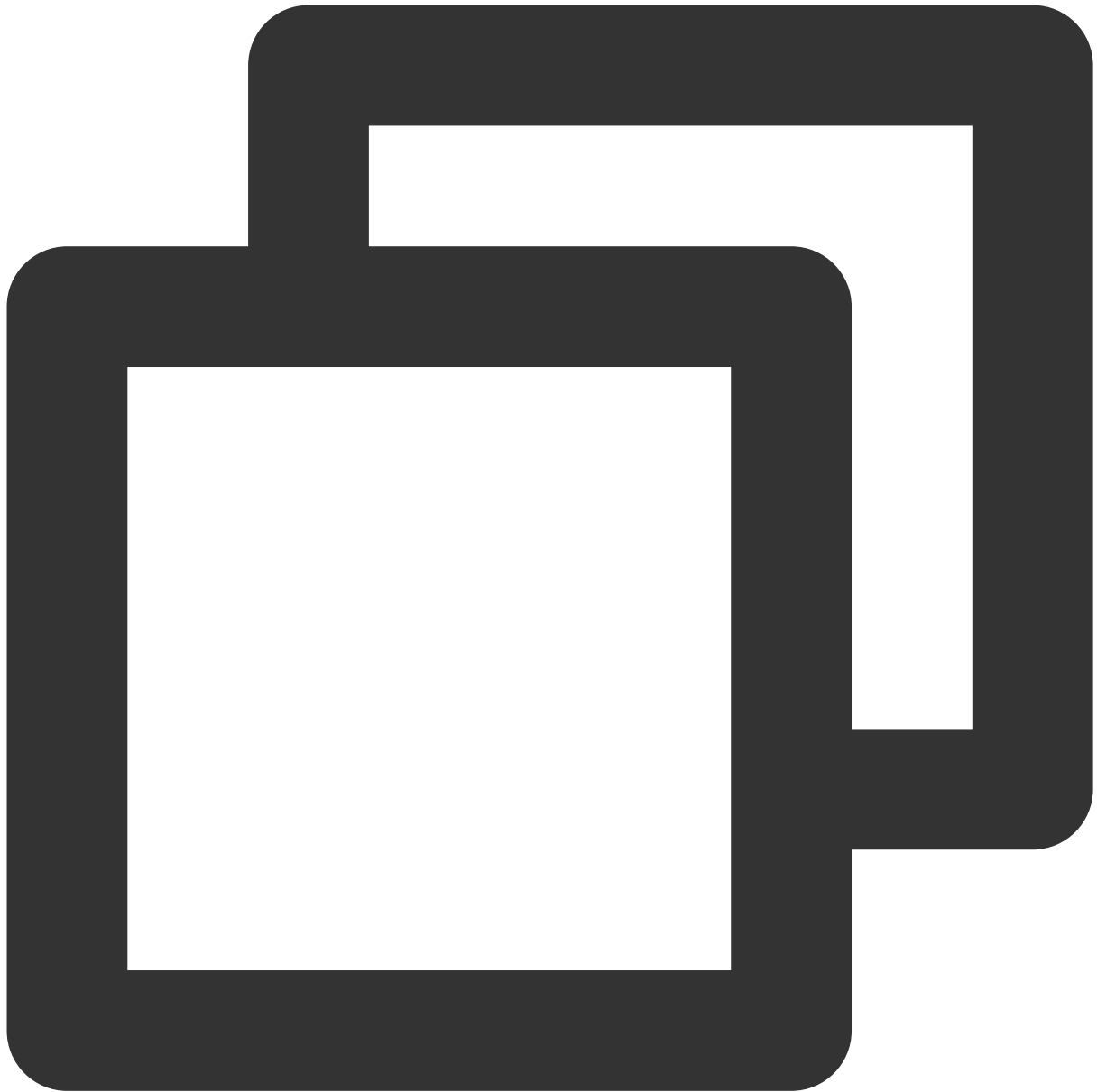
Big keys in the source database

Before the migration, check whether there are big keys in the source database. They may cause the buffer (client-output-buffer-limit) to overflow during the migration, leading to a migration failure.

For TencentDB databases, you can use the performance optimization feature of TencentDB for DBbrain to quickly analyze big keys. For more information, see [Memory Analysis](#).

For non-TencentDB databases, use RDBTools to analyze big keys in Redis.

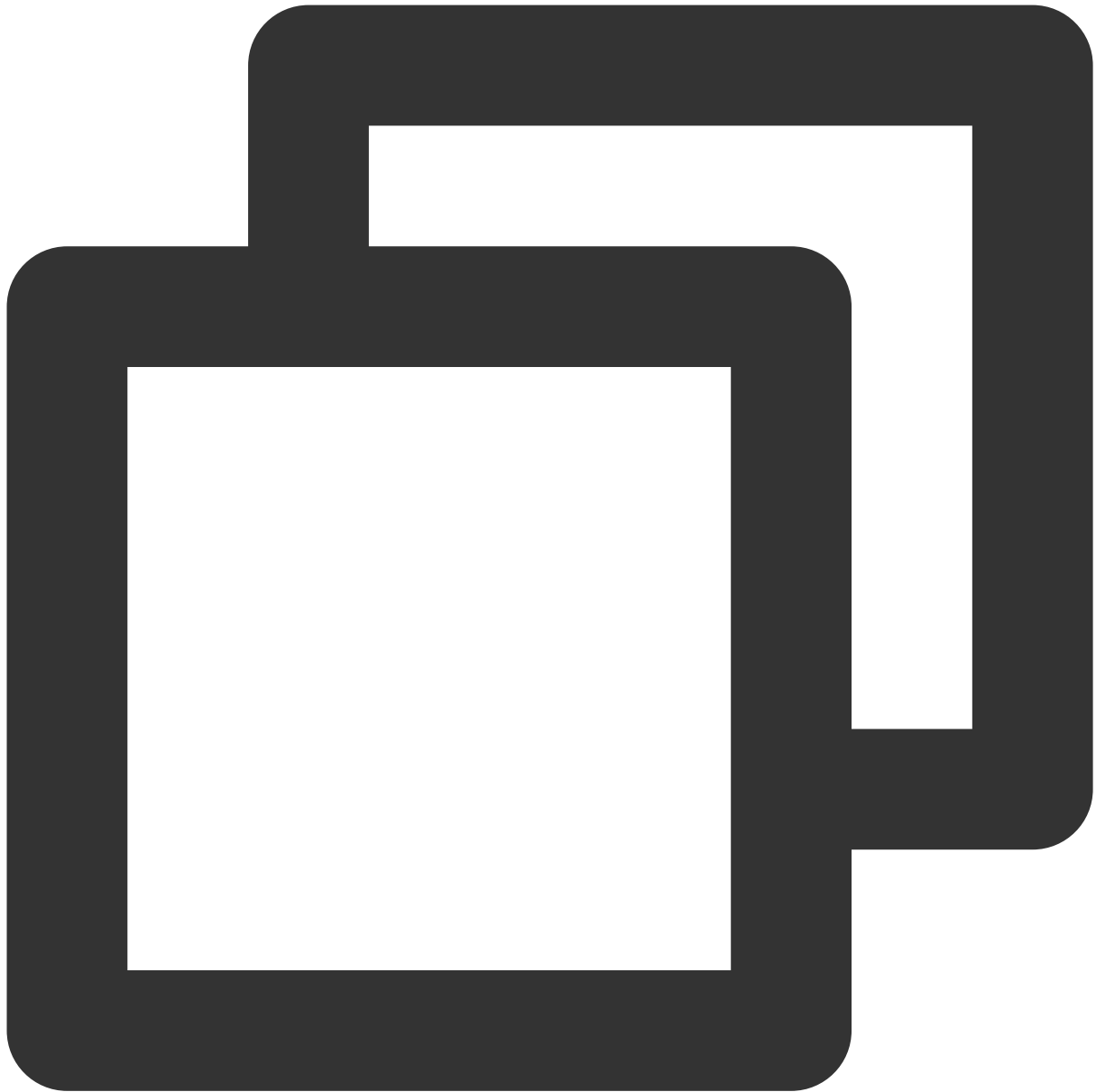
Evaluate large keys for splitting or cleaning. If you need to retain them, set the source buffer size (client-output-buffer-limit) to infinite.



```
config set client-output-buffer-limit 'slave 0 0 0'
```

Limit on the number of TCP connections in the source Linux kernel

If the number of concurrent business requests is high, check the limit on the number of connections in the Linux kernel before the migration. If this value is exceeded, the Linux server will actively disconnect from DTS.

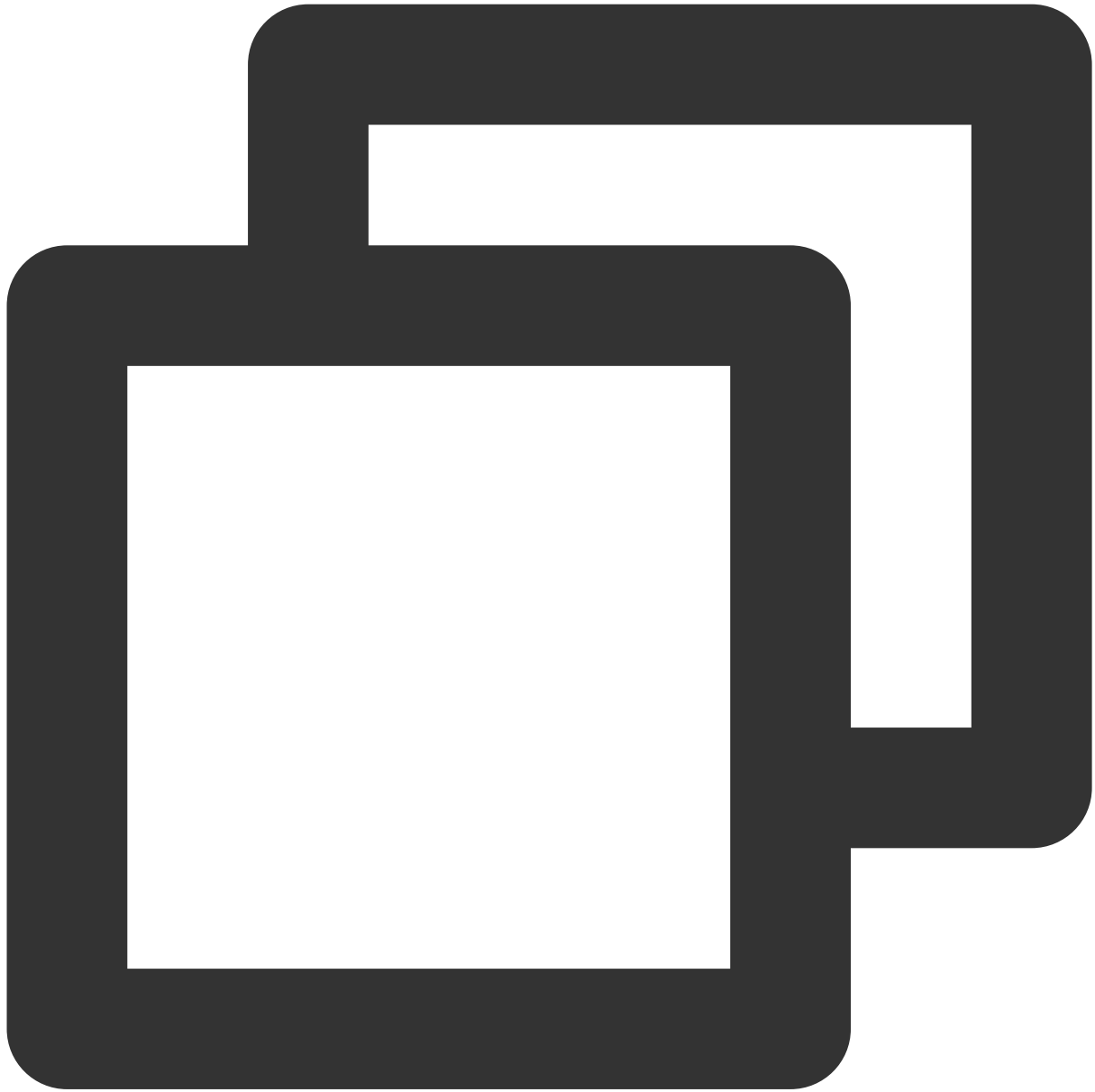


```
echo "net.ipv4.tcp_max_syn_backlog=4096" >> /etc/sysctl.conf
echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
echo "net.ipv4.tcp_abort_on_overflow=0" /etc/sysctl.conf
sysctl -p
```

Access permission of the source RDB file directory

Before the migration, check and make sure that the directory where RDB files are stored in the source database is readable; otherwise, the migration will fail.

If the RDB file directory is not readable, run the following command in the source database to set "diskless replication". Then, RDB files will be directly sent to DTS for storage, with no need to be stored in the source database first and then sent.



```
config set repl-diskless-sync yes
```

Command compatibility (for migration from standard architecture to cluster architecture)

The most challenging problem in migrating data from Standard Edition to Memory Edition (Cluster Architecture) is the command compatibility with usage specifications of Memory Edition (Cluster Architecture).

Multi-key operations

TencentDB for Redis Memory Edition (Cluster Architecture) only supports cross-slot multi-key access for `mget` , `mset` , `del` , and `exists` commands. In the source database, keys that need to engage in multi-key computing can be aggregated into the same slot through a hash tag. For more information on how to use hash tags, see [Redis cluster specification](#).

Transactional operations

Memory Edition (Cluster Architecture) supports transactions, but cross-slot access to keys in transactions is not supported.

Perform static and dynamic evaluations as instructed in [Check on Migration from Standard Architecture to Cluster Architecture](#).

Migration Directions

1. Create a migration task

1.1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and click **Create Migration Task** on the right.

1.2. On the **Create Migration Task** page, select the types and regions of the source and target databases and click **Buy Now**.

Configuration Item	Description
Billing Mode	Only pay-as-you-go is supported.
Source Instance Type	Select the source database type, which cannot be changed after purchase. Here, select Redis .
Source Instance Region	Select the source database region. If the source database is a self-built one, select a region nearest to it.
Target Instance Type	Select the target database type, which cannot be changed after purchase. Here, select Redis .
Target Instance Region	Select the target database region.
Migration Method	Select Migrate to the cloud .
Specification	Currently, only Xlarge is supported.

Quantity	You can purchase up to 10 migration tasks at a time.
----------	--

2. Set source and target databases

Configure the **Source Database Settings** and **Target Database Settings** and click **Test Connectivity**. After the test is passed, click **Save** to proceed to the next step.

Category	Configuration Item	Description
Task Configuration	Task Name	Set a task name that is easy to identify.
	Running Mode	<p>Select Immediate execution or Scheduled execution.</p> <p>To modify the scheduled task, you must click Scheduled start again after the verification is passed, so as to make the task start at the specified time.</p> <p>If the specified time has passed, the task will be started immediately. You can also click Immediate start to start the task immediately.</p>
Source Database Settings	Source Database Type	The source database type selected during purchase, which cannot be changed.
	Region	The region selected during purchase, which cannot be changed.
	Access Type	<p>For a third-party cloud database, you can select Public Network generally or select VPN Access, Direct Connect, or CCN based on your actual network conditions.</p> <p>In this scenario, select Public Network. For the preparations for different access types, see Overview</p> <p>Public Network: The source database can be accessed through a public IP.</p> <p>Self-Build on CVM: The source database is deployed in a CVM instance.</p> <p>Direct Connect: The source database can be interconnected with VPCs through Direct Connect.</p> <p>VPN Access: The source database can be interconnected with VPCs through VPN Connections.</p> <p>Database: The source database is a TencentDB instance.</p> <p>CCN: The source database can be interconnected with VPCs through CCN.</p>
	Node Type	<p>Single-Node Migration and Cluster Migration are supported.</p> <p>Cluster Migration is used as an example here. Currently, there are no limits on the number of shards and replicas in migration from Cluster Edition Redis to Cluster Edition Redis.</p>
	Node Info	Enter the addresses and passwords (IP:port:password or IP:port) of all shards of the source database cluster and separate the information of

		different nodes with line breaks. We strongly recommend you migrate data from a replica node of the source database to avoid any impact on business access to the source database.
Target Database Settings	Target Database Type	The target database type selected during purchase, which cannot be changed.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select Database .
	Database Instance	Select the instance ID of the target database.
	Password	Target database password.

3. Verify and start the task

On the task verification page, verify the task. After the verification is passed, click **Start Task**.

Failed: It indicates that a check item fails and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: It indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. You need to assess whether to ignore the alarm or fix the problem and continue the task based on the alarm message.

Return to the data migration task list, and you can see that the task has entered the **Preparing** status. After 1–2 minutes, the data migration task will be started. Then, full + incremental data will be migrated, that is, historical data in the source database written before migration and incremental data written during migration will be migrated together.

4. Complete the migration task

(1) (Optional) If you want to view, delete, or perform other operations on a task, click the task and select the target operation in the **Operation** column. For more information, see [Viewing Task](#).

(2) If the keys of the source and target databases are the same, click **Done** in the **Operation** column to stop the data migration task.

(3) After the migration task status becomes **Task successful**, verify the data in the target database. If the verification is passed, you can formally cut over the business. For more information, see [Cutover Description](#).

任务 ID / 名称	任务状态 / 进度	运行模式	规格	计费类型	最后一次校验结果	源库类型	目标库类型	源接入类型	地址	创建时间	操作
ots-xxxxxx	100 / 100  状态: 任务成功 开始: 2022-05-12 16:30:43 结束: 2022-05-12 16:46:28	立即执行	--	按量计费		Redis	Redis	云数据库	源: ots-xxxxxx 目标: ots-xxxxxx	2022-05-12 16:27:12	 

Event Alarming and Metric Monitoring

(1) DTS can automatically report event alarms triggered upon migration interruption to keep you informed of any exceptions. For detailed directions, see [Configuring Alarm Policy for Data Migration](#).

(2) DTS allows you to view the monitoring data of various metrics during migration to understand the performance metrics of the system. For more information, see [Viewing Monitoring Metric](#).

Fix for Verification Failure Network Connectivity

Last updated : 2024-07-08 20:28:47

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see [Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database](#).

The source IP addresses are blocked by the source database. For more information, see [Failed Connectivity Test > Source IP Addresses Blocked in Source Database](#).

The network port is closed. For more information, see [Failed Connectivity Test > Closed Network Port](#).

The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Source-Target Instance Version Compatibility

Last updated : 2024-07-08 20:28:47

Check Details

The migration from Redis to TencentDB for Redis is supported for source Redis 2.2.6 or later. The migration from Redis to TencentDB for KeeWiDB is supported for source Redis 4.0. To migrate to other versions, [submit a ticket](#) for application.

The target database version must be later than or equal to the source database version; otherwise, an alarm will be triggered for compatibility problems during verification.

The target database must have the latest proxy.

Troubleshooting

Check the source and target databases as instructed in [Databases Supported for Data Migration](#) and [Databases Supported for Data Sync](#). If the source or target database version is not supported, upgrade the target database version or use a database instance on a higher version.

Source Instance Parameter Check

Last updated : 2024-07-08 20:28:47

Check Details

In Redis migration scenarios, if the target database is TencentDB for Redis, the number of databases in the source instance must be not greater than that in the target instance.

Check whether the status of the source instance is normal.

Troubleshooting

Modify the number of databases of the source and target instances and restart the verification task.

Confirm the status of the source instance.

Target Instance Capacity Check

Last updated : 2024-07-08 20:28:47

Check Details

For migration from Redis to TencentDB for Redis/KeeWiDB, the space of the target database must be at least 1.5 or 2 times the volume of the data to be migrated in the source database, respectively.

Troubleshooting

Delete some data from the target database to free up space.

Upgrade the storage specification of the target database to use an instance with a larger capacity for migration.

Target Instance Status Check

Last updated : 2024-07-08 20:28:47

Check Details

Check whether the target database exists, and if not, an error will be reported.

Troubleshooting

Do not delete the target database during migration; otherwise, you need to create a migration task again.

Source Instance Type (Master or Replica)

Last updated : 2024-07-08 20:28:47

Check Details

The source database must be a replica node; otherwise, an alarm will be triggered. The alarm will not stop the task and can be ignored, but you should assess the impact of ignoring it.

During migration, DTS will perform a `BGSAVE` operation on the source database, which will use the database memory and resources. If the source database is a master node, business writes will be greatly affected.

If the source database is a TencentDB for Redis database, a replica node will be used for migration by default.

If the source database is a local self-built Redis database, the master node may be used for migration. If an alarm is triggered during the check, we recommend that you change the source database to a replica node.

Troubleshooting

Reconfigure the migration task by changing the parameters of the source database to the information of a replica node.

Migrating to KeeWiDB

Migration Redis to KeeWiDB

Last updated : 2024-07-08 20:28:47

The steps of data sync from Redis to TencentDB for KeeWiDB are basically the same as those described in [Migration from Redis to TencentDB for Redis](#).

Migration Tendis to KeeWiDB

Last updated : 2024-07-08 20:28:47

The steps of data migration from Tendis to TencentDB for KeeWiDB are basically the same as those described in [Migration from Redis to TencentDB for Redis](#).

Task Management

Task Status Description

Last updated : 2024-07-08 20:31:47

Billable task status description

Note:

Full migration is free of charge. Fees are incurred only during normal incremental migration (including data consistency check). When the task ends or is paused, billing will also stop.

If you select **Structural migration** or **Full migration** for **Migration Type**, no fees will be incurred.

If you select **Full + Incremental migration** for **Migration Type**, during **source database export** and **data import**, full migration is performed free of charge. During subsequent **incremental sync**, incremental migration is performed and billing starts. When you click **Complete**, the task ends and billing stops.

Status	Description	Supported Operations
Creation completed	A data migration resource has been purchased.	View, Configure, Adjust Specification, Terminate
Checking	The migration task is being checked.	View, Adjust Specification, Terminate
Verification passed	The migration task passed the verification.	Immediate start, View, Configure, Verify, Adjust Specification, Terminate
Verification failed	The migration task failed the verification.	View, Configure, Verify, Adjust Specification, Terminate
Preparing	The verification has been completed, and data migration is ready to start.	View, Stop, Adjust Specification, Terminate
Running	The migration task is running.	View, Stop, Create Consistency Check Task (displayed only for supported links), Pause, Adjust Specification, Terminate
Pausing	The task is being paused.	View, Terminate
Paused	The task is paused. We recommend you pause a task for no longer than seven days; otherwise, the task will fail.	View, Stop, Resume, Terminate, Adjust Specification
Resuming	The paused migration task is being resumed. After it	View, Terminate

	is resumed, its status will become Running .	
Prepared	When the task enters the Incremental Sync stage, the Complete button will be displayed, and you can click it to end the migration task.	View, Complete, Stop, Create Consistency Check Task (displayed only for supported links), Pause, Adjust Specification, Terminate
Completing	The incremental migration task is being completed after you click Complete to end it in the *Incremental Sync** stage.	View, Adjust Specification, Terminate
Task successful	The migration task is successfully completed.	View, Adjust Specification, Terminate
Task error	The migration task was interrupted during migration due to an exception. You can click Retry to resume the task in the console.	View, Stop, Retry, Adjust Specification, Terminate
Resuming	The task is being resumed after you click Retry .	View, Adjust Specification, Terminate
Task failed	The migration task failed and couldn't be resumed.	View, Retry (only available for Redis migration), Adjust Specification, Terminate
Stopping	The task is being stopped after you click Stop . A stopped task cannot be resumed, but the migration link record will be retained.	View, Adjust Specification, Terminate
Stopped	The task is stopped after you click Stop .	View, Adjust Specification, Terminate
Adjusting specification	The task specification is being adjusted after you click Adjust Specification .	View
Isolating	The task is being isolated after you click Terminate . Its resources will be cleared automatically in seven days after termination. The task is being isolated by the system automatically one day after your account has overdue payments.	View
Isolated	The task is isolated after you terminate it or it is moved to the recycle bin one day after your account has overdue payments. It will be retained in the recycle bin for seven days.	View, Deisolate/Resume, Delete
Resuming	The task is being restored. If a pay-as-you-go task is isolated after you click Terminate , you can click Deisolate/Resume to restore it.	View

Deleting	The task is being deleted automatically seven days after it was isolated, and its data will be cleared permanently. The task is being deleted after you click Delete .	View
Topping up	The account balance is being topped up, and the task moved to the recycle bin due to overdue payments can be restored.	View

Free task status description

Status	Description	Supported Operations
Creation completed	A data migration resource has been purchased.	View, Configure, Delete
Checking	The migration task is being checked.	View, Delete
Verification passed	The migration task passed the verification.	View, Configure, Verify, Start, Delete
Verification failed	The migration task failed the verification.	View, Configure, Verify, Delete
Preparing	The verification has been completed, and data migration is ready to start.	View, Stop, Delete
Running	The migration task is running.	View, Stop, Delete
Prepared	When the task enters the Incremental Sync stage, the Complete button will be displayed, and you can click it to end the incremental migration task.	View, Stop, Complete, Delete
Completing	The incremental migration task is being completed after you click Complete to end it in the *Incremental Sync** stage.	View, Delete
Task successful	The migration task is successfully completed.	View, Delete
Task error	The migration task was interrupted during migration due to an exception. You can click Retry to resume the task in the console.	View, Stop, Retry, Delete
Resuming	The task is being resumed after you click Retry .	View, Delete

Task failed	Migration task failed	View, Delete
Stopping	The task is being stopped after you click Stop . A stopped task cannot be resumed, but the migration link record will be retained.	View, Delete
Stopped	The task is stopped after you click Stop .	View, Delete

Viewing Task

Last updated : 2024-07-08 20:31:47

Overview

After successfully creating a data migration task, you can view its details (including source and target database information and migration type), migration object, and progress.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and you can view a task in one of the following two methods:

Method 1: on the **Data Migration** page, select the target migration task and click the task name.

Method 2: on the **Data Migration** page, select the target migration task and click **More > View** in the **Operation** column.

Create Migration Task							
Edit Tag							
DBrain							
<input type="checkbox"/>	Task ID / Name	Task Status / Progress	Running Mode	Last Check Result	Source Database Type	Target Database Type	Source Access Type
<input type="checkbox"/>	dts-xxxxxx NewDTS	(3 / 3) Current step: binlog_sinker Status: Task running Start: 2021-12-01 19:38:46 End: --	Immediate execution	no job	MySQL	MySQL	Database
<input type="checkbox"/>	dts-xxxxxx NewDTS	(12 / 12) Status: Verification failed	Immediate execution	no job	MySQL	MySQL	Database
<input type="checkbox"/>	dts-xxxxxx NewDTS	Status: Creation completed	Immediate execution	no job	MySQL	MySQL	Database

2. View the task details.

The screenshot shows the 'Task Details' tab selected in the top navigation bar. The page is divided into three sections: Basic Info, Source Database Info, and Target Database Info.

Basic Info

Task ID	dts-
Task Name	te-
Running Mode	Immediate execution
Migration Type	Full + Incremental migration
Tag	--

Source Database Info

Source Database Type	MySQL
Service Provider	Others
Access Type	Database
Region	South China (Guangzhou)
Database Instance	cdb-r- : Instance Monitoring
Account	root

Target Database Info

Target Database Type	MySQL
Access Type	Database
Region	South China (Guangzhou)
Database Instance	cdb-1 : Instance Monitoring
Account	root

3. View the migration object.

The screenshot shows the 'Migration Object' tab selected in the top navigation bar. The page displays the migration object configuration.

Migration Object

Specify object ⓘ

[Unfold all](#) [Fold all](#)

► db-dst

4. View the data migration progress details. The **Migration Progress** tab displays information such as source and target databases/tables, number of estimated records, number of completed records, and migration status.

Task Details		Migration Object	Migration Progress	Data Consistency Check	Monitoring Data	Task Log		
Source		Target	Object Type	Migration Stage ▾	Estimated Records	Completed Reco...	Status ▾	Start Time
Database: db-dst Table: tb		Database: db-dst Table: tb	Table	Export	3	3	Completed	2021-12-01 19:40:01
Database: db-dst Table: tb		Database: db-dst Table: tb	Table	Import	3	3	Completed	2021-12-01 19:41:19

Configuring Task

Last updated : 2024-07-08 20:31:47

Overview

After purchasing migration resources, you need to configure the detailed parameters of a new migration task. You can also modify the parameters of a completed configuration task, such as migration object and type.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **More > Configure** in the **Operation** column.
 2. On the migration task page, configure or modify the relevant parameters and save your editing.
- For the requirements and instructions on configuring different links, see [Data Migration \(NewDTS\)](#).

Verifying Task

Last updated : 2024-07-08 20:31:47

Overview

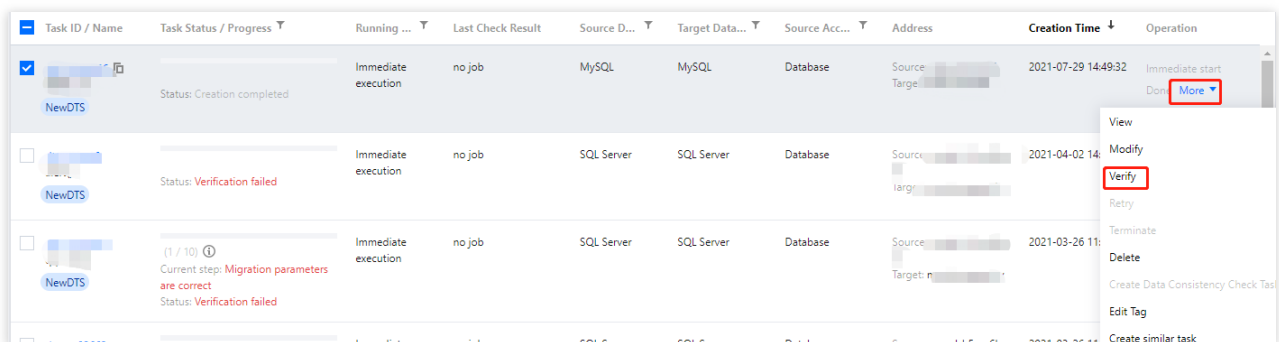
If a verification task failed, you need to initiate it again after fixing the problem as prompted.

Prerequisites

You have fixed the error reported by the verification task.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **More > Verify** in the **Operation** column.



Task ID / Name	Task Status / Progress	Running ...	Last Check Result	Source D...	Target Data...	Source Acc...	Address	Creation Time	Operation
<input checked="" type="checkbox"/> [Task ID] NewDTS	Status: Creation completed	Immediate execution	no job	MySQL	MySQL	Database	Source: [Address] Target: [Address]	2021-07-29 14:49:32	Immediate start Don't [More]
<input type="checkbox"/> [Task ID] NewDTS	Status: Verification failed	Immediate execution	no job	SQL Server	SQL Server	Database	Source: [Address] Target: [Address]	2021-04-02 14:...	View Modify Verify Retry Terminate Delete Create Data Consistency Check Task Edit Tag Create similar task
<input type="checkbox"/> [Task ID] NewDTS	(1 / 10) ⓘ Current step: Migration parameters are correct Status: Verification failed	Immediate execution	no job	SQL Server	SQL Server	Database	Source: [Address] Target: [Address]	2021-03-26 11:...	
<input type="checkbox"/> [Task ID] NewDTS		Immediate execution	no job	SQL Server	SQL Server	Database	Source: [Address] Target: [Address]	2021-03-26 11:...	

2. In the pop-up window, you can create a verification task and query the verification result.

Start Task

Last updated : 2024-07-08 20:31:47

Overview

After completing the verification task, users can either start the task directly from the verification operation page or from the task list page.

The validity period for verification is 2 hours. If the task is not started immediately after verification and is started after 2 hours, the task needs to be re-verified.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **More > Immediate start** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.

Retrying Task (MySQL)

Last updated : 2024-07-08 20:31:48

Overview

After a migration task is started, if it is interrupted due to an exception, you can manually retry it without the need to pull data all over again.

Full stage: The chunk-based import/export mechanism is implemented in this stage. Each chunk is assigned a unique identifier. With these identifiers, DTS can identify the unsynced chunk and resume the data sync from there when a disrupted task is restarted.

Incremental stage: The checkpoint-based mechanism is implemented in this stage. When DTS parses the source database binlogs before writing them to the target database, a checkpoint message is inserted to indicate the offset of the data being synced every 10 seconds. With these checkpoint messages, DTS can resume the data sync from the checkpointed offset when a disrupted task is restarted.

Below describes the scenarios where the task retry is supported for data migration:

Link	Supported Scenario	Description
Data migration links among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.	Full export Full import Incremental sync	Description for full export: Retry is only supported for exporting unlocked tables with primary keys. Retry is only supported when a task is interrupted by table locking failures due to time-consuming SQL statements in the source instance. It doesn't work in other cases.
Migration links among MySQL, TDSQL-C for MySQL, MariaDB, Percona, and TDSQL for MySQL.	Incremental sync	-

Application Scope

Data migration links among MySQL, TDSQL-C for MySQL, MariaDB, Percona, and TDSQL for MySQL.

Directions

Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **Retry** in the **Operation** column.

Retrying Task (Redis)

Last updated : 2024-07-08 20:31:48

Overview

After a sync task is started, if it is interrupted due to an exception, you can manually retry it.

For Redis migration links, the task retry mechanism is to execute the data migration task again after the task interruption, with the previously migrated data cleared or overwritten.

This document describes manual retry operations. You can also set automatic retry when configuring the task as instructed in [Migration from Redis to TencentDB for Redis](#). The manual and automatic retry mechanisms are the same.

Note

Only the tasks in the **Task error** (`error`) status can be retried.

Directions

Note

As the retry operation will clear or overwrite the data in the target database, you need to configure the target database read/write policies for DTS.

For tasks created after the retry feature was released in June 2023, such policies can be set in the task configuration stage

For tasks created before the retry feature was released in June 2023, such policies need to be set on the **Task Details** page.

Configuring target database read/write policies for new tasks

Below are brief instructions. For detailed directions, see [Migration from Redis to TencentDB for Redis](#).

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, click **Create Migration Task**, and purchase a task.
2. In the task list, select the purchased task and click **Configure** in the **Operation** column.
3. In the **Set source and target databases** step, set the automatic retry time.

1 设置源和目标数据库

2 设置迁移选项及选择迁移对象

3 校验任务

任务设置

任务名称 *

运行模式 *

立即执行

定时执行

自动重试 *

5

+

分钟

任务出错后，将在任务断开的 5 分钟内按照预置的重试策略进行重试，如果在 5 分钟内无法拉起任务。

提示：您正在使用数据迁移（NewDTS）。
为了您的数据安全，请在创建数据迁移任务前，仔细阅读[《数据迁移》](#)

4. In the **Set migration options and select migration objects** step, set the target database read/write policy.

Target Database Write Mode

Clear target instance: This option indicates that when you sync the data from the source database to the target database, the existing data in the target database will be cleared to allow for the synced data to be written.

Overwrite: This option indicates that when you sync the data from the source database to the target database, keys in the target database will be overwritten by those in the source database.

Enable Target Database Read/Write: To use the task retry feature, you must enable this option to allow DTS to write data to the target database.

1 设置源和目标数据库

2 设置迁移选项及选择迁移对象

3 校验任务

迁移类型

全量 + 增量迁移

迁移对象

整个实例

目标库写入模式 *

清空目标实例

覆盖写

启用清空目标实例模式，将清空目标库内的数据后重新写入

启用目标库读写 *

迁移注意事项，请参见 [迁移常见问题](#)

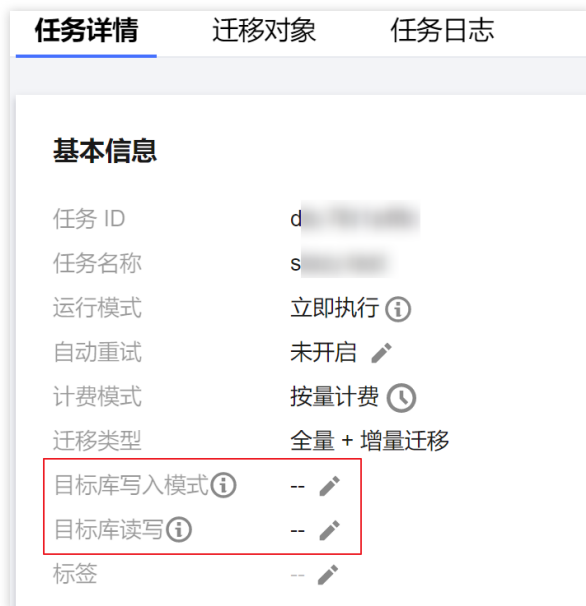
上一步

保存

5. Verify the task and start it.

Configuring target database read/write policies for existing tasks

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select a target task, and click the task ID.
2. On the **Task Details** page, modify the **Target Database Write Mode** and **Target Database Read/Write** parameters.



Retrying the task

1. Select **Data Migration** on the left sidebar and select a target migration task.
2. In the **Operation** column, click **More > Retry**.

任务 ID / 名称	任务状态 / 进度	运行模式	规格	计费类型	最后一次校验结果	源实例类型	目标实例类型	源接入类型	目标接入类型	地址	创建时间	操作
dts-...-test01	状态: 创建完成	立即执行 ⓘ	Xlarge	按量计费 ⌚		Redis	Redis	--	--	源: -- 目标: --	2023-06-08 18:00:07	配置 编辑
dts-...	状态: 创建完成	立即执行 ⓘ	Small	按量计费 ⌚	查看更多	MySQL	MySQL	--	--	源: -- 目标: --	2022-10-28 10:12:52	创建实例 立即执行 完成 校验 重试
dts-...-5 test	(5 / 7) ⓘ 当前步骤: 同步构建中 (0%) 状态: -- 开始: 2020-07-06 16:15:36	立即执行 ⓘ	--	按量计费		MySQL	MySQL	云数据库	云数据库	源: cdb-... 目标: cdb-...	2020-07-06 16:15:36	终止 暂停 恢复

Speed Limit (MySQL)

Last updated : 2024-08-13 17:29:12

Overview

In the DTS data migration task configuration, or when the task is running, the task concurrency number and RPS can be flexibly adjusted based on the data write volume of the source database, and the resource configuration of both the source and target databases.

Note:

This document only provides guidelines for operating DTS performance parameters. To comprehensively enhance the DTS transmission performance, it is necessary to analyze the load and network conditions of the source/target, etc.

Application Scope

Data migration link between MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL.

Notes

1. You can only adjust the speed during the task configuration process, or when the task status is Running.
2. When reducing the task speed, please observe the delay data volume and delay time. DTS by default only caches the intermediate data for 7 days. A long-term large delay volume may cause the intermediate cache data to expire and the task to fail completely.
3. Adjusting the speed during the full export stage will cause the task to restart, and a partial restart may require starting the sync all over again, as follows.

MySQL series data migration tasks default to lock-free sync. In lock-free sync scenarios, during the structure export stage and the non-primary key table export stage, restarting the task requires starting all over again; and during the primary key table export stage, restarting the task will resume sync, without the need to start all over again.

3.1 Restart during full export stage

During the structure export stage and the non-primary key table export stage, restarting the task requires starting all over again;

During the primary key table export stage: If the time from the start of sync (X) to the time the retry operation is performed (Y) is within 48 hours (including 48 hours), breakpoint resume is supported. If it exceeds 48 hours, the task needs to start all over again. Once the task retries or automatically restarts and begins sync from scratch, the time X will be recalculated.

3.2 Restart during full import and incremental sync stages: support breakpoint resume, which does not affect data transfer.

Adjustment Principles

The default number of concurrent threads for a DTS task's full export is 8, which generally does not need to be modified. If adjustment is necessary, please combine monitoring indicators and source/target database configurations, and considering the precondition of not affecting the performance of the source/target database, enhance the DTS transfer speed to the greatest extent.

DTS transfer performance is related to the configuration of the source/target database, DTS task specifications, network latency, network bandwidth, etc. Therefore, when improving DTS transfer performance, simply increasing the number of threads or RPSes may not be effective. It is necessary to analyze where the performance bottleneck lies and solve the specific problem accordingly.

Task Stage	Adjust Scenario	Adjustment Plan	Adjustment Result
Task Configuration Stage	Source database configuration is too low to handle the increased load from DTS, resulting in the source database crashing after the DTS task starts.	Reduce the number of export threads for the source database to lower the export RPS.	Valid.
	Target database configuration is too low to handle the increased load from DTS, resulting in the target database crashing after the DTS task starts.	Reduce the number of import threads for the target database to lower the import RPS.	Valid.
Full Volume Stage	Source/target database configuration is high, yet DTS full volume transfer speed is slow	<p>If the source database configuration is high, increase the number of export threads for the source database</p> <p>If the target database configuration is high, increase the number of import threads for the target database.</p> <p>The default RPS is very large in the full volume stage, and generally, there's no need to</p>	<p>It may not always be effective.</p> <p>If adjustments do not take effect, please investigate factors such as network latency and bandwidth, as these can also affect the transfer performance.</p>

		adjust the RPS to increase the transfer speed.	
	Target database configuration is low, and DTS writes to the target database too quickly, impacting the target database load (e.g., high CPU occupancy rate)	Reduce the number of import threads for the target database to lower the import RPS.	Valid.
Incremental Stage	Source database configuration is high, source database write data volume is large, and network latency is high	Check if the number of RPSes has reached the upper limit. Different specifications of DTS tasks correspond to different RPS upper limits. If the number of RPSes has reached the upper limit, first upgrade the DTS task specifications (through Configuration Adjustment operation).	If the upgrade task does not take effect, please investigate factors such as network latency and bandwidth, as these can also affect the transfer performance.
	Target database configuration is low, and DTS writes to the target database too quickly, impacting the target database load (e.g., high CPU occupancy rate)	1. Reduce the DTS task specification. 2. Decrease the concurrent thread count for importing into the target database.	Valid.
	The target database configuration is low, DTS has too many connections, and the target database reaches the thread limit	Reduce the number of threads for target database import.	Valid.
	DTS writes to the target database, causing lock contention with the database's own writes (e.g., lock wait timeout error)	Reduce the number of threads for importing into the target database to minimize the lock competition.	Valid.

Speed Adjustment in Task Configuration

During the initial configuration stage of the sync task, in the Advanced Options on the **Set sync options and objects** page, you can set the transfer speed parameters.

Advanced Options

Migrate Account ⓘ * ☒

Limit Transfer Speed ☒

Concurrent Threads During Full Export ⓘ *

Full Export RPS ⓘ

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Full Import ⓘ *

Full Import RPS ⓘ

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Incremental Import *

Speed Adjustment When the Task Is Running

You can adjust the speed during the full data sync stage by adjusting the speed parameters for both full and incremental (full export thread count and RPS, full import thread count and RPS, and incremental import thread count). During the incremental data sync stage, only the incremental speed parameters (incremental import thread count) can be modified, with specific operations as follows.

1. Log in to the [DTS console](#), choose **Data Sync** page in the left sidebar, and select the specified sync task.
2. Adjust the speed during the full data sync stage.

In the **Operation** column, select **More > Limit Speed**.

Adjust the speed parameters in the pop-up interface, and click **OK** after completion.

Limit Task Speed

! If you adjust the task speed in the current stage, the full data export will start all over again. Please proceed with caution.

! When the source or target database instance is overloaded, you can reduce the thread count to alleviate resource occupation. When the source or target database instance is underloaded, you can increase the thread count to accelerate the data transfer. For details, see [here](#) .

Note: If you reduce the full import/export speed when the data volume is high, the data transfer may take longer than the max allowed time, causing previously imported or exported data to expire. We recommend that you upgrade the source and target database specifications as needed.

You have selected 1 task. [Show less](#) ▼

Task ID	Task Name	Billing Mode	Current Specific...
syr		Pay as you go	Micro

Concurrent Threads During Full Export **!** *

— 8 +

Full Export RPS **!**

— 0 +

thou: ▼ Counts/s

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Full Import **!** *

— 8 +

Full Import RPS **!**

— 0 +

thou: ▼ Counts/s

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Incremental Import *

— 32 +

OK

Cancel

3. Adjust the speed during the incremental data sync stage.

In the **Operation** column, select **More > Limit Speed**.

Limit Task Speed

In most cases, there's no need to adjust the incremental transfer speed. For details, see [here](#).

You have selected 1 task. [Show less](#)

Task ID	Task Name	Billing Mode	Current Specific...
sys-123456789	task-123456789	Pay as you go	Micro

Concurrent Threads During Incremental Import

-

32

+

OK

Cancel

Complete Task

Last updated : 2024-07-08 20:31:47

Overview

During the full + incremental data migration, once the full migration is completed, the task will automatically start the incremental migration, which will not stop on its own and must be ended manually.

For a pay-as-you-go migration task, the billing stops only after you click **Complete**. Therefore, we recommend that you click **Complete** promptly after the data migration is completed to avoid unnecessary costs.

Prerequisite

The data migration type is full + incremental migration.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, and select the specified migration task.
2. When the task enters the incremental stage, the grayed out **Complete** button will turn blue. Click **Complete** to stop the migration task at an appropriate time.

We recommend that you check the task status and ensure the following conditions are met before stopping the task to avoid data inconsistency due to a large data gap or database time lag.

The source-target database data gap is 0 MB.

The source-target database time lag is 0s.

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

Terminating Task

Last updated : 2024-07-08 20:31:48

Overview

You can terminate a started migration task.

Note:

Note that once terminated, the task cannot be resumed, and executed operations cannot be rolled back.

Directions

Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **More** > **Terminate** in the **Operation** column.

Deleting Task

Last updated : 2024-07-08 20:31:47

Overview

You can delete a stopped or failed task. Once deleted, the task will no longer exist or occupy any resources.

Prerequisites

The migration task has been stopped or failed.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **More > Delete** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.

Pausing Task

Last updated : 2024-07-08 20:31:47

Overview

After a sync task is started, if there are too many connections to the source database or the business needs to be adjusted, you can pause the task and resume it after the source database can handle the load or the business adjustment is completed.

Application Scope

Data migration links among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Note

Only migration tasks in **Running** or **Prepared** status can be paused.

After a task is paused, DTS will continue to pull the source database data but will not write data to the target database. After the task is resumed, DTS will continue the task.

In the full export stage, after a task is resumed, the checkpoint restart is not supported, and DTS needs to export all the source database data again.

In the full import and incremental sync stages, after a task is resumed, DTS will continue with the data transfer based on the offset of the binlog synced before the task is paused.

During migration, binlog records are retained for up to seven days. If the pause lasts longer than seven days, the task will fail and cannot be restarted. We recommend that you keep the pause period below three days.

Because DTS continues to pull the data and binlog of the source database during the task pause, billable links will still be billed when the migration task is paused.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **Pause** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.

暂停迁移任务



您已选 1 个任务，[收起详情](#) ▼

任务 ID	任务名	现有规格	计费模式
dts-r	迁移任务	2xlarge	按量计费

- 服务暂停期间，DTS 服务会持续拉取源端的数据和 binlog，但不会向目标库中写入数据，恢复任务后，将基于任务暂停前的 binlog 数据点继续执行该任务
- 迁移期间 binlog 记录最多保留 7 天，若暂停时长超过 7 天则会导致任务失败，无法重新启动；建议任务暂停时间不超过 3 天
- 对于计费链路，迁移任务暂停期间仍会保持计费

确定

取消

Resuming Task

Last updated : 2024-07-08 20:31:47

Overview

This document describes how to resume a paused migration task.

Application scope

Data migration links among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Notes

After a task is paused, DTS will continue to pull the data and binlog of the source database but will not write data to the target database. After the task is resumed, DTS will continue the task based on the binlog data points before the pause.

During migration, binlog records are retained for up to seven days. If the pause lasts longer than seven days, the task will fail and cannot be restarted. We recommend that you keep the pause period below three days.

Billable links will still be billed when the migration task is paused.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **Resume** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.

Changing Configuration

Last updated : 2024-07-08 20:31:48

Overview

When your business requirements change, you can modify the selected migration link specification through the operation below.

Note

It takes 3-5 minutes for the configuration changes you make to take effect, and the task will be paused during this period.

Application Scope

This operation is available for data migration tasks that have been billed.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target task instance, and click **More > Adjust Specification** in the **Operation** column.
2. In the pop-up window, select the specification to modify and click **OK**.

Terminating Task

Last updated : 2024-07-08 20:38:41

Overview

You can terminate the instance tasks that have stopped or failed if they are no longer needed.

After being terminated, the instance will enter the isolated status and will be automatically deleted after 7 days, so proceed with caution. If you have accidentally deleted an task, you can restore it as instructed in [Deisolating Task](#).

In pay-as-you-go billing mode, the fees for one hour's usage will be frozen during task creation and will be unfrozen when you terminate the instance or the instance is isolated and terminated due to overdue payments. This is the case even in scenarios where no fees are incurred, for example, incremental migration is not selected in a migration task.

We recommend that you terminate completed or unwanted tasks promptly to unfreeze fees in time.

Application Scope

This operation is available for data migration tasks that have been billed.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target task instance, and click **More > Terminate** in the **Operation** column.
2. In the pop-up window, read and click **I have read and agreed to Termination Rules** and click **Terminate Now**.

Deisolating Task

Last updated : 2024-07-08 20:31:47

Overview

When a pay-as-you-go migration task is isolated due to accidental termination, you can resume the task through the operation below. Once resumed, the task will return to the status before it was terminated.

Application Scope

This operation is available for pay-as-you-go migration tasks.
Only NewDTS migration tasks support the deisolation.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target task instance, and click **More > Deisolate/Resume** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Resume**.

Deleting Task

Last updated : 2024-07-08 20:31:48

Overview

You can delete an instance only when it has been moved to the recycle bin and becomes isolated due to termination or overdue payments.

By default, the instance will stay in the recycle bin for seven days before being automatically deleted. If you need to delete it immediately, you can perform the operation below.

Application Scope

This operation is available for data migration tasks that have been billed.

Directions

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target task instance, and click **More > Delete** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Delete**.

Managing Tag

Last updated : 2024-07-08 20:31:47

Overview

You can set tags for each migration task. If there are many tasks, you can filter tasks in the same category by tag to manage them easily.

Editing and Adding Tag

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **Edit Tag** at the top of the page.
2. In the pop-up window, select a tag. If you want to add a tag, click **Add**.

Edit Tags ✕

The tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to [Manage Tags](#) 🔗

1 resource selected

Tag key ▼

Tag value ▼ ✕

[+ Add](#)

OK

Cancel

Deleting Tag

1. Log in to the [DTS console](#), select **Data Migration** on the left sidebar, select the target migration task, and click **Edit Tag** at the top of the page.
2. In the pop-up window, click the "x" icon after a tag to delete it.

Edit Tags ✕

The tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to [Manage Tags](#) 🔗

1 resource selected

DBA ▼	Harris ▼	✕
Tag key ▼	Tag value ▼	✕

[+ Add](#)

OK Cancel

Creating a Similar Task

Last updated : 2024-07-08 20:31:48

Overview

Creating a similar task allows users to quickly create a task with the same configuration as the current one. The default options filled in the new task, such as Database Type, Connection Method, Billing Mode, and Migration Type, will all be consistent with those of the historical tasks. Users can also modify these options as needed.

In scenarios such as task failure and many-to-one sync, a similar task can be created quickly to configure a new task.

Directions

1. Log in to the [DTS console](#), choose **Data Migration** page in the left sidebar, select the specified task instance, and then select **More > Create similar task** in the **Operation** column.
2. In the pop-up dialog box, after the information is confirmed, click **Create similar task**.

Viewing Log

Last updated : 2024-07-08 20:31:47

Overview

During a migration task, you can view the migration logs to know the task progress.

Directions

1. Log in to the [DTS console](#), and you can view a task in the following two ways:
Method 1: on the **Data Migration** page, select the target migration task and click the task name.
Method 2: on the **Data Migration** page, select the target migration task and click **More > View** in the **Operation** column.
2. View the migration logs.
- Switch the tab to view the task logs.

Task Details	Migration Object	Migration Progress	Data Consistency Check	Task Log
				<div>AllTodayYesterdayLast 7 daysLast 30 daysSelect date📅</div>
Operation Time			Status ▾	
2021-10-25 21:38:32			Configured	
2021-10-25 21:32:32			Configured	
2021-09-27 20:17:16			Verification failed	
2021-09-27 20:16:48			Checking	
2021-09-27 20:16:40			Configured	