Data Transfer Service

Data Migration

Product Documentation

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

Last updated：2025-06-04 17:01:21

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 5.5, 5.6, 5.7, 8.0 (IDC self-built MySQL/Self-built MySQL on CVM)	TencentDB for MySQL 5.5, 5.6, 5.7, 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, 8.0			
TencentDB for MySQL 5.5, 5.6, 5.7, 8.0 Migration between databases under the same Tencent Cloud root account Migration 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 on CVM 5.5, 10.0-10.6	TencentDB for MySQL 5.5, 5.6, 5.7, 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) Migration between databases under the same Tencent Cloud root account. Migration between databases under different Tencent Cloud root accounts		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 on CVM 5.5, 5.6, 5.7, 8.0	TencentDB for MySQL 5.5, 5.6, 5.7, 8.0	Structure migration Full migration Full + incremental migration	Migration from MariaDB/Percona to TencentDB for MySQL

TDSQL-C for MySQL > MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TDSQL-C for MySQL 5.7, 8.0 Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts	TencentDB for MySQL 5.7, 8.0	Structure migration Full migration Full + incremental migration	Migration from TDSQL-C for MySQL to TencentDB for MySQL

TDSQL for MySQL > MySQL

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

Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts		Full + incremental migration	

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 Self-built MySQL on CVM 5.5, 5.6, 5.7, 8.0	TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	Structure migration Full migration Full + incremental migration	Migration from MySQL to TencentDB for MariaDB
TencentDB for MySQL 5.5, 5.6, 5.7, 8.0 Migration between databases under the same Tencent Cloud root account Migration 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 on CVM 5.5, 10.0-10.6	TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	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)			

Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts			

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 on CVM 5.5, 5.6, 5.7, 8.0	TencentDB for MariaDB (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1)	Structure migration Full migration Full + incremental migration	Migration from MySQL to TencentDB for MariaDB (MySQL/MariaDB/Percona/TDSQL-C for MySQL)

TDSQL-C for MySQL > MariaDB

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

TDSQL for MySQL > MariaDB

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

Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts	5.7, MariaDB 10.1)	Full + incremental migration	

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 on CVM 5.5, 5.6, 5.7, 8.0	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, 8.0			
TencentDB for MySQL 5.5, 5.6, 5.7, 8.0 Migration between databases under the same Tencent Cloud root account Migration 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	TDSQL-C for MySQL	Structure	Migration from

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

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 on CVM 5.5, 5.6, 5.7, 8.0	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-C MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TDSQL-C for MySQL (database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts	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 for MySQL > TDSQL-C for MySQL

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

Migrating to TDSQL for MySQL

MySQL > TDSQL for MySQL

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

MariaDB > TDSQL for 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 on CVM 5.5, 10.0-10.6	TDSQL for MySQL (database version: MariaDB 10.1)	Structure migration 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) Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts			

Percona > TDSQL for 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 for MySQL (database version: Percona 5.7)	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL for MySQL

TDSQL-C for MySQL > TDSQL for MySQL

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

Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts	MySQL 8.0, Percona 5.7, MariaDB 10.1)	Full + incremental migration	

TDSQL for MySQL > TDSQL for MySQL

Source Database and Version	Target Database and Version	Migration Type	Reference Documentation
TDSQL for MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts	TDSQL for 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 for 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 on CVM 5.5, 5.6, 5.7, 8.0	TDSQL for 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, 8.0 Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts			

MariaDB > TDSQL for 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 on CVM 5.5, 10.0-10.6	TDSQL for 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) Migration between databases under the same Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts			

Percona > TDSQL for 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 on CVM 5.5, 5.6, 5.7, 8.0	TDSQL for MySQL (TDStore)	Structure migration Full migration Full + incremental migration	Migration from MySQL/MariaDB/Percona/TDSQL for MySQL to TDSQL for MySQL (TDStore)

TDSQL for MySQL > TDSQL for 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)	TDSQL for 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 Tencent Cloud root account Migration between databases under different Tencent Cloud root accounts		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 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 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	Migration from TDSQL-C for PostgreSQL to PostgreSQL

		Full + incremental migration	

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 on CVM 3.0, 3.2, 3.4, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0	TencentDB for MongoDB 3.0, 3.2, 3.4, 3.6, 4.0, 4.4、6.2、7.0, 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,	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

2022 on CVM			
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, 6.2, 7.0 Self-built Redis on CVM 2.8, 3.0, 3.2, 4.0, 5.0, 6.0, 6.2, 7.0	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0, 6.2, 7.0	Full + incremental migration	Migration from Redis to TencentDB for Redis
Third-Party cloud vendors AWS ElasticCaChe for Redis 4.0,5.0,6.0, 6.2, 7.0			
TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.0, 6.2, 7.0 Migration between databases under the same root account Database migration between different root accounts			

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			

Tendis > KeeWiDB

Source Database	Target Database	Migration Type	Reference Documentation
TencentDB for Tendis 4.9 Migration between databases under the same root account Database migration between different root accounts	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.

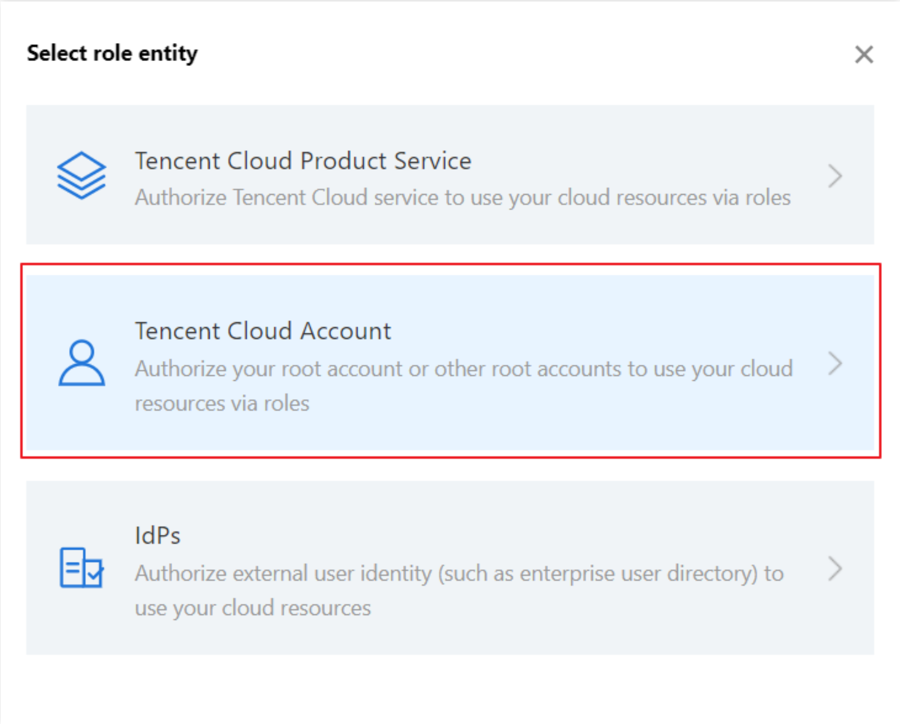
If you use a root account to execute the migration task, before executing the task, ask the root account of the source database to grant the root account of the target database access to the source database.

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**.



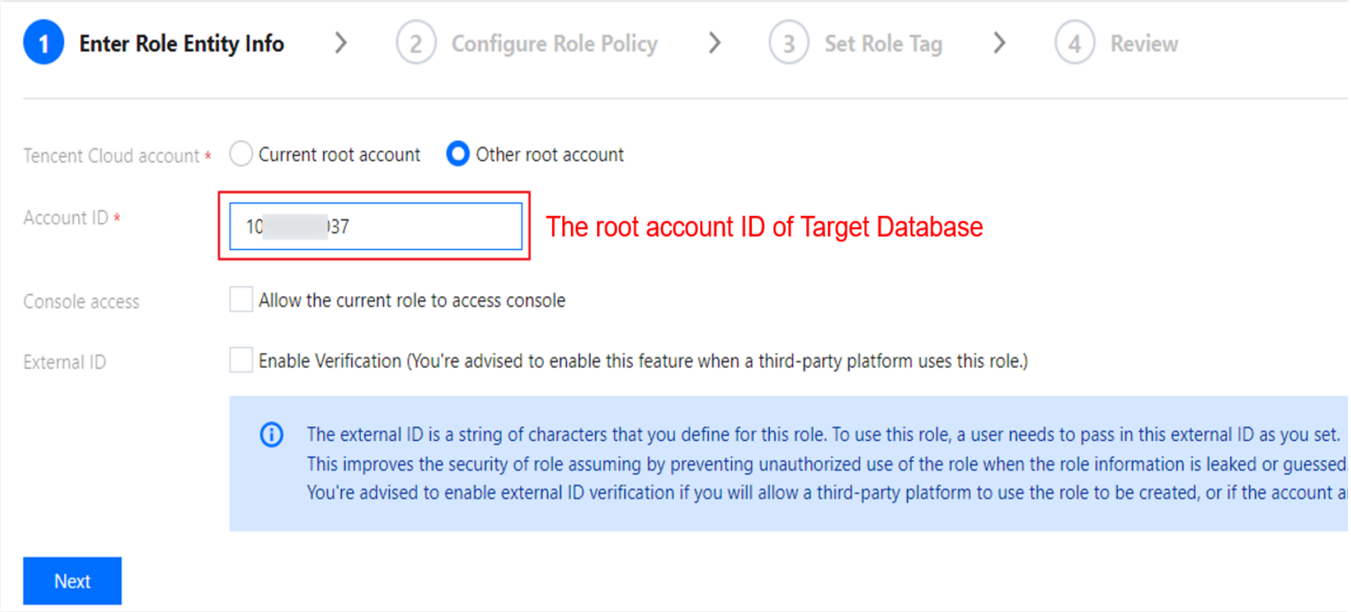
Select role entity

Tencent Cloud Product Service
Authorize Tencent Cloud service to use your cloud resources via roles

Tencent Cloud Account
Authorize your root account or other root accounts to use your cloud resources via roles

IdPs
Authorize external user identity (such as enterprise user directory) to use your cloud resources

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



Tencent Cloud account * ○ Current root account ● Other root account

Account ID * 10[redacted]37

The root account ID of Target Database

Console access □ Allow the current role to access console

External ID □ Enable Verification (You're advised to enable this feature when a third-party platform uses this role.)

The external ID is a string of characters that you define for this role. To use this role, a user needs to pass in this external ID as you set. This improves the security of role assuming by preventing unauthorized use of the role when the role information is leaked or guessed You're advised to enable external ID verification if you will allow a third-party platform to use the role to be created, or if the account a

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.

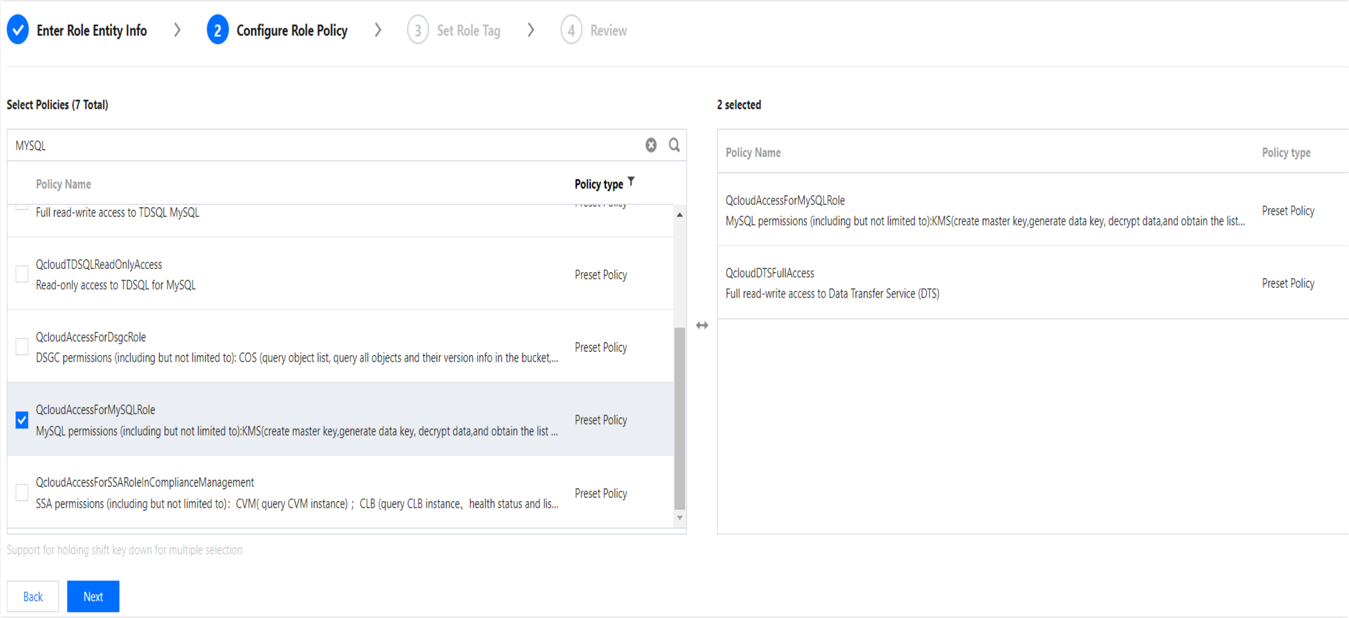
Note:

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.



Select Policies (7 Total)

MYSQL

	Policy Name	Policy type
□	Full read-write access to TDSQL MySQL	
□	QcloudTDSQLReadOnlyAccess Read-only access to TDSQL for MySQL	Preset Policy
□	QcloudAccessForDsgcRole DSGC permissions (including but not limited to): COS (query object list, query all objects and their version info in the bucket,...	Preset Policy
☑	QcloudAccessForMySQLRole MySQL permissions (including but not limited to):KMS(create master key,generate data key, decrypt data,and obtain the list ...	Preset Policy
□	QcloudAccessForSSARoleInComplianceManagement SSA permissions (including but not limited to)：CVM(query CVM instance)；CLB (query CLB instance、health status and lis...	Preset Policy

Support for holding shift key down for multiple selection

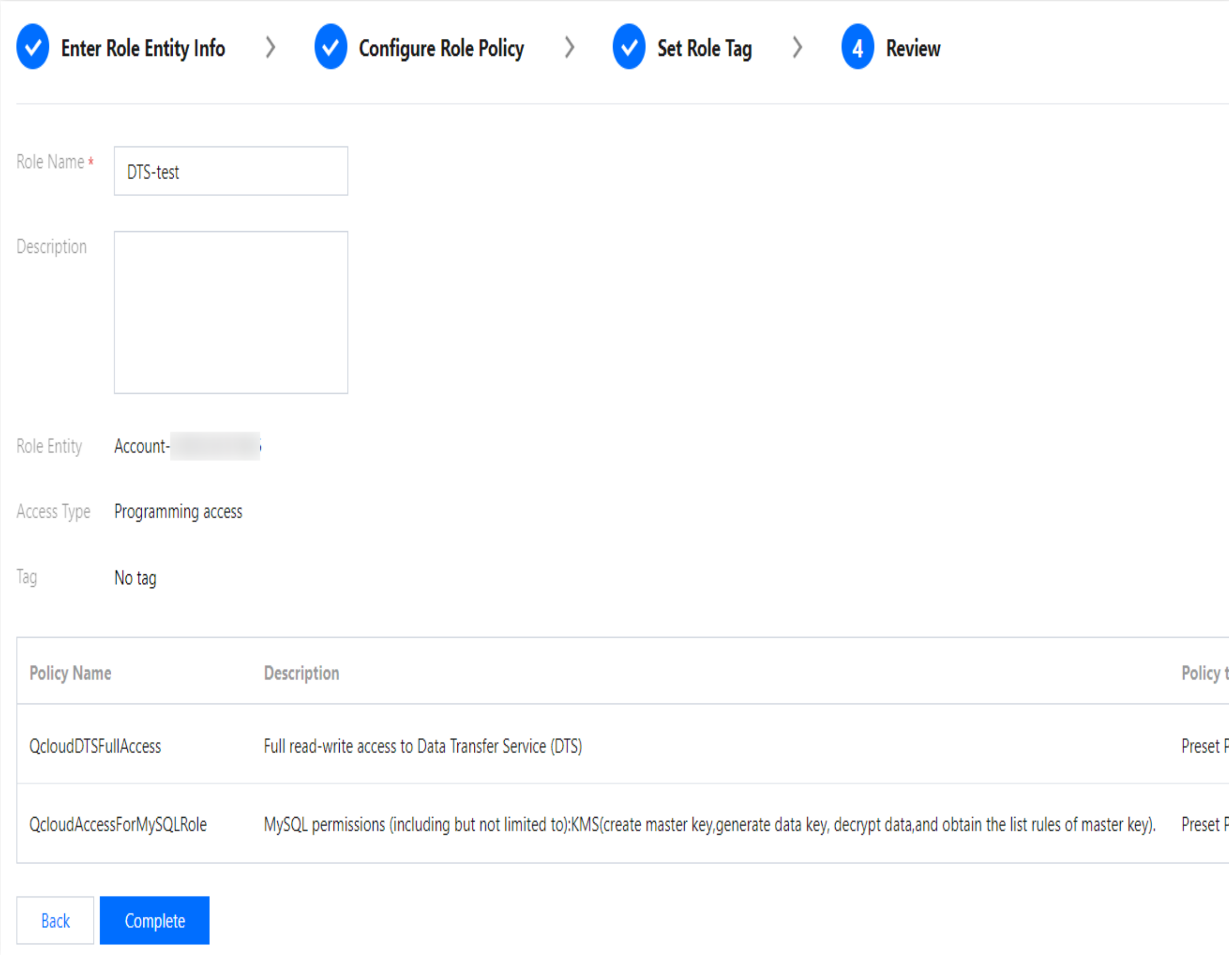
2 selected

Policy Name	Policy type
QcloudAccessForMySQLRole MySQL permissions (including but not limited to):KMS(create master key,generate data key, decrypt data,and obtain the list...	Preset Policy
QcloudDTSFullAccess Full read-write access to Data Transfer Service (DTS)	Preset Policy

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 Review

Role Name * DTS-test

Description

Role Entity Account-[redacted]

Access Type Programming access

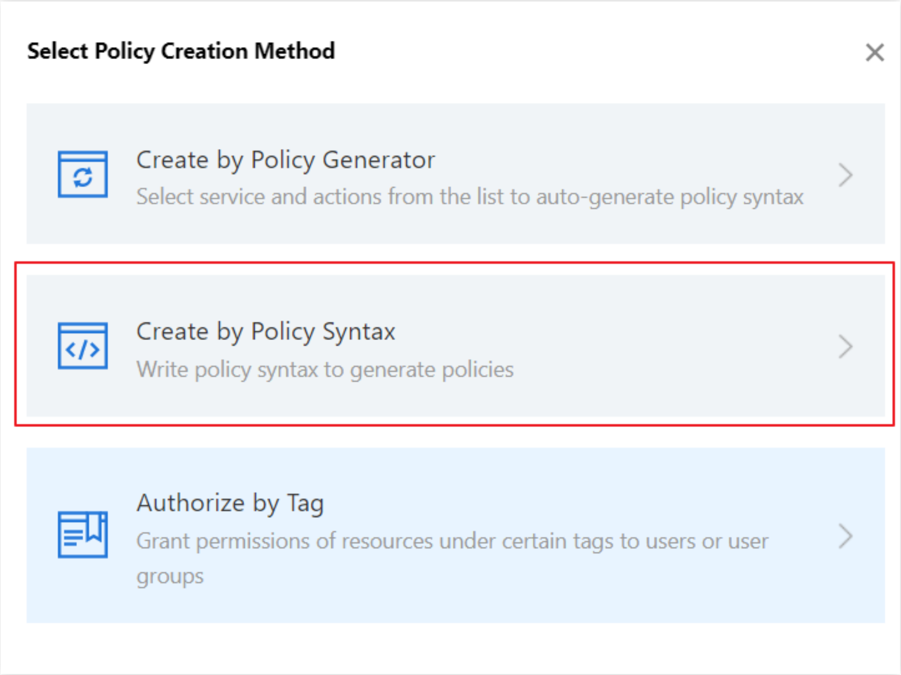
Tag No tag

Policy Name	Description	Policy t
QcloudDTSFullAccess	Full read-write access to Data Transfer Service (DTS)	Preset P
QcloudAccessForMySQLRole	MySQL permissions (including but not limited to):KMS(create master key,generate data key, decrypt data,and obtain the list rules of master key).	Preset P

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**.



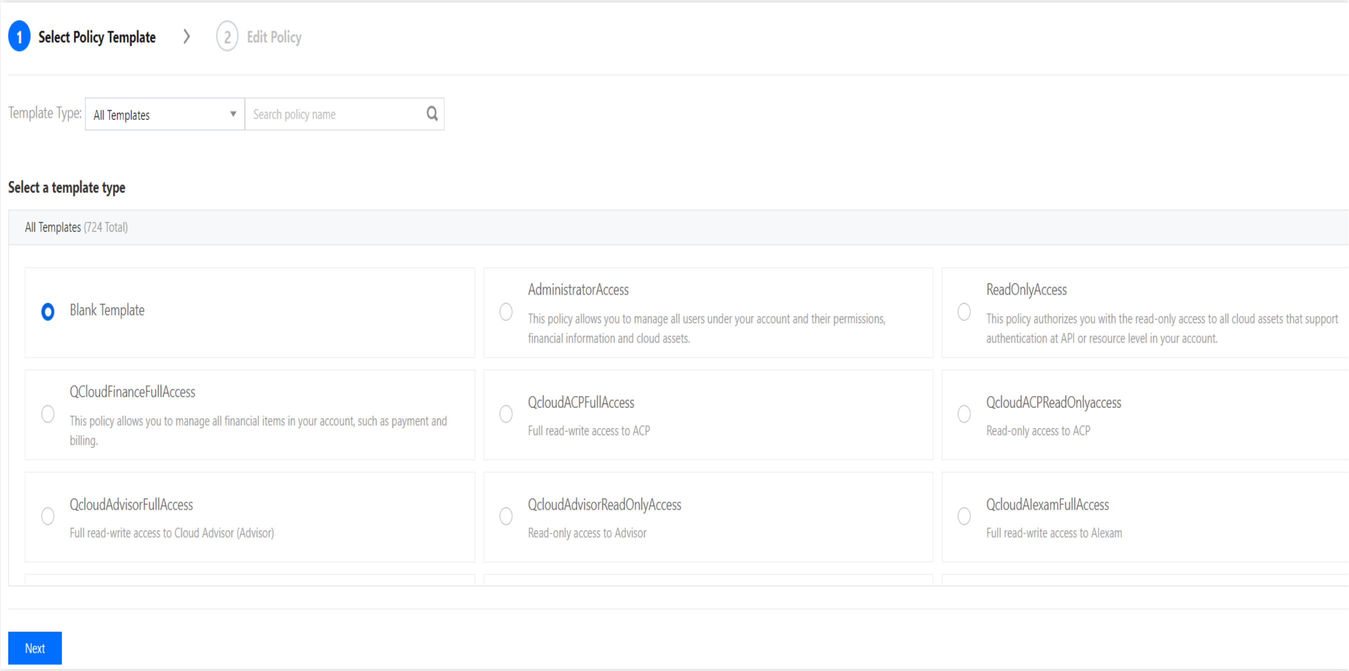
Select Policy Creation Method

Create by Policy Generator
Select service and actions from the list to auto-generate policy syntax

Create by Policy Syntax
Write policy syntax to generate policies

Authorize by Tag
Grant permissions of resources under certain tags to users or user groups

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



Select a template type

All Templates (724 Total)

● Blank Template

○ AdministratorAccess
This policy allows you to manage all users under your account and their permissions, financial information and cloud assets.

○ ReadOnlyAccess
This policy authorizes you with the read-only access to all cloud assets that support authentication at API or resource level in your account.

○ QCloudFinanceFullAccess
This policy allows you to manage all financial items in your account, such as payment and billing.

○ QcloudACPFullAccess
Full read-write access to ACP

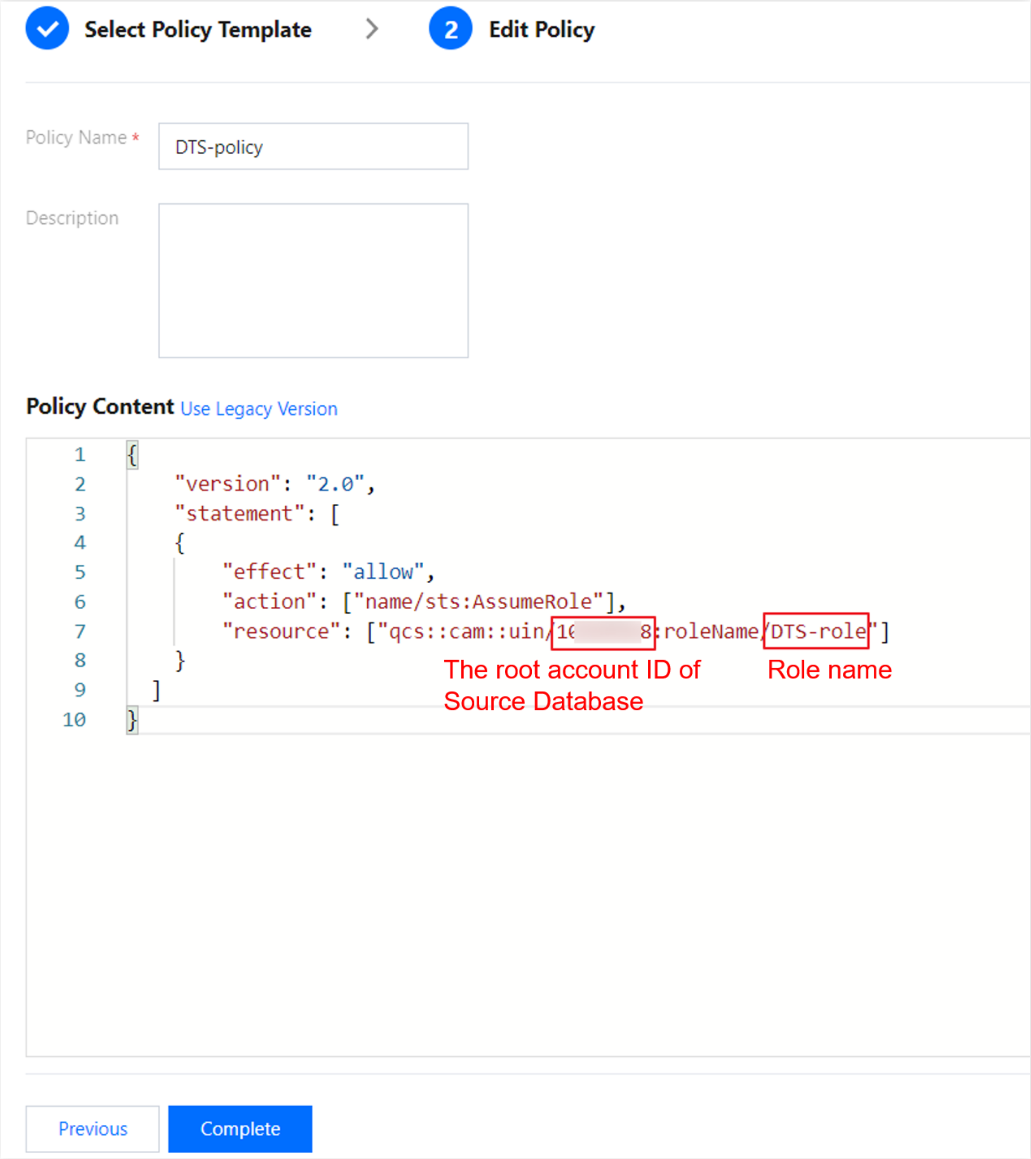
○ QcloudACPReadOnlyaccess
Read-only access to ACP

○ QcloudAdvisorFullAccess
Full read-write access to Cloud Advisor (Advisor)

○ QcloudAdvisorReadOnlyAccess
Read-only access to Advisor

○ QcloudAlexamFullAccess
Full read-write access to Alexam

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.



Policy Name * DTS-policy

Description

Policy Content Use Legacy Version

```
{
    "version": "2.0",
    "statement": [
    {
        "effect": "allow",
        "action": ["name/sts:AssumeRole"],
        "resource": ["qcs::cam::uin/1[redacted]8:roleName/DTS-role"]
    }
  ]
}
```

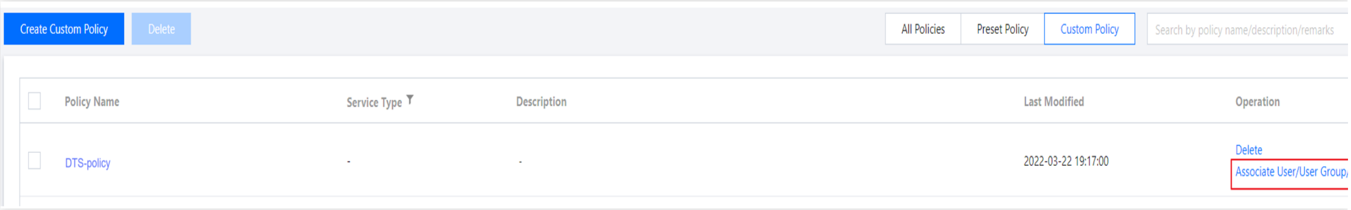
The root account ID of Source Database

Role name

10. Sample policy syntax:

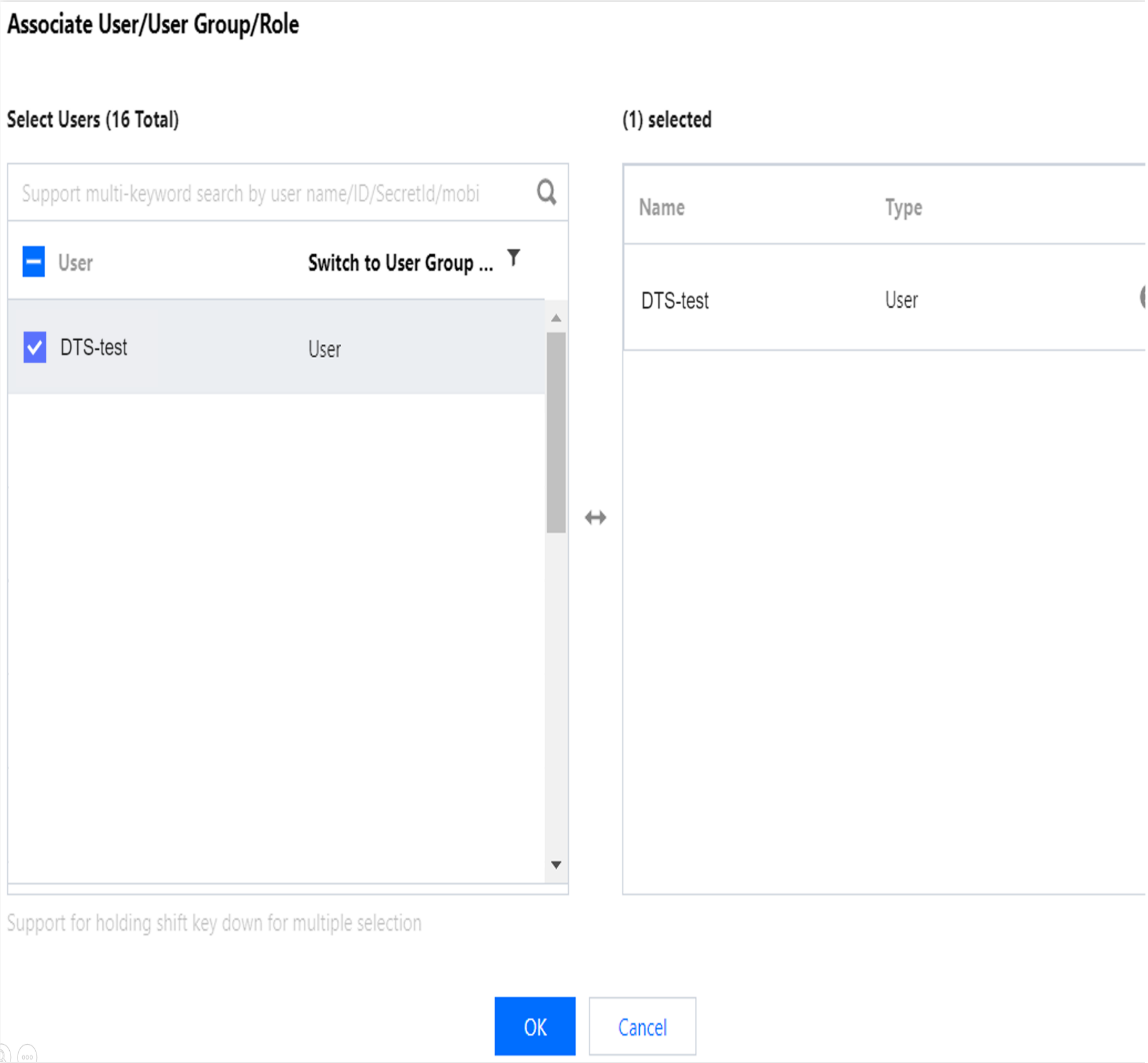
```
{
 "version": "2.0",
 "statement": [
 {
     "effect": "allow",
     "action": ["name/sts:AssumeRole"],
  "resource": ["qcs::cam::uin/10*******8:roleName/DTS-role"]
 }
 ]
}
```

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



□	Policy Name	Service Type	Description	Last Modified	Operation
□	DTS-policy	-	-	2022-03-22 19:17:00	Delete Associate User/User Group/

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

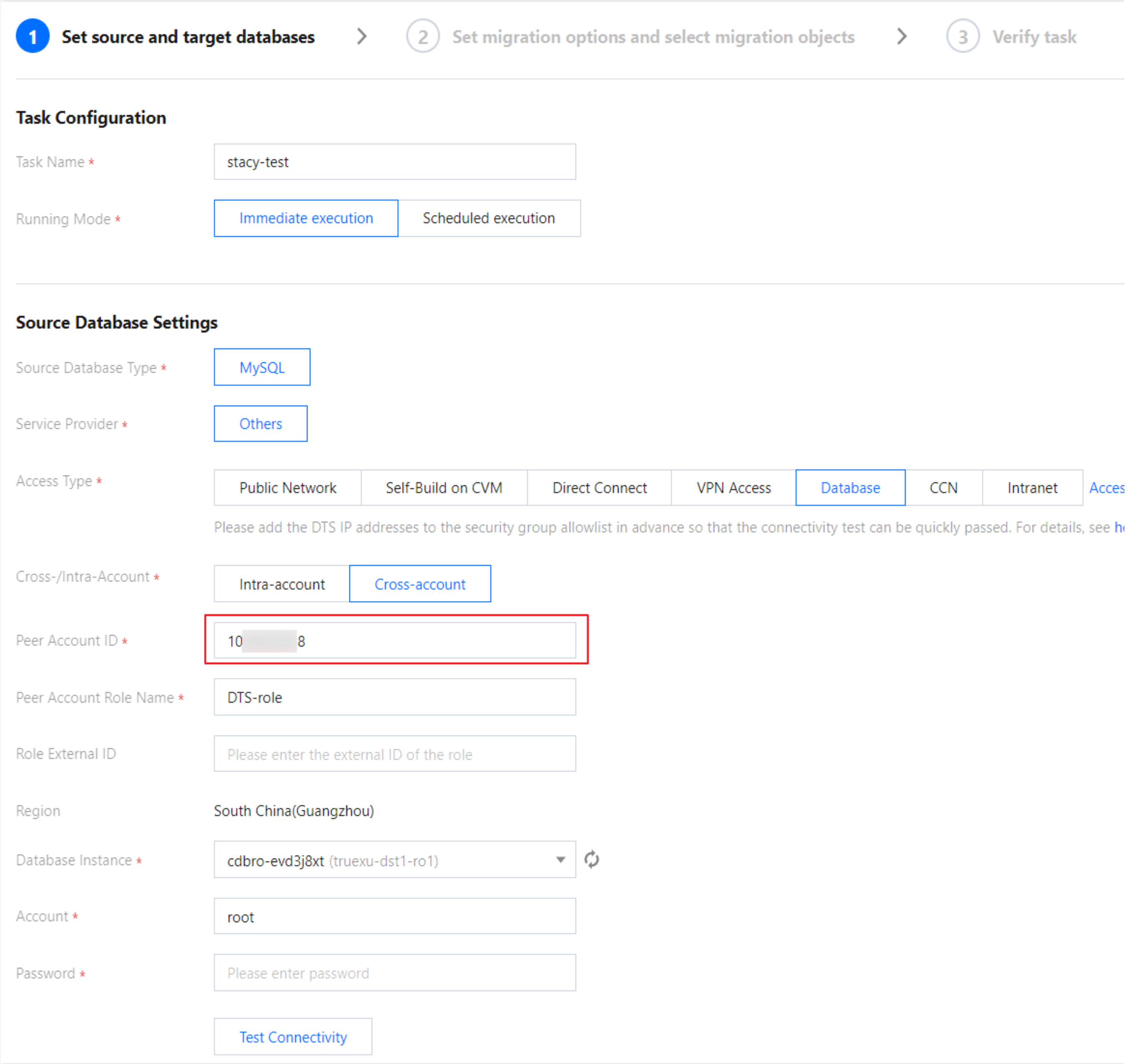

Associate User/User Group/Role
Select Users (16 Total)
(1) selected
Support multi-keyword search by user name/ID/SecretId/mobi
Switch to User Group ...
Support for holding shift key down for multiple selection
OK
Cancel

☐	User	
☑	DTS-test	User

Name	Type
DTS-test	User

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 | Acces

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 h

Cross-/Intra-Account * Intra-account | Cross-account

Peer Account ID * 10[redacted]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

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：2024-12-25 14:55:07

Overview

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.

Business Impact

1. DTS performs a read operation of the source database’s entire data during full data migration, which increases the source database’s load. If the database specifications are low, it is recommended to schedule the migration during off-peak hours or reduce the DTS speed before you start the task.

The impact varies with the specifications of the source database. For example, with a source database of 8 cores and 16 GB, DTS tasks adopt 8-thread concurrency (adjustable) by default. In a network without bottlenecks, the impact on the source database’s performance is as follows:

DTS full export phase: Use approximately 18%-45% of the source database’s CPU, increase query pressure by about 40-60 MB/s, and occupy roughly 8 active session connections.

DTS incremental export phase: Poses minimal pressure on the source database, with only one connection monitoring the binlog of the source database in real time.

2. 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.

3. 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.

Preparation

1. Ensure that the access channel between DTS and the database is established based on the access type you intend to use. For details, see Network Preparation Work.

2. Grant the required permissions to the task execution account in the source database, as detailed below:

Migration of the entire instance:

```
CREATE USER 'migration_account'@'%' IDENTIFIED BY 'migration password';
GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT, REPLICATION SLAVE, SHOW DATABASES, S
//If the Source Database is Alibaba Cloud Database, SHOW DATABASES authorization is
//If you choose to migrate triggers and events, TRIGGER and EVENT permissions are r
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, S
//If the Source Database is Alibaba Cloud Database, SHOW DATABASES authorization is
//If you choose to migrate triggers and events, you need to grant both TRIGGER and
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration_account'@'%';
GRANT SELECT ON `mysql`.* TO 'migration_account'@'%';
GRANT SELECT ON libraries to be migrated.* TO 'migration_account';
```

3. Grant the task execution account the following required permissions in 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, UPDATE.

Usage Instructions

Migration Objects

1. Basic tables, views, functions, triggers, stored procedures, and events can be migrated. System database tables, such as `information_schema` , `sys` , `performance_schema` , `__cdb_recycle_bin__` , `__recycle_bin__` , `__tencentdb__` , and `mysql` can not be migrated.
2. Related data objects should be migrated together to avoid migration failure. Common relationships include views referencing tables, views referencing other views, and primary-foreign key relationships.
3. It is recommended to migrate tables with primary keys or non-null unique keys to prevent data duplication.
4. When migrating views, stored procedures, and functions, DTS checks whether user1, corresponding to the `DEFINER` ([DEFINER = user1]) in the source database, matches the migration account user2. If they do not match, after migration, DTS modifies the `SQL SECURITY` attribute of user1 in the target database from `DEFINER` to `INVOKER` ([INVOKER = user1]), and sets the `DEFINER` in the target database to the migration account user2 ([DEFINER = migration account user2]). If the view definitions in the source database are too complex, the migration task may fail.
5. When the source database is Alibaba Cloud RDS or PolarDB, since RDS and PolarDB add additional primary key columns to tables without primary keys or non-null unique keys in the Binlog (which are not visible in the table structure), DTS may fail to recognize them. Therefore, it is recommended to avoid migrating tables without primary keys.
6. Only databases using the InnoDB, MyISAM, or TokuDB engines can be migrated. Tables using other database engines will be skipped by default. If the source database contains compressed TokuDB engine data, the target database should support compression mode for migration; otherwise, the task will fail.
7. When the target database is a Tencent Cloud MySQL instance using the RocksDB engine, the source InnoDB/TokuDB engines can be converted to the target RocksDB engine. However, there are specific constraints for converting the InnoDB to RocksDB engine:
7.1 If the primary key or part of the primary key in a table to be migrated from the source database is of the types TINYTEXT, TEXT, MEDIUMTEXT, or LONGTEXT, these types do not support value equality on the target side. Therefore, SQL statements related to value equality (such as DELETE FROM table_name WHERE primary_key_column = 'some text') will not work after synchronization, potentially leading to data inconsistency between the source and target databases.
7.2 During data consistency verification, if the primary key or part of the primary key in a table to be migrated from the source database is of the types TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT, and the table contains more than 50,000 rows, the table will be skipped, and no consistency verification will be performed.
7.3 Due to differences in the underlying storage of DOUBLE data types between RocksDB and InnoDB engines, if the source DOUBLE value approaches its upper limit (such as -1.7976931348623157E+308), false negatives may occur when you create consistency verification tasks. While the actual data may be consistent, the verification results might show discrepancies.
7.4 Tables involving partitions are not supported for migration.

Data Types

1. During incremental migration, if the source database generates Binlog statements in the `STATEMENT` format, migration will fail.
2. If there are gaps in the GTID of the source database's Binlog, it may impact migration task performance and lead to task failure.
3. Scenarios where DML and DDL statements are included in the same transaction are not supported. In such cases, the task will return an error.
4. Geometry-related data types are not supported. Tasks involving these data types will fail.
5. `ALTER VIEW` statements are not supported. Such statements will be skipped and not migrated.
6. JSON data type specifics for migration linkages between MySQL and TDSQL-C MySQL
6.1 For data requiring precision, it is recommended to use types such as DECIMAL for separate recording rather than JSON. This is because MySQL's handling of floats in JSON can result in a loss of precision for numbers with six or more decimal places.
6.2 DTS handling of floats in JSON types
When MySQL 5.7 or earlier is the target database, due to issues inherent in MySQL 5.7, floats x.0 in JSON on the source may be converted to x on the target during synchronization. For example, source data {"a":12.0} would be synchronized to the target as {"a":12}.
6.2.1 When MySQL 5.7 or earlier is the source database and MySQL 8.0 or later is the target database, due to issues inherent in MySQL 5.7, the float x.0 in JSON on the source might appear as x when read via SELECT. However, the binlog still records the value as a floating-point type, and DTS processes it accordingly, resulting in x.0 on the target database. For example, source data {"a":12.0} may appear as {"a":12} in SELECT results but will be synchronized as {"a":12.0} to the target.
6.2.2 Regardless of the version, if the float in the JSON type from the source database has more than one decimal place, such as {"a":12.00} or {"a":12.000}, it will be synchronized to the target database as {"a":12.0}.

Additional Instructions

1. If the source database is a non-GTID instance, DTS does not support HA switch at the source side. If the source MySQL switches, it may cause the DTS incremental migration to be interrupted.
2. It is recommended not to perform dual writes on the target database during the task, as this may cause task failures or inconsistencies in the synchronized data.
Full synchronization phase: When DTS **writes the initial full data** to the target database and detects primary key conflicts, the task will **report an error** .
Incremental synchronization phase: When DTS **writes newly added data from the full synchronization phase** into the target database and detects primary key conflicts, it will **overwrite** the target database with data from the source. When DTS **writes newly added data from the incremental synchronization phase** to the target database and detects primary key conflicts, the task will **report an error** .

Operation Restrictions

Avoid performing the following operations during migration to prevent migration task failures:

1. In scenarios involving structural migration and full migration, DDL operations are not supported during the Full Export step of the migration task.
2. Do not modify or delete user information (including usernames, passwords, and permissions) or port numbers in the source or target databases.
3. Avoid executing operations to clear Binlog on the source database.

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 `CREATE TABLE table name AS SELECT` is not supported.

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**.

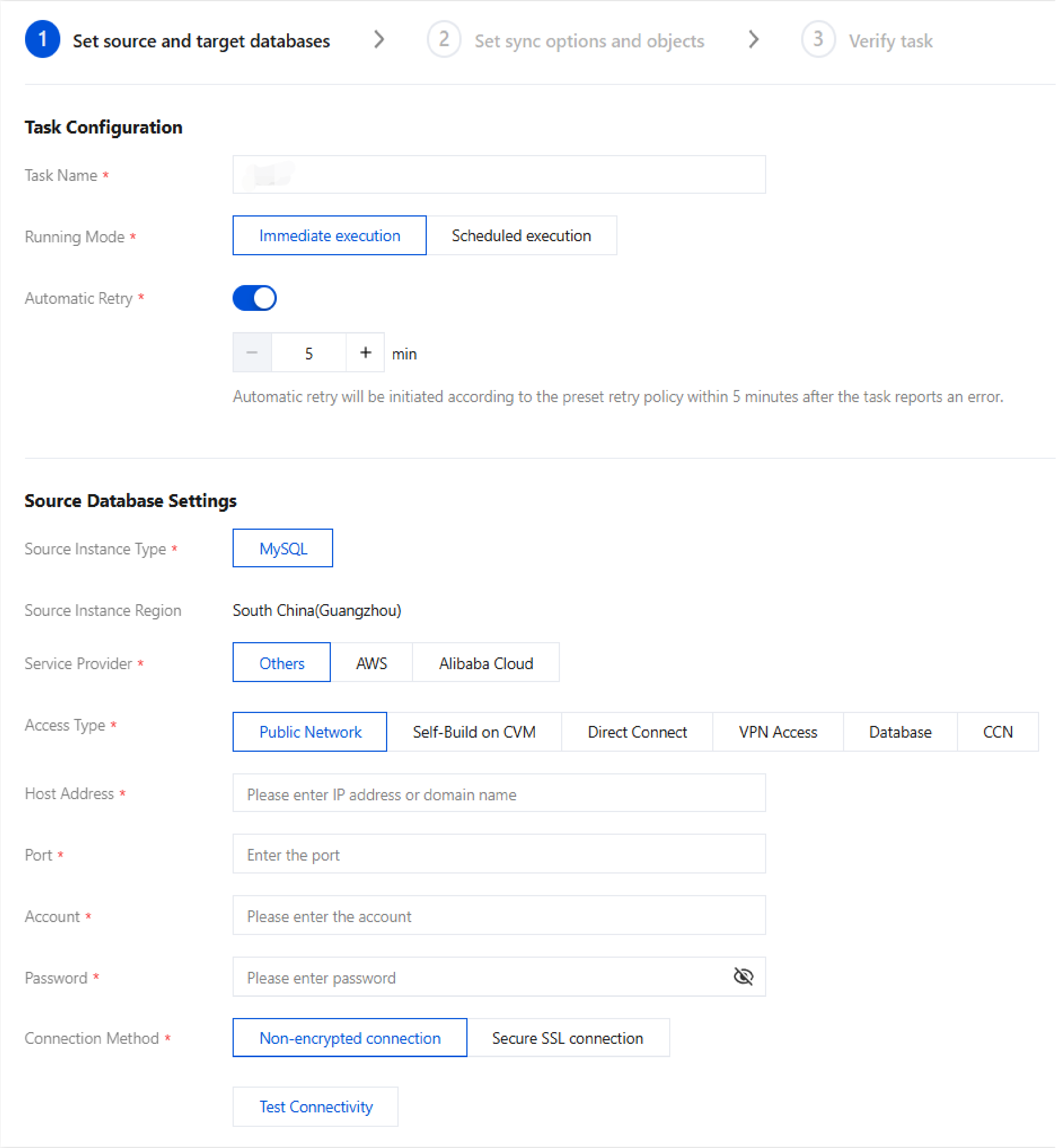
Source Instance Region	This refers to the source region of the DTS service. If the source database is TencentDB, select its region here. If the source database is self-built or in another cloud, select the region closest to it, so that DTS can choose the optimal migration path to reduce the migration 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. After the purchase is completed, the page automatically redirects to the data migration task list. Select the task you just purchased for configuration. If you have purchased tasks across multiple regions or for cross-region DTS, the task list displays them based on the target instance’s region. You can switch regions at the top of the page to locate the purchased tasks.

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 **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 sync options and objects
3 Verify task

Task Configuration

Task Name *

Running Mode * Immediate execution | Scheduled execution

Automatic Retry *

- 5 + min

Automatic retry will be initiated according to the preset retry policy within 5 minutes after the task reports an error.

Source Database Settings

Source Instance Type * MySQL

Source Instance Region South China(Guangzhou)

Service Provider * Others | AWS | Alibaba Cloud

Access Type * Public Network | Self-Build on CVM | Direct Connect | VPN Access | Database | CCN

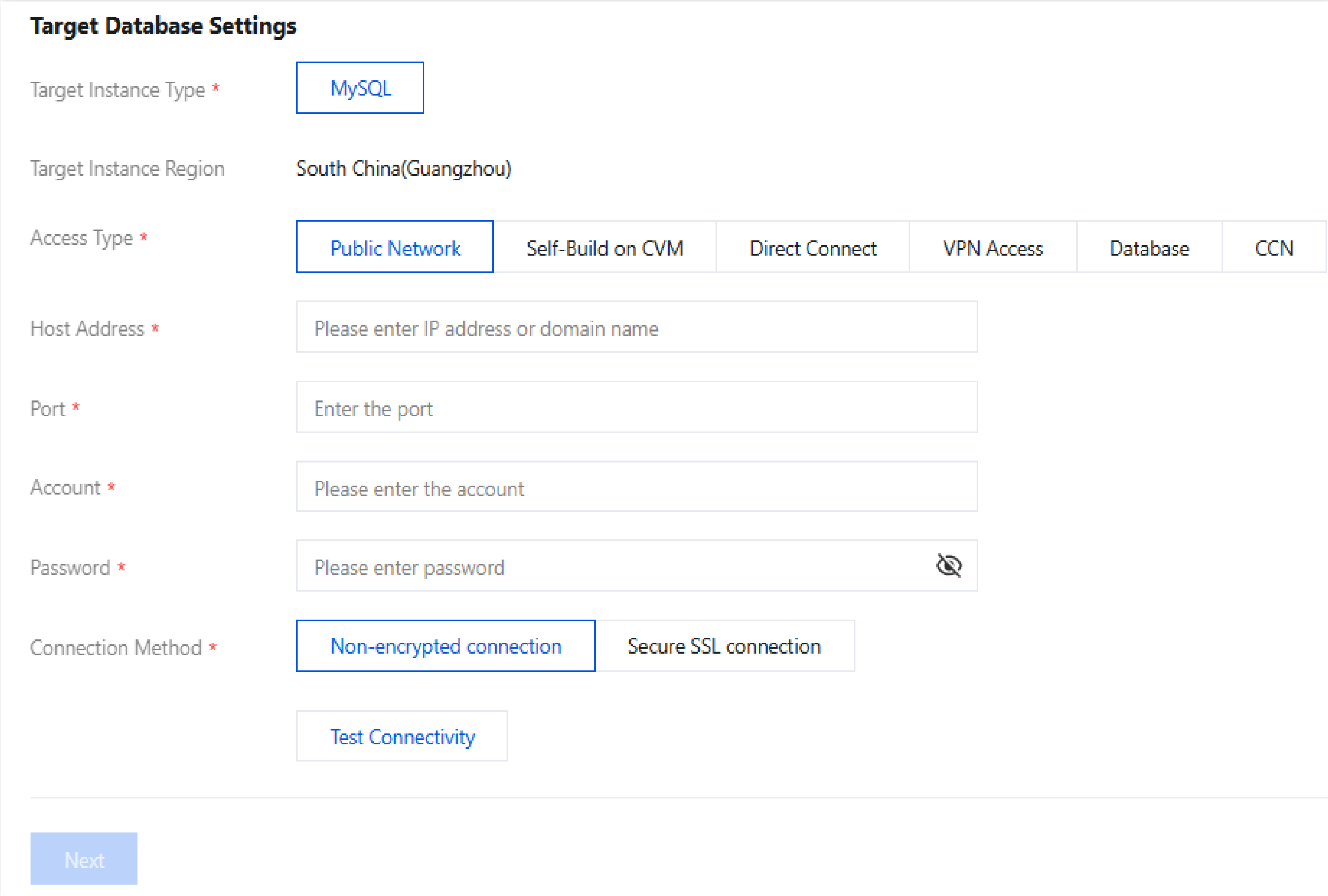
Host Address * Please enter IP address or domain name

Port * Enter the port

Account * Please enter the account

Password * Please enter password

Connection Method * Non-encrypted connection | Secure SSL connection



Target Database Settings

Target Instance Type *	MySQL
Target Instance Region	South China(Guangzhou)
Access Type *	Public Network / Self-Build on CVM / Direct Connect / VPN Access / Database / CCN
Host Address *	Please enter IP address or domain name
Port *	Enter the port
Account *	Please enter the account
Password *	Please enter password
Connection Method *	Non-encrypted connection / Secure SSL connection

Task Configuration

Parameter	Description
Task Name	DTS automatically generates a task name. It is recommended to change it to a meaningful name for easier task identification.
Running Mode	Immediate execution: The task is started immediately after passing the pre-verifications. Scheduled execution: Set a specific start time for the task. The task does not start immediately after passing the pre-verifications but waits until the scheduled time.
Automatic Retry	Once it is enabled, if a synchronization task is temporarily interrupted due to network issues or similar causes, DTS will automatically retry and resume the task within the specified time frame without requiring manual intervention. The supported range is 5 to 720 minutes.

Source Database Settings

Parameter	Description
Source Instance Type	The type of source instance selected at the time of purchase, which cannot be modified.
Source Instance	The region of the source instance selected at the time of purchase, which cannot be

Region	modified.
Service Provider	For self-built databases (including those on CVMs), Tencent Cloud database instances, and lightweight databases, select Others. For third-party cloud provider databases, select the corresponding service provider.
Access Type	Choose based on the deployment setup of the source database. Different deployment scenarios require specific network configurations. For details, see Network Preparation Work Overview. For source databases hosted on self-built IDC environments or other cloud providers, the access types include Public network/DC/VPN access/CCN. Public Network: The source database can be accessed via a public IP address. Direct Connect: The source database can connect to Tencent Cloud VPC using DC. VPN Access: The source database can connect to Tencent Cloud VPC using VPN Connections. CCN: The source database can connect to Tencent Cloud VPC using CCN. VPC: Both the source data and target database are deployed on Tencent Cloud and within a VPC. If you need to use the VPC access type, submit a ticket for application. If the source database is self-built on a CVM, select Self-built on CVM as the access type. If the source database is a Tencent Cloud database instance, select Cloud Database as the access type.
Public Network	When Public Network is selected as the **Access Type**, configure the following parameters: Host Address: The IP address or domain name of the source database. Port: The port used by the source database.
Self-Build on CVM	When Self-Build on CVM is selected as the **Access Type**, configure the following parameters: CVM Instance: The instance ID of the CVM. Port: The port used by the source database.
Direct Connect	When Direct Connect is selected as the **Access Type**, configure the following parameters: VPC-Based Direct Connect Gateway: Only VPC Direct Connect Gateway is supported for DC. Confirm the network type associated with the gateway. VPC: Select the VPC and subnet. Host Address: The IP address of the source database. Port: The port used by the source database.
VPN Access	When VPN Access is selected as the **Access Type**, configure the following parameters: VPN gateway: Select the VPN gateway based on its ID. VPC: Select the VPC and subnet associated with the VPN gateway. Host Address: The IP address of the source database. Port: The port used by the source database.
Database	When Database is selected as the **Access Type**, configure the following parameters:

	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. For cross-account operations, see Cloud Database Cross-Account Instance Synchronization. Database Instance: The instance ID of the source database.
CCN	When CCN is selected as the **Access Type**, configure the following parameters: CCN access supports both same-account and cross-account configurations. Due to the complexity of network configuration, see Migrate Self-built Database to Tencent Cloud Database via CCN for detailed guidance. Host Network Environment: Select based on the actual scenario. For example, select Tencent Cloud if the source database is a Tencent Cloud database instance, Self-built IDC if it is a self-built IDC database, or the corresponding network for databases from other cloud providers. Host Address: The host IP address of the source database. Port: The port used by the source database. CCN Instance Account Type My account: The CCN resources and target database belong to the same Tencent Cloud root account. Other account: The CCN resources and target database belong to different Tencent Cloud root accounts. VPC-Based CCN Instance: Name of the CCN instance. CCN-associated VPC and subnet: The CCN-associated VPC refers to the VPC connected to the synchronization linkage within the CCN. Select a VPC from all the VPCs associated with the CCN, excluding the VPC of the source database. Region of the VPC: The region of the VPC used for connection should match the region of the source database selected during task purchase. If there is a mismatch, DTS will automatically update the source database region to align with the VPC region.
Account/Password	Account/Password: The account and password of the source database.
Connection Method	Currently, if users want to experience the SSL secure connection feature, submit a ticket for application. SSL secure connection refers to encrypting the transmission linkage between DTS and the database using SSL. Enabling SSL secure connection may increase the connection response time of the database. Generally, Tencent Cloud’s private network linkage is relatively secure, and enabling SSL secure connection is not necessary. However, for scenarios using public network, DC, or other transmission methods where higher data security is required, enabling SSL secure connection is recommended. Ensure that SSL encryption is enabled in the database before selecting SSL secure connection.

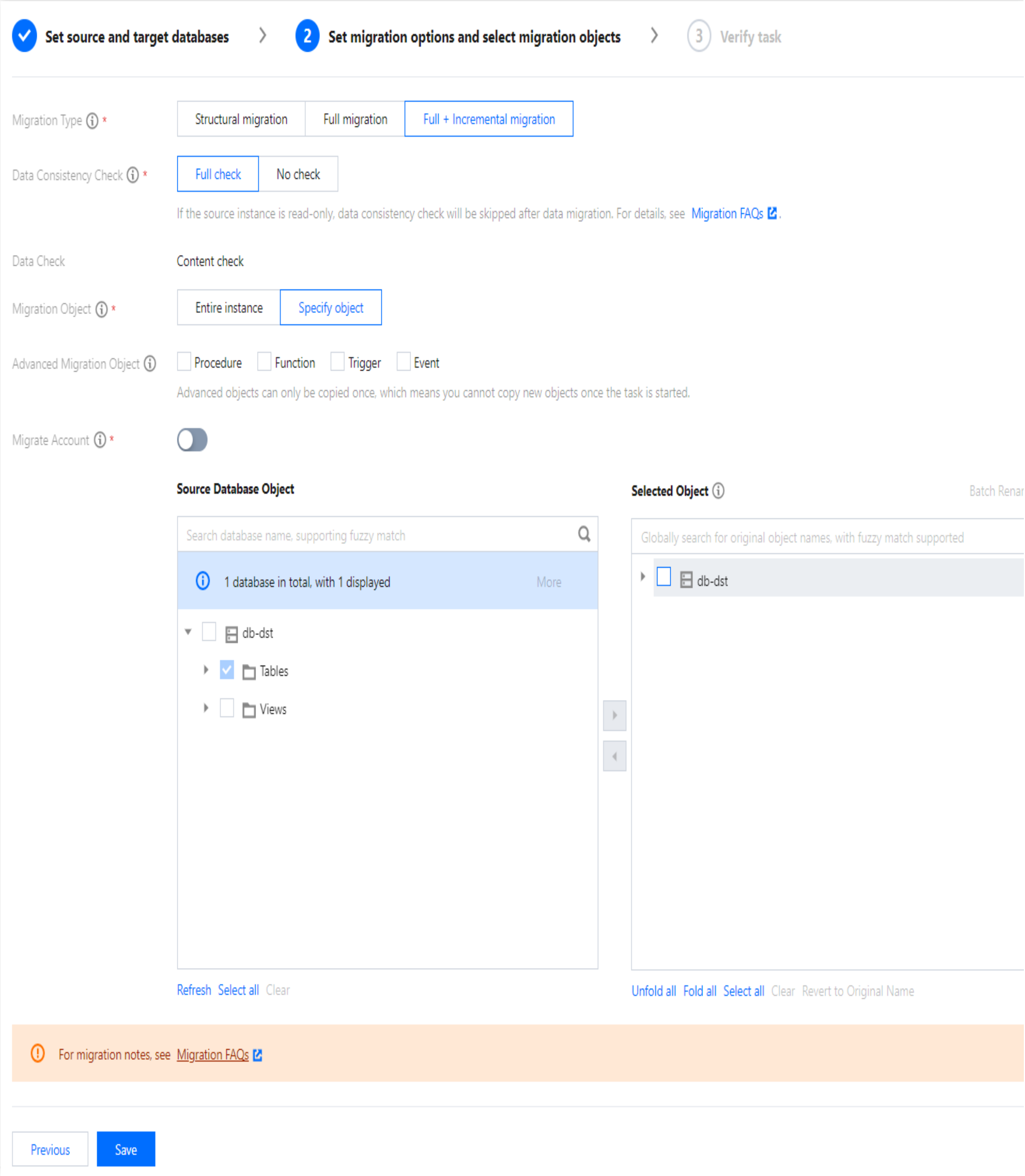
Target Database Settings

The parameter configuration for the target database is similar to that of the source database. Select the access type based on the actual situation; further explanation is omitted here.

5. 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.



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 Renar

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

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. 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. 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.
Data Consistency Check	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. 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. 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.
Migration Object	Entire instance: Migrate the entire database instance excluding the system databases such as `information_schema` , `mysql` , `performance_schema` , and `sys` . Specified objects: Migrate specified objects.
Advanced Migration Object	Procedures, functions, triggers, and events can be migrated. 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. 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. For more information, see Migrating Advanced Object.
Selected Object	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. We recommend you not rename tables when migrating advanced objects; otherwise, the migration may fail.
Sync Online DDL Temp Table	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. 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. For more information, see Migrating Online DDL Temp Table.

Advanced Options

Configuration Item	Description
Migrate accounts	Check this feature if you need to migrate account information from the source database to the target database. For more details, see Account Migration. In migration linkages with AWS as the source, users who migrate accounts should manually install the auth_socket plugin in the target database.
Limit transmission rate	This configuration is generally not required. It should only be adjusted when the database's configuration is insufficient to handle the default DTS concurrent thread count and RPS. For more details, see Rate Limiting.

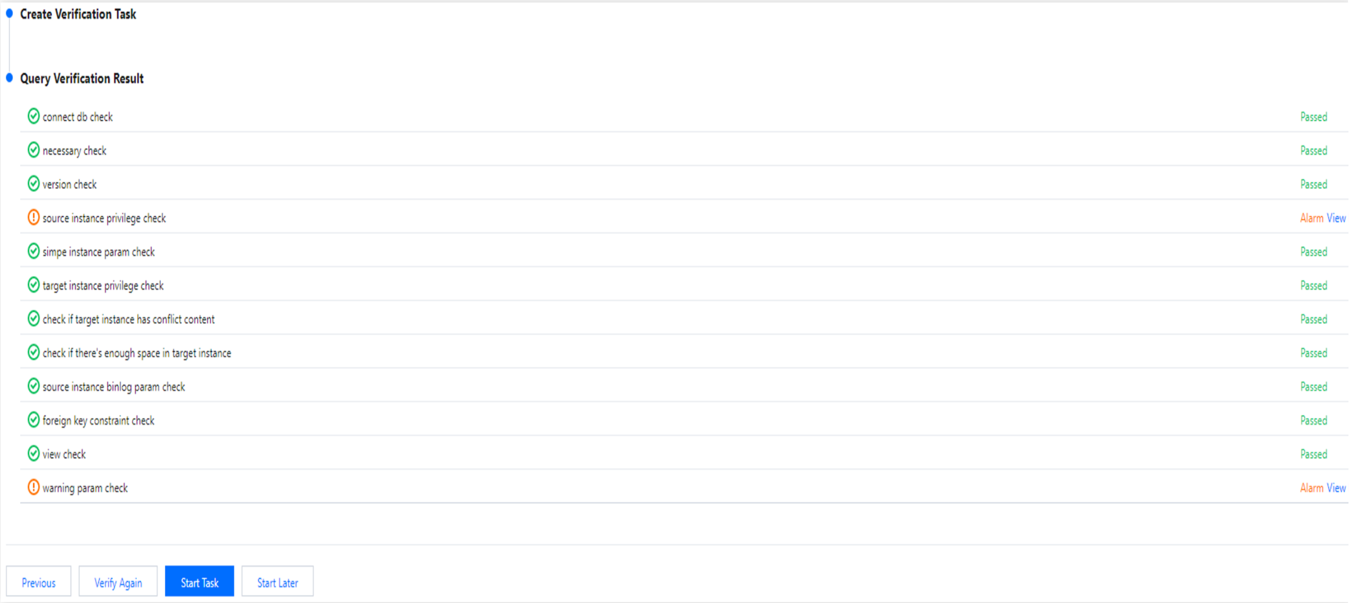
6. On the task verification page, verify the task. After the verification is passed, click **Start Task**.

If the verification fails, fix the problem as instructed in Check Item Overview and initiate the verification again. 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	Passed
✓ necessary check	Passed
✓ version check	Passed
! source instance privilege check	Alarm View
✓ simpe instance param check	Passed
✓ target instance privilege check	Passed
✓ check if target instance has conflict content	Passed
✓ check if there's enough space in target instance	Passed
✓ source instance binlog param check	Passed
✓ foreign key constraint check	Passed
✓ view check	Passed
! warning param check	Alarm View

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.

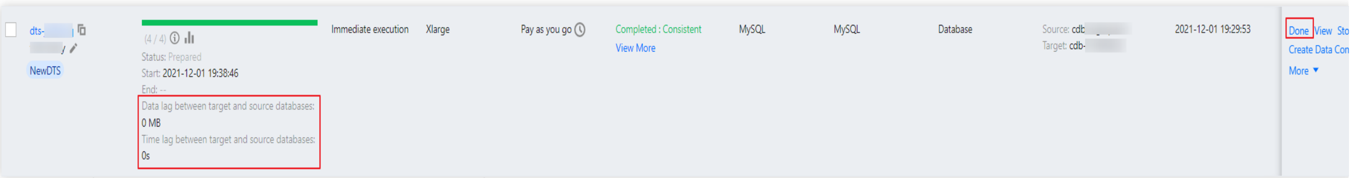
Select **Structural migration** or **Full migration**: Once completed, the task will be stopped automatically.

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 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.



dts-[redacted]
[redacted]
NewDTS

(4 / 4)
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

Immediate execution
Xlarge
Pay as you go
Completed : Consistent
View More
MySQL
MySQL
Database
Source: cdb[redacted]
Target: cdb-[redacted]
2021-12-01 19:29:53

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 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 VIEW,PROCESS ON *.* TO 'migration account'@'%';
// If the source database is a TencentDB for MariaDB database, you need to
submit a ticket to authorize `RELOAD`; otherwise, you can authorize by
referring to the sample code
// If you select to migrate triggers and events, you need grant both the
`TRIGGER` and `EVENT` permissions.
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 VIEW, PROCESS ON *.* TO 'migration account'@'%';
// If the source database is a TencentDB for MariaDB database, you need to
submit a ticket to authorize `RELOAD`; otherwise, you can authorize by
referring to the sample code
// If you select to migrate triggers and events, you need grant both the
`TRIGGER` and `EVENT` permissions.
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_tame_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 `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 Requirement
Requirements for the source database	The source and target databases can be connected. The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected. Requirements for instance parameters: The `server_id` parameter in the source database must be set manually and cannot be 0. `row_format` for the source databases/tables cannot be set to `FIXED` . The values of the `lower_case_table_names` variable in both the source and target databases must be the same. The `connect_timeout` variable in the source database must be greater than or equal to 10. We recommend you enable `skip-name-resolve` to reduce the possibility of connection timeout. Requirements for binlog parameters: The `log_bin` variable in the source database must be set to `ON` . The `binlog_format` variable in the source database must be set to `ROW` . The `binlog_row_image` variable in the source database must be set to `FULL` . On MariaDB 10.2 or later and Percona 5.6 or later, if the `gtid_mode` variable is not `ON` , an alarm will be triggered. We recommend you enable `gtid_mode` . You cannot set filter conditions with `do_db` and `ignore_db` .

	If the source instance is a replica database, the `log_slave_updates` variable must be set to `ON` . 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. Foreign key dependency: Foreign key dependency can be set to only one of the following two types: `NO ACTION` and `RESTRICT` . During partial table migration, tables with foreign key dependency must be migrated. The migration precision of DTS for data in `FLOAT` type is 38 digits, and for data in `DOUBLE` type is 308 digits. You should check whether this meets your requirements. The environment variable `innodb_stats_on_metadata` must be set to `OFF` .
Requirements for the target database	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. The target database cannot have migration objects such as tables and views with the same name as those in the source database. The `max_allowed_packet` parameter of the target database must be set to 4 MB or above.

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 to MySQL from TDSQL-C for MySQL

Last updated：2025-01-02 11:15:51

The steps of data migration from TDSQL-C for MySQL to TencentDB for MySQL are basically the same as those of migration from MySQL to 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/TDSQL-C MySQL)

Last updated：2025-06-04 17:07:08

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 VIEW,PROCESS ON *.* TO 'migration account'@'%';
// If the source database is a TencentDB for MariaDB database, you need to
submit a ticket to authorize `RELOAD`; otherwise, you can authorize by
referring to the sample code
// If the source database is an Alibaba Cloud database, you don't need to grant
the `SHOW DATABASES` permission; otherwise, you need to do so. For more
information on authorizing an Alibaba Cloud database, visit
https://help.aliyun.com/document_detail/96101.html.
// If you select to migrate triggers and events, you need grant both the
`TRIGGER` and `EVENT` permissions.
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 VIEW,PROCESS ON *.* TO 'migration account'@'%';
// If the source database is a TencentDB for MariaDB database, you need to
submit a ticket to authorize `RELOAD`; otherwise, you can authorize by
referring to the sample code
// If the source database is an Alibaba Cloud database, you don't need to grant
the `SHOW DATABASES` permission; otherwise, you need to do so. For more
information on authorizing an Alibaba Cloud database, visit
https://help.aliyun.com/document_detail/96101.html.
// If you select to migrate triggers and events, you need grant both the
`TRIGGER` and `EVENT` permissions.
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_tame_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.

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	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 `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	The source and target databases can be connected. The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected. Requirements for instance parameters: The `server_id` parameter in the source database must be set manually and cannot be 0. `row_format` for the source databases/tables cannot be set to `FIXED` . The values of the `lower_case_table_names` variable in both the source and target databases must be the same. The `connect_timeout` variable in the source database must be greater than or equal to 10. We recommend that you enable `skip-name-resolve` to reduce the possibility of connection timeout. Requirements for binlog parameters: The `log_bin` variable in the source database must be set to `ON` . The `binlog_format` variable in the source database must be set to `ROW` . The `binlog_row_image` variable in the source database must be set to `FULL` . On MySQL 5.6 or later, if the `gtid_mode` variable is not `ON` , an alarm will be triggered. We recommend you enable `gtid_mode` . You cannot set filter conditions with `do_db` and `ignore_db` . If the source instance 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. Foreign key dependency: Foreign key dependency can be set to only one of the following two types: `NO ACTION` and `RESTRICT` . During partial table migration, tables with foreign key dependency must be migrated. The migration precision of DTS for data in `FLOAT` type is 38 digits, and for data in `DOUBLE` type is 308 digits. You should check whether this meets your requirements. The environment variable `innodb_stats_on_metadata` must be set to `OFF` .
Requirements for the target database	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. The target database cannot have migration objects such as tables and views with the same name as those in the source database. 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 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	Select the source database type, which cannot be changed after purchase. Here, select **MariaDB**. 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.
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 **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 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. 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.
	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.
	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.
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 account.
	Connection	To try out the SSL connection feature, submit a ticket for application.

	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 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.

Configuration Items	Description
Migration Type	Select an option as needed. Structural migration: Structured data such as databases and tables in the database will be migrated. 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.
Migration Object	Entire instance: Migrate the entire database instance excluding the system databases such as `information_schema` , `mysql` , `performance_schema` , and `sys` . Specified objects: Migrate specified objects.
Advanced Migration Object	Procedures, functions, triggers, and events can be migrated. 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. 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. For more information, see Migrating Advanced Object.
Selected Object	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. We recommend that you not rename tables when migrating advanced objects; otherwise, the migration may fail.
Sync Online DDL Temp Table	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. 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. 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.

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 **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：2025-01-21 19:33:56

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,SHOW VIEW,PROCESS ON *.* TO 'migration account'@'%';
// If the source database is a TDSQL for MySQL database, you need to submit a
ticket to authorize `RELOAD`; otherwise, you can authorize by referring to the
sample code
GRANT INSERT, UPDATE, DELETE, DROP, SELECT, INDEX, ALTER, CREATE ON
`__tencentdb__`.* 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

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	The source and target databases can be connected. Requirements for instance parameters: `table_row_format` cannot be set to `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 in the target database must be at least 4 MB. The `connect_timeout` variable in the source database must be greater than or equal to 10. Requirements for binlog parameters: The `binlog_format` variable in the source database must be set to `ROW` . The `log_bin` variable in the source database must be set to `ON` . The `binlog_row_image` variable in the source database must be set to `FULL` . On v5.6 or later, if the `gtid_mode` variable in the source database is not `ON` , a warning will be triggered. We recommend you enable `gtid_mode` . It is not allowed to set `do_db` and `ignore_db` . If the source instance 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. Foreign key dependency: Foreign key dependency can be set to only one of the following two types: `NO ACTION` and `RESTRICT` . During partial table migration, tables with foreign key dependency must be migrated. The environment variable `innodb_stats_on_metadata` must be set to `OFF` .
Requirements for the target database	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. The target database version must be later than or equal to the source database version. The target database space must be at least 1.2 times the space of databases/tables to be migrated in the source database. The target database cannot have tables that conflict with the source database.

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	Select the source database type, which cannot be changed after purchase. Here, select **MySQL**. 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.
Source Instance Region	Select the source database region.
Target Instance Type	Select the target database type, which cannot be changed after purchase. Here, select **TDSQL for 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.

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**.

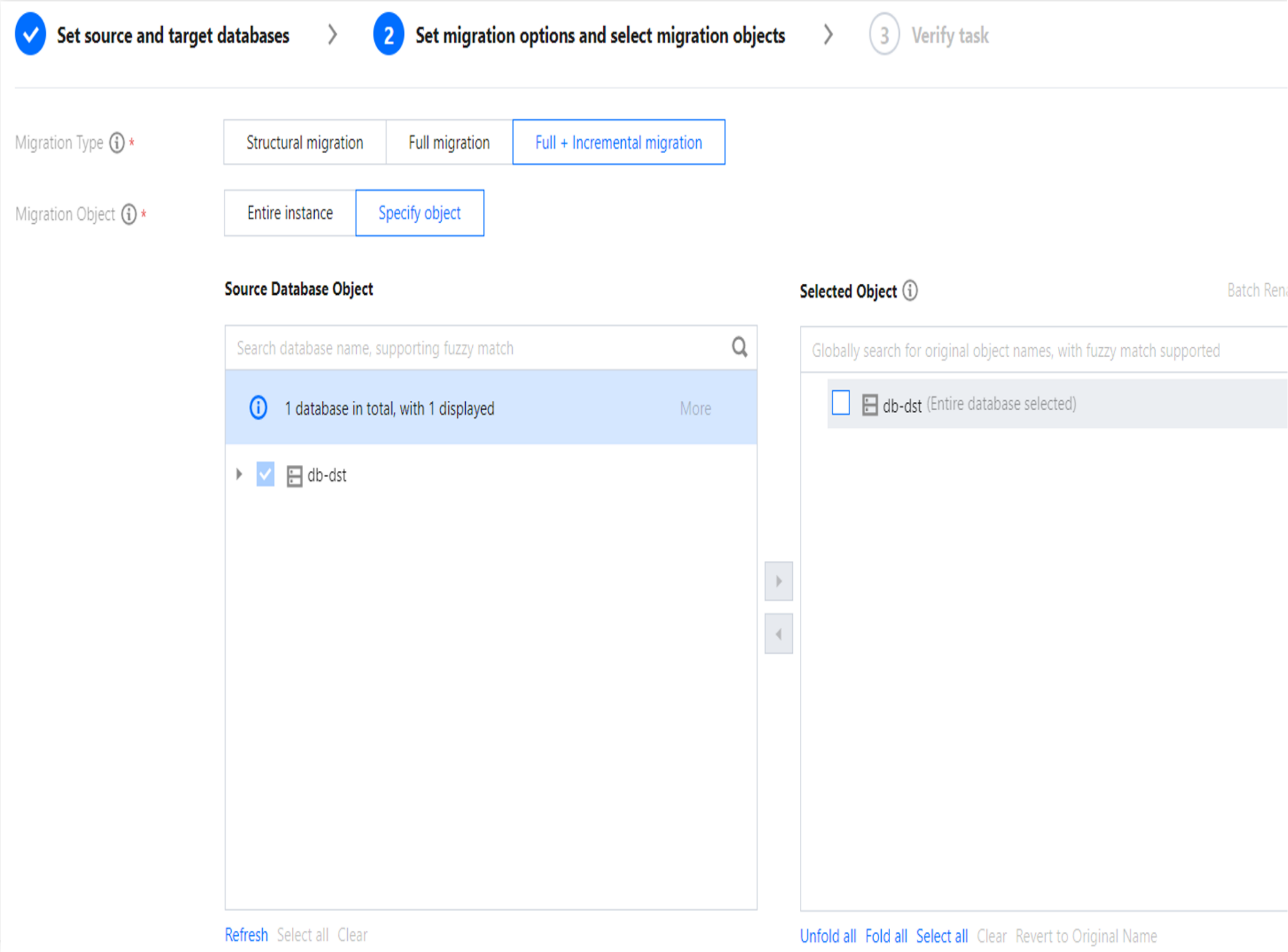
	Database Instance	Select the instance ID of the source database.
	Account	Account of the source TDSQL for MySQL database, which must have the required permissions.
	Password	Password of the source TDSQL for MySQL database account.
Target Database Settings	Target Database Type	The target database type 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 task

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

Refresh Select all Clear

Selected Object

Globally search for original object names, with fuzzy match supported

□ db-dst (Entire database selected)

Unfold all Fold all Select all Clear Revert to Original Name

Configuration Items	Description
Migration Type	Select an option as needed. Structural migration: Structured data such as databases and tables in the database will be migrated. 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.
Migration Object	Entire instance: Migrate the entire database instance excluding the system databases such as `information_schema` , `mysql` , `performance_schema` , and `sys` . Specified objects: Migrate specified objects.
Specify object	Select the objects to be migrated in **Source Database Object** and move them 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 **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]`).

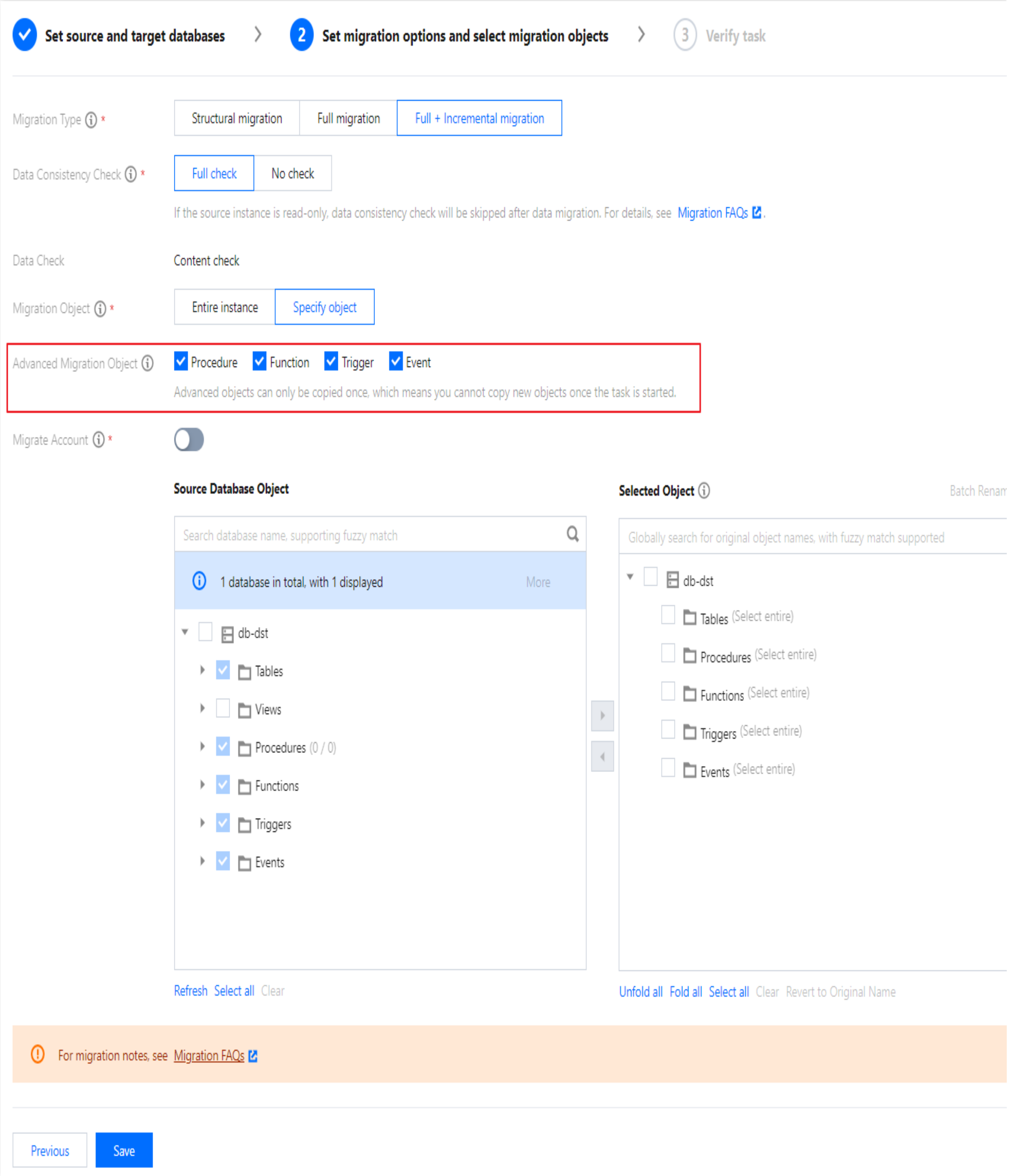
Procedures and functions will be migrated during **source database export**. If there are no incremental migration tasks, triggers and events will be migrated when the task stops; otherwise, they will be migrated after you click **Done**, in which case the transition will take a longer time.

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 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
☑ Procedures (0 / 0)
☑ Functions
☑ Triggers
☑ Events
Refresh Select all Clear
Selected Object
Batch Renam
Globally search for original object names, with fuzzy match supported
□ db-dst
□ Tables (Select entire)
□ Procedures (Select entire)
□ Functions (Select entire)
□ Triggers (Select entire)
□ Events (Select entire)
Unfold all Fold all Select all Clear Revert to Original Name
Previous
Save

For migration notes, see Migration FAQs

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 TDStore > TDSQL for TDStore

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

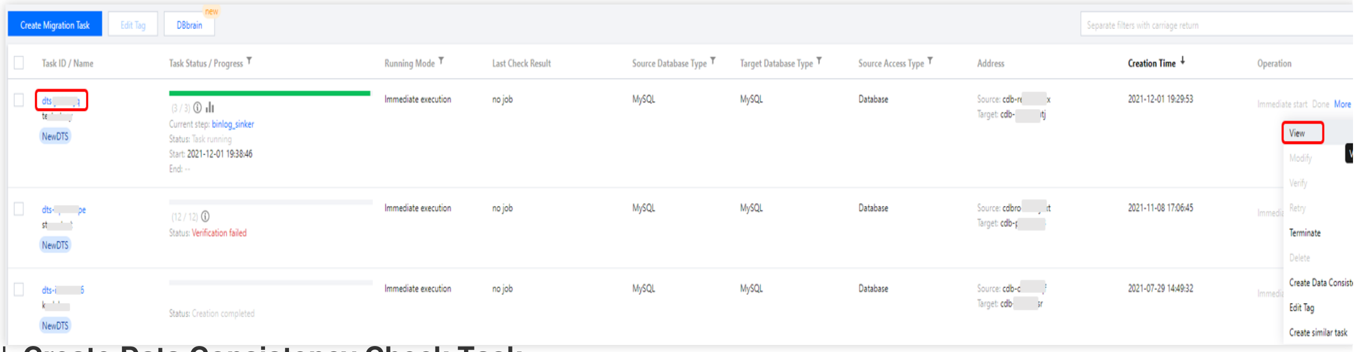
Automatic triggering

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.

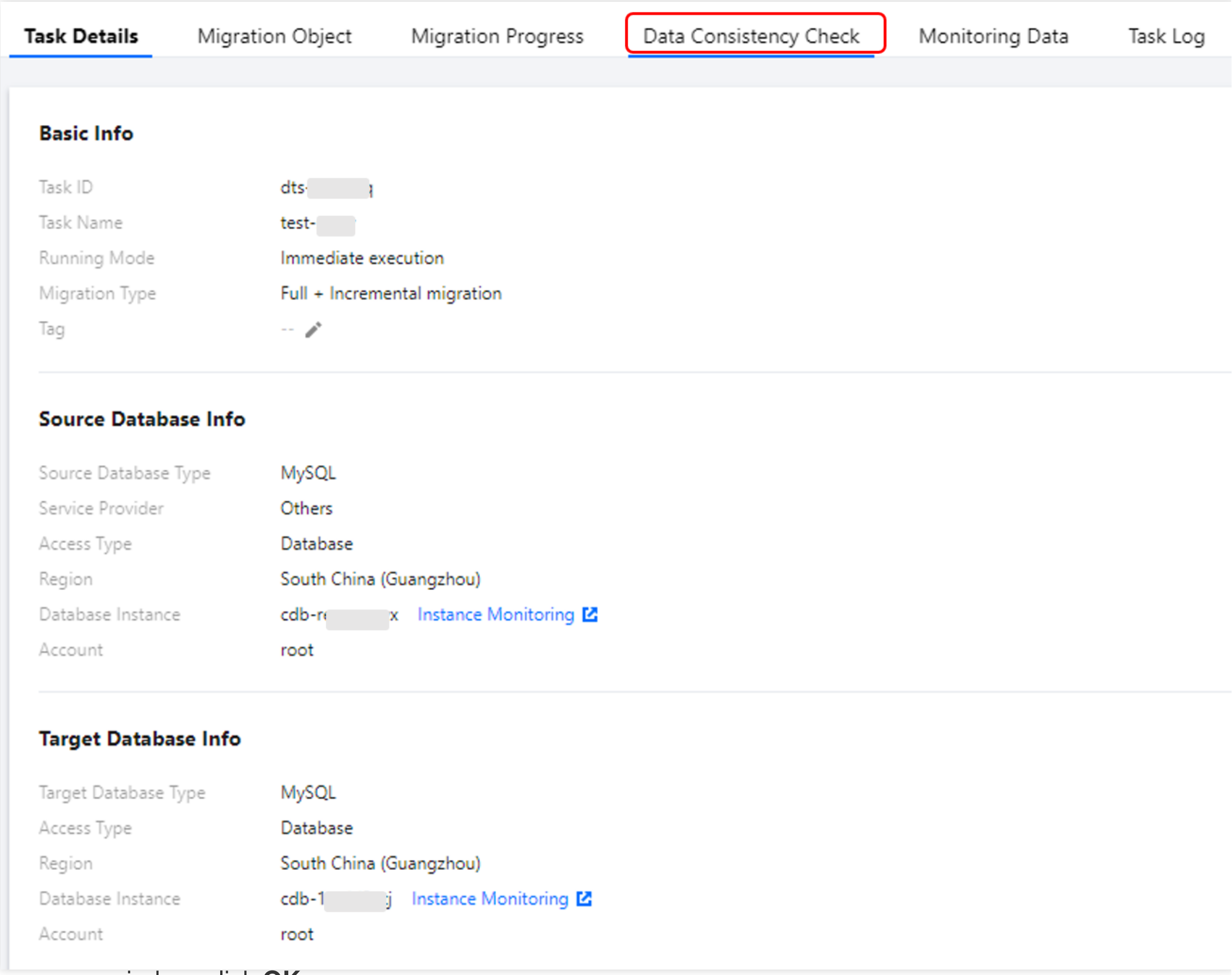


Task ID / Name	Task Status / Progress	Running Mode	Last Check Result	Source Database Type	Target Database Type	Source Access Type	Address	Creation Time	Operation
dts[redacted] te[redacted] NewDTS	(3 / 3) Current step: binlog_sinker Status: Task running Start: 2021-12-01 19:38:46 End: --	Immediate execution	no job	MySQL	MySQL	Database	Source: cdb-r[redacted]x Target: cdb-[redacted]tj	2021-12-01 19:29:53	Immediate start Done More
dts-[redacted]pe st[redacted] NewDTS	(12 / 12) Status: Verification failed	Immediate execution	no job	MySQL	MySQL	Database	Source: cdbro[redacted]t Target: cdb-[redacted]	2021-11-08 17:06:45	Immedi
dts-i[redacted]5 k[redacted] NewDTS	Status: Creation completed	Immediate execution	no job	MySQL	MySQL	Database	Source: cdb-c[redacted] Target: cdb[redacted]r	2021-07-29 14:49:32	Immedi

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

Note:

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.



Basic Info

Task ID dts[redacted]
Task Name test-[redacted]
Running Mode Immediate execution
Migration Type Full + Incremental migration
Tag --

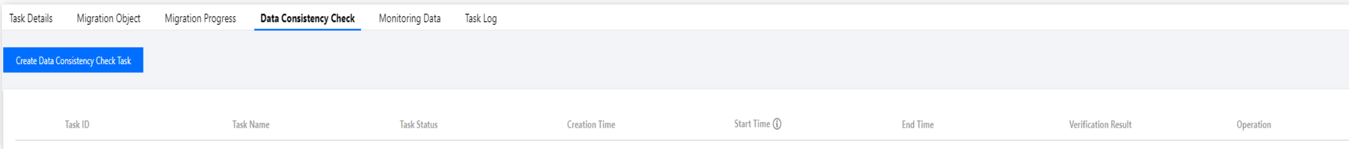
Source Database Info

Source Database Type MySQL
Service Provider Others
Access Type Database
Region South China (Guangzhou)
Database Instance cdb-r[redacted]x Instance Monitoring
Account root

Target Database Info

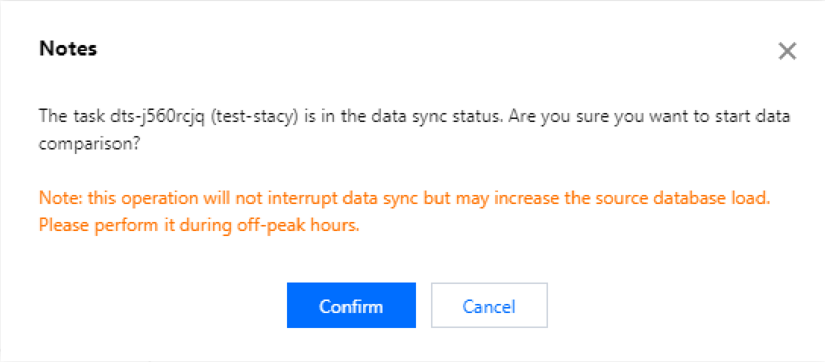
Target Database Type MySQL
Access Type Database
Region South China (Guangzhou)
Database Instance cdb-1[redacted]j Instance Monitoring
Account root

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



Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	End Time	Verification Result	Operation

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



Notes

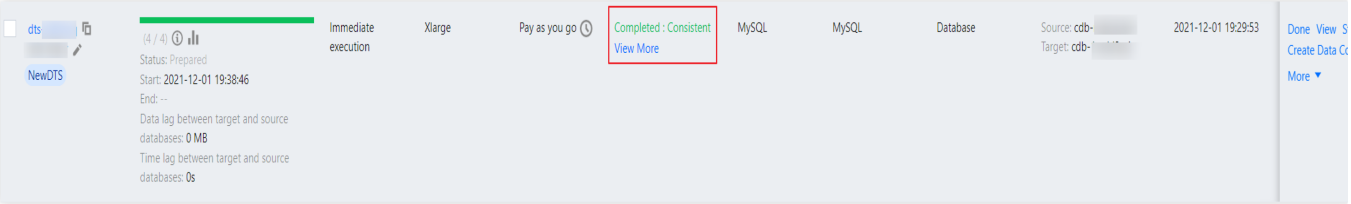
The task dts-j560rcjq (test-stacy) is in the data sync status. Are you sure you want to start data comparison?

Note: this operation will not interrupt data sync but may increase the source database load. Please perform it during off-peak hours.

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.

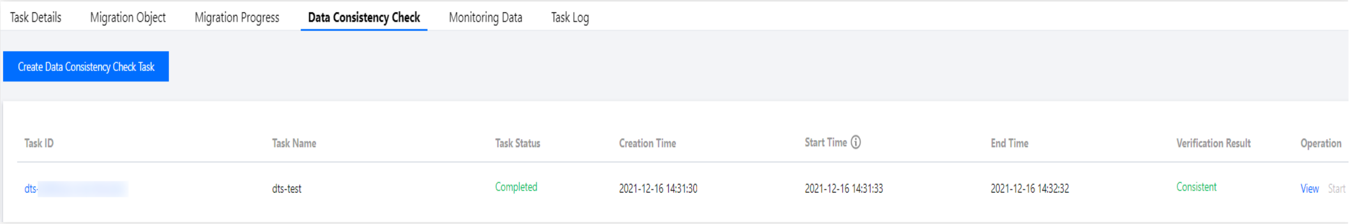


dts-[redacted]
[redacted]
NewDTS

(4 / 4)
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

Immediate execution | Xlarge | Pay as you go | Completed : Consistent View More | MySQL | MySQL | Database | Source: cdb-[redacted] Target: cdb-[redacted] | 2021-12-01 19:29:53

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

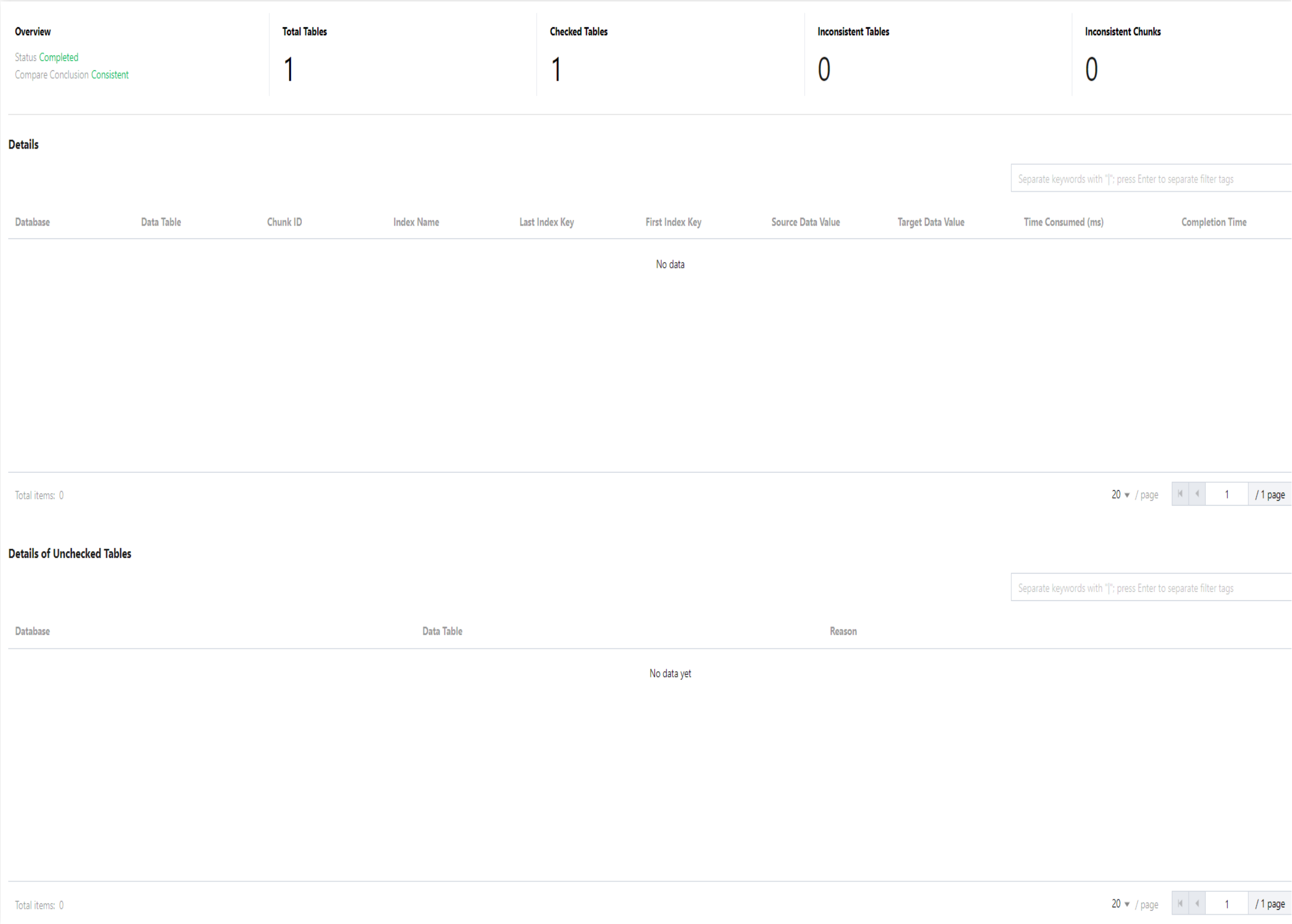


Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	End Time	Verification Result	Operation
dts-[redacted]	dts-test	Completed	2021-12-16 14:31:30	2021-12-16 14:31:33	2021-12-16 14:32:32	Consistent	View Start

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	Checked Tables	Inconsistent Tables	Inconsistent Chunks
1	1	0	0

Details

Separate keywords with "|"; press Enter to separate filter tags

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

Details of Unchecked Tables

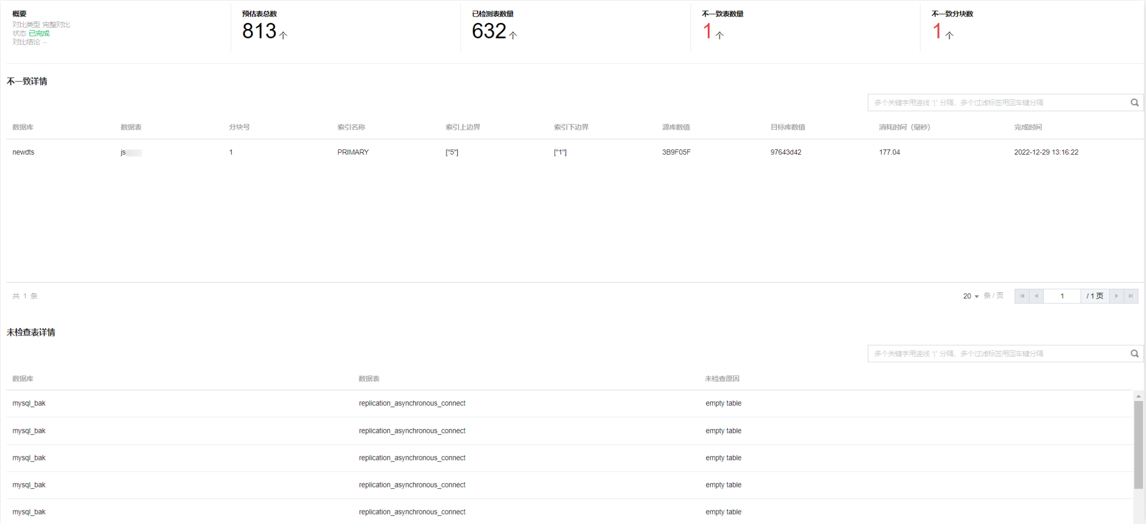
Separate keywords with "|"; press Enter to separate filter tags

Database	Data Table	Reason

No data yet

Total items: 0

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



概要
对比类型 完整对比
状态 已完成
对比结论 --

预估表总数
813 个

已检测表数量
632 个

不一致表数量
1 个

不一致分块数
1 个

不一致详情

多个关键字用竖线 "|" 分隔，多个过滤标签用回车键分隔

数据库	数据表	分块号	索引名称	索引上边界	索引下边界	源库数值	目标库数值	消耗时间（毫秒）	完成时间
newdts	js[redacted]	1	PRIMARY	["5"]	["1"]	3B9F05F	97643d42	177.04	2022-12-29 13:16:22

共 1 条

20 条 / 页 1 / 1 页

未检查表详情

多个关键字用竖线 "|" 分隔，多个过滤标签用回车键分隔

数据库	数据表	未检查原因
mysql_bak	replication_asynchronous_connect	empty table
mysql_bak	replication_asynchronous_connect	empty table
mysql_bak	replication_asynchronous_connect	empty table
mysql_bak	replication_asynchronous_connect	empty table
mysql_bak	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.

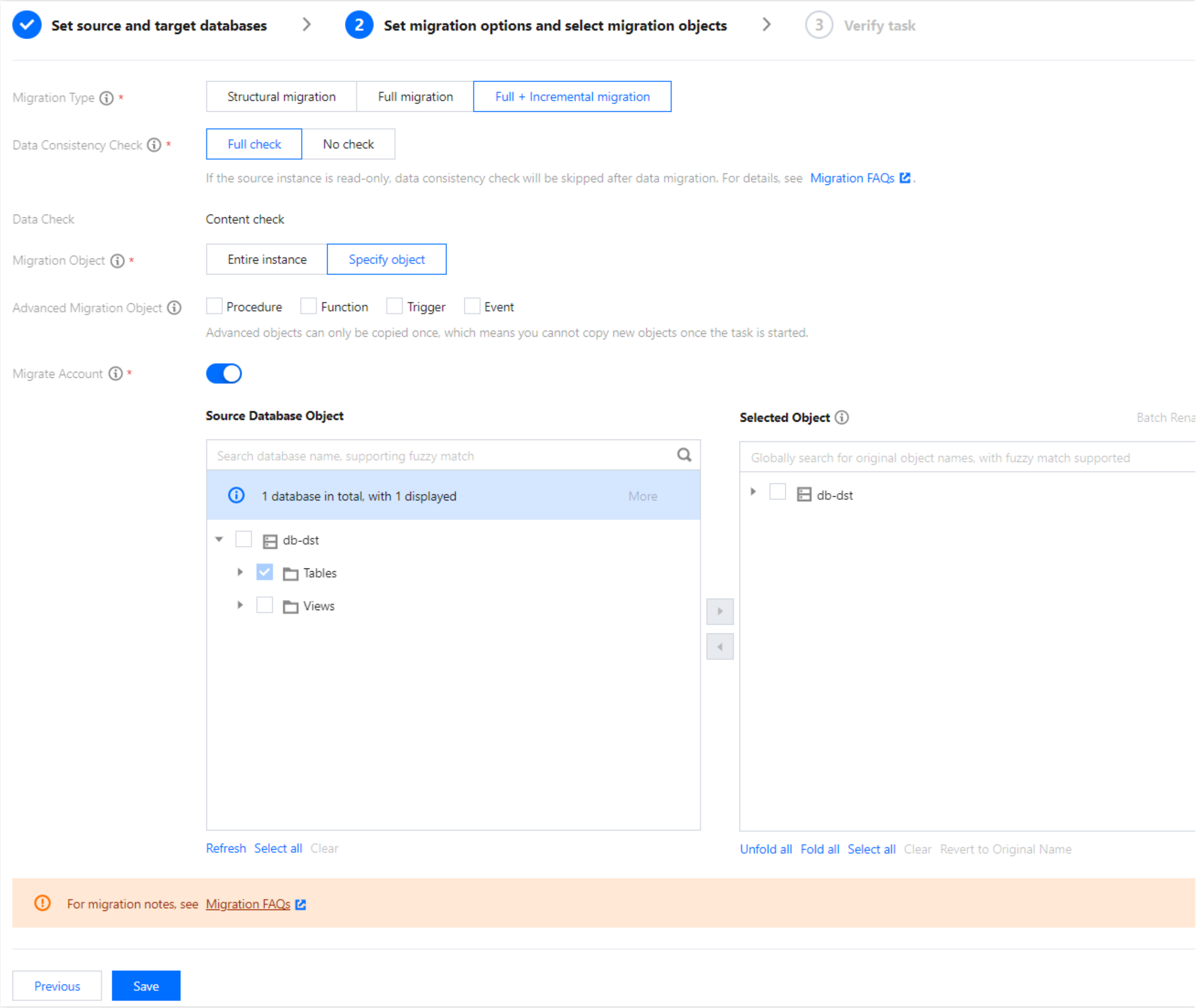
If the target database is MySQL 8.0 and has an account with stored procedures of the account to be migrated in the source database, as DTS currently cannot migrate stored procedures, an error will be reported for the `CREATE USER` statement. In this case, you need to delete the account in the target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



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 Rena

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

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

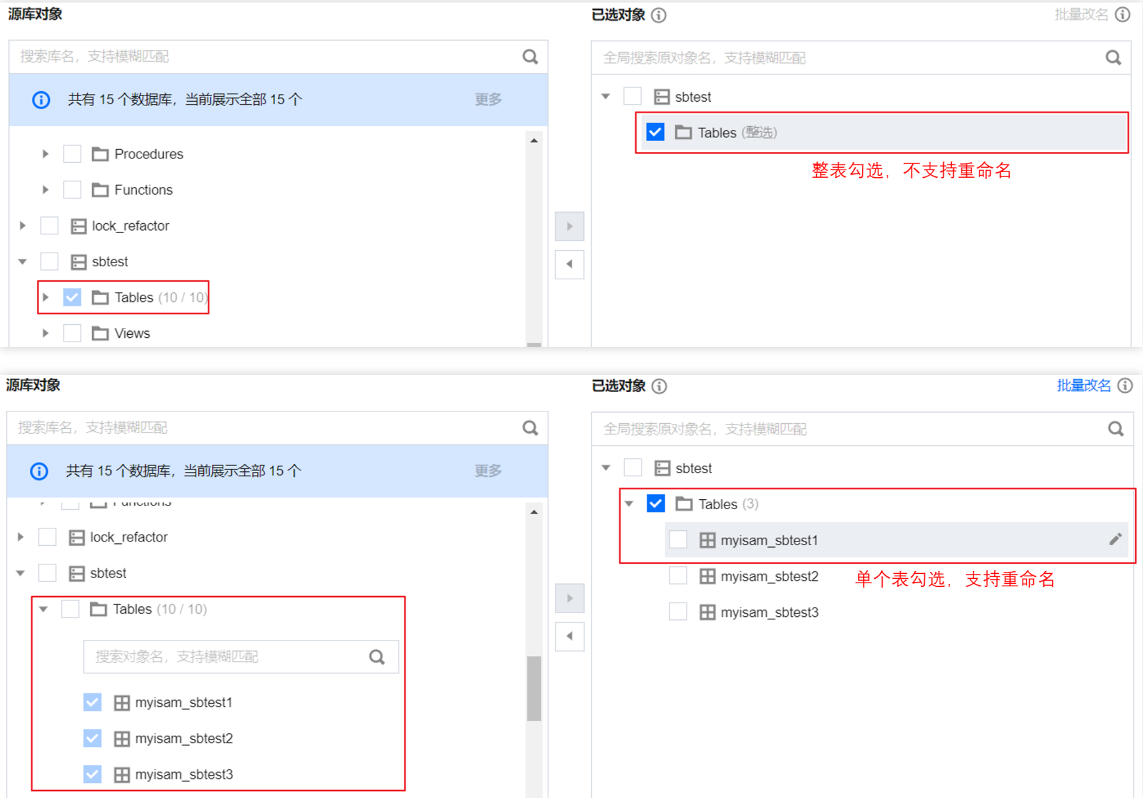
Overview

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.



整表勾选，不支持重命名

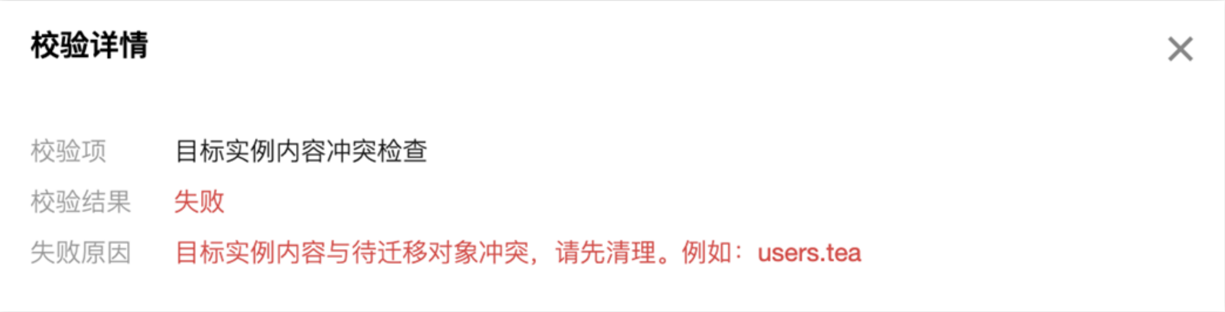
单个表勾选，支持重命名

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.



校验详情

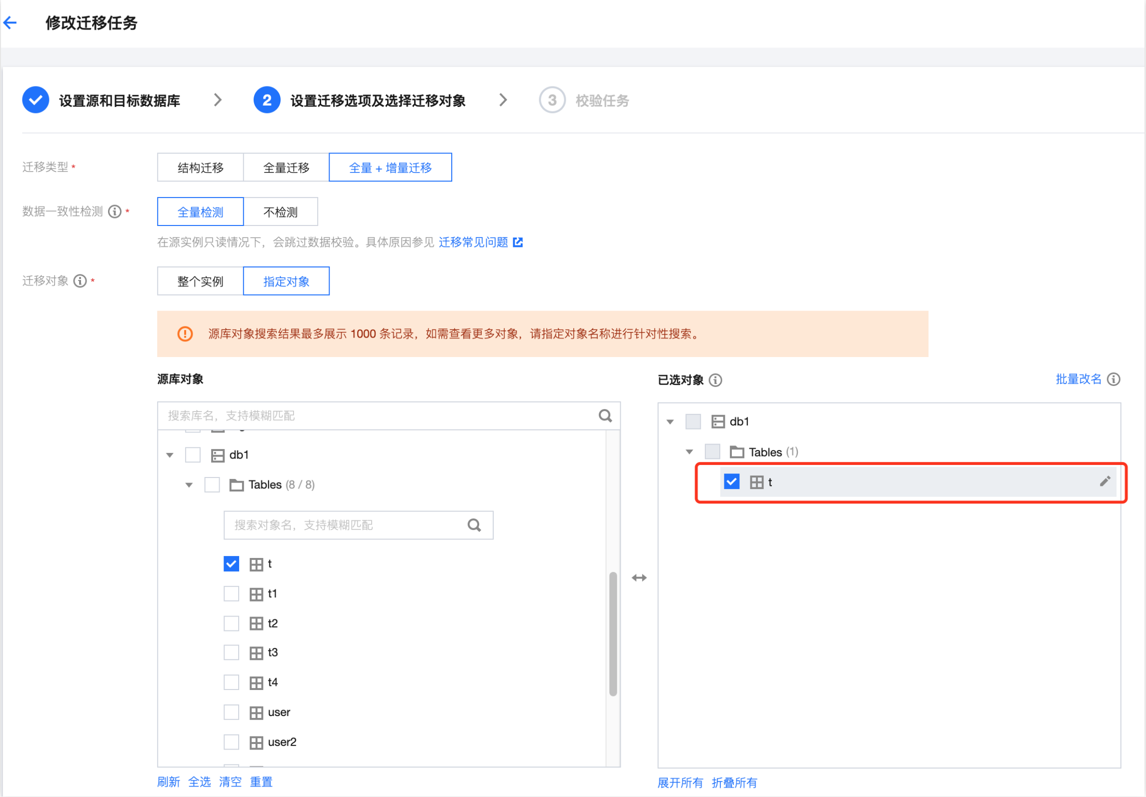
校验项 目标实例内容冲突检查

校验结果 失败

失败原因 目标实例内容与待迁移对象冲突，请先清理。例如：users.tea

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.



修改迁移任务

✓ 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 * 结构迁移 全量迁移 [✓ 全量 + 增量迁移]

数据一致性检测 * [✓ 全量检测] 不检测

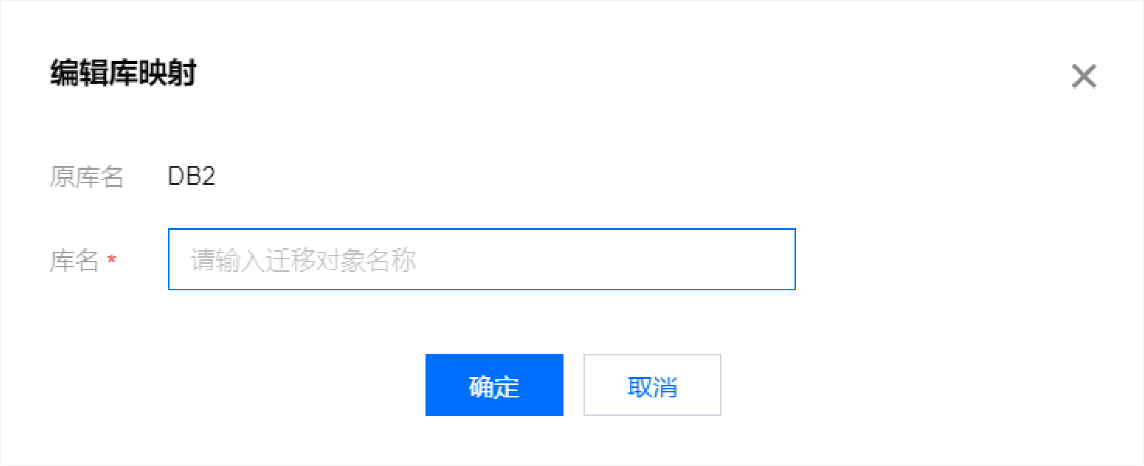
在源实例只读情况下，会跳过数据校验。具体原因参见 迁移常见问题

迁移对象 * 整个实例 [✓ 指定对象]

源库对象搜索结果最多展示 1000 条记录，如需查看更多对象，请指定对象名称进行针对性搜索。

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.

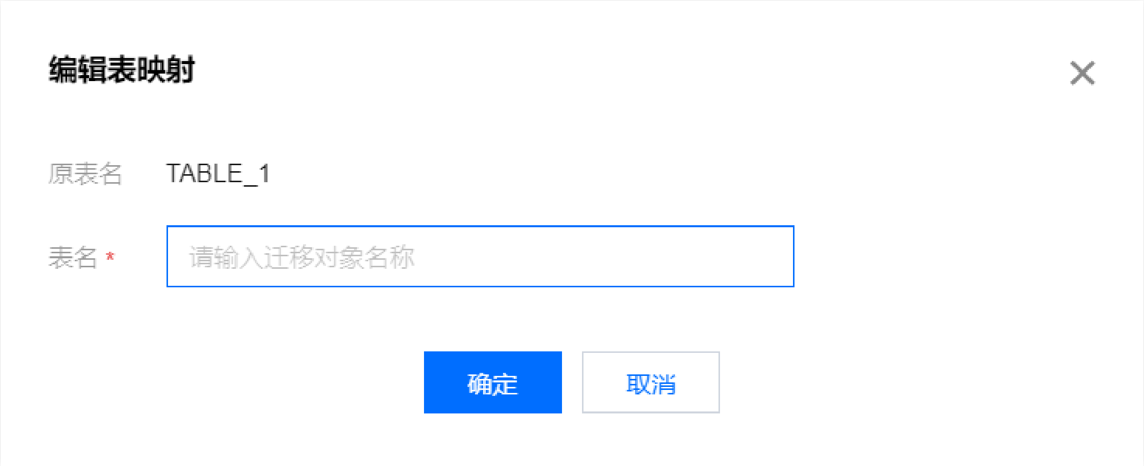


编辑库映射

原库名 DB2

库名 * 请输入迁移对象名称

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



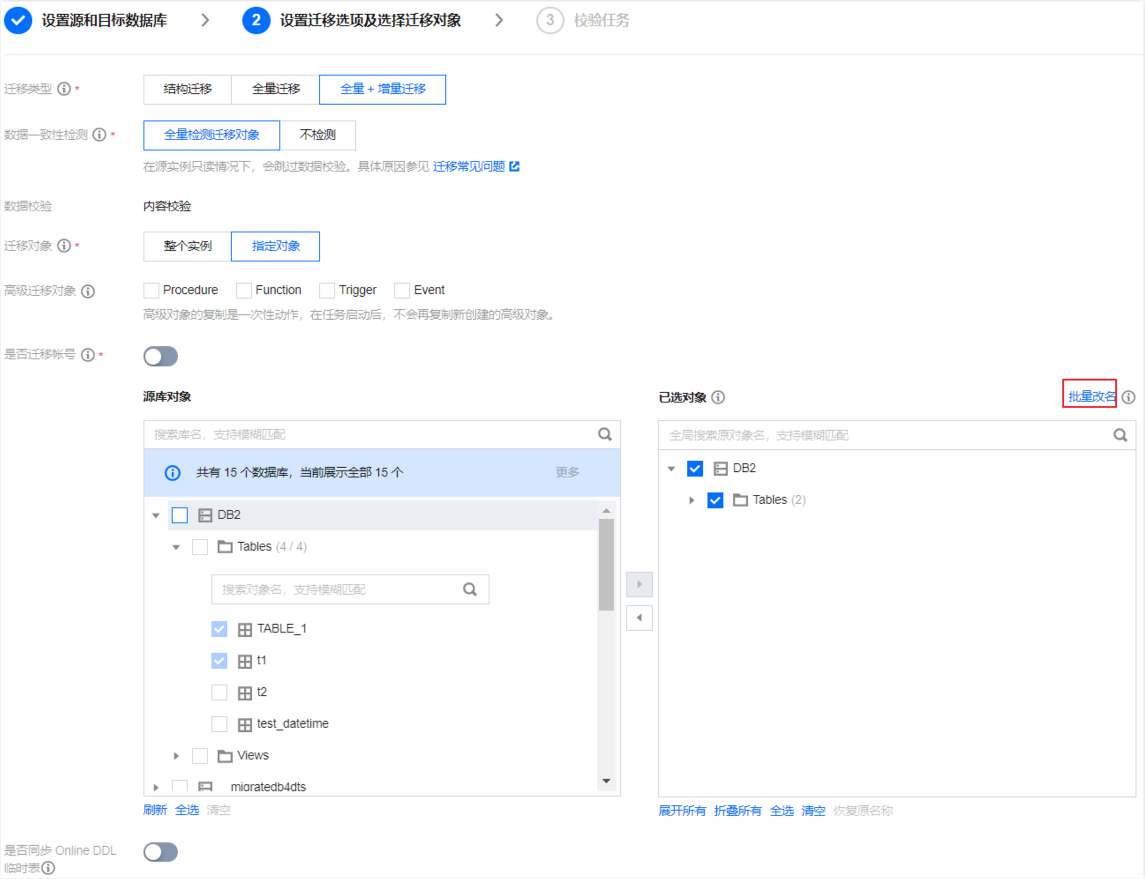
编辑表映射

原表名 TABLE_1

表名 * 请输入迁移对象名称

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.



迁移类型 *：全量 + 增量迁移

数据一致性检测 *：全量检测迁移对象

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 *：指定对象

高级迁移对象：□ 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。

共有 15 个数据库，当前展示全部 15 个

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

Renaming Scope

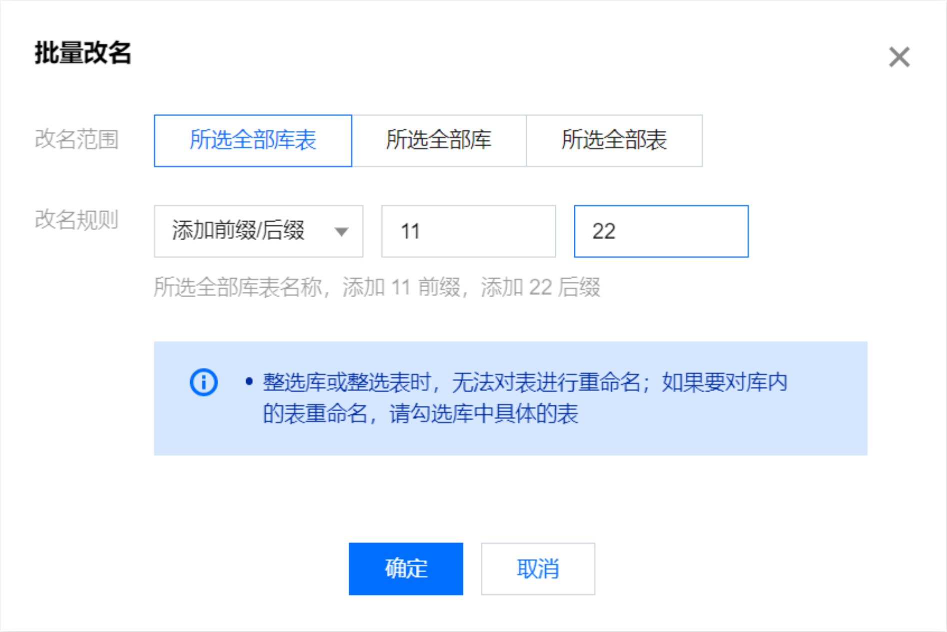
All selected databases and tables: All the selected databases and tables will be renamed in batches.

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

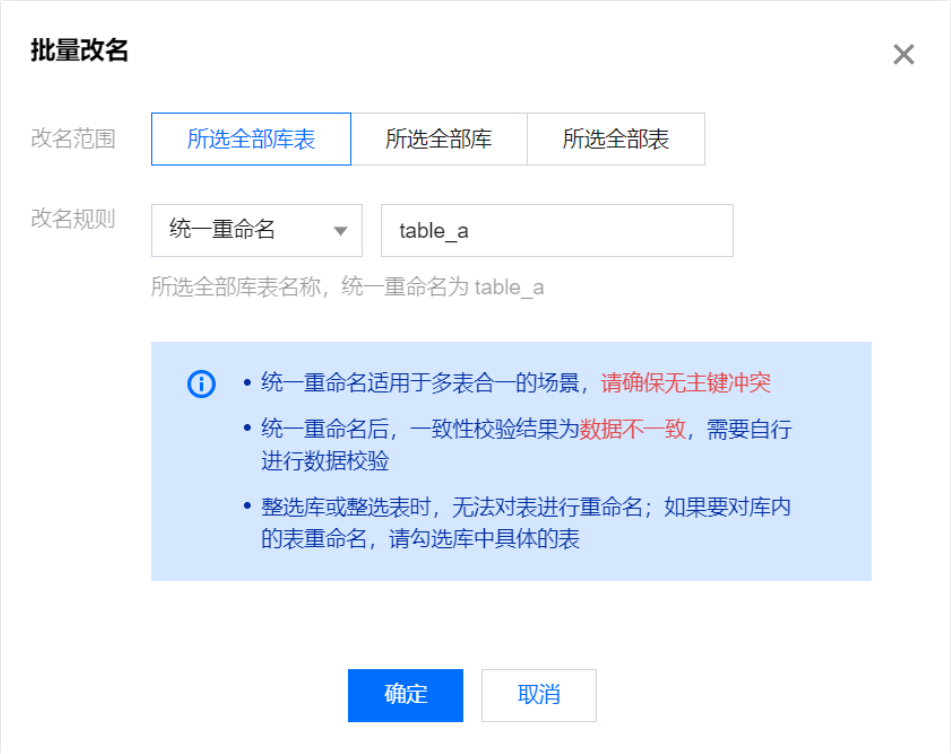
改名规则: 添加前缀/后缀 ▾ | 11 | 22

所选全部库表名称，添加 11 前缀，添加 22 后缀

- 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

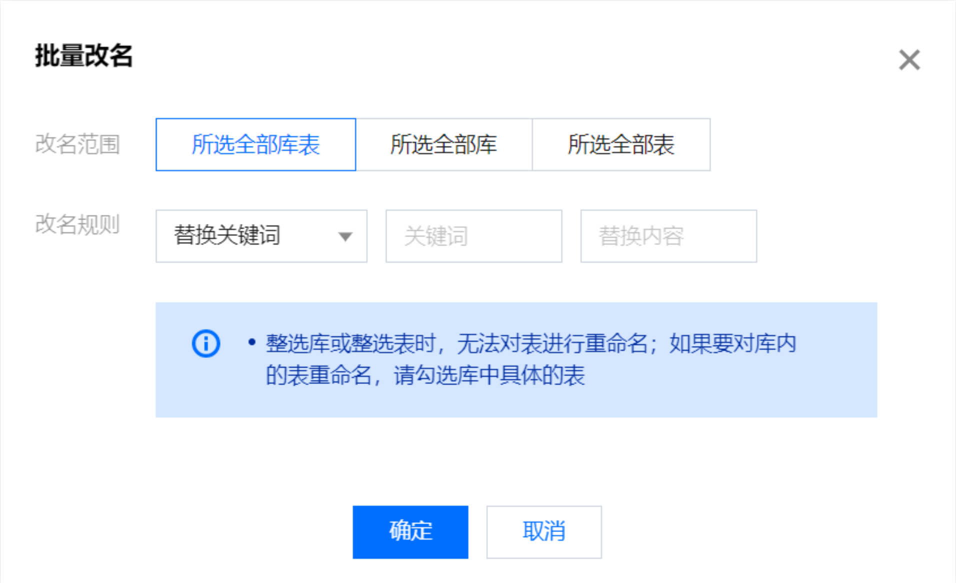
改名规则: 统一重命名 ▾ | table_a

所选全部库表名称，统一重命名为 table_a

- 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

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**

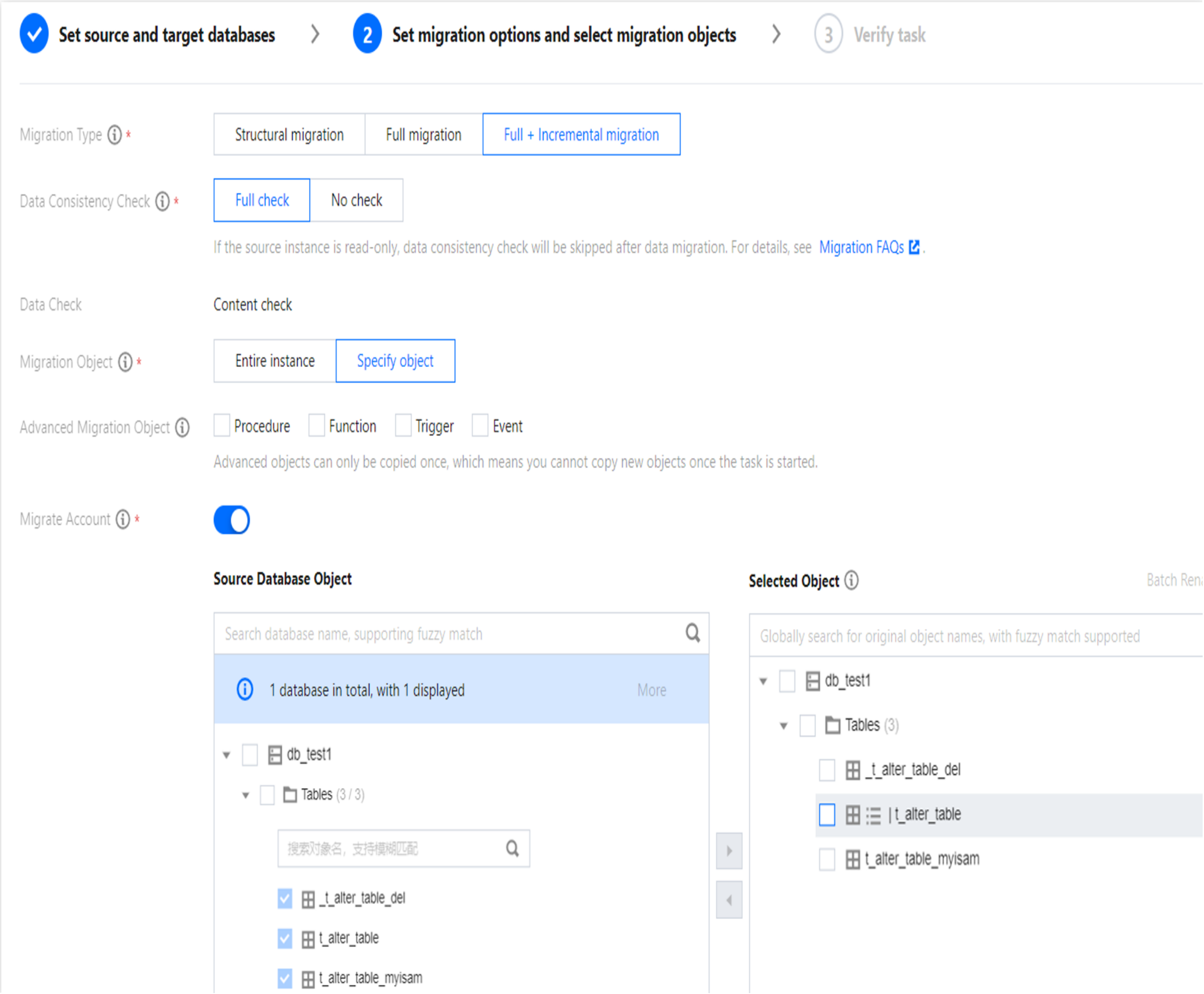
If you select **gh-ost**, DTS will migrate the temp tables (`_table name_ghc` , `_table name_gho` , and `_table name_del`) generated by the gh-ost tool to the target database.

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.

Note:

In **Selected Object** on the right, you need to selected 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.



✓ 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_test1
  - □ Tables (3 / 3)
    - 搜索对象名，支持模糊匹配
    - ☑ _t_alter_table_del
    - ☑ t_alter_table
    - ☑ t_alter_table_myisam

Selected Object ⓘ

Batch Ren

Globally search for original object names, with fuzzy match supported

- □ db_test1
  - □ Tables (3)
    - □ _t_alter_table_del
    - □ | t_alter_table
    - □ t_alter_table_myisam

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.

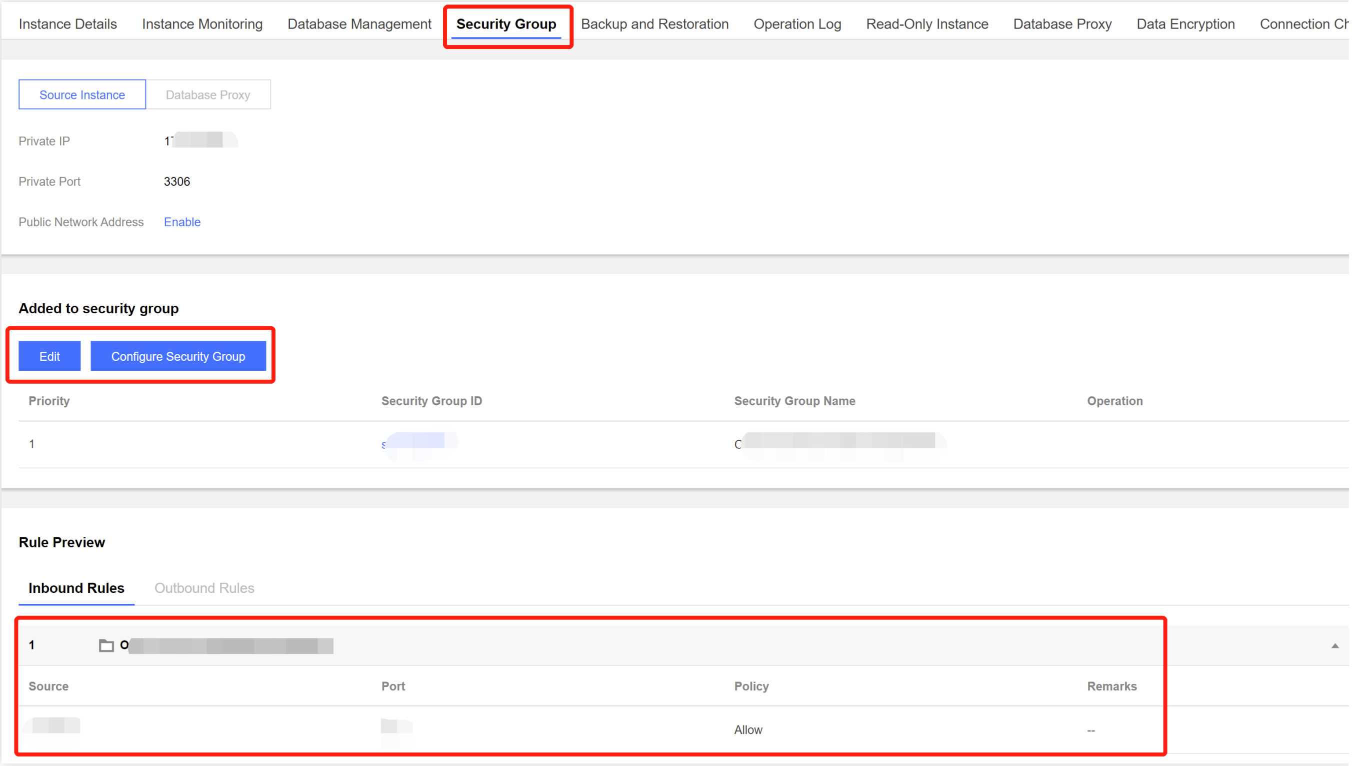
Windows: open Control Panel and find the Windows Defender Firewall and check whether firewall policies are configured.

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.



Private IP 1[redacted]

Private Port 3306

Public Network Address Enable

Added to security group

Priority	Security Group ID	Security Group Name	Operation
1	[redacted]	[redacted]	

Rule Preview

1 o[redacted]

Source	Port	Policy	Remarks
[redacted]	[redacted]	Allow	--

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 will be blocked (incorrect configuration)
root@%           // Authorize users to access all IPs, which should include the SNAT IPs (correct configuration)
```

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 entered in the data migration task
```

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 database account entered in the data migration task
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 here is only an example
binlog_row_image = FULL
expire_logs_days = 3     //Modify the binlog retention period (at least 3 days
preferably).
```

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. The value here is only an example.
```

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`).

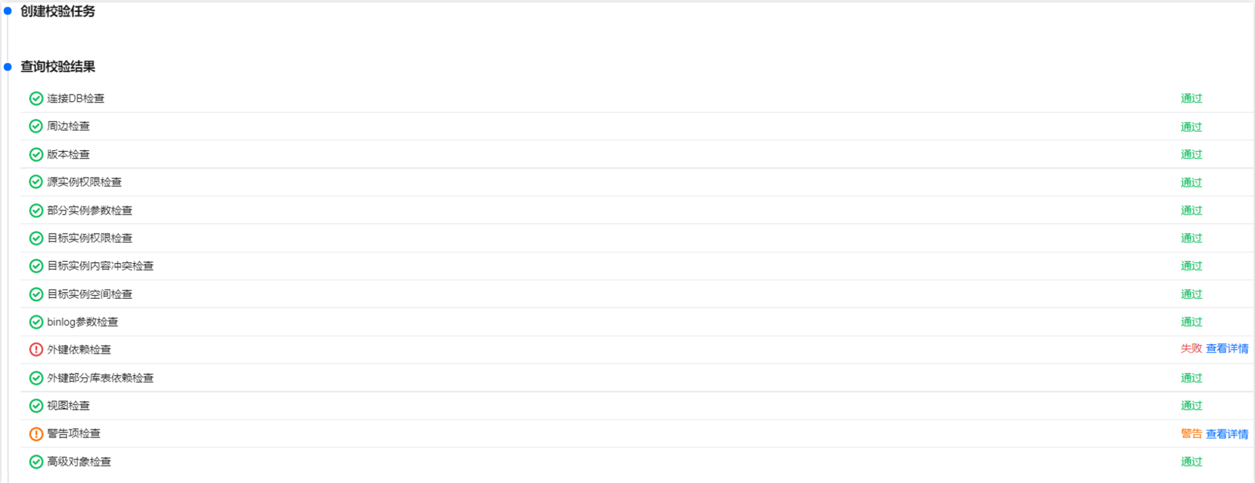
`RESTRICT` : When a record is deleted or updated in the parent table, if it is associated with records in the child table, the deletion request in the parent table will be denied.

`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.



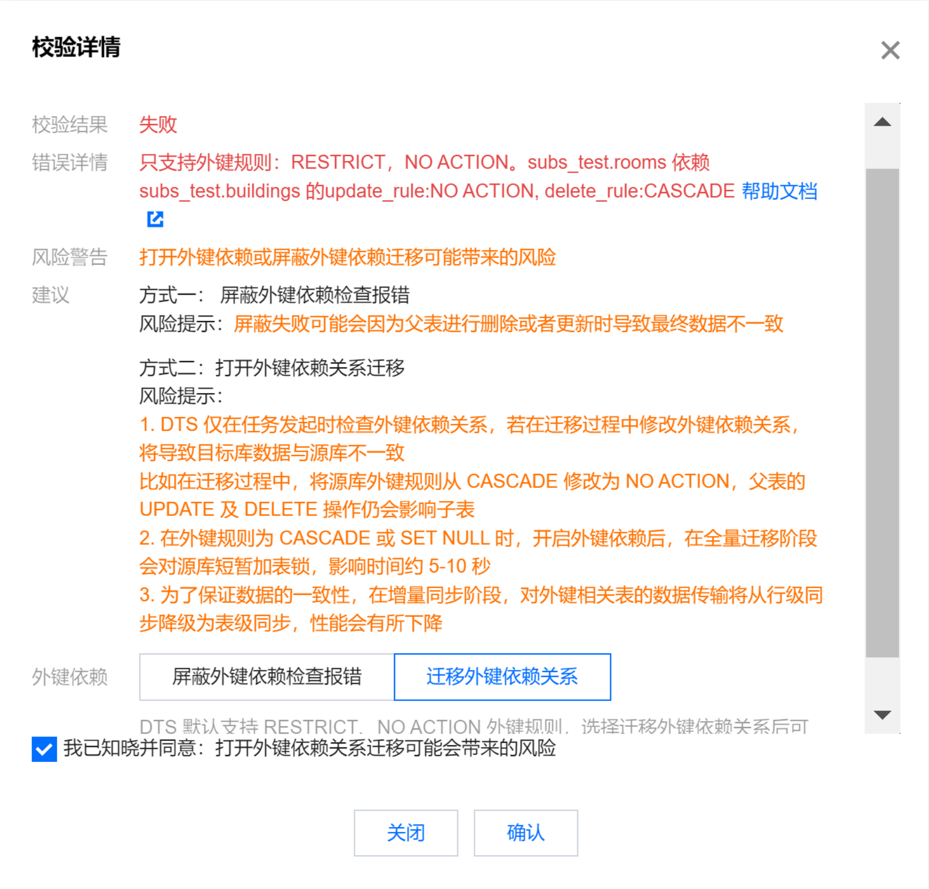
- 创建校验任务
- 查询校验结果

✓ 连接DB检查	通过
✓ 周边检查	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	After this option is selected, the system will ignore existing check errors and proceed with the migration task. If the source database has foreign key rules `CASCADE` , `SET NULL` , or `SET DEFAULT` , 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.
Migrate foreign key dependency	After this option is selected, `CASCADE` , `SET NULL` , and `SET DEFAULT` 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. 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. If you change the foreign key dependency check rule in the source database from `CASCADE` to `NO ACTION` during migration, updates and deletions performed on the parent table in the target database will still affect child tables.

	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
比如在迁移过程中，将源库外键规则从 CASCADE 修改为 NO ACTION，父表的 UPDATE 及 DELETE 操作仍会影响子表
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

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:

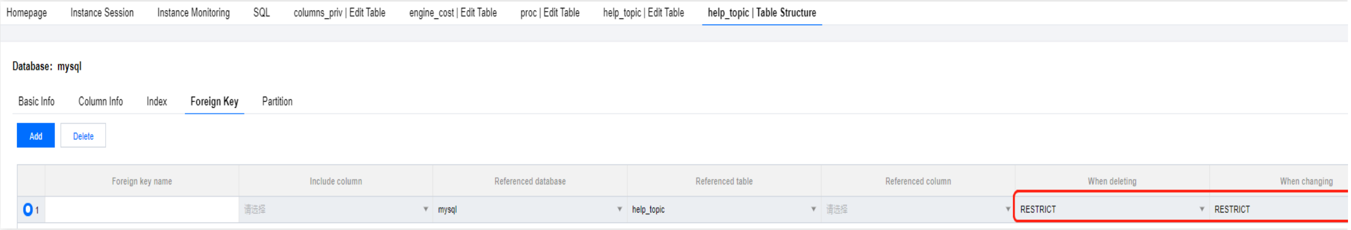
TDSQL for MySQL data migration and sync: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT` ; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task. You can fix the error as follows:

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.



Database: mysql

	Foreign key name	Include column	Referenced database	Referenced table	Referenced column	When deleting	When changing
○ 1		请选择	mysql	help_topic	请选择	RESTRICT	RESTRICT

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 name 1`(`column name 1`) references `table name 2`(`column name 1`)
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.

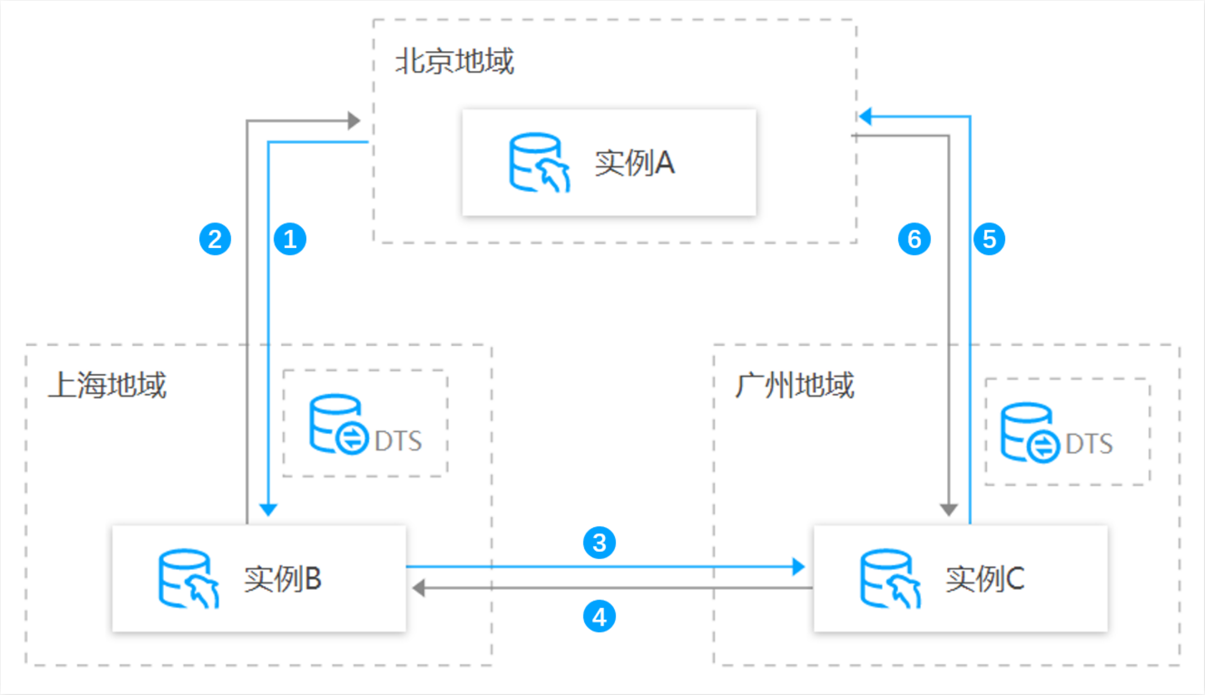
DDL Ring Linkage Synchronization Detection for Same Database and Table Objects

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.

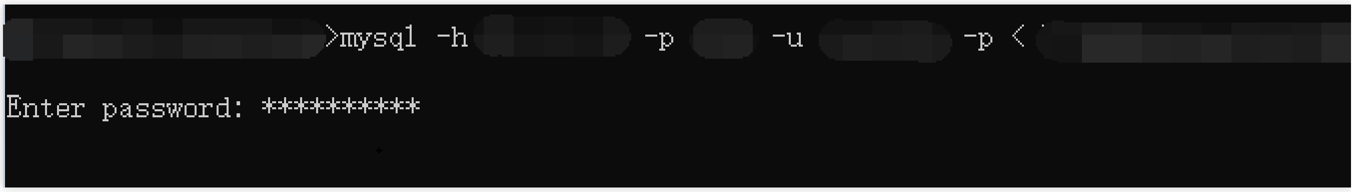
2. Import data to the target database with the MySQL command line tool as follows:

```
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.

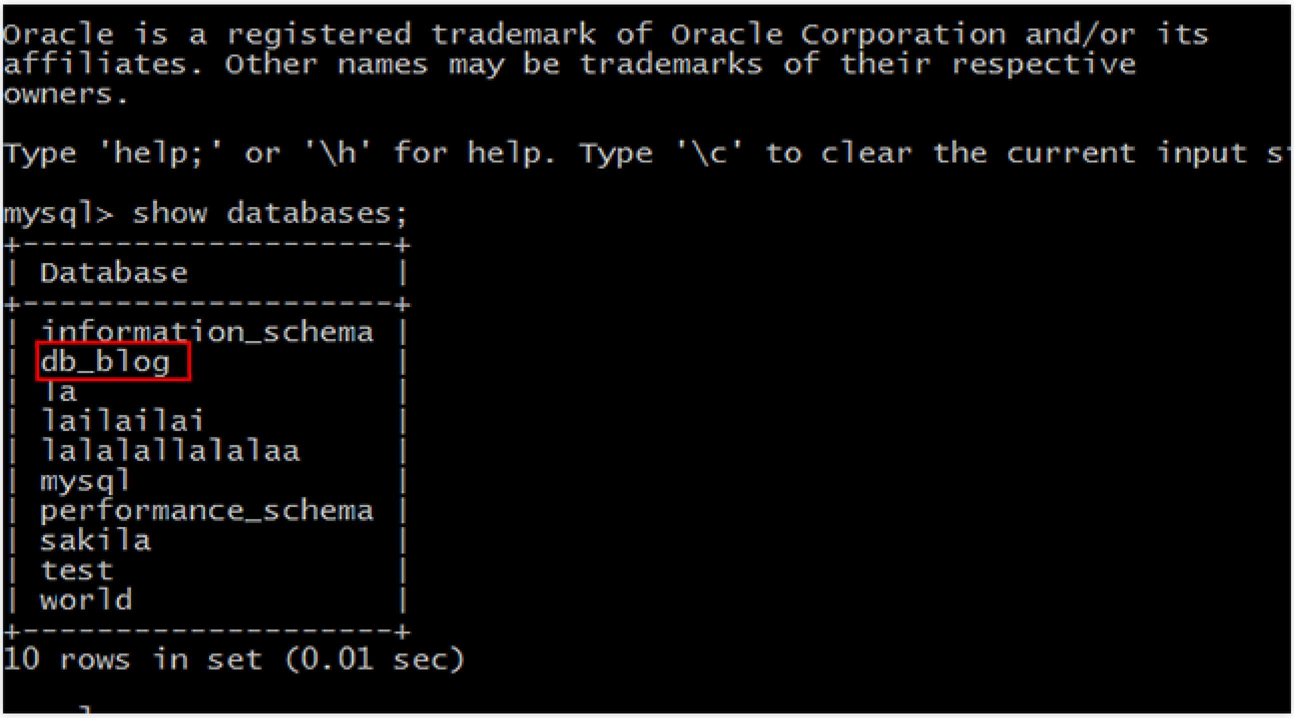
Migrating data (Windows)

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.



```
[redacted]>mysql -h [redacted] -p [redacted] -u [redacted] -p < [redacted]

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.

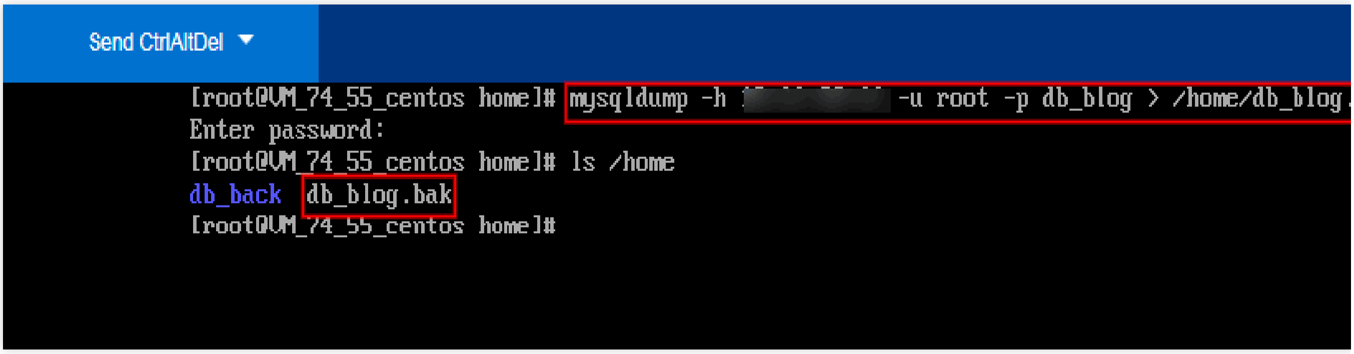


```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db_blog            |
| la                 |
| lailailai          |
| lalalallalalaa     |
| 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

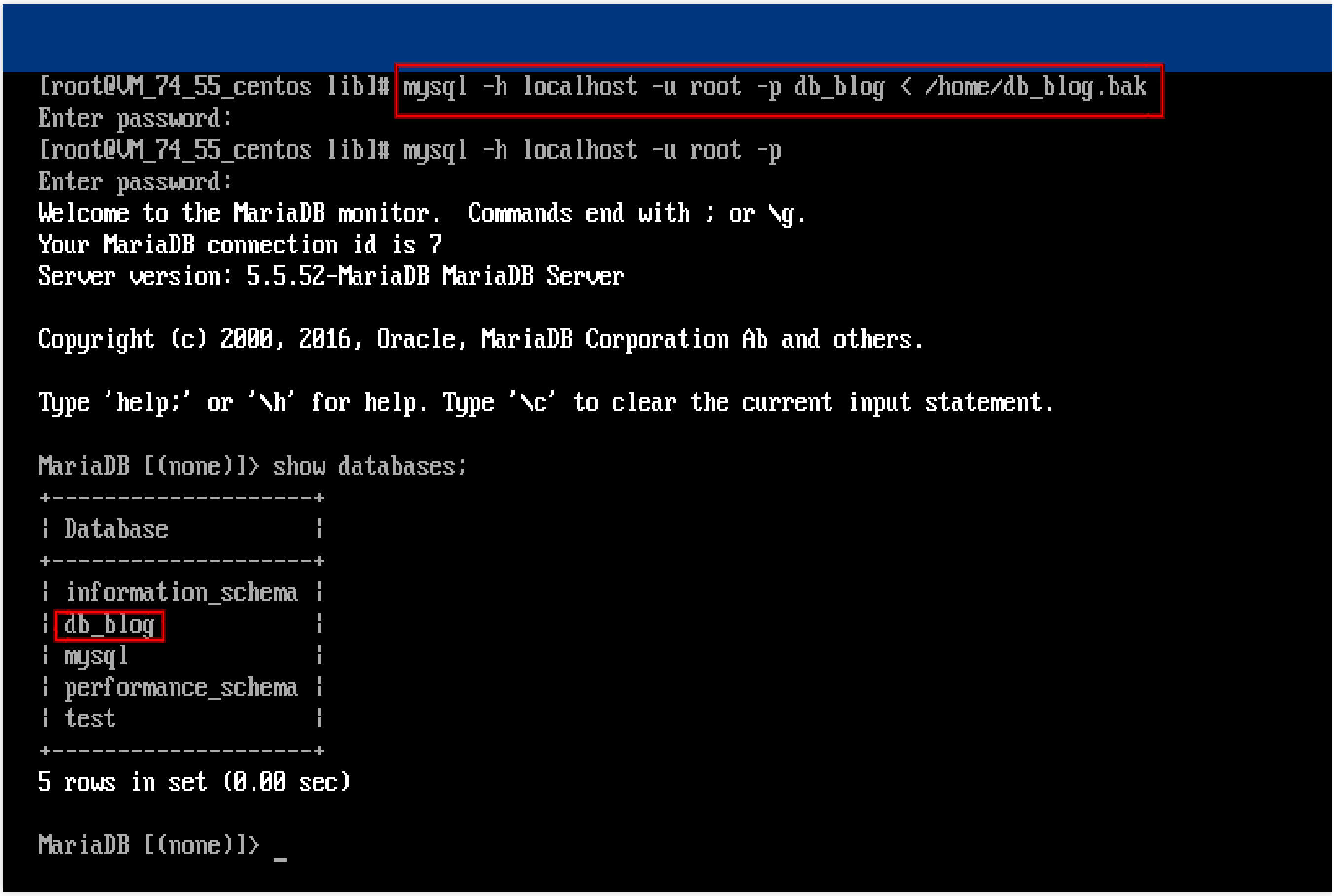
1. Log in to the CVM instance and generate the SQL file to be imported with the MySQL command line tool "mysqldump". Take the `db_blog` database in TencentDB as an example:



```
[root@VM_74_55_centos home]# mysqldump -h [redacted] -u root -p db_blog > /home/db_blog.
Enter password:
[root@VM_74_55_centos home]# ls /home
db_back  db_blog.bak
[root@VM_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@VM_74_55_centos lib]# mysql -h localhost -u root -p db_blog < /home/db_blog.bak
Enter password:
[root@VM_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	The source and target databases can be connected. The server where the source database resides must have enough outbound bandwidth; otherwise, the migration speed will be affected. Requirements for instance parameters: The `wal_level` parameter in the source database must be set to `logical` during incremental migration. The value of `max_replication_slots` in the source database must be greater than the number of databases to be migrated during incremental migration. The value of `max_wal_senders` in the source database must be greater than the number of databases to be migrated during incremental migration.
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	Select a type based on your scenario. In this scenario, select **Database**. For the preparations for different access types, see Overview. 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. 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.
	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	Select an option based on your scenario. Structural migration: Structured data such as databases and tables in the database will be migrated. 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.
Migration Object	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.

	Specified objects: Migrate specified objects.
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 modify the owner of the resources first (with a table as an
example below):
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 MongoDB

Supported Capabilities

Last updated：2024-09-10 17:26:58

Supported Scenarios and Versions

Support migration between different architectures: Replica Set > Replica Set / Sharded Cluster, Sharded Cluster > Replica Set / Sharded Cluster, Single Node > Replica Set / Sharded Cluster.

Source Database	Target Database	Scene Description
Self-hosted MongoDB database (Self-hosted on IDC, Self-hosted on CVM) 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0	TencentDB for MongoDB 4.0, 4.2, 4.4, 5.0, 6.0	Migrate on-premises databases to the cloud.
Third-Party cloud provider MongoDB 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0	TencentDB for MongoDB 4.0, 4.2, 4.4, 5.0, 6.0	Migrate databases from other cloud providers to TencentDB for MongoDB.
TencentDB for MongoDB 3.2, 3.6, 4.0, 4.2, 4.4, 5.0, 6.0	TencentDB for MongoDB 4.0, 4.2, 4.4, 5.0, 6.0	Migration between Tencent Cloud regions or across different regions Database migration within the same Tencent Cloud root account or between different root accounts Cross-version migration between TencentDB for MongoDB instances Migration between Replica Set and Sharded Cluster within TencentDB for MongoDB instances

Supported Features

Feature Category	Feature Sub-Item or Description	Supported Capabilities

Migration Object	-	Databases, collections
Migration Type	-	Full migration Full + Incremental migration
Key Task Management Operations	Retry	Supported
	Create similar task	Supported
Incremental Synchronization	DML synchronization (INSERT / UPDATE / ELETE)	Supported
	DDL synchronization	INDEX: createIndexes, createIndex, dropIndex, and dropIndexes COLLECTION: createCollection, drop, collMod, renameCollection, and convertToCapped DATABASE: dropDatabase and copyDatabase Support DDL operations for both Replica Set and Sharded Cluster.
Consistency Check	Validation object	Entire migration object / Custom object
	Validation method	Row count comparison / Content validation / Sampling comparison

User Guide

Last updated：2024-09-10 17:28:19

Impact on the Source Database

When DTS performs full data synchronization, it consumes certain resources from the source database, which may lead to an increase in the source database's load and put additional pressure on it. If your database configuration is insufficient, it is recommended to perform data migration during off-peak hours.

Impact on the Target Database

During the migration process, DTS will use a system service account to create a table under the TencentDTSData database on the target end, using the task ID (e.g., table name dts-xxxxx). This table is used to record checkpoints, enabling checkpoint restart in case of task interruption.

Migration Architecture

1. Sharded Migration Related Notes:
1.1 Before a sharded cluster is migrated, it is recommended to clean up orphaned documents in the source cluster in advance. Otherwise, data inconsistency issues may arise after migration. For instructions on how to clean up orphaned documents, see the MongoDB official documentation cleanupOrphaned.
1.2 During a sharded migration, do not enable sharding on the source databases and tables being migrated. This is to prevent discrepancies in data distribution between the source and target databases. If sharding is enabled on the source during migration, check the sharding status on the target. If sharding is not enabled on the target, you will need to manually enable it. For detailed instructions on how to enable sharding, see the MongoDB official documentation on Shard a Collection.
1.3 If the source is a TencentDB for MongoDB 3.2 sharded cluster, all shard keys will be treated as hashed shard keys by default during migration. If you want to use range shard keys on the target, create the range shard keys on the target database in advance before the data migration is started.
2. Since a single node does not have an Oplog, incremental migration is not supported when the self-hosted instance is a single node.

Notes

1. Do not perform the following operations during migration, as they may cause the migration task to fail.

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

Do not perform Oplog clearance operations on the source database.

During the data migration, do not delete the TencentDTSData database on the target database.

2. Be cautious when you operate on the target database during the data migration to avoid data inconsistencies.

Migration Operation Guide

Last updated：2024-10-08 14:18:59

Overview

DTS-based MongoDB data migration supports both full and incremental data migration, meaning that the historical data from the source database as well as the newly written data during the migration process can be migrated together.

This document provides instructions on how to use the DTS data migration feature to migrate data from MongoDB to TencentDB for MongoDB.

Preparation

1. Ensure that the access channel between DTS and the database is established in advance according to the type of access you plan to use. For detailed instructions, see Network Preparation Work.

IDC self-hosted databases/other cloud provider databases: The available access methods include Public Network/Direct Connect/VPN Access/CCN.

Self-built databases on CVM: The access method should be Self-Build on CVM.

TencentDB instances: The access method should be Database.

2. It is recommended to create a read-only account in the source database for migration purposes. See the following method.

```
use admin
db.createUser({user: "username",pwd: "password",roles:[{role:
"readAnyDatabase", db: "admin"},{role: "read", db: "local"}]})
```

3. If the target database is a TencentDB instance, you can use the mongouser account for migration, or you can create your own account. The method for creating an account is as follows.

```
db.createUser({user:"username",pwd:"password",roles:
[{role:"readWriteAnyDatabase",db:"admin"}]})
```

Operation Steps

1. Log in to the DTS console, choose the **Data Migration** page in the left sidebar, and click **Create Migration Task** to enter the Create Migration Task page.

2. On the Create Migration Task page, select the source instance type and region, the target instance type and region, and the specifications. Then, click **Purchase Now**.

Configuration Parameter	Description
Creation Mode	Create new task: Create a brand new task. Create similar task: Quickly create a task with the same configuration as a previous task. In the new task, the options for database type, access method, billing mode, and migration type are pre-filled to match the previous task. Users can modify these options as needed based on the current situation.
Source Instance Type	Select based on the type of your source database. This cannot be changed after purchase. For this scene, select MongoDB.
Source Instance Region	Select the region where the source database is located. If the source is a self-hosted database, select the region closest to the self-hosted database.
Target Instance Type	Select based on the type of your target database. This cannot be changed after purchase. For this scene, select MongoDB.
Target Instance Type	Select the region where the target database is located.
Version	The default is NewDTS; no modification is needed.
Specification	Currently, only the Medium specification is supported.
Task Name	Select **Name After Creation**. The task name will be the same as the task ID by default. After the migration task is created, you can rename the task. Select **Name Now**, and enter the task name in the input field below.

3. After the purchase is completed, the page will automatically redirect to the data migration task list. Select the task you just purchased to proceed with the configuration.

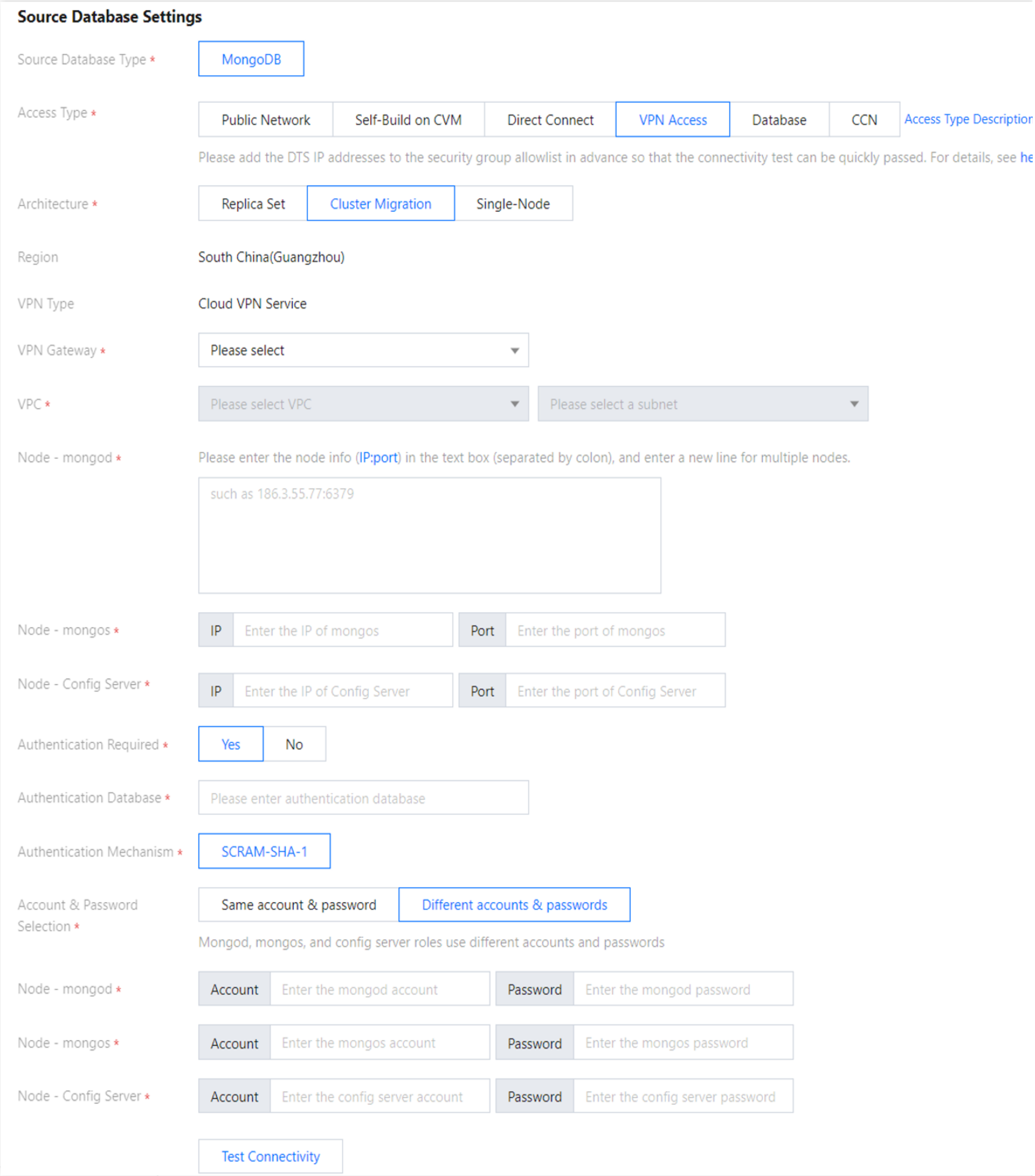
If you have purchased multiple regions or are configuring cross-region tasks, the task list will display tasks based on the region of the target instance. You can switch regions at the top of the page to find the tasks you have purchased.

4. On the set source and target database page, complete the task settings, source database settings, and target database settings.

Note:

Enter the read-only account that was previously created for the source database; otherwise, the pre-check step will fail.

If the source or target database is a TencentDB instance, DTS will use a system service account to export or write data during the migration process. For example, if the source database is a TencentDB instance, DTS will connect to the source database using the read-only account provided by the user, and also use a system service account to export data from the source. Similarly, if the target database is a TencentDB instance, DTS will connect to the target database using the account provided by the user, and use a system service account to write data to the target.



Source Database Settings

Source Database Type * MongoDB

Access Type * Public Network | Self-Build on CVM | Direct Connect | [✓ VPN Access] | Database | CCN Access Type Descriptior

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 he

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 for multiple nodes.

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

Configuration	Description

Parameter	
Task Name	Set a business-relevant name for easy task identification.
Run Mode	Immediate execution: The task will start immediately after the pre-check passes. Scheduled execution: Set a specific start time for the task. The task will not start immediately after the pre-check passes, but will instead start at the scheduled time.
Access Type	Select based on your specific scene. For preparation steps related to different access types, see the Overview. Public Network: The source database can be accessed via a public IP. Self-Build on CVM: The source database is deployed on a Tencent Cloud Service CVM. Direct Connect: The source database can connect to Tencent Cloud Virtual Private Cloud via Direct Connect. VPN Access: The source database can connect to Tencent Cloud Virtual Private Cloud via VPN Connections. Database: The source database is a TencentDB instance. CCN: The source database can connect to Tencent Cloud Virtual Private Cloud via Cloud Connect Network. If the source database is a self-hosted IDC database or a database from another cloud provider, you can choose Public Network/Direct Connect/VPN Access/CCN as the access method. If the source database is self-hosted on a CVM, select Self-Hosted on CVM as the access method. If the source database is a TencentDB instance, select Database as the access method.
Architecture	This parameter is displayed when **Access Type** is set to Self-Build on CVM/Direct Connect/VPN Access/CCN. Select it according to your actual situation. Replica set: Refers to an architecture where the source database is a replica set, consisting of one Primary node and one or more Secondary nodes. Cluster migration: Refers to an architecture where the source database is a sharded cluster, consisting of components such as mongos nodes, config servers, and shard nodes. Single node: Refers to an architecture where the source database cluster consists of a single node that handles both read and write operations. **Note:** Once you select an architecture type, and the connectivity test passes. It cannot be changed to another architecture type; otherwise, the task will encounter errors.
Cluster Migration	When **Architecture** is set to Cluster migration, the following parameters need to be configured. Node - mongod: Enter the IP and port, or the domain name and port, of the mongod node. For multiple nodes, enter each on a new line; only one mongod needs to be entered for each shard. Example: 1xx.xx.55.77:6xx9. Node - mongos: Enter the IP and port, or the domain name and port, of the mongos node. Node - Config Server: Enter the IP and port, or the domain name and port, of the Config Server node.

Public Network	When **Access Type** is set to Public Network, the following parameters need to be configured. Host address: The IP address or domain name of the source database. Port: The port used by the source database.
Self-Build on CVM	When **Access Type** is set to Self-Build on CVM", the following parameters need to be configured. CVM instance: The instance ID of the CVM. Port: The port used by the source database.
Direct Connect	When **Access Type** is set to Direct Connect, the following parameters need to be configured. VPC Direct Connect Gateway: Only the VPC Direct Connect Gateway is supported for Direct Connect. Confirm the network type associated with the gateway. VPC: Select the VPC and subnet. Host address: The IP address or the domain name of the source database. Port: The port used by the source database.
VPN Access	When **Access Type** is set to VPN Access, the following parameters need to be configured. VPN gateway: Select the VPN gateway instance that is accessed through the VPN gateway. VPC: Select the VPC and subnet. Host address: The IP address or the domain name of the source database. Port: The port used by the source database.
Database	When **Access Type** is set to Database, the following parameters need to be configured. Cloud database instance: Select the instance ID of the source database.
CCN	When **Access Type** is set to CCN, the following parameters need to be configured. When using CCN for access, both same-account CCN and cross-account CCN are supported. Due to the complexity of network configuration, see Migrating Self-built Databases to Tencent Cloud via Cloud Connect Network. Host network environment: Select according to your actual situation. For example, if the source database is a TencentDB instance, select Tencent Cloud; if the source database is a self-hosted IDC database, select Self-built IDC; if the source database is from another cloud provider, select the corresponding network. Host address: The host IP address or the domain name of the source database. Port: The port used by the source database. CCN instance account My account: The CCN resources and DTS belong to the same Tencent Cloud root account. Other account: The CCN resources and DTS belong to different Tencent Cloud root accounts. VPC CCN: The name of the CCN instance.

	Access VPC: Select the VPC and subnet to be accessed. The Access VPC refers to the VPC that is connected to the DTS migration linkage within the CCN. Choose a VPC other than the one that the source database belongs to from all the VPCs associated with the CCN. Access VPC region: The region of the Access VPC should match the region of the source database selected when purchasing the task. If they do not match, DTS will change the region of the source database selected in the purchased task to the Access VPC region.
Authentication Required	Whether security authentication for the username and password of the source database is required. If Required is selected, the following parameters need to be filled in. Authentication database: The name of the database that requires authentication, i.e., the database name associated with the account executing the migration task. Only admin is supported. Authentication mechanism: Currently, only SCRAM-SHA-1 is supported. Account and password options Same account and password: Select this parameter if the same account and password are used for the roles of mongod, mongos, and Config Server, and enter the unified account and password. Different accounts and passwords: Select this parameter if different accounts and passwords are used for the roles of mongod, mongos, and Config Server, and enter the account and password for each mongod, mongos, and Config Server separately.
Account/Password	Account/Password: The account and password of the source database.

5. Test the connectivity between the source and target instances.

If the connectivity test fails, see Connectivity Test Failure for troubleshooting.

6. On the Set migration options and select migration objects page, configure the migration options and select the migration objects.

Configuration Parameter	Description
Migration Type	Select based on your scene. Full migration: Migrates the entire database, including only the data that exists in the source database at the time the task is initiated, and not including any new data written to the source database after the task starts. Full + Incremental migration: Migrates both the existing data in the source database at the time the task is initiated and any new data written to the source database in real-time after the task starts. If data is being written to the source database during the migration and you need a seamless migration without downtime, select this option.
Data Consistency Check	Data consistency check is supported only when the **Migration Type** is configured as Full + Incremental Migration, allowing a detailed comparison of data between the source and target databases after migration. When the **Migration Type** is configured as Full Migration, consistency check is disabled by default.

	Full object consistency check: After the incremental synchronization is completed, DTS will automatically trigger a consistency check task. No check: Data consistency will not be checked. If needed, you can manually trigger a consistency check after the incremental synchronization is completed. For details, see Creating Data Consistency Check Task.
Data Check	When **Data Consistency Check** is set to Full migration object detection, the default consistency check type displayed is Content Validation.
Migration Objects	Entire instance: Migrates the entire instance, excluding system databases, such as system objects in PostgreSQL. However, it will migrate roles and user metadata definitions. Specified objects: Migrates only the specified objects.
Specified Objects	Select the objects to be migrated from the source database and move them to the Selected Object box.

7. On the validation task page, complete the pre-migration validation, and then click **Start Task**.
If the validation task fails, you can see Handling Methods for Validation Failure to resolve the issue and then reinitiate the validation task.
Failed: Indicates that the validation item did not pass the check, blocking the task. The issue should be fixed before re-running the validation task.
Warning: Indicates that the validation item does not fully meet the requirements. You can continue the task, but it may have some impact on the business. The user needs to evaluate whether to ignore the warning or fix the issue before proceeding, based on the provided information.
8. Return to the migration task list, where the synchronization task will be running.
If you need to view task details, delete tasks, or perform other actions, click the corresponding task operation. For more information, see Task Management. If a task error occurs, see Error Handling.
9. End the task.
Select **Full migration**: The task will automatically end upon completion, with no need for manual intervention.
Select **Full + Incremental migration**: After the full migration is completed, the task will enter the incremental data synchronization phase. Incremental data synchronization will not end automatically and requires the user to manually end the task.
Once incremental synchronization is completed (i.e., the status is Ready to Complete) and the time delay between the target and source databases is 0 seconds, click **Complete** in the **Operation** column to end the migration task.
10. (Optional) If a cutover is required, you can perform the official cutover of your business once the task status changes to **Task successful** after the task is completed. For more details, see Cutover Instructions.

Fix for Verification 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, {numInitialChunks: number of preset chunks})
```

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

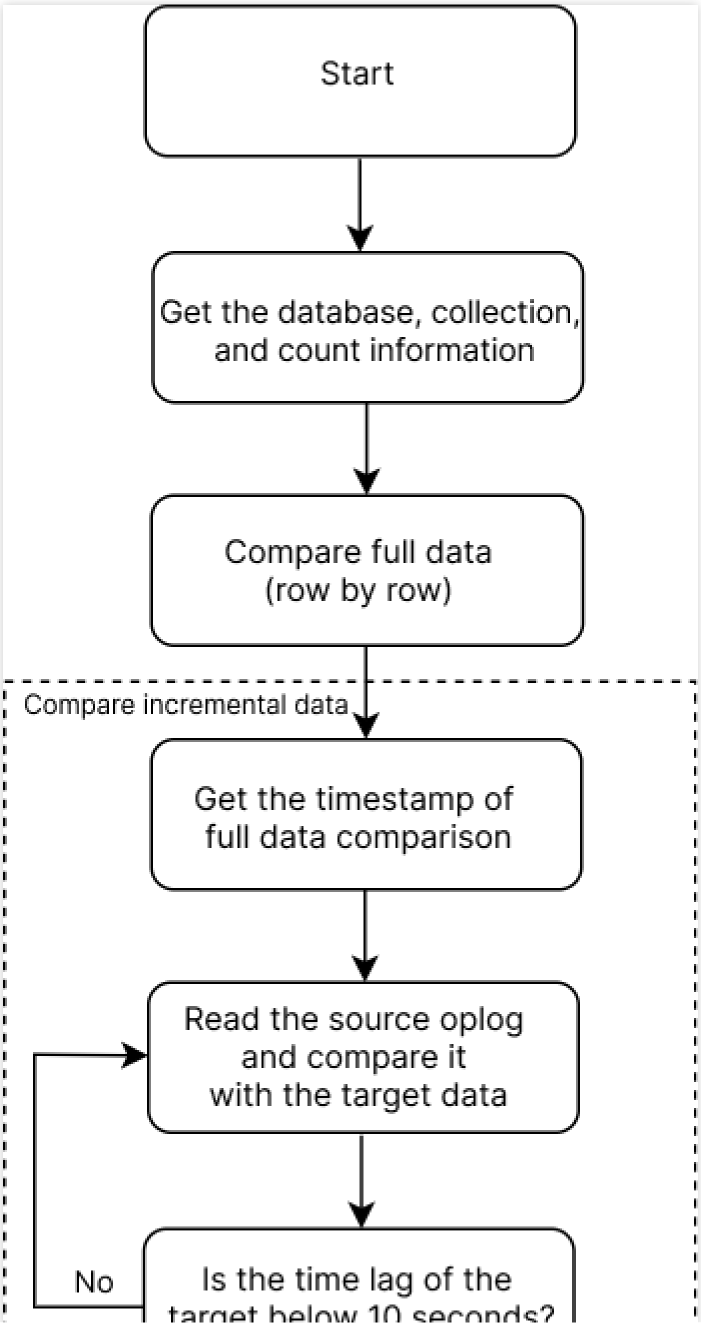
1. Data consistency check compares only the objects selected in the source database and objects migrated to the target database. If you write data into the target database during migration, then the written data will not be included in the consistency check.
2. A data consistency check task may increase 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_5xxxxxxxxx4231: Saves inconsistent metadata detected
result_5xxxxxxxxx4231: Records the results after phase validation
status_5xxxxxxxxx4231: Records validation progress

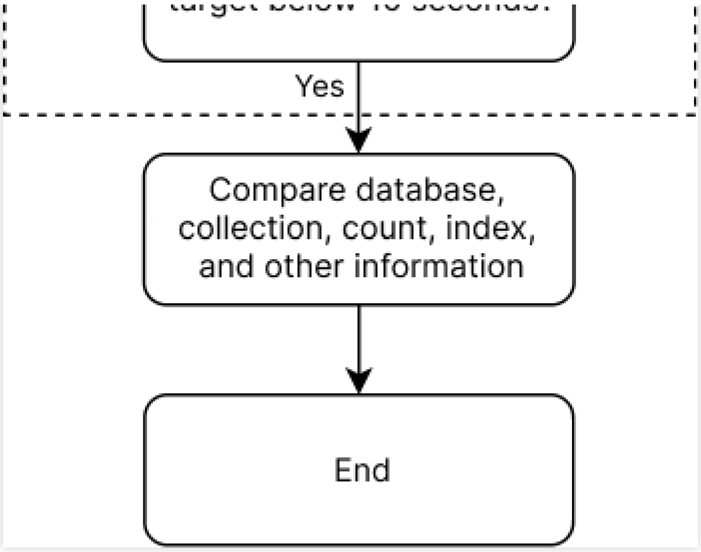
Restrictions

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.


Start
Get the database, collection, and count information
Compare full data (row by row)
Compare incremental data
Get the timestamp of full data comparison
Read the source oplog and compare it with the target data
No
Is the time lag of the target below 10 seconds?



Yes
Compare database, collection, count, index, and other information
End

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:

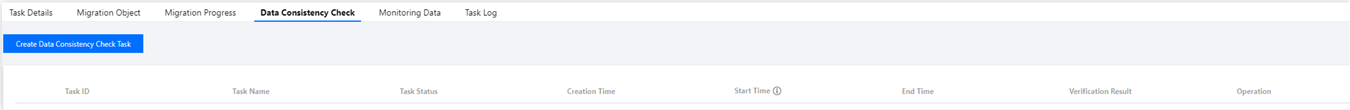
In this case, the full data and all the database information will be checked by default. If you need to filter check objects, create a data consistency check task manually.

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**.

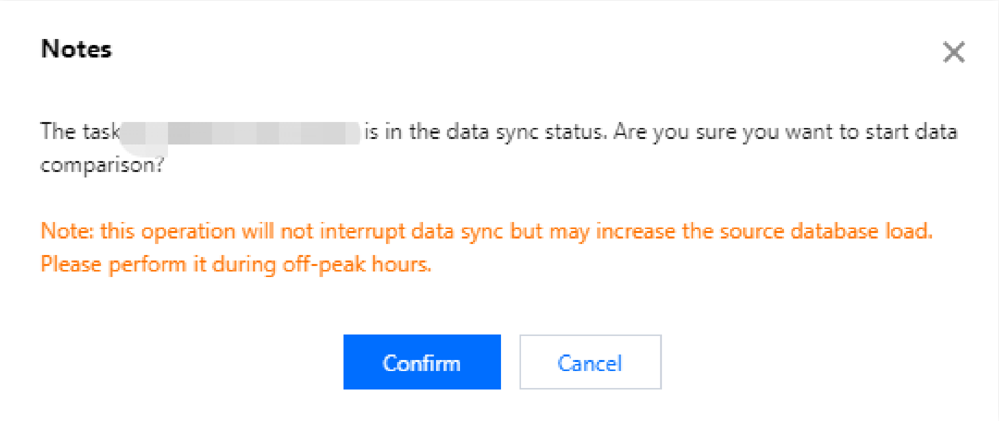
Note:

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.



Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	End Time	Verification Result	Operation

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



Notes

The task [redacted] is in the data sync status. Are you sure you want to start data comparison?

Note: this operation will not interrupt data sync but may increase the source database load. Please perform it during off-peak hours.

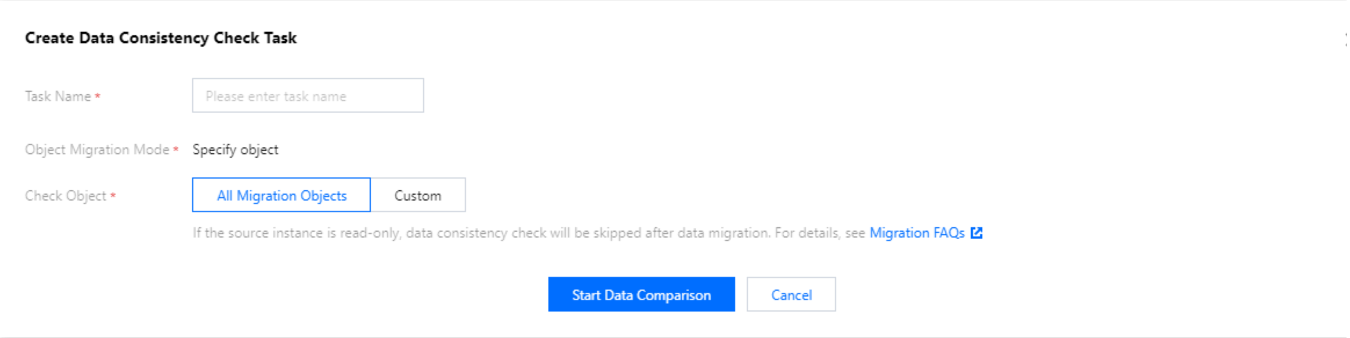
5. After configuring the data consistency check parameters, click **Start Data Comparison**.

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.



Create Data Consistency Check Task

Task Name * Please enter task name

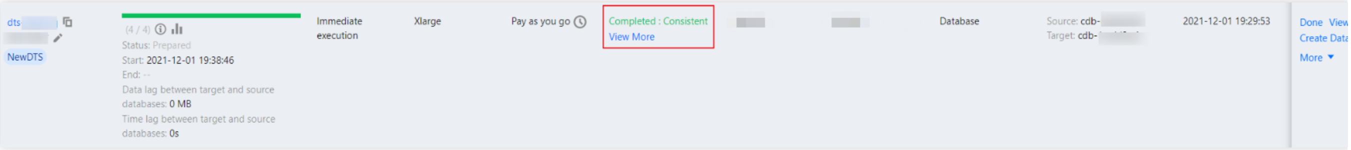
Object Migration Mode * Specify object

Check Object * All Migration Objects | Custom

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

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-[redacted]
[redacted]
NewDTS

(4 / 4)
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

Immediate execution

Xlarge

Pay as you go

Completed : Consistent
View More

[redacted]

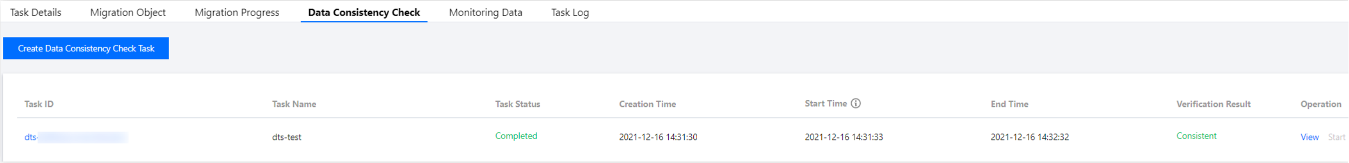
[redacted]

Database

Source: cdb-[redacted]
Target: cdb-[redacted]

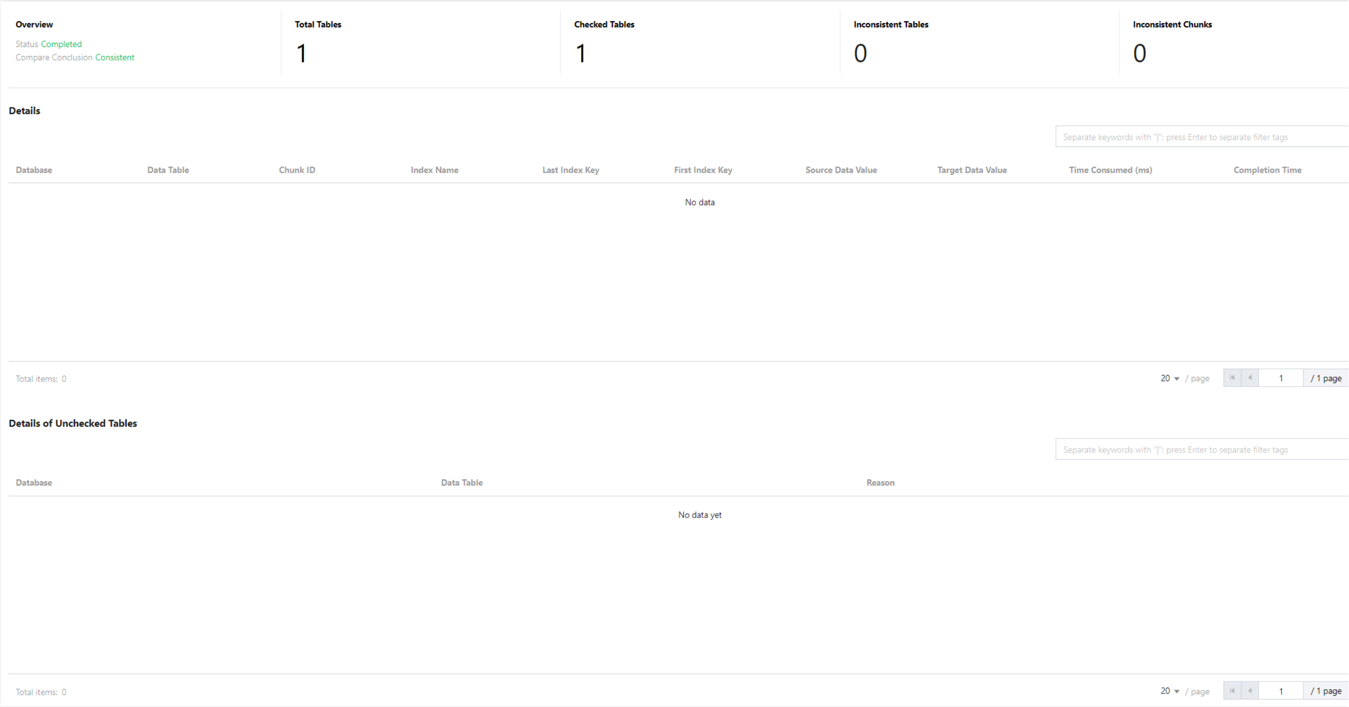
2021-12-01 19:29:53

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



Task ID	Task Name	Task Status	Creation Time	Start Time	End Time	Verification Result	Operation
dts-[redacted]	dts-test	Completed	2021-12-16 14:31:30	2021-12-16 14:31:33	2021-12-16 14:32:32	Consistent	View Start

If the data is consistent, the result will be like:



Overview
Status Completed
Compare Conclusion Consistent

Total Tables	Checked Tables	Inconsistent Tables	Inconsistent Chunks
1	1	0	0

Details

Separate keywords with "|"; press Enter to separate filter tags

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 page

Details of Unchecked Tables

Separate keywords with "|"; press Enter to separate filter tags

Database	Data Table	Reason
	No data yet	

Total items: 0 20 ▾ / page 1 / 1 page

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.

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:

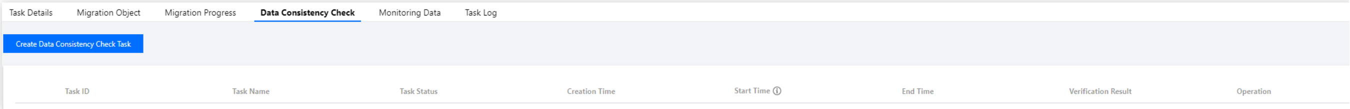
In this case, the full data and all the database information will be checked by default. If you need to filter check objects, create a data consistency check task manually.

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**.

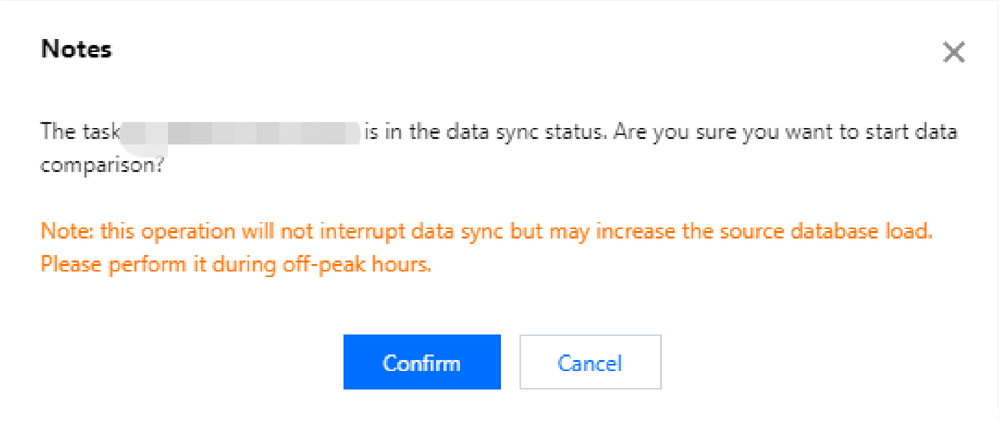
Note:

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.



Task ID	Task Name	Task Status	Creation Time	Start Time ⓘ	End Time	Verification Result	Operation

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



Notes

The task [redacted] is in the data sync status. Are you sure you want to start data comparison?

Note: this operation will not interrupt data sync but may increase the source database load. Please perform it during off-peak hours.

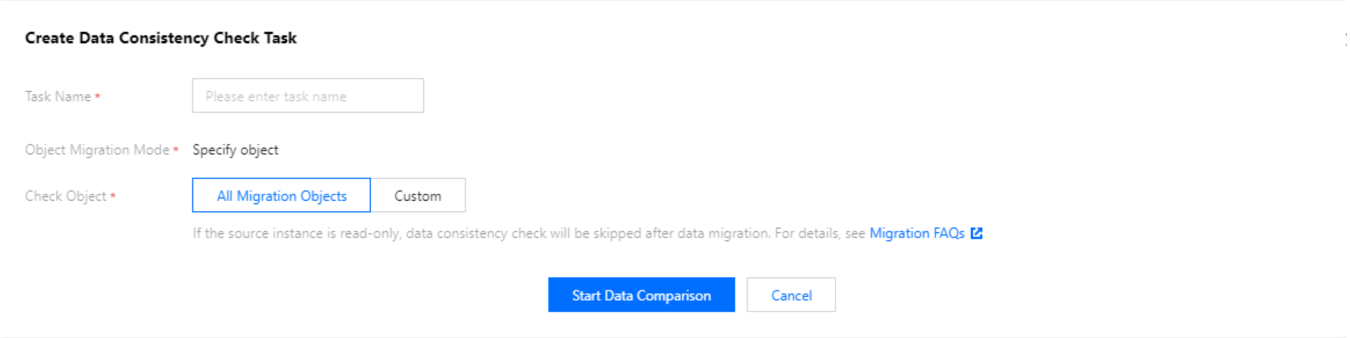
5. After configuring the data consistency check parameters, click **Start Data Comparison**.

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.



Create Data Consistency Check Task

Task Name * Please enter task name

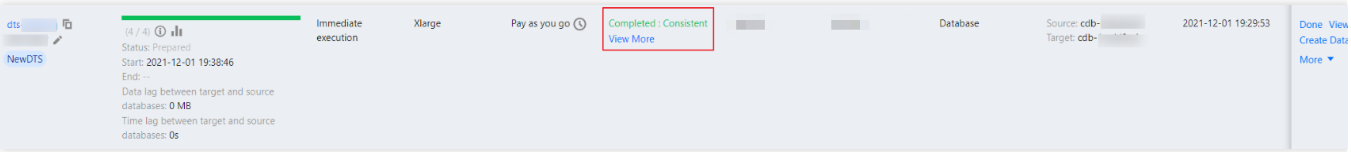
Object Migration Mode * Specify object

Check Object * All Migration Objects | Custom

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

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.

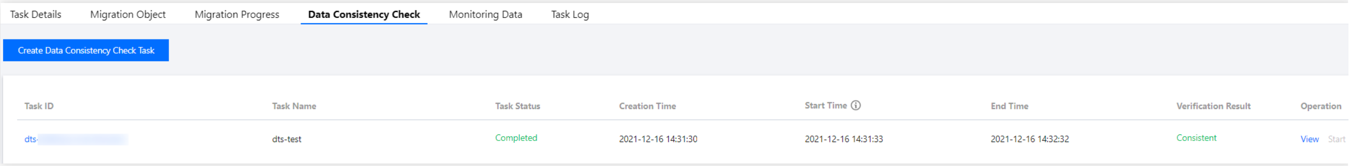


dts-[redacted]
NewDTS

(4 / 4)
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

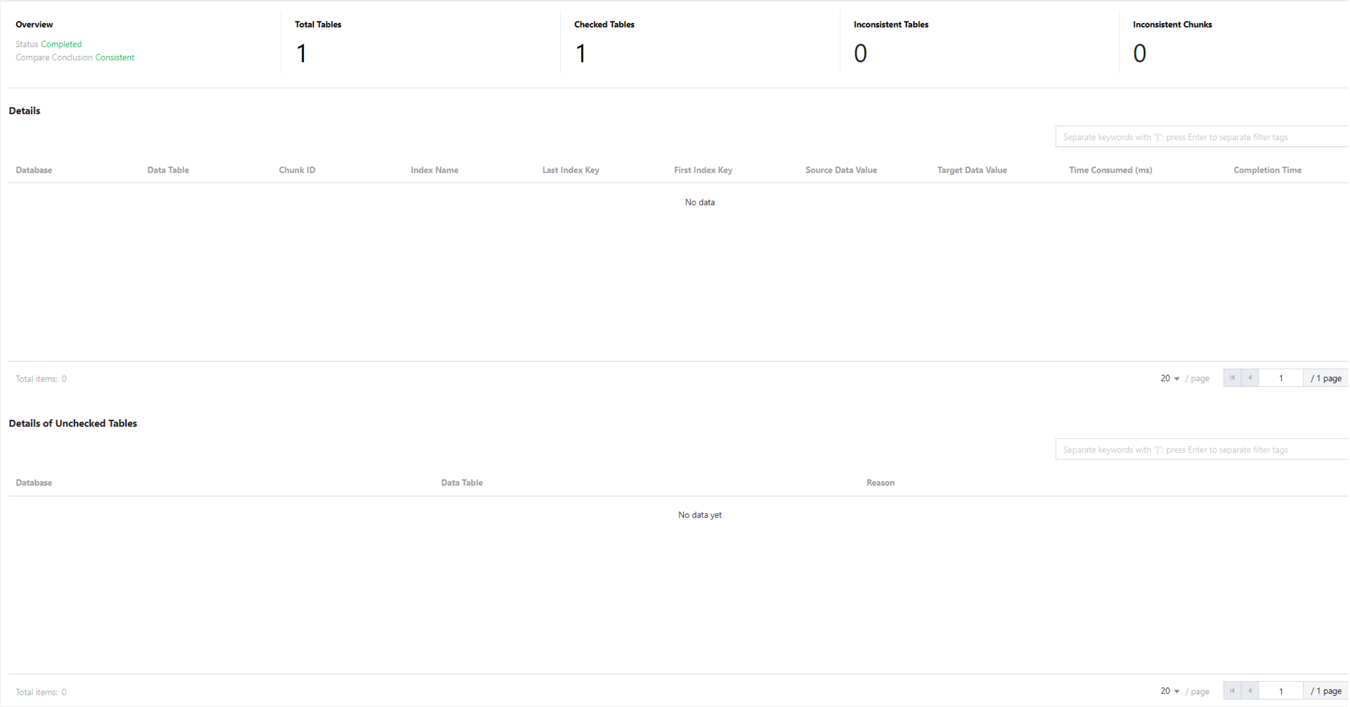
Immediate execution | Xlarge | Pay as you go | Completed : Consistent View More | [redacted] | [redacted] | Database | Source: cdb-[redacted] Target: cdb-[redacted] | 2021-12-01 19:29:53

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



Task ID	Task Name	Task Status	Creation Time	Start Time	End Time	Verification Result	Operation
dts-[redacted]	dts-test	Completed	2021-12-16 14:31:30	2021-12-16 14:31:33	2021-12-16 14:32:32	Consistent	View Start

If the data is consistent, the result will be like:



Overview
Status Completed
Compare Conclusion Consistent

Total Tables	Checked Tables	Inconsistent Tables	Inconsistent Chunks
1	1	0	0

Details

Separate keywords with "|"; press Enter to separate filter tags

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

Details of Unchecked Tables

Separate keywords with "|"; press Enter to separate filter tags

Database	Data Table	Reason
No data yet		

Total items: 0

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

Instructions

Last updated：2024-09-25 11:06:01

Category	Description
Source/Target Type	1. Source Type: Self-built databases (IDC self-built and CVM self-built) SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, and 2022. TencentDB (intra-account and cross-account) for SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, and 2022. 2. Target Type: TencentDB (intra-account and cross-account) for SQL Server 2008R2, 2012, 2014, 2016, 2017, 2019, and 2022. 3. Data Transfer Service (DTS) migration is not supported in network shared storage disk environments.
Cross-region Migration	Currently, cross-region migration is supported in the Chinese Mainland and Hong Kong (China) regions. Other regions do not support cross-region migration.
Migration Object	1. Only database-level migration is supported, that is, all objects in the database must be migrated together. Single-table migration is not supported. 2. The migration of basic database and table objects is supported, but migration of instance-level jobs, triggers, DB links (link servers), or user permissions is not supported.**These objects need to be rebuilt after the migration is complete.**.
Source Database Impact	1. During full data migration, DTS consumes certain source instance resources, which may increase the load and pressure of the source database. If your database configuration is low, we recommend you perform the migration during off-peak hours. 2. The full data migration process is implemented through locked migration, and write operations will be briefly (in seconds) blocked during table locking. 3. Due to changes in the physical environment of data files after migration, database indexes will become invalid. You need to rebuild the indexes after migration; otherwise, database performance may significantly decline. 4. By default, a lock-free synchronization method is used. In the full data export stage, a global lock (FTWRL) will not be applied on the source database; only table locks are applied on tables without a primary key.
Supported	1. DDL

SQL Statements	TABLE: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, and RENAEM TABLE VIEW: CREATE VIEW, A LTER VIEW, and DROP VIEW INDEX: CREATE INDEX, and DROP INDEX DATABASE: CREATE DATABASE, ALTER DATABASE, and DROP DATABASE 2. DML INSERT, UPDATE, DELETE, and REPLACE
Limits on Operations	1. Only one migration task can be initiated at any time for the same source instance. 2. 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. 3. Do not perform transaction log backup during incremental sync; otherwise, the transaction log will be truncated and become discontinuous. 4. If you only perform full data migration, do not write new data into the source database during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure data consistency in real time, we recommend that you select full + incremental data migration. 5. For full + incremental data migration, when the task status is **completing** after clicking Complete, do not write new data into the source database. We recommend stopping data writes for two minutes after clicking **Complete**; otherwise, the data in the source and target databases may be inconsistent.

Migration Operation Guide

Last updated：2024-09-25 14:34:56

Operation scenarios

This document provides operation guidance for migrating data from SQL Server to TencentDB for SQL Server using the DTS data migration feature.

Preparations

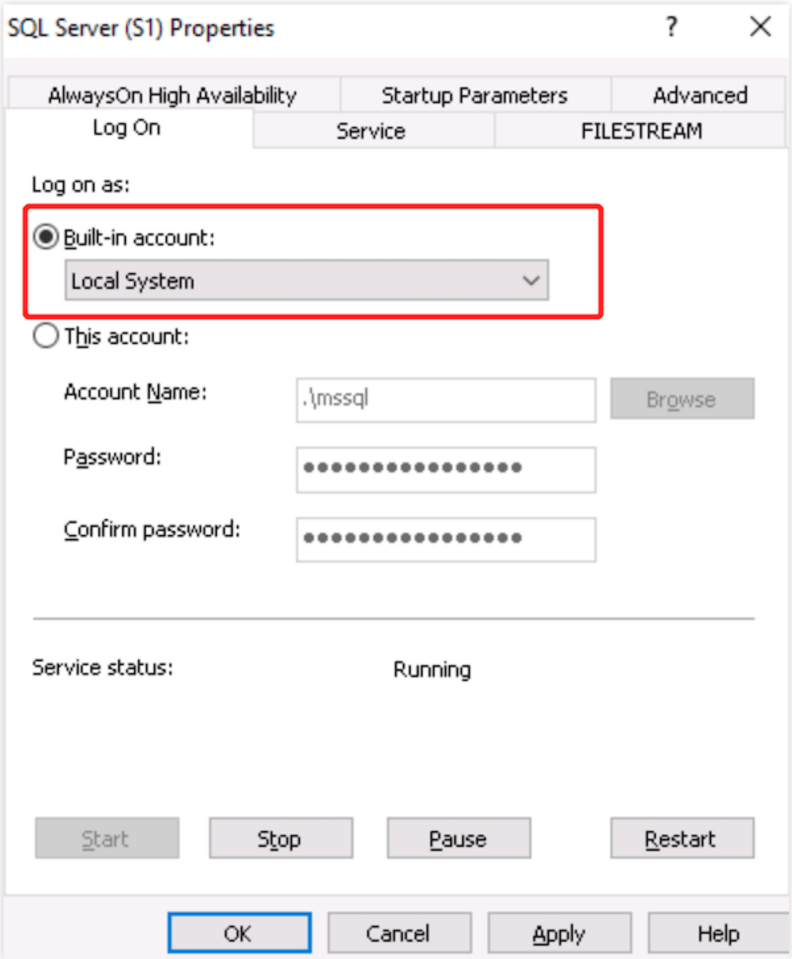
1. Please carefully read the Usage Instructions to understand the feature constraints and precautions.
2. In advance, ensure the access path between DTS and the database is established according to the access type you need. For details, refer to Network Preparation Work.

IDC self-built database: You can choose "Public Network/Direct Connect/VPN Access/Cloud Connect Network (CCN)" as the access type.

Self-built databases on cloud virtual machine (CVM): Choose "Self-Built on CVM" as the access type.

TencentDB instances: Select "TencentDB" as the access type.

3. The service where the source database is located must have the file-sharing port 445 enabled.
4. The source database must be set to "Full Recovery Mode", and it is recommended to make a full backup before migration.
5. The local disk space of the source database must be large enough to accommodate the size of the database to be migrated.
6. When the source instance is a non-TencentDB for SQL Server instance (self-built instance on Public Network/CVM) or a TencentDB for SQL Server Basic Edition instance, the target end must use an account with sysadmin permissions for migration and be able to run the xp_cmdshell stored procedure. When the source instance is a TencentDB for SQL Server High Availability Edition or Cluster Edition instance, there are no permission restrictions on the target end account.
7. The SQL service startup account on the migration source needs to be changed to the built-in account Local System. There are no restrictions on the account of the source database to be migrated, but it must have sysadmin permissions.



Log on as:

● Built-in account:

Local System

○ This account:

Account Name: .\mssql

Password: ●●●●●●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●●●●●●

Service status: Running

As shown in the figure, start the SQL service on the migration source. In the startup configuration, select **Log on as**, select **Built-in account**, and change it to Local System startup.

Note:

After modifying the account, you need to restart the SQL server service.

Environment Requirements

Note:

The following environment requirements will be automatically checked by the system before starting the migration task. The system will report an error for requirements not met. If users can identify the issue, they can refer to Verification Item Check Requirements to make necessary modifications by themselves. If not, wait for the system check to complete and then follow the error prompts to make the necessary modifications.

Type	Environmental Requirements
Source Database Requirements	The service where the source instance is located needs to have the file-sharing port 445 enabled. The source and target databases' networks must be interconnected.

	The server where the source database is located must have sufficient outbound bandwidth; otherwise, the migration rate will be affected.
Target Database Requirements	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 that of the source instance. The target database cannot have databases with the same name as those in the source database. The disk space of the target database must be larger than the size of the source database, specifically 1.5 times the size of the source database. The target database cannot have access requests or active businesses; otherwise, the migration will fail.

Migration Operation

1. Log in to DTS console, select **Data Migration** on the left navigation bar, and click **Create Migration Task** to enter the Create Migration Task page.
2. On the Create Migration Task page, select the source instance type and region, the target instance type, region, specification, etc., and then click **Buy Now**.

Configuration Item	Description
Source instance type	Please select according to your source database type. Once purchased, it cannot be modified. For this scenario, 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	Please select according to your target database type. Once purchased, it cannot be modified. For this scenario, select "SQL Server".
Target instance region	Select the target database region.
Specification	Currently, only the fixed specification Medium is supported.

3. On the Set Source and Target Database page, complete task settings, source database settings, and target database settings. Once the connectivity test between the source and target databases is passed, click **Create**.

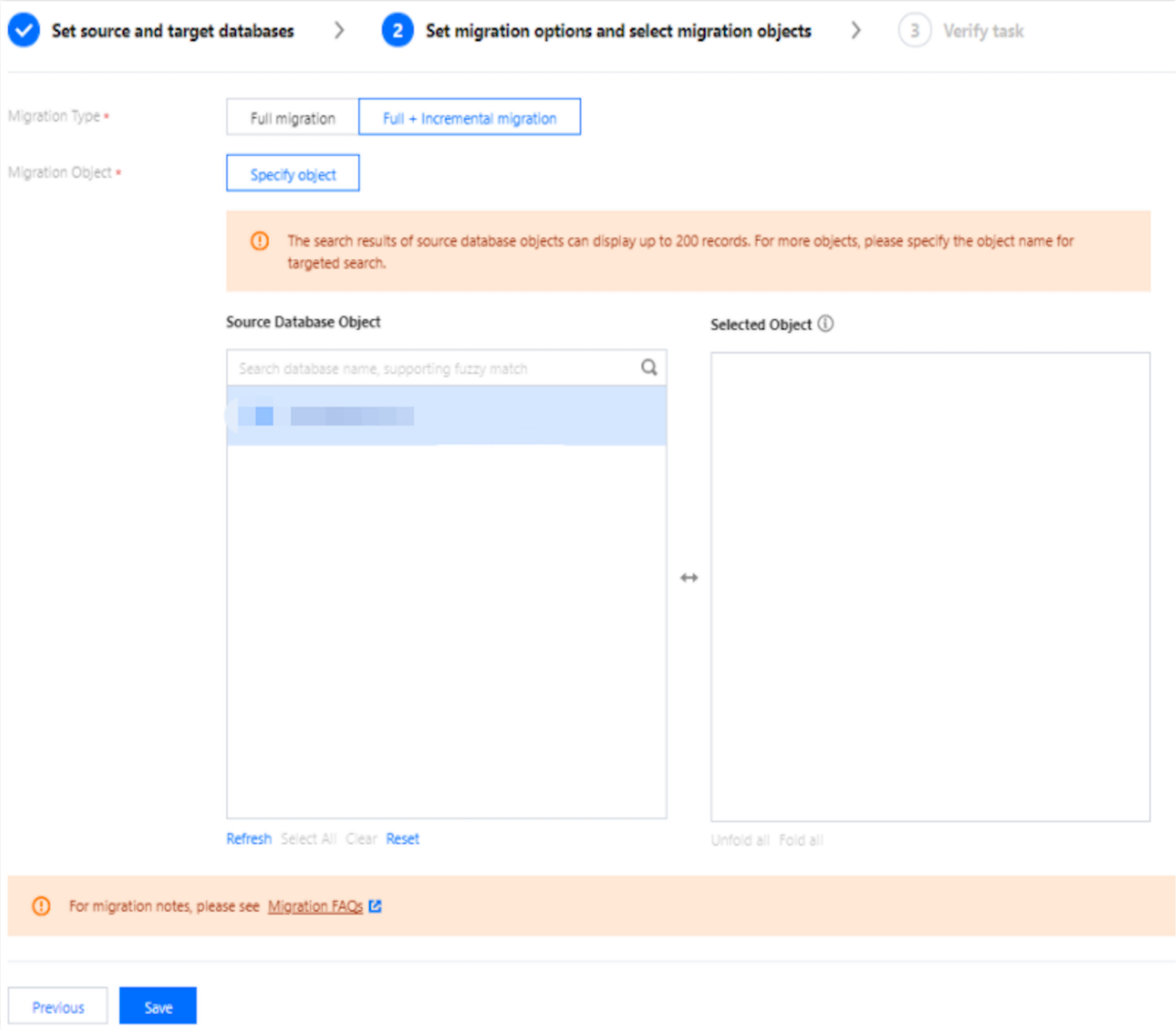
Note:

If the connectivity test fails, troubleshoot and fix the issues as prompted in the Repair Guidance, then try again.

Settings Type	Configuration Item	Description
Task configuration	Task name	Set a business-significant name for easy task identification.
	Running node	Execute immediately: Starts the task immediately after task validation is passed. Scheduled execution: A task execution time must be configured, and the task will start when the time is reached.
	Tag	Tags are used to manage resources by category from different dimensions. If the existing tags do not meet your requirements, please go to the Console to manage tags.
Source database settings	Source database type	The source database type selected at the time of purchase. It cannot be modified.
	Region	The source database region selected at the time of purchase. It cannot be modified.
	Access type	Please choose according to your scenario. This scenarios takes "TencentDB" as an example. For preparation work of different access types, please refer to Preparation Overview. Public network: The source database can be accessed via a public network IP. Self-Built on CVM: The source database is deployed on Tencent CVM. Direct connect: The source database can be connected to Tencent Cloud Virtual Private Cloud (VPC) via direct connect. VPN access: The source database can be connected to Tencent Cloud VPC via VPN connections. TencentDB: The source database is a TencentDB instance. CCN: The source database can be connected to Tencent Cloud VPC via Cloud Connect Network.
	Cross-account/intra-account	This Account: The source database instance and the target database instance belong to the same Tencent Cloud root account. Cross-account: The source database instance and the target database instance belong to different Tencent Cloud root accounts. The following takes intra-account migration as an example. For cross-account operations, please refer to TencentDB Cross-Account Migration Guide.
	Database instance	Select the instance ID of the source database.
	Account	The account of the source SQL Server database. The account must have the required permissions.

	Password	The password of the source SQL Server database account.
Target database settings	Target database type	The target database type selected at the time of purchase. It cannot be modified.
	Region	The target database region selected at the time of purchase. It cannot be modified.
	Access Type	Select according to your scenario. For this scenario, select "TencentDB".
	Database instance	Select the instance ID of the target database.
	Account	The database account of the target database. It must have the required permissions.
	Password	The password of the target database account.

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



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

Migration Type * Full migration | Full + Incremental migration

Migration Object * Specify object

The search results of source database objects can display up to 200 records. For more objects, please specify the object name for targeted search.

Source Database Object
Selected Object
Search database name, supporting fuzzy match
Refresh Select All Clear Reset
Unfold all Fold all

For migration notes, please see Migration FAQs

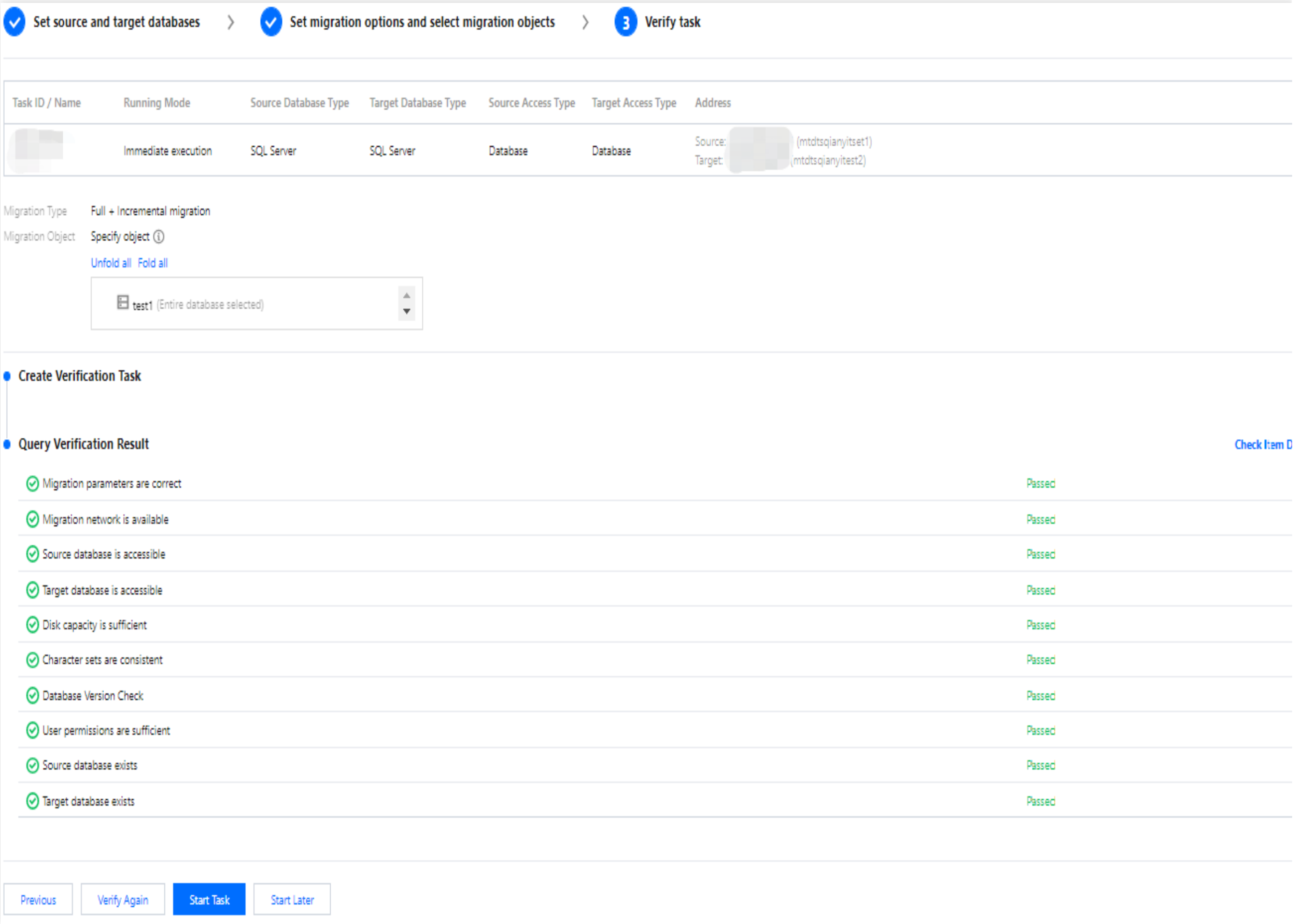
Configuration Item	Description
Migration type	Please choose according to your scenario. Full migration: The entire database is migrated. The data to be migrated includes only the existing content of the source database at the time of task initiation and does not include the incremental data written to the source database after the task initiation. Full + incremental migration: The data to be migrated 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.
Migration object	Only database-level migration is supported, meaning that all objects in the specified database need to be migrated together. Select the database to be migrated from the source database objects, then move it to the Selected Object box.

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

If the task verification fails, refer to Pre-Verification Failure Handling, fix the issue, and reinitiate the verification task.

Failed: Indicates that the verification items failed the check, the task is blocked, and you need to fix the problem and run the verification task again.

Alarm: Indicates that the verification items do not fully meet the requirements. You can continue with the task, but it may have some impact on the business. Users need to evaluate based on the prompts whether to ignore the warning or fix the issues before continuing.



✓ Set source and target databases › ✓ Set migration options and select migration objects › 3 Verify task

Task ID / Name	Running Mode	Source Database Type	Target Database Type	Source Access Type	Target Access Type	Address
[redacted]	Immediate execution	SQL Server	SQL Server	Database	Database	Source: [redacted] (mtdtsqianyitset1) Target: [redacted] (mtdtsqianyitest2)

Migration Type Full + Incremental migration

Migration Object Specify object ⓘ

Unfold all Fold all

test1 (Entire database selected)

Create Verification Task

Query Verification Result

Check Item D

✓ Migration parameters are correct	Passed
✓ Migration network is available	Passed
✓ Source database is accessible	Passed
✓ Target database is accessible	Passed
✓ Disk capacity is sufficient	Passed
✓ Character sets are consistent	Passed
✓ Database Version Check	Passed
✓ User permissions are sufficient	Passed
✓ Source database exists	Passed
✓ Target database exists	Passed

6. Return to the data migration task list. The task enters the ready-to-run state. After 1 to 2 minutes, the data migration task will officially start.

If you need to view the task, delete the task, or perform other operations, please click the corresponding task and perform operations in the **Operation** column. For details, refer to Task Management.

If there is an error in the task, please refer to Error Handling.

7. Assess whether to end the task.

Select **Full migration**: Once the task is completed, it will end automatically; no manual action is required.

Select **Full + incremental migration**: After full migration is completed, it will automatically enter the incremental data synchronization stage. Incremental data synchronization will not end automatically; you need to verify the migration results and manually click **Complete** to end incremental data synchronization. If business switching is needed, refer to Cutover Instructions.

Post-migration Operations

After completing the migration using DTS, it is recommended to perform the following checks on the target database:

Permission completeness. Permissions will affect operations performed on the database. The migration only restores data. To restore other service-level permissions, such as database users and login user names, you need to recreate them and associate them with database accounts.

Indexes: Reindexing is recommended. As the physical environment of the data files changes, database index statistics may not be updated in a timely manner. It is advised to perform reindexing; otherwise, database performance may degrade.

Instance-level objects. After the migration is completed, users need to re-create these by themselves.

Related APIs

For DTS-related APIs, please refer to: Viewing Related APIs.

Pre-Validation Failure Handling

Migration Parameter Check

Last updated：2024-11-01 15:10:27

Check Details

The target instance cannot contain the database to be migrated; otherwise, an error will be reported with the details: "The database to be migrated already exists in the target instance."
The provided migration account must be able to be logged in normally; otherwise, an error will be reported with the details: "The provided migration account cannot be logged in."
The name of the database to be migrated cannot contain sensitive characters; otherwise, an error will be reported with the details: "The name of the database to be migrated contains sensitive characters."
The name of the database to be migrated should comply with standards; otherwise, an error will be reported with the details: "The name of the database to be migrated does not comply with standards. Database names must consist of letters, numbers, and underscores, and must start with a letter."
The name of the database to be migrated should be at least 1 character long; otherwise an error will be reported with the error details: "The name of the database to be migrated must be at least 1 character long."
The name of the database to be migrated cannot exceed 64 characters; otherwise an error will be reported with the error details: "The name of the database to be migrated cannot exceed 64 characters."

Fixing Method

Serial Number	Error Details	Fixing Method
1	The database to be migrated already exists in the target instance.	Please change the database to be migrated in the source instance or delete the database with the same name as the database to be migrated in the target instance.
2	The provided migration account cannot be logged in.	1. Please check if the source instance is already in the Running state. 2. Please check if the port of the source instance is being blocked by a firewall security group. 3. Please check if the port of the source instance is entered incorrectly. 4. Please check if the account and password of the source instance are entered incorrectly.

3	The name of the database to be migrated contains sensitive characters.	Please change the database to be migrated in the source instance or rename the database to remove illegal characters before migration.
4	The name of the database to be migrated does not comply with standards. Database names must consist of letters, numbers, and underscores, and must start with a letter.	Please change the database to be migrated in the source instance or rename the database according to naming standards before migration.
5	The name of the database to be migrated must be at least 1 character long.	Please change the database to be migrated in the source instance or rename the database according to naming standards before migration.
6	The name of the database to be migrated cannot exceed 64 characters.	Please change the database to be migrated in the source instance or rename the database according to naming standards before migration.

Migration Network Check

Last updated：2024-09-25 11:17:46

Check details

Check if the internal migration network is connected. If the check is not completed, you may see the error message: "Waiting for network check." Please wait for a moment without taking any action. If there is a problem with the internal migration network, you will see the error details: "Network check failed."

Fixing Method

If a migration network check error is reported: "Network check failed," please submit a ticket to get a solution.

Source Database Connectivity Check

Last updated：2024-09-25 11:18:47

Check Details

The Data Transfer Service (DTS) server needs to be able to connect to the source database; otherwise an error will be reported with the error details: "Check if the DTS server can connect to the source database."
The original server needs to be connected; otherwise, an error will be reported with the error details: "Insufficient migration account permissions. Unable to complete the migration."

Fixing Method

Serial Number	Error Details	Fixing Method
1	Check if the DTS server can connect to the source database.	Please check if the SQL service startup account for the source instance is using the built-in Local System account. You need to select "Built-in account" in "Log on as" in the startup configuration and modify it to start with the Local System account. Please check if xp_cmdshell is enabled on the source instance. xp_cmdshell needs to be enabled. Please ensure the migration account has sysadmin permissions.
2	Insufficient migration account permissions. Unable to complete the migration.	

Target Database Connectivity Check

Last updated：2024-09-25 11:19:56

Check Details

The Data Transfer Service (DTS) server needs to be able to connect to the target database; otherwise, an error will be reported with the error details: "Check if the DTS server can connect to the target database.".

Fixing Method

If a target database connectivity check error is reported: "Check whether the DTS server can connect to the target database," please submit a ticket to get a solution.

Disk Space Check

Last updated：2024-09-25 11:21:03

Check Details

Check whether the disk space on the server of the source database is sufficient. During migration of a self-built source instance, 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; otherwise, an error will be reported with the error details: "Check whether the disk space on the destination server is sufficient.".

Fixing Method

If a disk space check error is reported: "Check whether the disk space on the destination server is sufficient," please check the purchased disk space size of the target instance. It is recommended that the disk space of the target instance is not smaller than the disk space of the source instance, preferably 1.5 times the size of the source instance's disk space. You can expand the disk space size of the target instance through the Adjusting Instance Specification feature in the console.

Character Set check

Last updated：2024-09-25 11:22:20

Check Details

The character sets of the source database and target database need to be consistent, otherwise, an error will be reported with the details: "Check if the character sets are consistent.".

Fixing Method

If a character set check error is reported: "Check if the character sets are consistent," please check if the character sets at the database level are consistent. If they are not consistent, please adjust the character sets of the source database and the destination database to be consistent.

Database Version Check

Last updated：2024-09-25 11:23:26

Check Details

The source database version cannot be later than the target database version; otherwise, an error will be reported with the error details: "Check if the database version numbers are consistent.".

Fixing Method

If a data version check error is reported: "Check if database version numbers are consistent," please verify the database versions of the source and target instances. A later version instance cannot migrate to an earlier version instance. For example, a source instance of version 2012 cannot migrate to a target instance of version 2008. The target instance's version must be later than or equal to the source instance's version for migration to proceed. You can use the console's Adjusting the Instance Version feature to upgrade the target instance's version.

User Permission Check

Last updated：2024-09-25 11:24:25

Check Details

Check if the target instance permissions exist. If not, an error will be reported with the error details: "Check if target instance permissions exist".

Fixing Method

If a user permission check error is reported: "Check if target instance permissions exist", please submit a ticket to get a solution.

Source Database Existence Check

Last updated：2024-09-25 11:25:22

Check details

Check if the source database to be migrated no longer exists. If it does not exist, an error will be reported with the error details: "Check if the source database exists".

Fixing Method

If a source database existence check error is reported: "Check whether the source database exists," please confirm whether the database to be migrated exists in the source instance. If it does not exist, you need to reconstruct the migration task. It is recommended not to delete the source database during the migration process.

Target Database Existence Check

Last updated：2024-09-25 11:26:20

Check Details

When migrating the source database to the target instance, ensure that there are no databases with the same name as the source database to be migrated in the target instance; otherwise, an error will be reported with the error details: "Check if the target database exists.".

Fixing Method

If a target database existence check error is reported: "Check if the target database exists," it means that the database to be migrated already exists in the target instance. Please delete or rename the database with the same name in the target instance and then retry.

Error Handling

Last updated：2024-09-25 11:27:15

Problem Scenario

After the migration task officially starts, it will go through the following **Migration Steps**.

1. Prohibit backup Jobs.
2. Back up databases.
3. Transfer backup files.
4. Restore databases.
5. Deploy real-time synchronization (This step is involved only when the migration type is: full + incremental migration).

During the above migration steps, if a task failure occurs, click **Error Message** to understand the cause of the task failure, and click **Error Details** for handling suggestions.

Problem Handling

The following table lists the error messages of migration task failures and handling suggestions.

Serial Number	Error Message	Suggestions
1	Task failed. Please submit a ticket to get a solution.	Please submit a ticket to get a solution.
2	The database to be migrated was not found in the source instance.	The database to be migrated was not found in the source instance. Please confirm whether the database to be migrated exists in the source instance.
3	The migration initialization operation cannot be performed on the source instance.	The migration initialization operation cannot be performed on the source instance: 1. Please check if the SQL service startup account for the source instance is using the built-in Local System account. You need to select "Built-in account" in "Log on as" in the startup configuration and modify it to start with the Local System account. 2. Please check if the source instance has xp_cmdshell enabled. You need to enable xp_cmdshell.

		3. Please ensure that the migration account has sysadmin permissions.
4	Later-version instances cannot migrate to earlier-version instances.	Later-version instances cannot migrate to earlier-version instances, e.g., a source instance of version 2012 cannot migrate to a target instance of version 2008. The target instance version must be greater than or equal to the source instance version for migration to occur. You can upgrade the target instance version using the Adjusting Instance Version feature in the console.
5	Backup jobs for the source instance cannot be disabled.	Please submit a ticket to get a solution.
6	Creating full backup files for the source database failed.	Creating full backup files for the source database failed: 1. Please execute net share in the cmd terminal of the source instance to check if a shared folder named backup exists. 2. Please check if the backup files are generated in the shared folder of the source instance. 3. Please check if the source instance does not have enough disk space to create a backup.
7	Source instance failed to transfer backup files.	Source instance failed to transfer backup files: 1. Please check if the file-sharing service is enabled on the source instance. 2. Please check if the system account beginning with "ls" is correctly created in the source instance. 3. Please execute net share in the cmd terminal of the source instance to check if a shared folder named "backup" exists. 4. Please check if the system account beginning with "ls" has full control permissions for the shared folders.
8	Failed to recover the database using the backup file on the target instance.	Failed to recover the database using the backup file on the target instance: 1. Please check if the file-sharing service is enabled on the source instance. 2. Please check if the system account beginning with "ls" has been correctly created in the source instance. 3. Please execute net share in the cmd terminal of the source instance to check if a shared folder named 'backup' exists. 4. Please check if the system account beginning with "ls" has full control permissions for the shared folders.
9	Incremental synchronization deployment failed.	Please submit a ticket to get a solution.
10	Failed to synchronize	Failed to synchronize incremental logs:

	incremental logs.	1. Network connectivity exception occurs during synchronization and sharing incremental files failed. Please redo the migration. 2. During incremental migration, transaction logs were truncated. Please stop the log backup job on the source instance and redo the migration.
11	After synchronization, data inconsistency exists.	After synchronization, data inconsistency exists. Full backup data synchronization has been completed. Please manually check the migrated data. If there are significant data differences, re-migration is required. The possible causes of data inconsistency are as follows: 1. The source database was not stopped for writes: Before clicking "Complete" for the task, the source database needs to be stopped for writes for 3-5 minutes to avoid data verification inconsistencies. 2. Incremental log synchronization failure: During incremental synchronization, log backup and log truncation operations on the source instance need to be stopped.

Migrating to TencentDB for Redis

Migration from Redis to Redis

Supported Capabilities

Last updated：2024-09-10 17:14:48

Supported Scenes and Versions

Source	Target	Scene Description
Self-built Redis database (IDC Self-built/CVM Self-built) 2.8, 3.0, 3.2, 4.0, 5.0, 6.2, 7.0	Tencent Cloud Redis 2.8, 3.0, 4.0, 5.0, 6.2, 7.0	Migration of off-cloud databases to the cloud
Other third-party cloud providers' Redis AWS ElasticCache for Redis 4.0, 5.0, 6.0, 6.2, 7.0 Codis	Tencent Cloud Redis 2.8, 3.0, 4.0, 5.0, 6.2, 7.0	Migration of Redis databases from other cloud providers to TencentDB instances
TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.2, 7.0	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, 5.0, 6.2, 7.0	Migration of TencentDB for Redis within the same region or between different regions Migration of TencentDB for Redis across versions Migration of TencentDB for Redis within the same root account or between different root accounts Migration between TencentDB for Redis standard architecture and cluster architecture
	Self-built Redis database (IDC Self-built/CVM Self-built) 2.8, 3.0, 3.2, 4.0, 5.0, 6.2, 7.0	Migration of TencentDB for Redis to a self-built database

Supports migration between different architectures, including standard architecture > standard architecture, standard architecture > cluster architecture, and cluster architecture > cluster architecture. However, heterogeneous migration (e.g., standard architecture > cluster architecture) may have compatibility issues. For common compatibility issues, see Check on Migration from Standard Architecture to Cluster Architecture.

Standard architecture refers to self-built standalone versions, primary-secondary modes, and TencentDB for Redis Memory Edition (Standard Architecture). Cluster architecture includes Redis Cluster native solutions, Codis cluster solutions, and Twemproxy cluster solutions.

Supported Features

Feature Category	Feature Sub-item or Description	Supported Capabilities
Migration object	-	Databases and collections
Migration type	-	Only supports Full + Incremental Migration.
Checkpoint restart	-	Not supported
Key task management operations	Retry	Supported
	Speed limit	Supported
	Create similar task	Supported
Topology synchronization	One-to-one	Supported
	Bidirectional synchronization	Not supported

User Guide

Last updated：2024-09-24 15:31:05

Impact on the Source Database

During full data migration, DTS will consume some resources from the source instance, which may lead to an increase in the load on the source instance and add pressure to the database. If your database has a low configuration, it is recommended to perform the migration during off-peak hours.

Migration Restrictions and Notes

1. DTS migration requires the source instance to support the PSYNC command.
2. The source instance should be in a normal operating status to support migration. Migration is not possible if the password has not been initialized or if other tasks (such as scaling) are in progress.
3. If the source database is AWS ElasticCache for Redis, it requires submitting a ticket for application. The corresponding constraints and limitations are as follows:
3.1 Only ElasticCache for Redis is supported for migration; other AWS Redis products are currently not supported.
3.2 TLS connections are currently not supported. If TLS is enabled on the source database, it should be disabled before migration. However, since disabling TLS on AWS ElasticCache for Redis also disables Auth user authentication, this may expose your data to risks. Therefore, it is recommended to first create a security group, then disable TLS, and add the DTS Service IP to the source database's security group allowlist.
3.3 Since AWS ElasticCache for Redis replica nodes does not support the PSYNC command, migration can only be performed using the primary node.
4. DTS tasks cannot form circular synchronization. Data written during circular synchronization will not be terminated and will continuously loop, leading to a write storm.
5. DTS does not support resumable transmission. Since there are no Binlog files, all incremental data is synchronized in memory. If the connection is lost, the incremental data is lost from memory. Therefore, once a task is initiated, it cannot be interrupted or paused.
6. Migration of Modules is not supported.
7. If global replication is enabled for Tencent Cloud Redis, DTS only supports selecting a primary instance when connecting to the source database. Selection of a read-only instance only facilitates the migration of full data, as migration of incremental data is not supported currently.
8. Do not perform the following operations during migration, as they may result in migration task failure.
Do not modify or delete user information (including usernames, passwords, permissions) and port numbers in the source and target databases.

Do not delete the target database during the data migration phase.

Do not make changes to the target database during the data migration phase to prevent inconsistencies in the final data.

Migration Operation Guide

Last updated：2024-09-24 15:33:35

Operation Scenes

DTS-based Redis data migration supports full + incremental data migration, allowing both the historical data in the source database before the migration and any new data written during the migration process to be migrated together. This document introduces how to use the DTS data migration feature to migrate data from Redis to TencentDB for Redis.

Preparations

1. Establish the access channel between DTS and the database in advance according to the type of access you need. For detailed instructions, see Network Preparation Work.

IDC self-built databases/Databases from other cloud providers: The available access methods include Public Network/Direct Connect/VPN Access/CCN".

When public network instances are migrated, ensure that the source instance is accessible over the public network and that the connection remains stable. Network fluctuations or failures can cause the migration to fail, and if that happens, you need to restart the migration task.

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

TencentDB Instance: Select Database as the access method.

2. Pre-Migration Check

You should check and make sure that the following items are passed before the migration; otherwise, the migration may fail.

2.1. 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'
```

2.2. 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
```

2.3. 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
```

2.4. (Optional) For scenes involving migration from a standard architecture to a cluster architecture, check for command compatibility issues. For detailed static and dynamic evaluations, see Standard to Clustered Architecture Migration Assessment.

Directions

Step 1:Create a migration task

1. log in to DTS Console, select **Data Migration** page, then on the right side click **Create Migration Task**.

2. On the new migration task page, select the task creation parameters, and then click **Buy Now**.

Configuration Parameter	Description
Service Type	Select **Data Migration**.
Creation Mode	Create a task: Create a completely new task. Create a similar task: Quickly create a task with the same configuration as that of a previous task. The new task's pre-filled options such as database type, access method, billing mode, and migration type are consistent with that of the historical task. Users can modify the settings if needed.
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.
Version	Currently, only **NewDTS** is supported.
Specification	Currently, only the **Xlarge** specification is supported.
Tag	Set tag keys and values for the migration task to efficiently manage multiple tasks using tags.
Task Name	Select **Name After Creation**: The default task name will be the same as the task ID. You can rename the task after the migration task is created. Select **Name Now**: Enter the task name in the input box below.
Terms of service	Make sure to check the box for I have read and agree to <Cloud Database Service Terms>.
Quantity	You can purchase up to 10 migration tasks at a time.

3. After the purchase is completed, the page will automatically redirect to the Data Migration Task List.

Step 2:Set source and target databases

1. In the Data Migration Task List, find the task you just created, and click **Configure** in the **Operation** column.
2. Go to the **Set source and target databases** tab, fill in the source and target database information,and then click **Test Connectivity**. Once the test passes, click **Save** to proceed to the next step.

Task Configuration

Configuration Parameter	Description
Task Name	Set a task name that is easy to identify.
Running Mode	Immediate execution: The task will start immediately after the pre-check validation passes. Scheduled execution: Set a specific time for the task to start. The task will not start immediately after the pre-check validation passes; instead, it will start at the scheduled time.
Automatic Retry	If it is selected, the migration task will automatically retry within the specified time range in case of task interruptions caused by network exceptions or similar issues. No manual intervention is

	required. The retry mechanism restarts the data migration from the beginning. During the retry process, any data that was previously migrated to the target end will either be cleared or overwritten.

Source Database Settings

Configuration Parameter	Description
Source Database Type	The source database type selected during purchase, which cannot be changed.
Service Provider	If the source database is a self-built database on IDC or CVM, or a TencentDB database, select "**Others**". If the source database is an Alibaba Cloud database, Huawei Cloud, etc., select "**Others**". If the source database is an AWS database, select AWS. Currently, only AWS ElastiCache for Redis is supported; other AWS Redis databases are not supported. Depending on whether the PSYNC command is prefixed, there are different selections. For the PSYNC command with a prefix, select **AWS**. For the PSYNC command without a prefix, select **Others**.
Region	The region selected for the source database at the time of purchase cannot be modified.
Access Type	Select the network type through which the source database will connect to Tencent Cloud. Different access types require different network connectivity operations. For details, see Preparation Overview. If the source database is an IDC self-built database or a database from another cloud providers, the access methods available include Public Network/Direct Connect/VPN Access/CCN. Public Network: The source database can be accessed through a public IP. 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. If the source database is self-built on CVM, select Self-Build on CVM as the access method. If the source database is a TencentDB instance, select Database as the access method.
Node Type	**Access Type:** This parameter is displayed when an access type other than"Database" is selected. Choose according to your specific situation. Single-node migration: Refers to a cluster architecture with one primary node and multiple replicas. Cluster migration: Refers to a cluster architecture consisting of multiple shards, with each shard having several replica nodes. When cluster migration is selected, you will need to fill in the **Node Info**.

	Enter the addresses and passwords for all shard nodes in the source database cluster (IP:port: password or IP:port). For multiple nodes, enter each on a new line. If the source database is AWS, you need to enter the PSYNC name (IP:port:PSYNC name: password or IP:port:PSYNC name). The PSYNC name is the prefix used in the PSYNC command and should be requested from AWS. If the PSYNC command is not prefixed, it is necessary to set the preceding **Service Provider** parameter to **Others** and perform the migration as per an ordinary cluster. It is recommended to use the replica nodes (secondary nodes) of the source database. Migrating from the replica nodes can help avoid impacting the source database's business access.
Public Network	When Public Network is selected as the **Access Type**, the following parameters need to be configured. Host Address: The IP address or domain name of the source database. Port: The port used by the source database. Password: The password for the migration account of the source database.
Self-Build on CVM	When Self-Build on CVM is selected as the **Access Type**, the following parameters need to be configured. CVM instance: The instance ID of the CVM. Port: The port used by the source database. Password: The password for the migration account of the source database.
Direct Connect	When Direct Connect is selected as the **Access Type**, the following parameters need to be configured. VPC Direct Connect Gateway: Only VPC Direct Connect gateways are supported for Direct Connect. Ensure that the gateway is associated with the correct network type. VPC: Select the VPC and subnet. Host Address: The IP address of the source database. Port: The port used by the source database. Password: The password for the migration account of the source database.
VPN Access	When VPN Access is selected as the **Access Type**, the following parameters need to be configured. VPN Gateway: VPN Gateway. Select the VPN gateway instance that connects to the source database network. VPC: Select the VPC and subnet. Host Address: The IP address of the source database. Port: The port used by the source database. Password: The password for the migration account of the source database.
Database	When Database is selected as the **Access Type**, the following parameters need to be configured. Cross-account Same account: The source database instance and the target database instance belong to the same Tencent Cloud account.

	Cross-account: The source database and the target database instances belong to different Tencent Cloud accounts. For detailed instructions on cross-account operations, see Cloud Database Migration Across Accounts. Cloud Database Instance: Select the instance ID of the source database. Password: The password for the migration account of the source database.
CCN	When the **Access Type** is set to CCN, the following parameters need to be configured. When CCN is used for connection, it supports both same-account and cross-account CCN. Due to the complexity of network configurations, see Migrate Self-built Database to Tencent Cloud Database via CCN. Host network environment: Select according to your actual situation. If the source database is a TencentDB instance, select Tencent Cloud; if the source database is a self-built IDC database, select Self-built IDC; if the source database is from another cloud provider, select the corresponding network. Host Address: The IP address of the source database host. Port: The port used by the source database. Password: The password for the migration account of the source database. CCN instance account My Account: The CCN resources and DTS belong to the same Tencent Cloud root account. Other Account: The CCN resources and DTS belong to different Tencent Cloud root accounts. VPC CCN : The name of the CCN instance. VPC access: Select the VPC and subnet to access. The Access VPC refers to the VPC that connects to the DTS migration linkage within the CCN. Select a VPC other than the one to which the source database belongs from all the VPCs associated with the CCN. VPC access region: The region of the source database selected at the time of task purchase should match the region of the VPC you are connecting to. If they do not match, DTS will automatically change the region of the source database selected during task purchase to match the VPC region.

Target Database Settings

The target database parameter settings are similar to those of the source database. Only the differences are described below.

Configuration Parameter	Description
Access Type	Select based on your scene. The configuration principles for different access types are similar to those for the source database. Downward migration is supported. When the **Access Type** of the source database is set to Database, the target database can be selected as Public Network/Self-Build on CVM/Direct Connect/VPN Access/ CCN.
Node Type	This parameter appears when any **Access Type** other than "Database" is selected. Select based on the actual situation of the target database. Supported options include Single-node Migration, Redis Cluster Migration, and Proxy Cluster Migration.

Step 3: Configuring Migration Options and Selecting Migration Objects

In the **Set migration options and select migration objects** tab, configure the parameters and then click **Next**.

Basic Options

Configuration Item	Description
Migration Type	The default is full + incremental migration and cannot be modified. Full + incremental migration means migrating the historical data of the source database before the migration and the new data written to the source database during the migration.
Migration Object	The default is to select the entire instance and cannot be modified.
Target Database Writing Mode	Clear Target Instance: When the source database data is synchronized to the target database, the existing data in the target database will be cleared and then rewritten. Overwrite: When the source database data is synchronized to the target database, the keys from the source database will overwrite the keys in the target database.
Enable Target Database Read/Write	Allow DTS or other services to write data to the target database. This parameter needs to be enabled when using the task retry feature.
Automatic expiry of keys	When it is selected, expired keys in the target database will be automatically purged. This may result in inconsistencies between the data in the source and target databases, so proceed with caution.

Advanced Options

Configuration Item	Description
Limit Data Transfer Rate	This generally does not need to be set. However, if the DTS transfer rate is impacting the load on the target database, you can set a rate limit. Number of shards in the target database: If the target database is a TencentDB instance, the number of shards is automatically retrieved and cannot be modified. Traffic limit per shard in the target database: Enter the traffic limit per shard, with the setting range between 0-100 MB/s. If the target instance has a cluster architecture, then the maximum traffic per second for a single instance = number of shards * If the target instance has a cluster architecture, the maximum traffic per second for the entire instance is calculated as: number of shards x traffic limit per shard (MB/s).

Step 4: Validating and Starting the Task

1. On the task verification page, the system will automatically perform a pre-check. After the validation is successful, click **Start Task** to immediately enable the data migration. (If a scheduled execution time was set during task configuration, you can also select **Start Later**, and the task will be enabled at the scheduled time.)

Note:

If the validation fails, see Processing of Pre-verification Items Not Passed.

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.

2. 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.

Step 5: Ending the Task

1. If you need to view task progress details, delete the task, or perform other operations, click the corresponding button in the **Operation** column. For more details, see Task Management.

2. When the task status is Ready to complete, it indicates that the incremental synchronization is finished, and you can end the task. Click **Complete** in the **Operation** column to end the migration task.

It is recommended to end the task when the data difference between the target and source databases is 0 KB, and the time difference is 0 seconds.

If you need to perform a business cutover after the migration is completed, see Cutover Instructions.

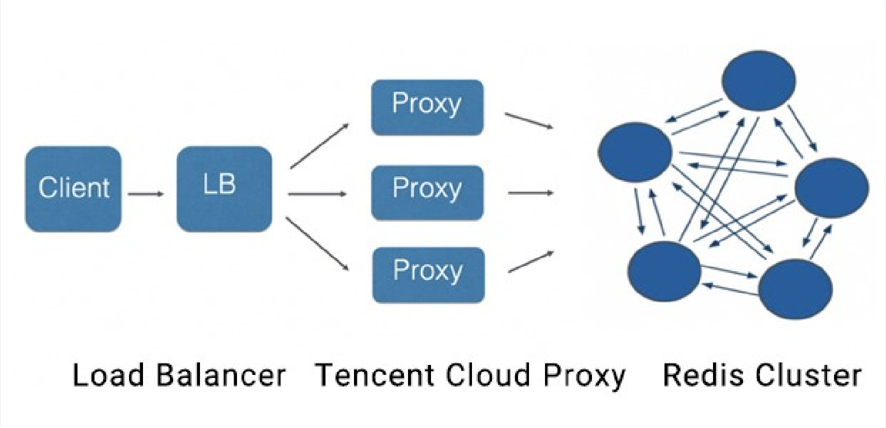
Check for Migration from Standard Architecture to Cluster Architecture

Last updated：2024-09-10 17:48:04

Standard Edition can be your self-created Redis Standalone Edition, master/replica mode, or TencentDB for Redis Memory Edition (Standard Architecture). This document describes the compatibility issues in migrating data from Redis Standard Edition to TencentDB for Redis Memory Edition (Cluster Architecture).

Compatibility Description

TencentDB for Redis Memory Edition (Cluster Architecture) adopts the cluster architecture consisting of Tencent Cloud's proprietary proxy and Redis Community Cluster Edition, which is 100% compatible with Redis Community Cluster Edition commands.



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). You need to pay attention to the following usage specification issues:

Multikey operation

TencentDB for Redis Memory Edition (Cluster Architecture) uses the hash algorithm to distribute keys to 16,384 slots. For more information on the principle, please see Redis Cluster Specification.

Redis Community Cluster Edition: it does not support any cross-slot multi-key access commands.

TencentDB for Redis Memory Edition (Cluster Architecture): it supports cross-slot multikey access of the `MGET` , `MSET` , and `DEL` commands. This mainly works by using Tencent Cloud's proprietary proxy to implement aggregated command computing among multiple nodes.

Hash tag: in your business, 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.

Cross-slot command list:

Command Group	Command	Cross-slot support in Memory Edition (Cluster Architecture)
Keys Group	del	✓
	exists	✓
	rename	x
	renamenx	x
	unlink	x
List Group	rpoplpush	x
	blpop	x
	brpop	x
	brpoplpush	x
Sets Group	sdiff	x
	sdiffstore	x
	sinter	x
	sinterstore	x
	smove	x
	sunion	x
	sunionstore	x
Sorted Sets Group	zinterstore	x
	zunionstore	x
Strings Group	bitop	x
	mget	✓
	mset	✓
	msetnx	x
Hyperloglog Group	pfcount	x

	pfmerge	x
Scripting Group	eval	x
	evalsha	x
	script exists	x
Stream Group	xread	x
	xreadgroup	x

Support for Lua

Memory Edition (Cluster Architecture) supports Lua commands, but cross-slot access to keys in Lua scripts is not supported.

The `Key` parameter must be passed in for the `EVAL` and `EVALSHA` commands; otherwise, they cannot be executed.

The subcommands `LOAD` , `FLUSH` , `KILL` , and `EXIST` of `SCRIPT` will be distributed to all master nodes in the cluster through the proxy.

```
> eval "return {KEYS[1],KEYS[2],ARGV[1],ARGV[2]}" 2 key1 key2 first second
1) "key1"
2) "key2"
3) "first"
4) "second"
```

Note：

The `key1` and `key2` parameters must be passed in when you use Lua.

Transaction support

Memory Edition (Cluster Architecture) supports transactions, but cross-slot access to keys in transactions is not supported.

You need to first run the `watch key` command and then the `multi` and `exec` commands in the current version. This operation will be optimized in future versions to eliminate need to run `watch key` first.

Custom commands

Through VIP encapsulation, TencentDB for Redis Memory Edition (cluster architecture) provides a user experience in cluster mode comparable to the standard edition, making it much easier for use in different scenarios. To increase the transparency to OPS, custom commands can be used. Access to each node in the cluster is supported by adding a parameter "node ID" on the right of the original command parameter list, such as `COMMAND arg1 arg2 ... [node ID]` . The node ID can be obtained through the `cluster nodes` command or in the console.

```
10.1.1.1:2000> cluster nodes25b21f1836026bd49c52b2d10e09fbf8c6aa1fdc
10.0.0.15:6379@11896 slave 36034e645951464098f40d339386e9d51a9d7e77 0
1531471918205 1 connectedda6041781b5d7fe21404811d430cdffea2bf84de
10.0.0.15:6379@11170 master - 0 1531471916000 2 connected 10923-
1638336034e645951464098f40d339386e9d51a9d7e77 10.0.0.15:6379@11541
myself,master - 0 1531471915000 1 connected 0-
546053f552fd8e43112ae68b10dada69d3af77c33649 10.0.0.15:6379@11681 slave
da6041781b5d7fe21404811d430cdffea2bf84de 0 1531471917204 3
connected18090a0e57cf359f9f8c8c516aa62a811c0f0f0a 10.0.0.15:6379@11428 slave
ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 0 1531471917000 2
connectedef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 10.0.0.15:6379@11324 master -
0 1531471916204 0 connected 5461-10922

Native command: `info server`
Custom command:
info server ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171SCAN
Sample:
scan 0 238b45926a528c85f40ae89d6779c802eaa394a2
scan 0 match a* 238b45926a528c85f40ae89d6779c802eaa394a2KEYS
Sample:
keys a* 238b45926a528c85f40ae89d6779c802eaa394a2
```

Client access method

We recommend you use a Standard Edition (e.g., Jedis but not JedisCluster) client to access TencentDB for Redis Memory Edition (Cluster Architecture), as this access method is more efficient and simpler. You can also access through cluster clients, such as JedisCluster.

Codis compatibility

TencentDB for Redis Memory Edition (Cluster Architecture) is 100% compatible with Codis Server commands with no modification to your business required. You can use DTS to quickly migrate data to TencentDB for Redis, which has the following advantages over Codis:

Compatibility with more versions. Codis supports only Redis 3.2 or below, while TencentDB for Redis Memory Edition (Cluster Architecture) supports Redis 4.0 and 5.0 and will be continuously updated in sync with the Redis Community.

TencentDB for Redis Cluster Edition is compatible with more commands. Codis does not support blocking commands such as `BLPOP` and `SUBSCRIBE` .

If a big key occurs in data migration with Codis, the service may become unavailable. In contrast, TencentDB for Redis supports lossless expansion with no fear for big keys.

Compatibility Check

Currently, no tools can be used to exactly determine whether there will be compatibility problems in data migration from Standard Edition to Cluster Edition. You can use the following two tools to evaluate the compatibility before migration. We recommend you perform static evaluation, dynamic evaluation, and business verification before migration to ensure that data can be smoothly migrated.

Static evaluation

1. Download the cluster_migrate_online_check.py static verification tool and use it to run the `info commandstats` command to check whether Standard Edition has ever executed cross-slot commands in order to determine whether there is compatibility problem.

```
Usage:
./cluster_migrate_check.py host port password
```

Note：

Enter the corresponding Redis Standard Edition information for `host` , `port` , and `password` .

2. Check whether each item can pass as instructed in Compatibility Description above.

Dynamic evaluation

Download the cluster_migrate_online_check dynamic verification tool and use it to simulate the execution of the `psync` command on the client so as to sync incremental data from Standard Edition to the TencentDB for Redis Memory Edition (Cluster Architecture) in real time. By performing real-time sync, you can check whether there is compatibility problem in write commands. This tool cannot test the compatibility of read commands.

Note：

The psync command for TencentDB for Redis requires submission of a submitting a ticket for application.

The proxy version of TencentDB for Redis needs to be later than or equal to 5.6.5 to be compatible with the Redis config set command. For versions earlier than 5.6.5, the proxy should be upgraded first.

The steps are as follows:

1. Activate TencentDB for Redis Memory Edition (Cluster Architecture) in the console.

2. Use the tool to sync data from Standard Edition to TencentDB for Redis Memory Edition (Cluster Architecture) in real time.

3. After a period of verification (such as 6 or 24 hours), if the tool does not report any errors, the write commands do not have compatibility problems; otherwise, you can get the information of incompatible commands in the error message.

```
Usage:
./cluster_migrate_online_check srcip:srcport srcpasswd dstip:dstport dstpasswd
Environment variable parameters:
export logout=1  // It is used to print command in the console, which is
disabled by default
```

```
export pipeline = 2000  // Number of concurrent pipelines, which is 1,000 by
default
```

Note：

`srcip:srcport` : Redis Standard Edition address information, which is required.

`dstip:dstport` : TencentDB for Redis Memory Edition (Cluster Architecture) address information, which is optional. If it is left empty, the tool can be used as a monitor.

4. Check whether each item can pass as instructed in Compatibility Description above.

Business verification

To ensure successful data migration, we recommend you test the business in the test environment. You can connect the business in the testing environment to the TencentDB for Redis Memory Edition (Cluster Architecture) and confirm whether all features can work properly before data migration.

Self-Created Instance Migration Failure

The `client-output-buffer-limit` parameter value is too small. You are recommended to set it to 512 MB or 1,024 MB by running the following command:

```
config set client-output-buffer-limit "slave 1073741824 1073741824 600"
```

Parameters have not been passed in for the `EVAL` command.

Key Prefix Filtering

Last updated：2024-11-26 11:24:29

Overview

DTS supports configuring prefix filtering rules for multi-key commands in Redis data migration linkages, making it easier for users to manage keys in their databases.

Key Prefix Filtering Rules

Supports configuring single or multiple key prefix matching rules, with multiple prefixes separated by the Enter key. When migration rules and filtering rules are used together, filtering rules take precedence over migration rules.

Key Prefix Filtering Rule Examples

1. MSET/DEL/UNLINK commands use exact matching rules.
Migration rule: If a multi-key command on the source matches a single key, only that key is migrated to the target; other keys are not migrated.
Filtering rule: If a multi-key command on the source matches a single key, only that key is filtered and not migrated to the target; other keys will still be migrated.

Prefix Type	Source Database Key Command	Prefix Set in DTS	Result on Target
Migration key prefix	MSET dtsnew "111" abcd "222"	dts	MSET dtsnew "111"
Filtering key prefix	MSET redisnew "111" efgh "222"	redis	MSET efgh "222"

2. Commands other than MSET/DEL/UNLINK.
Migration rule: If a multi-key command on the source matches any migration prefix, the entire command will be migrated to the target.
Filtering rule: A multi-key command on the source will only be filtered if all keys match the filter prefix. If only one key matches, the command will not be filtered and will still migrate to the target.

Prefix Type	Source Database Key	Prefix Set in DTS	Result on Target

	Command		
Migration key prefix	SUNION dtsnew abcd	dts	SUNION dtsnew abcd
Filtering key prefix	SUNION redisnew efgh	redis	SUNION redisnew efgh
Filtering key prefix	SUNION redisnew efgh	redis efg	Unsynced

Directions

To configure a Redis migration task, go to the **Set migration options and select migration objects** page, check **Enable Key Filter**, set the filtering rules, and proceed with the subsequent configuration steps.

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.

Handling of Common Errors

Last updated：2024-09-10 17:49:04

Issue 1

Description

The following error message is displayed during migration with DTS:

```
[launch]state:6 #rdb rdbfile:./tmp1600869159_89068.rdb rdbsize:2753701723
rdb_writed_size:1606959104 rdb_parsed_size:0 rdb_parsed_begin:0
rdb_parsed_time:0 #replication
master_replid:0549e2f0bdf373cef0c4c89bb0ce9e1757c4b105
repl_offset:1327777565448 write_command_count:0 finish_command_count:0
last_replack_time:0 #queue send_write_pos:0 send_read_pos:0
response_write_pos:0 response_read_pos:0 errtime:1600870264 errmsg:read rdb eof
save rdb fail ready shutdown dts
```

Causes

Check the log of the source Redis database. If it contains the following message, the `client-output-buffer-limit` configured for the source database is exceeded.

```
psync scheduled to be closed ASAP for overcoming of output buffer limits
```

Solution

Run the following command to set the `client-output-buffer-limit` to infinite and initiate the DTS task again.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

Issue 2

Description

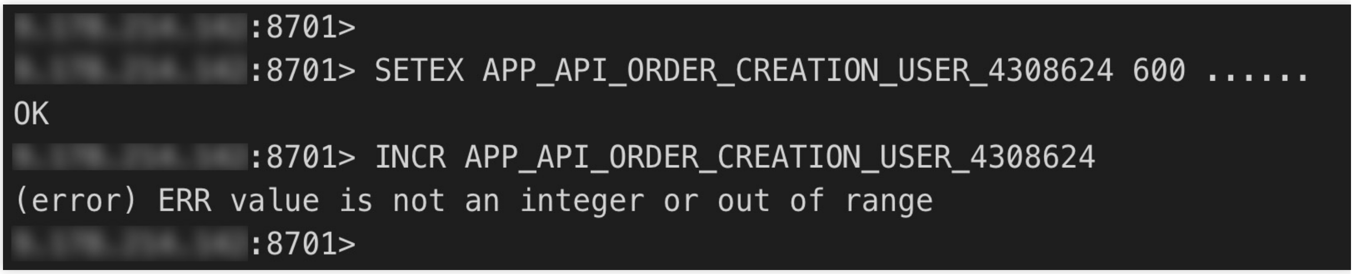
The following error message is displayed during migration with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1600395232_34851.rdb rdbsize:107994104
rdb_writed_size:107994104 rdb_parsed_size:107994104 rdb_parsed_begin:1600395238
rdb_parsed_time:5 #replication
```

```
master_replid:995dba8ccffb7cc32a7c85de7b1632b952b74496 repl_offset:23851025
write_command_count:940765 finish_command_count:940763
last_replack_time:1600395298 #queue send_write_pos:440766 send_read_pos:440765
response_write_pos:440765 response_read_pos:440764 errtime:1600395297
errmsg:get rsp error:ERR value is not an integer or out of range command:*2 $4
INCR $35 APP_API_ORDER_CREATION_USER_4260882
```

Causes

By capturing packets on two DTS Syncer instances in the region, it was found that the value of the key was characters rather than a number, causing the INCR execution to fail.



```
[redacted]:8701>
[redacted]:8701> SETEX APP_API_ORDER_CREATION_USER_4308624 600 ......
OK
[redacted]:8701> INCR APP_API_ORDER_CREATION_USER_4308624
(error) ERR value is not an integer or out of range
[redacted]:8701>
```

Solution

Delete the relevant key and initiate the DTS migration again.

Issue 3

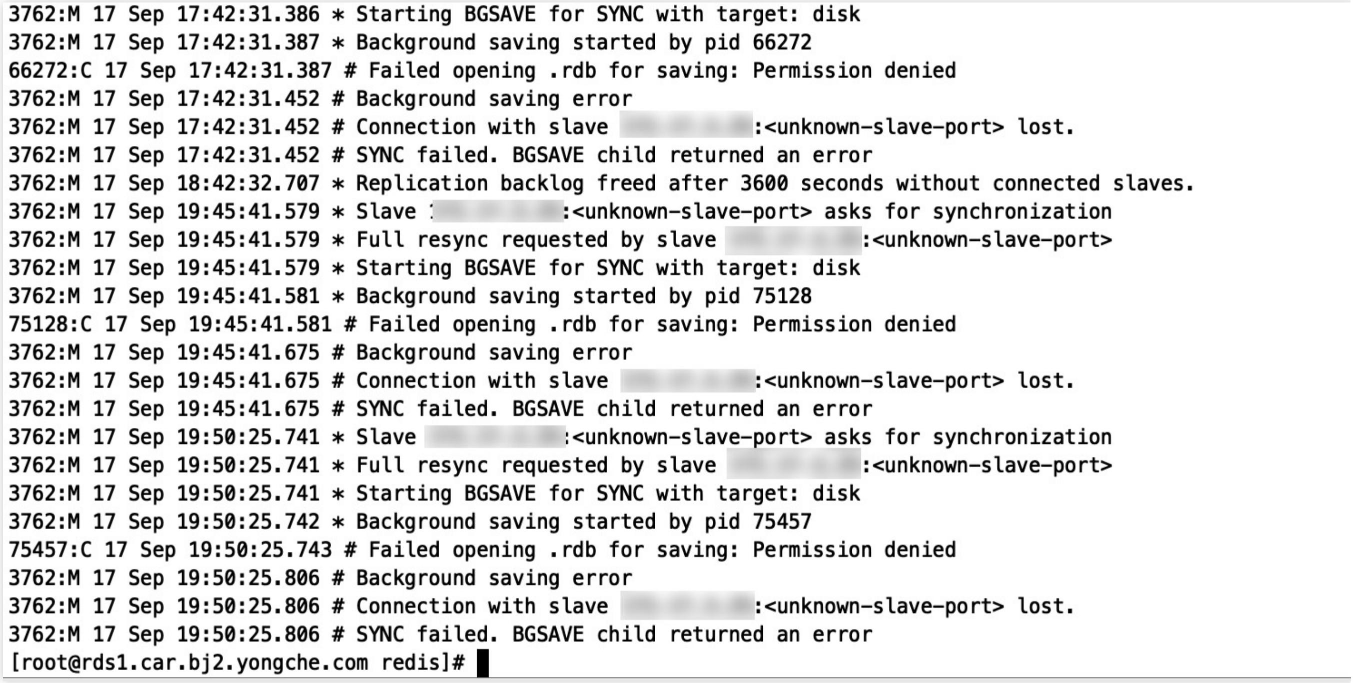
Description

The following error message is displayed during migration with DTS:

```
errmsg:Error reading bulk length while SYNCing:Operation now in progress read
rdb length from src fail save rdb fail ready shutdown dts
```

Causes

The error message of the source instance reveals that the RDB file does not have permission to access the directory.



```
3762:M 17 Sep 17:42:31.386 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 17:42:31.387 * Background saving started by pid 66272
66272:C 17 Sep 17:42:31.387 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 17:42:31.452 # Background saving error
3762:M 17 Sep 17:42:31.452 # Connection with slave [redacted]:<unknown-slave-port> lost.
3762:M 17 Sep 17:42:31.452 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 18:42:32.707 * Replication backlog freed after 3600 seconds without connected slaves.
3762:M 17 Sep 19:45:41.579 * Slave [redacted]:<unknown-slave-port> asks for synchronization
3762:M 17 Sep 19:45:41.579 * Full resync requested by slave [redacted]:<unknown-slave-port>
3762:M 17 Sep 19:45:41.579 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:45:41.581 * Background saving started by pid 75128
75128:C 17 Sep 19:45:41.581 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:45:41.675 # Background saving error
3762:M 17 Sep 19:45:41.675 # Connection with slave [redacted]:<unknown-slave-port> lost.
3762:M 17 Sep 19:45:41.675 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 19:50:25.741 * Slave [redacted]:<unknown-slave-port> asks for synchronization
3762:M 17 Sep 19:50:25.741 * Full resync requested by slave [redacted]:<unknown-slave-port>
3762:M 17 Sep 19:50:25.741 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:50:25.742 * Background saving started by pid 75457
75457:C 17 Sep 19:50:25.743 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:50:25.806 # Background saving error
3762:M 17 Sep 19:50:25.806 # Connection with slave [redacted]:<unknown-slave-port> lost.
3762:M 17 Sep 19:50:25.806 # SYNC failed. BGSAVE child returned an error
[root@rds1.car.bj2.yongche.com redis]#
```

Solution

Run the following command to set diskless replication and initiate the DTS task again.

```
config set repl-diskless-sync yes
```

Issue 4

Description

The following error message is displayed during migration with DTS:

```
[launch]state:6 #rdb rdbfile:./tmp1597977351_20216.rdb rdbsize:24282193511
rdb_writed_size:18683334200 rdb_parsed_size:0 rdb_parsed_begin:0
rdb_parsed_time:0 #replication
master_replid:1b0da9f595cc40b795803eba3c9bea3aad1a1d68 repl_offset:921330115650
write_command_count:0 finish_command_count:0 last_replack_time:0 #queue
send_write_pos:0 send_read_pos:0 response_write_pos:0 response_read_pos:0
errtime:1597978778 errmsg:write rdb data fail:456!=1696 error:No space left on
device save rdb fail ready shutdown dts
```

Causes

The disk space in the DTS Syncer instance is insufficient.

Solution

Clear the disk in the DTS Syncer instance or mount a new disk and then initiate the DTS task again.

Issue 5

Description

The following error message is displayed during migration with DTS:

```
[launch]state:4/5 #rdb rdbfile: rdbsize:0 rdb_writed_size:0 rdb_parsed_size:0
rdb_parsed_begin:0 rdb_parsed_time:0 #replication
master_replid:d3e707ec0e72c3908b0ce70dd2460f48086c5386 repl_offset:683087907631
write_command_count:0 finish_command_count:0 last_replack_time:0 #queue
send_write_pos:0 send_read_pos:0 response_write_pos:0 response_read_pos:0
errtime:1654369638 errmsg:Error reading bulk length while SYNCing:Operation now
in progress read rdb length from src fail save rdb fail ready shutdown dts
```

Causes

The issue is caused by insufficient memory on the source, leading to RDB generation failure, or an unstable network.

Solution

1. Adjust the kernel connection limits and buffer limits on the source.

Disconnect the source database and adjust the limit on the connections to the source system kernel.

```
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
```

Execute the following command to set the source's client-output-buffer-limit to unlimited.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

2. If the issue persists, increase the memory on the machine where the source node is located to ensure that the node has sufficient resources to generate the RDB file.

Issue 6

Description

The following error message is displayed during migration from Redis Memory Edition (Standard Architecture) to Cluster Architecture with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1645683629_34614.rdb rdbsize:781035471
rdb_writed_size:781035471 rdb_parsed_size:781035471 rdb_parsed_begin:1645683632
rdb_parsed_time:25 #replication
```

```
master_replid:5abe7987b1e263582c68835412d2963eeb0a3d60 repl_offset:895807918761
write_command_count:6102523 finish_command_count:6102137
last_replack_time:1645683656 #queue send_write_pos:101832 send_read_pos:101742
response_write_pos:101742 response_read_pos:101357 errtime:1645683657
errmsg:get rsp error:CROSSSLOT Keys in request don't hash to the same slot
command:*3 $6 RENAME $16 dispatch:km:pool $34 dispatch:km:tmp-
pool:1645683651224 ready shutdown dts send replconf ack to src fail:Bad file
descriptor
```

Causes

The database involves multi-key, transactional, or cross-slot operations. For more information, see Check on Migration from Standard Architecture to Cluster Architecture.

Solution

Migrate the data to a Standard Architecture instance in the cloud, or change the business logic to clear multi-key operations.

Issue 7

Description

The following error message is displayed during migration with DTS:

```
[launch]state:7 #rdb rdbfile:./tmp1633836033_79441.rdb rdbsize:1008499748
rdb_writed_size:1008499748 rdb_parsed_size:607311937
rdb_parsed_begin:1633836038 rdb_parsed_time:0 #replication
master_replid:d42935b9537b1d76ddd9e99e7cb8d4bc22a3e0c3
repl_offset:4649070152868 write_command_count:1569934
finish_command_count:1546843 last_replack_time:1633836088 #queue
send_write_pos:69933 send_read_pos:69934 response_write_pos:69934
response_read_pos:46843 errtime:1633836089 errmsg:send replconf ack to src
fail:Connection reset by peer rdb parse error: Wrong RDB checksum rdb load fail
ready shutdown dts
```

You can see the following information in the execution log of the source node:

```
44:M 05 Jun 03:31:06.728 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:31:06.978 * Background saving started by pid 89
89:C 05 Jun 03:32:08.417 # Error moving temp DB file temp-89.rdb on the final
destination 20617.20324.rdb (in server root dir /opt/data/dump): No such file
or directory
44:M 05 Jun 03:32:08.698 # Background saving error
```

```
44:M 05 Jun 03:32:08.698 # Connection with slave 10.xx.xx.119:<unknown-slave-port> lost.
44:M 05 Jun 03:32:08.698 # SYNC failed. BGSAVE child returned an error
44:M 05 Jun 03:50:24.626 * Slave 10.xx.xx.119:<unknown-slave-port> asks for synchronization
44:M 05 Jun 03:50:24.626 * Full resync requested by slave 10.xx.xx.119:<unknown-slave-port>
44:M 05 Jun 03:50:24.626 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:50:24.880 * Background saving started by pid 90
90:C 05 Jun 03:51:22.585 * DB saved on disk
90:C 05 Jun 03:51:22.739 * RDB: 280 MB of memory used by copy-on-write
44:M 05 Jun 03:51:23.008 * Background saving terminated with success
44:M 05 Jun 03:51:27.898 * Synchronization with slave 10.xx.xx.119:<unknown-slave-port> succeeded
44:M 05 Jun 03:52:19.531 # Connection with slave client id #317862457 lost.
```

Causes

This is often because the connection of the DTS task to the source node timed out due to network environment issues, big keys contained in the database, or `client-output-buffer-limit` overflows on the source node.

Solution

Check the source network environment for any issue. For detailed directions, see Database Connection Check.

Clear big keys in the source database. You can quickly locate, assess, and delete big keys as instructed in Memory Analysis.

Run the following command to set the `client-output-buffer-limit` to infinite on the source node.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

Issue 8

Description

The following error message is displayed during migration with DTS:

```
[launch]state:7 #rdb rdbfile:./tmp1654365384_70581.rdb rdbsize:1664871634
rdb_writed_size:1664871634 rdb_parsed_size:1266531 rdb_parsed_begin:1654365387
rdb_parsed_time:0 #replication
master_replid:d3e707ec0e72c3908b0ce70dd2460f48086c5386 repl_offset:683001122815
write_command_count:17818 finish_command_count:11224 last_replack_time:0 #queue
send_write_pos:30251 send_read_pos:17767 response_write_pos:17767
response_read_pos:11213 errtime:1654365387 errmsg:rdb parse error: Short read
or OOM loading DB. Unrecoverable error rdb load fail ready shutdown dts
```

Causes

If this error message is displayed when you retry a failed DTS task, it generally indicates that the target node is not empty or the memory is full.

Solution

Clear the target node and try again, or submit a ticket to apply for DTS overwrite.

Issue 9

Description

The following error message is displayed during migration with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1653290250_19158.rdb rdbsize:1721160435
rdb_writed_size:1721160435 rdb_parsed_size:1721160435
rdb_parsed_begin:1653290255 rdb_parsed_time:124 #replication
master_replid:ed87c56060bc5f9b28da6d7ef2f83a15d56a4827 repl_offset:239048673513
write_command_count:360526495 finish_command_count:360520725
last_replack_time:1654350624 #queue send_write_pos:406694 send_read_pos:406694
response_write_pos:406694 response_read_pos:400925 errtime:1654350625
errmsg:redisBufferRead read rsp from target fail:1:Connection reset by peer
ready shutdown dts send replconf ack to src fail:Bad file descriptor
```

Causes

The Redis node of the target instance experienced an HA master/replica switch, or the proxy node experienced a failover, causing the sync task to fail.

Solution

Create a new DTS task and configure the new node after the HA switch as the target node for data migration.

Issue 10

Description

When the memory eviction policy of the target instance is set to `allkey-lru` during migration with DTS, the following error message is displayed:

```
[launch]state:8 #rdb rdbfile:./tmp1638263556_29975.rdb rdbsize:597343276
rdb_writed_size:597343276 rdb_parsed_size:428299275 rdb_parsed_begin:1638263575
```

```
rdb_parsed_time:7 #replication
master_replid:ae0dfc45f72f3ee8642c8e31e493b6442179734f repl_offset:34832262785
write_command_count:6811 finish_command_count:6798 last_replack_time:1638263582
#queue send_write_pos:6811 send_read_pos:6811 response_write_pos:6811
response_read_pos:6799 errtime:1638263583 errmsg:get rsp error:OOM command not
allowed when used memory > 'maxmemory'. command:*3 $3 SET $26
all_business_newmikoxmsong $1189783
[{"id":3,"label":"AI\\u5e73\\u53f0\\u90e8","node_level":0,"leaf":false,"childre
n":[{"id":"887381","label":"[OMG][\\u4f53\\u80b2\\u641c\\u7d22]
[CMDB]","node_level":1,"leaf":false,"children":[{"id":"722605","label":"
[\\u4e2a\\u6027\\u5316\\u63a8\\u8350\\u4e2d\\u5fc3]
[\\u817e\\u8baf\\u7f51\\u4f53\\u80b2APP\\u63a8\\u8350]","node_level":2,"leaf":f
alse,"children":[],"collet":0}],"collet":0},{"id":"460871","label":"[OMG]
[\\u817e\\u8baf\\u89c6\\u9891\\u641c\\u7d22]
[CMDB]","node_level":1,"leaf":false,"children":[{"id":"383393","label":"
[\\u641c\\u7d22\\u4e1a\\u52a1\\u4e2d\\u5fc3]
[\\u817e\\u8baf\\u89c6\\u9891\\u641c\\u7d22]","node_level":2,"leaf":false,"chil
dren":[],"collet":0}],"collet":0}]},
{"id":8,"label":"IDC\\u5e73\\u53f0\\u90e8","node_level":0,"leaf":false,"childre
n":
[{"id":"452519","label":"IDC\\u5e73\\u53f0\\u90e8\\u81ea\\u7528[CMDB]","node_le
vel":1,"leaf":false,"children":
[{"id":"453099","label":"IDC\\u7cfb\\u7edf\\u5f00\\u53d1","node_level":2,"leaf"
:false,"children":[],"collet":0}],"collet":0}]},{"id":14 ready shutdown dts
send replconf ack to src fail:Bad file descriptor
```

Causes

The memory of the target instance is smaller than the memory used by the data to be migrated from the source database.

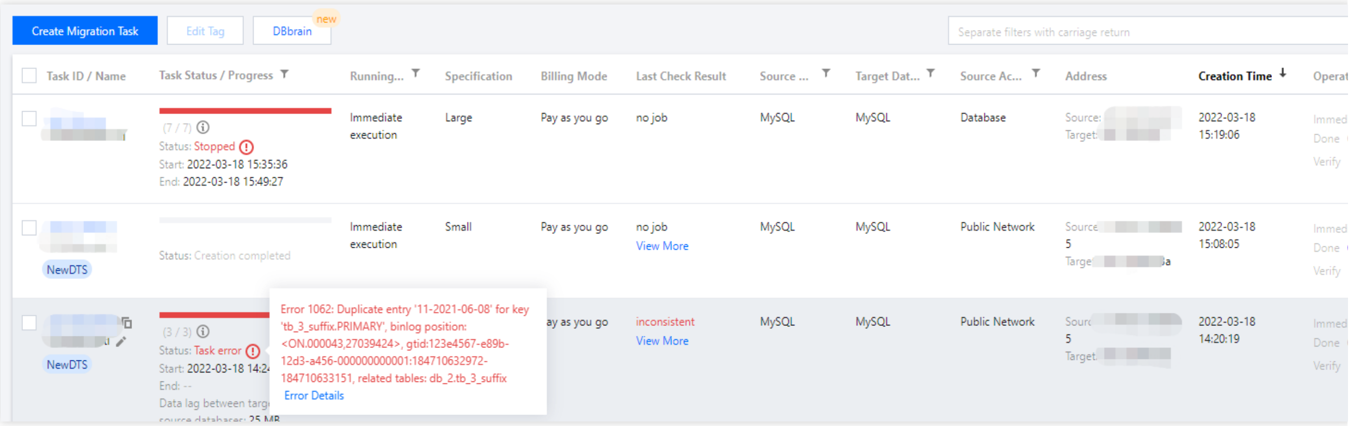
Solution

Expand the memory of the target instance and then initiate a new DTS migration task. For detailed directions, see Changing Instance Specification.

Issue 11

Description

The following error message is displayed during migration with DTS:



□ Task ID / Name	Task Status / Progress	Running...	Specification	Billing Mode	Last Check Result	Source ...	Target Dat...	Source Ac...	Address	Creation Time	Operat
□ [redacted]	(7 / 7) Status: Stopped Start: 2022-03-18 15:35:36 End: 2022-03-18 15:49:27	Immediate execution	Large	Pay as you go	no job	MySQL	MySQL	Database	Source: [redacted] Target: [redacted]	2022-03-18 15:19:06	Immed Done Verify
□ [redacted] NewDTS	Status: Creation completed	Immediate execution	Small	Pay as you go	no job View More	MySQL	MySQL	Public Network	Source [redacted] 5 Targe [redacted] a	2022-03-18 15:08:05	Immed Done Verify
□ [redacted] NewDTS	(3 / 3) Status: Task error Start: 2022-03-18 14:2 End: -- Data lag between targ source databases: 25 MB			ay as you go	inconsistent View More	MySQL	MySQL	Public Network	Sourc [redacted] 5 Target [redacted]	2022-03-18 14:20:19	Immed Done Verify

Error 1062: Duplicate entry '11-2021-06-08' for key 'tb_3_suffix.PRIMARY', binlog position: <ON.000043,27039424>, gtid:123e4567-e89b-12d3-a456-000000000001:184710632972-184710633151, related tables: db_2.tb_3_suffix
Error Details

Causes

If an error is reported when you start a DTS migration task with a proxy, it is usually because the bandwidth of the proxy is insufficient.

Solution

Expand the bandwidth of the proxy or perform migration tasks serially in sequence. For detailed directions, see Bandwidth Adjustment.

Issue 12

Description

The following error message is displayed during migration with DTS:

```
[launch]SrcInstance nodes has changed.
```

Causes

The source node experienced an HA master/replica switch, causing the DTS task sync to fail.

Solution

Create a new DTS task and configure the new node after the HA switch as the target node for data migration.

Migrating to KeeWiDB

Migration from Redis to KeeWiDB

Supported Capabilities

Last updated：2024-09-10 17:17:37

Supported Scenarios and Versions

Source	Target	Description
Self-hosted Redis database (IDC self-hosted/CVM self-hosted) 4.0	Tencent Cloud KeeWiDB 4.0 and earlier (For other versions, submit a ticket for application.)	Migrate on-premises databases to the cloud.
Other Third-Party Cloud Providers (all) Redis 4.0	Tencent Cloud KeeWiDB 4.0 and earlier (For other versions, submit a ticket for application.)	Migrate Redis databases from other cloud providers to TencentDB instances.
Tencent Cloud Redis 4.0	Tencent Cloud KeeWiDB 4.0 and earlier (For other versions, submit a ticket for application.)	Migration within the same region or between different regions Cross-version migration Migration within the same Tencent Cloud root account or between different root accounts Migration between cluster architectures. When migrating from a standard architecture to a cluster architecture, perform the Standard Architecture to Cluster Architecture Check before proceeding.

Supported Capabilities

Feature Category	Feature Sub-Item or Description	Supported Capabilities
Migration Object	-	Databases, collections

Migration Type	-	Only supports Full + Incremental Migration.
Resumable Transmission	-	Not supported
Key Task Management Operations	Retry	Supported
	Speed limit	Not supported
	Create similar task	Supported
Topology synchronization	One-to-one	Supported
	Bidirectional synchronization	Not supported

User Guide

Last updated：2024-09-10 17:18:36

Impact on Source Database

When DTS performs full data migration, it will consume some resources from the source instance, which may result in increased load on the source instance and added pressure on the database. If your database configuration is low, it is recommended to carry out the migration during off-peak business hours.

Migration Restrictions and Notes

1. In TencentDB for KeeWiDB, the string length cannot exceed 16 MB. This limitation applies not only to the Key and Value of the String type but also to each element within composite data structures such as Hash, Set, Zset, and List. If this limit is exceeded, an error will occur during migration. Perform a self-check before migration.
2. DTS tasks cannot form a circular synchronization loop. Data written in a circular synchronization loop will not be terminated and will continue to execute in the loop, leading to a write storm.
3. DTS does not support resuming from breakpoints. Since there are no Binlog files, all incremental data is synchronized in the connected memory. If the connection is interrupted, the incremental data is lost from memory. Therefore, once a task is initiated, it cannot be paused or interrupted.
4. Migration of Modules is not supported.
5. Do not perform the following operations during migration; otherwise, the migration task will fail.

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

Do not delete the target database during data migration.

Do not make changes to the target data during data migration to avoid inconsistencies in the final results.

Operation Guide

Last updated：2024-09-11 12:46:02

Operation Scenarios

DTS-based Redis data migration supports both full and incremental data migration, allowing you to migrate the source database's historical data as well as any new data written to the source during the migration process.

This document provides guidance on using the DTS data migration feature to migrate data from Redis to TencentDB for KeeWiDB.

Preparations

1. Establish the access channel between DTS and the database in advance according to the access type you intend to use. For detailed instructions, see Network Preparation Work.

IDC Self-hosted database/Other cloud provider database: The available access methods include Public Network/Direct Connect/VPN Access/CCN.

When migrating an external instance, you should ensure that the source instance is accessible in the public network environment and that the connection remains stable. Any network fluctuations or failures may result in migration failure. If the migration fails, you will need to restart the migration task.

Self-hosted database on CVM: Select Self-Build on CVM as the access method.

TencentDB instance: Select Database as the access method.

2. Pre-migration check

The following checks need to be performed by the user before migration; otherwise, migration failure may occur.

2.1 Check for large Keys on the source

During migration, large keys may cause the client-output-buffer-limit buffer to overflow, leading to migration failure.

For TencentDB instances, use the diagnostic and optimization features of TencentDB for DBbrain (DBbrain) to quickly analyze large keys. For detailed instructions, see Memory Analysis.

For non-TencentDB databases, use rdbtools to analyze large Redis keys.

Evaluate large keys and consider splitting or cleaning them. If you choose to retain large keys, set the source buffer size (client-output-buffer-limit) to unlimited.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

2.2 Check the TCP connection limit of the source Linux Kernel

If your business has a high number of concurrent requests, check the Linux kernel's connection limit before migration. If the number of business request connections exceeds the kernel's connection limit, the Linux server will proactively 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
```

2.3 Check access permissions for the source RDB file directory

Before migration, ensure that the directory storing the RDB files on the source is readable. If the directory is not readable, the migration will fail due to the inability to access the RDB files.

If the RDB file directory is not readable, execute the following command on the source to set up diskless replication, which allows the RDB file to be sent directly to DTS without needing to be stored on the source disk first.

```
config set repl-diskless-sync yes
```

2.4 (Optional) For scenes involving migration from a standard architecture to a cluster architecture, check for command compatibility issues. For details, see Check on Migration from Standard Architecture to Cluster Architecture to perform both static and dynamic evaluations.

Operation Steps

Stepl: Creating a Migration Task

1. Log in to the DTS Console, navigate to the **Data Migration** page, and then click **Create Migration Task** on the right side.

2. On the Create Migration Task page, select the task creation parameters, and then click **Buy Now**.

Configuration Parameter	Description
Service Type	Select **Data Migration**.
Creation Mode	Create a task: Create a brand new task. Create a similar task: Quickly create a task with the same configuration as a previous task. The new task will have pre-filled options consistent with the previous task, such as database type, access method, billing mode, and migration type. Users can modify these settings as needed based on their requirements.
Billing Mode	Only **pay-as-you-go** is supported.
Source Instance Type	Select the type of your source database. After purchase, this cannot be changed. Select **Redis** here.
Source	Select the region where the source database is located. If the source is a self-hosted database,

Instance Region	select the region closest to the self-hosted database.
Target Instance Type	Select the type of your target database. After purchase, this cannot be changed. Select **KeeWiDB** here.
Target Instance Region	Select the region where the target database is located.
Version	Currently, only **NewDTS** is supported.
Specification	Currently, only the **Xlarge** specification is supported.
Tags	Set tag keys and tag values for the migration task to efficiently manage multiple tasks using tags.
Task Name	Select **Name After Creation**: The task name will be the same as the Task ID by default. After the migration task is created, you can rename the task. Select **Name Now**: Enter the task name in the input box below.
Terms and Conditions	Make sure to check the box I have read and agree <Cloud Database Service Terms>.
Quantity	A maximum of 10 migration tasks can be purchased at one time.

3. After the purchase is completed, the page will automatically redirect to the Data Migration Task List.

Step II: Configuring the Source and Target Databases

1. In the Data Migration Task List, find the task you just created and click **Configure** in the **Operation** column.
2. Enter the **Set source and target databases** tab, fill in the details for both the source and target databases, then click **Test Connectivity**. If the test passes, click **Save** to proceed to the next step.

Task Configuration

Configuration Parameter	Description
Task Name	Set a name with business significance for easy task identification.
Running Mode	Immediate execution: The task will start immediately after the pre-check passes. Scheduled execution: Set a specific time for the task to start. The task will not start after the pre-check passes but will start at the scheduled time.
Automatic Retry	After this option is selected, if the migration task is interrupted due to network anomalies or other issues, DTS will automatically retry within the specified time range without requiring user

	intervention. The retry mechanism restarts the data migration from the beginning. When retrying, any data that was previously migrated to the target will be cleared or overwritten.

Source Database Settings

Configuration Parameter	Description
Source Database Type	The selected source database type at the time of purchase cannot be modified.
Service Provider	If the source database is an IDC self-hosted database, a self-hosted database on CVM, or a TencentDB database, select **Others**. If the source database is Alibaba Cloud, Huawei Cloud, etc., select **Others**.
Region	The region selected for the source database at the time of purchase cannot be modified.
Access Type	Select the network type through which the source database will connect to Tencent Cloud. Different access types require different network configuration steps. For details, see Overview of Preparatory Work. If the source database is an IDC self-hosted database or a database from another cloud provider, the available access methods are Public Network/Direct Connect/VPN Access/CCN. Public network: The source database can be accessed via a public IP. Direct connect: The source database connects to Tencent Cloud Virtual Private Cloud via Direct Connect. VPN access: The source database connects to Tencent Cloud Virtual Private Cloud via VPN Connections. CCN: The source database connects to Tencent Cloud Virtual Private Cloud via Cloud Connect Network. If the source database is a self-hosted database on CVM, select Self-Hosted on CVM as the access method. If the source database is a TencentDB instance, select Database as the access method.
Node Type	This parameter is displayed when any **Access Type** other than Database is selected. Select according to your actual situation. Single node migration: Refers to a cluster architecture with one primary node and multiple replica nodes. Cluster migration: Refers to a cluster architecture composed of multiple shards, each containing several replica nodes. When you select Cluster Migration, you will need to fill in the **Node Info**. Enter the addresses and passwords for all shard nodes in the source database cluster (IP: port: password or IP: port). For multiple nodes, list them on separate lines. It is recommended to enter the addresses of the replica nodes (secondary nodes) of the source database. Migrating from replica nodes can help avoid impacting the source database's business operations.

Public Network	When **Access Type** is set to Public Network, the following parameters need to be configured: Host address: The IP address or domain name of the source database. Port: The port used by the source database. Password: The password for the source database migration account.
Self-Build on CVM	When **Access Type** is set to Self-Build on CVM, the following parameters need to be configured. CVM instance: The instance ID of the CVM Port: The port used by the source database Password: The password for the source database migration account
Direct Connect	When **Access Type** is set to Direct Connect, the following parameters need to be configured. VPC Direct Connect Gateway: Only the VPC Direct Connect Gateway is supported for Direct Connect. Confirm the network type associated with the gateway. VPC: Select the VPC and subnet. Host address: The IP address of the source database Port: The port used by the source database Password: The password for the source database migration account
VPN Access	When **Access Type** is set to VPN Access, the following parameters need to be configured. VPN gateway: VPN gateway. Select the VPN gateway instance that connects to the source database network. VPC: Select the VPC and subnet. Host address: The IP address of the source database Port: The port used by the source database Password: The password for the source database migration account
Database	When **Access Type** is set to Database, the following parameters need to be configured. Cross-account/Intra-account Same 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. For detailed instructions on cross-account operations, see Cloud Database Migration Across Accounts. Cloud database instance: Select the instance ID of the source database. Password: The password for the source database migration account.
CCN	When **Access Type** is set to CCN, the following parameters need to be configured. When you are using CCN access, both same-account and cross-account CCN connections are supported. Due to the complexity of network configuration, see Migrating Self-built Databases to Tencent Cloud via Cloud Connect Network for detailed instructions. Host network environment: Select based on your actual situation. If the source database is a TencentDB instance, select Tencent Cloud. If the source database is an IDC self-hosted database, select Self-hosted IDC. If the source database is from another cloud provider, select the corresponding network. Host address: The IP address of the source database

	Port: The port used by the source database Password: The password for the source database migration account CCN instance account My account: The CCN resources and DTS belong to the same Tencent Cloud root account. Other account: The CCN resources and DTS belong to different Tencent Cloud root accounts. VPC CCN: The name of the CCN instance Access VPC: Select the VPC and subnet to be accessed. The Access VPC refers to the VPC within the CCN that is connected to the DTS migration linkage. Among all the VPCs associated with the CCN, choose one that is not the VPC to which the source database belongs. Access VPC region: The region of the Access VPC need to match the region of the source database selected when purchasing the task. If they do not match, DTS will automatically change the region of the source database to the Access VPC region.

Target Database Settings

The target database parameters are configured similarly to the source database. The following will describe only the differences.

Configuration Parameter	Description
Access Type	Select based on your scene. The configuration principles for different access types are similar to those for the source database. Supports migration to the cloud. When the source database **access type** is set to Cloud Database, the target database can be set to Public Network/Self-Built on CVM/Direct Connect/VPN Access/CCN.
Node Type	This parameter is displayed when any **Access Type** other than CDB is selected. Choose according to the actual situation of the target database. Supported options include Single-node migration, Redis Cluster Migration, and Proxy Cluster Migration.

Step III: Configuring Migration Options and Selecting Migration Objects

On the **Set migration options and migration objects** tab, configure the parameters, then click **Next**.

Configuration Item	Description
Migration Type	The default selection is Full + Incremental Migration and cannot be modified. Full + Incremental Migration means that both the historical data from the source database before the migration and any new data written during the migration process will be migrated together.
Migration Object	The default selection is to migrate the entire instance and cannot be modified.

Step IV: Validating and Starting the Task

1. On the verification task page, the system will automatically perform a pre-check. Once the validation passes, click **Start Task** to immediately enable data migration. (If a scheduled execution time was set during task configuration, you can also choose **Start Later** here, and the task will be enabled at the scheduled time.)

Note:

If the validation result fails, see Processing of Pre-verification Items Not Passed.

Validation result is failure: This indicates that the validation check did not pass, and the task is blocked. You need to fix the issue and then re-run the validation task.

Validation result is warning: This indicates that the validation check did not fully meet the requirements. The task can continue, but it may have some impact on the business. Users need to evaluate whether to ignore the warnings or fix the issues before proceeding.

2. Return to the data migration task list, where the task will enter the **Preparing** status. After 1 to 2 minutes, the data migration task will officially start.

Step V: Ending the Task

1. If you need to view task progress details, delete the task, or perform other operations, click the corresponding button in the **Operation** column. For more details, see Task Management.

2. When the task status is Ready to Complete, it indicates that the incremental synchronization is finished, and you can end the task. Click **Complete** in the **Operation** column to end the migration task.

It is recommended to end the task when the data difference between the target and source databases is 0 KB and the time difference between them is 0 seconds.

If you need to perform a business cutover after the migration is completed, see Cutover Description.

Migration from Tendis to KeeWiDB

Last updated：2024-09-10 17:50:22

Supported Capabilities

Source	Target	Description
Tencent Cloud Tendis	Tencent Cloud KeeWiDB 4.0 and earlier (For other versions, Submit Ticket for application)	Migration scenes between Tencent Cloud Tendis and KeeWiDB instances include: Migration between the same region or different regions within Tencent Cloud Migration between the same or different accounts within Tencent Cloud Migration across different versions with Redis protocol compatibility Migration between cluster architectures

Usage Instructions and Operation Guidance

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

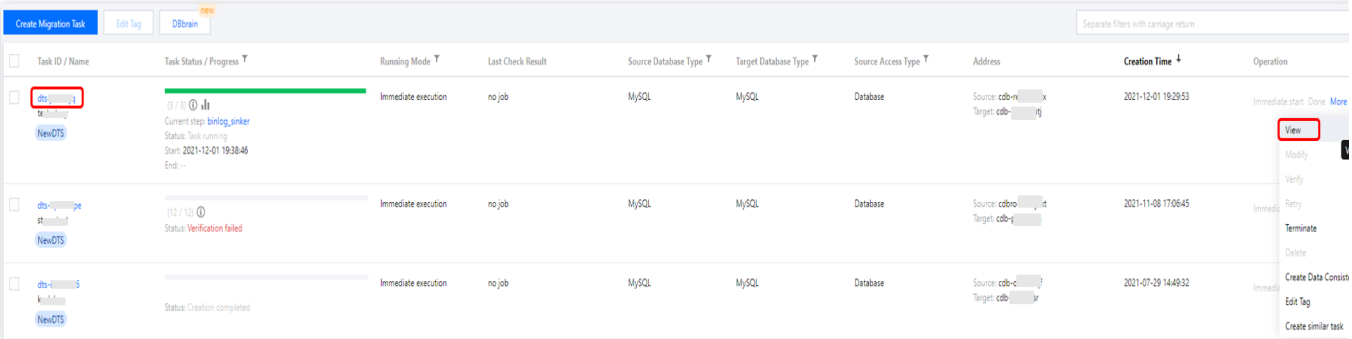
Last updated：2024-07-08 20:31:47

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.

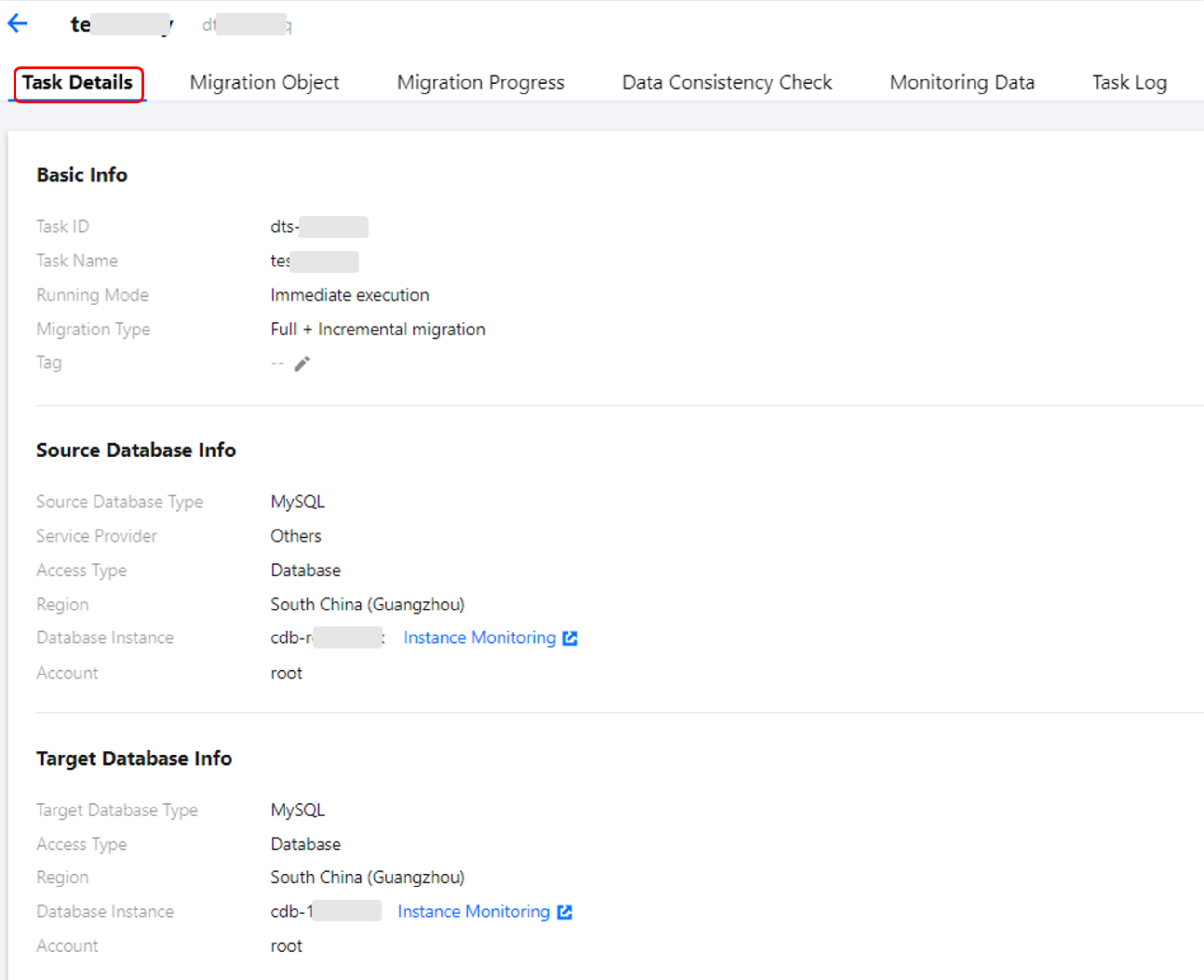
Directions

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.



□	Task ID / Name	Task Status / Progress	Running Mode	Last Check Result	Source Database Type	Target Database Type	Source Access Type	Address	Creation Time	Operation
□	dts[redacted] te[redacted] 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	Source: cdb-r[redacted]x Target: cdb-[redacted]tj	2021-12-01 19:29:53	Immediate start Done More
□	dts-[redacted]pe st[redacted] NewDTS	(12 / 12) Status: Verification failed	Immediate execution	no job	MySQL	MySQL	Database	Source: cdbro[redacted]t Target: cdb-[redacted]	2021-11-08 17:06:45	Immedi
□	dts-i[redacted]5 k[redacted] NewDTS	Status: Creation completed	Immediate execution	no job	MySQL	MySQL	Database	Source: cdb-c[redacted] Target: cdb[redacted]r	2021-07-29 14:49:32	Immedi

2. View the task details.



Basic Info

Task ID	dts-[redacted]
Task Name	tes[redacted]
Running Mode	Immediate execution
Migration Type	Full + Incremental migration
Tag	--

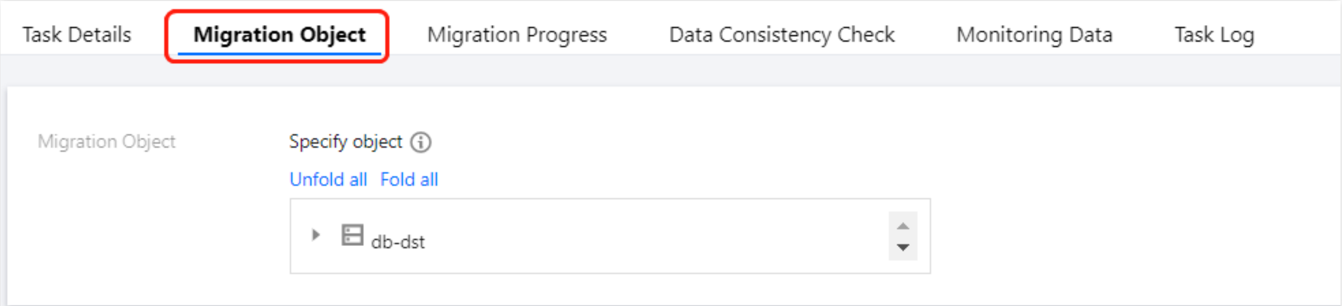
Source Database Info

Source Database Type	MySQL
Service Provider	Others
Access Type	Database
Region	South China (Guangzhou)
Database Instance	cdb-r[redacted] Instance Monitoring
Account	root

Target Database Info

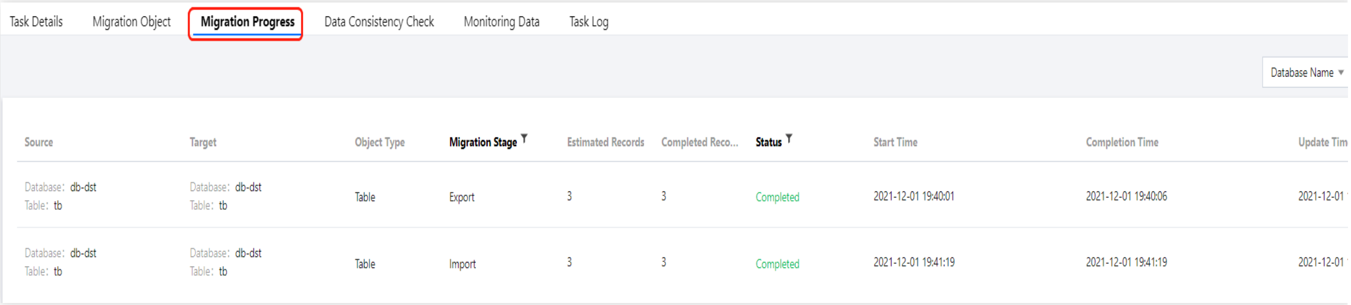
Target Database Type	MySQL
Access Type	Database
Region	South China (Guangzhou)
Database Instance	cdb-1[redacted] Instance Monitoring
Account	root

3. View the migration object.



Migration Object: Specify object

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.



Source	Target	Object Type	Migration Stage	Estimated Records	Completed Reco...	Status	Start Time	Completion Time	Update Tim
Database: db-dst Table: tb	Database: db-dst Table: tb	Table	Export	3	3	Completed	2021-12-01 19:40:01	2021-12-01 19:40:06	2021-12-01
Database: db-dst Table: tb	Database: db-dst Table: tb	Table	Import	3	3	Completed	2021-12-01 19:41:19	2021-12-01 19:41:19	2021-12-01

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

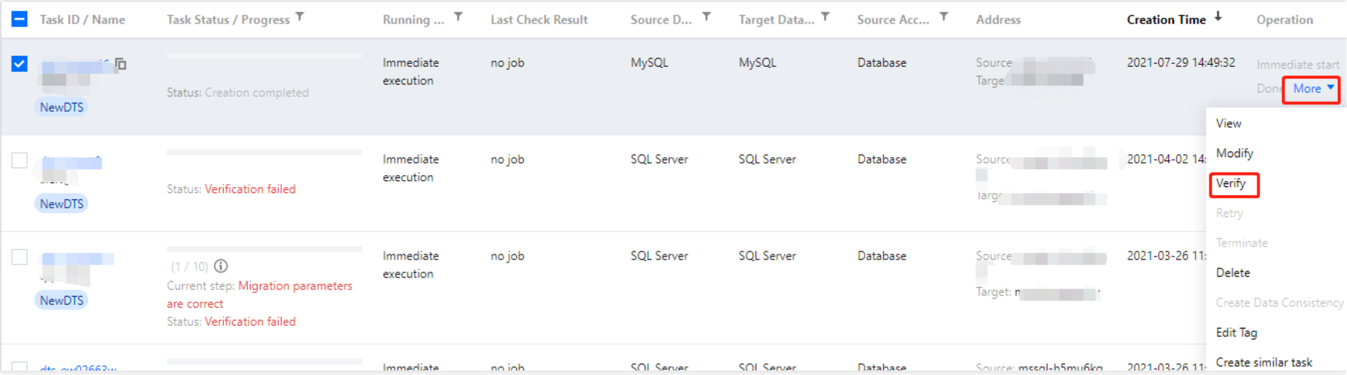
If a verification task failed, you need to initiate it again after fixing the problem as prompted.

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
☑ [redacted] NewDTS	Status: Creation completed	Immediate execution	no job	MySQL	MySQL	Database	Source [redacted] Targe [redacted]	2021-07-29 14:49:32	Immediate start Done More
☐ [redacted] NewDTS	Status: Verification failed	Immediate execution	no job	SQL Server	SQL Server	Database	Source [redacted] Targ [redacted]	2021-04-02 14:	
☐ [redacted] NewDTS	(1 / 10) Current step: Migration parameters are correct Status: Verification failed	Immediate execution	no job	SQL Server	SQL Server	Database	Source [redacted] Target: r [redacted]	2021-03-26 11:	
☐ dts-ow02663w		Immediate	no job	SQL Server	SQL Server	Database	Source: mssql-h5mu6kq	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.

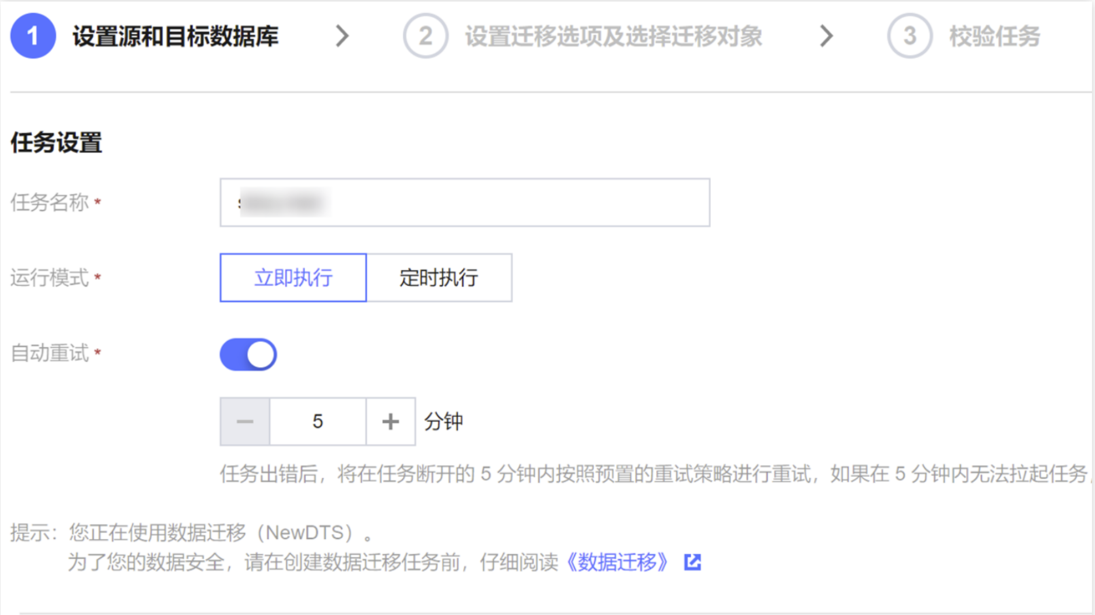
For tasks created after the retry feature was released in June 2023, such policies can be set in the task configuration stage

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 *	[redacted]
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》

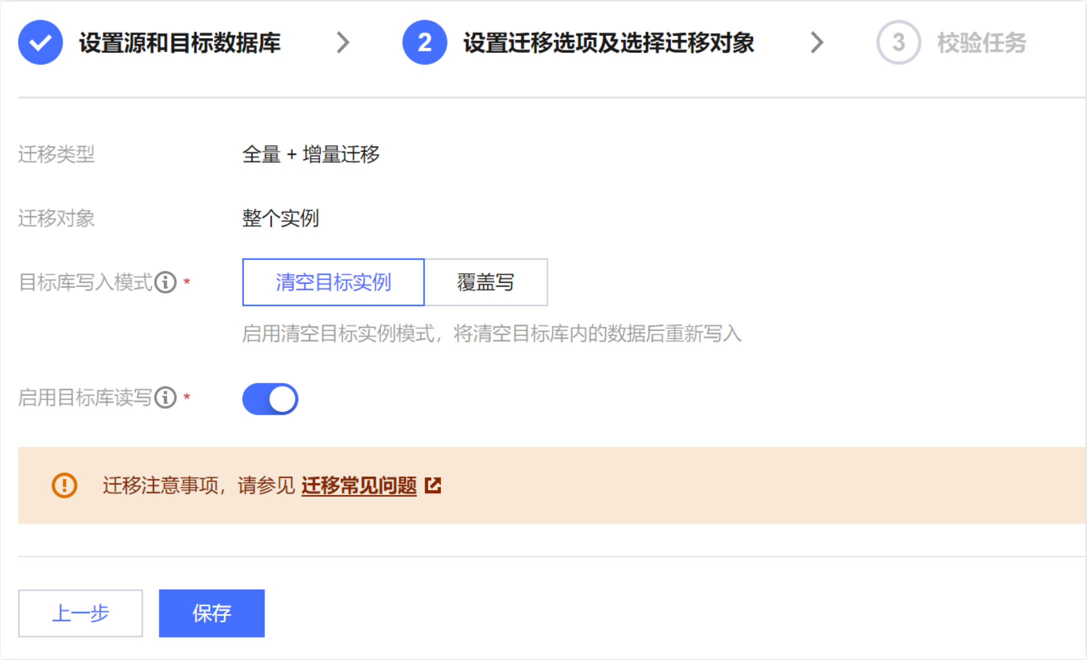
4. In the **Set migration options and select migration objects** step, set the target database read/write policy.

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.



✓ 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

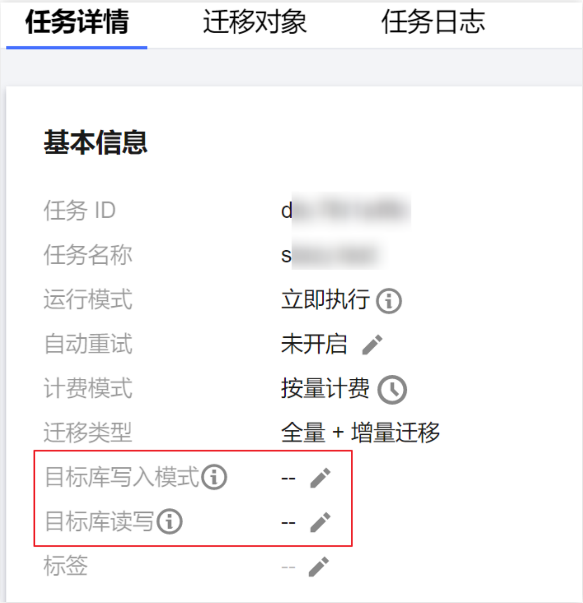
迁移类型	全量 + 增量迁移
迁移对象	整个实例
目标库写入模式 *	清空目标实例 覆盖写
	启用清空目标实例模式，将清空目标库内的数据后重新写入
启用目标库读写 *	●

迁移注意事项，请参见 迁移常见问题

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.



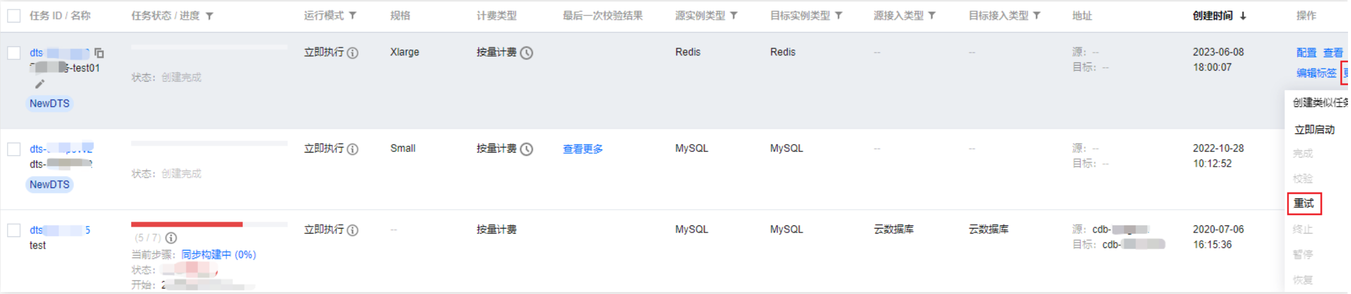
任务详情 迁移对象 任务日志

基本信息

任务 ID	d[redacted]
任务名称	s[redacted]
运行模式	立即执行 ⓘ
自动重试	未开启
计费模式	按量计费
迁移类型	全量 + 增量迁移
目标库写入模式ⓘ	--
目标库读写ⓘ	--
标签	--

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[redacted] [redacted]-test01 NewDTS	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 查看 编辑标签
dts-[redacted] dts-[redacted] NewDTS	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	
dts[redacted]5 test	(5 / 7) ⓘ 当前步骤：同步构建中 (0%) 状态：[redacted] 开始：2[redacted]	立即执行 ⓘ	--	按量计费		MySQL	MySQL	云数据库	云数据库	源：cdb-[redacted] 目标：cdb-[redacted]	2020-07-06 16:15:36	

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

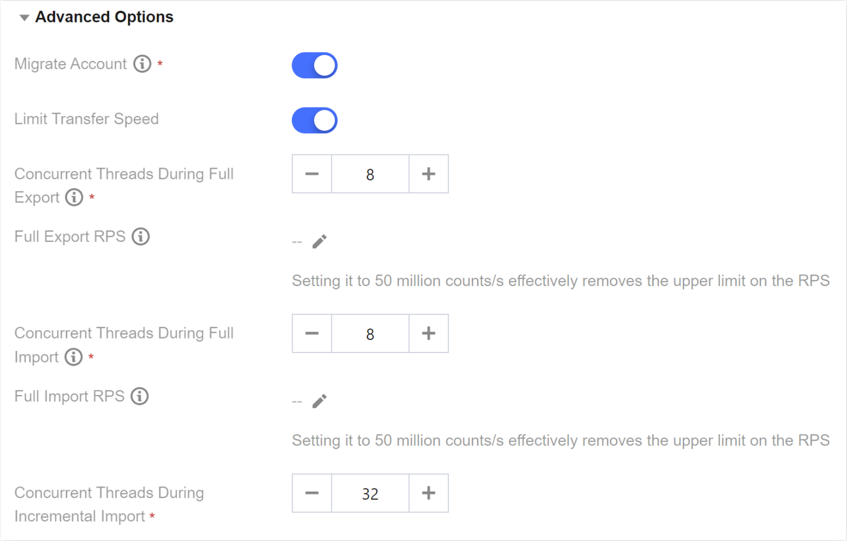
The default number of concurrent threads for a DTS task's full export is 8, which generally does not need to be modified. If adjustment is necessary, please combine monitoring indicators and source/target database configurations, and considering the precondition of not affecting the performance of the 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	If the source database configuration is high, increase the number of export threads for the source database If the target database configuration is high, increase the number of import threads for the target database. The default RPS is very large in the full volume stage, and generally, there's no need to	It may not always be effective. If adjustments do not take effect, please investigate factors such as network latency and bandwidth, as these can also affect the transfer performance.

		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 * : on

Limit Transfer Speed: on

Concurrent Threads During Full Export *: 8

Full Export RPS: --

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Full Import *: 8

Full Import RPS: --

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Incremental Import *: 32

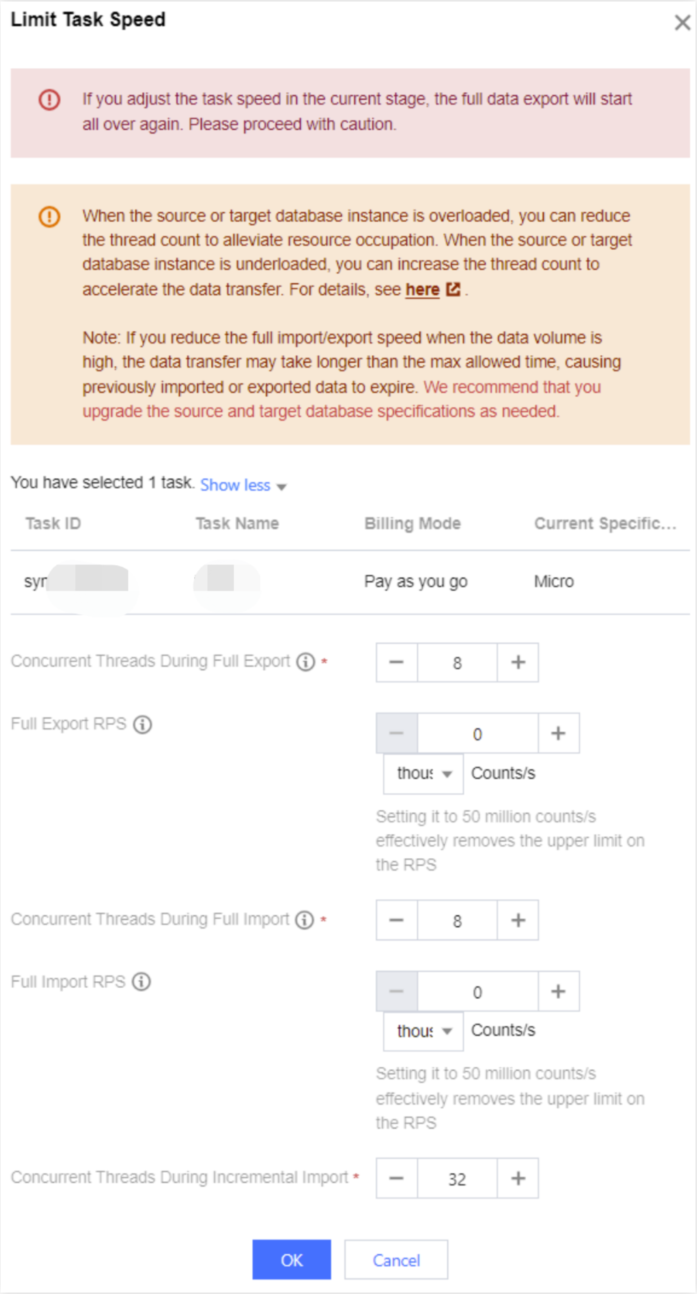
Speed Adjustment When the Task Is Running

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[redacted]	[redacted]	Pay as you go	Micro

Concurrent Threads During Full Export * : 8

Full Export RPS: 0 thous Counts/s

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Full Import * : 8

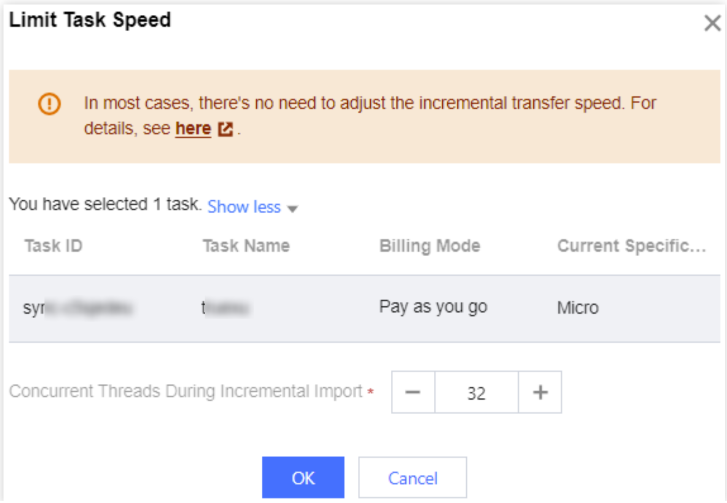
Full Import RPS: 0 thous Counts/s

Setting it to 50 million counts/s effectively removes the upper limit on the RPS

Concurrent Threads During Incremental Import * : 32

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...
[redacted]	[redacted]	Pay as you go	Micro

Concurrent Threads During Incremental Import * 32

Speed Limit (Redis)

Last updated：2024-09-10 17:31:04

Operation Scenarios

In DTS data migration task configuration or while the task is running, you can flexibly adjust the write speed to the target database based on its resource configuration.

Notes

The Redis speed limit feature was launched in March 2024. It only supports setting speed limits for tasks initiated after the launch. Existing tasks created before the launch are not supported. If you need to modify existing tasks, Submit a Work Order to upgrade the task.

Operation Steps

The following examples demonstrate how to set speed limits for a task while it is running.

1. Log in to the DTS Console, choose the **Data Migration** page in the left sidebar, and choose the specified migration task on the right.
2. In the **Operation** column, select **More** > **Limit Speed**.
3. Adjust the speed parameters in the pop-up window, and click **OK** after completion.

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.

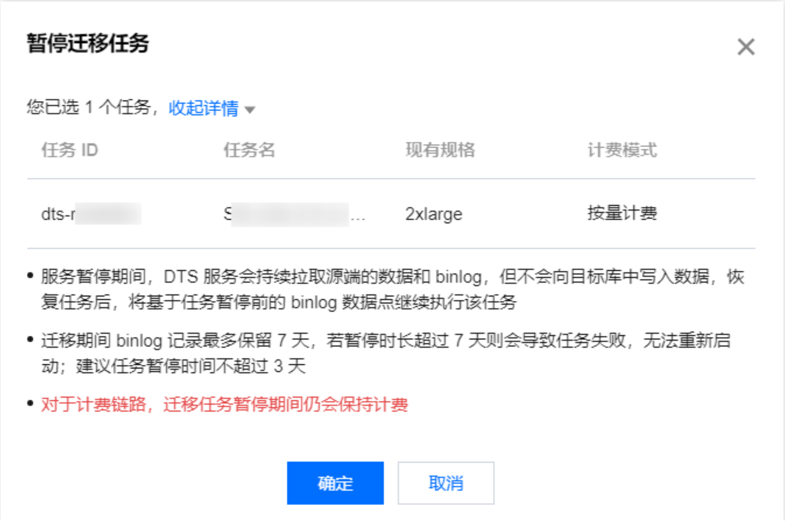
In the full import and incremental sync stages, after a task is resumed, DTS will continue with the data transfer based on the offset of the binlog synced before the task is paused.

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[redacted]	S[redacted]...	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

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 deioslation.

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

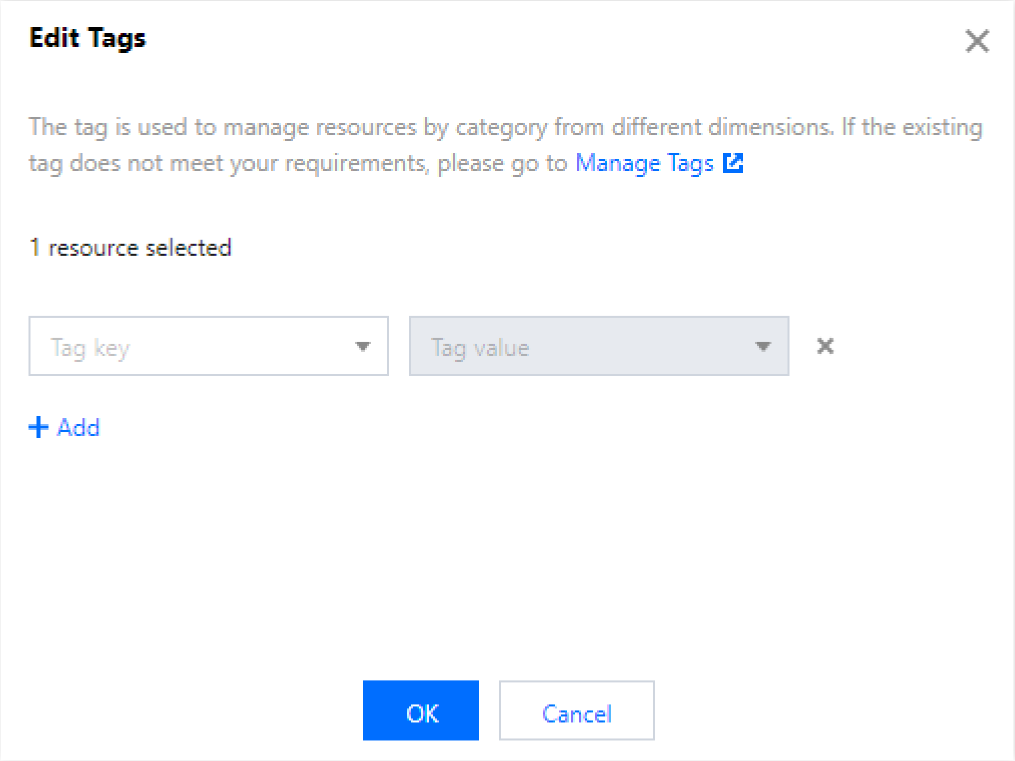
Last updated：2024-07-08 20:31:47

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**.



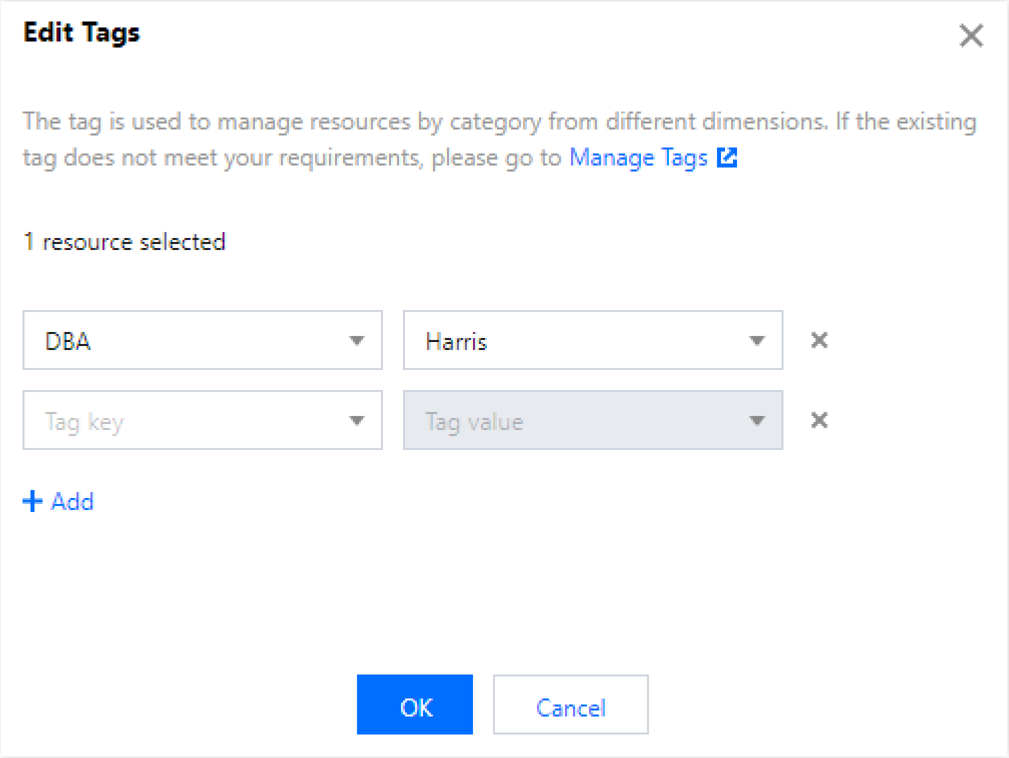
Edit Tags

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

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

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

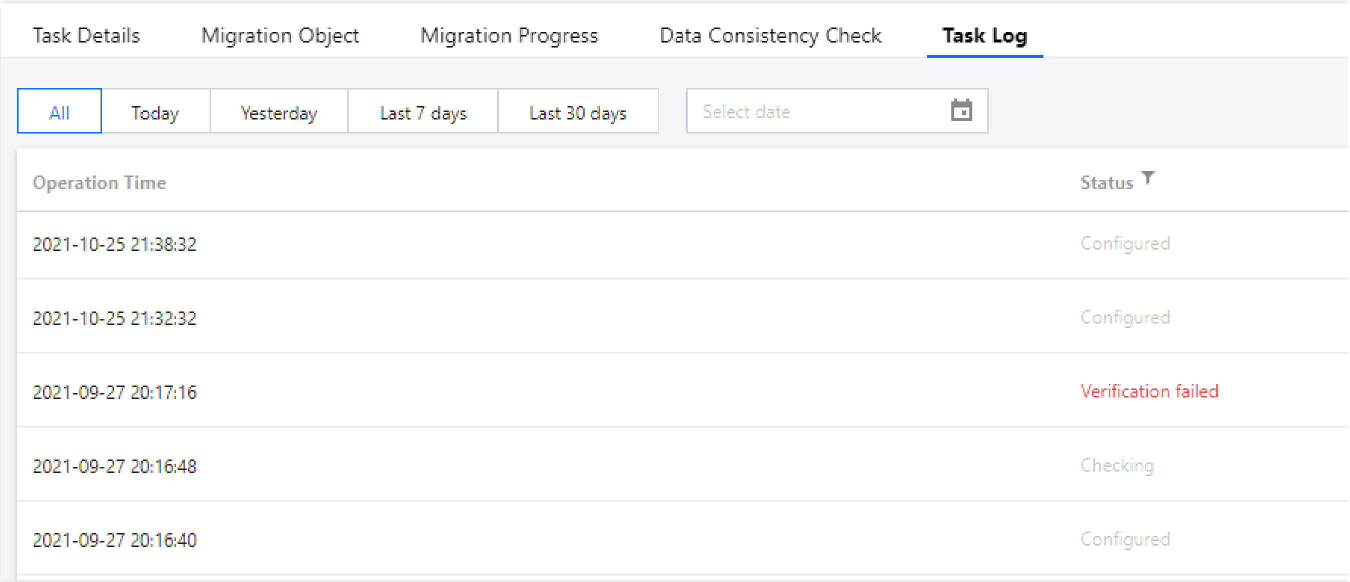
1. Log in to the DTS console, and you can view a task in the following two ways:

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.



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