Data Transfer Service

Data Sync

Product Documentation

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice

Tencent Cloud

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

Service Statement

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

Contents

Data Sync
  Databases Supported by Data Sync
  Cross-Account TencentDB Instance Sync
  Sync to MySQL series
    Sync to MySQL
      Sync from MySQL/MariaDB/Percona to TencentDB for MySQL
      Sync from TDSQL for MySQL to TencentDB for MySQL
      Sync from TDSQL-C for MySQL to TencentDB for MySQL
    Sync to MariaDB
      Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to TencentDB for MariaDB
      Sync from TDSQL for MySQL to TencentDB for MariaDB
    Sync to TDSQL-C for MySQL
      Sync from TDSQL-C for MySQL to TDSQL-C for MySQL
      Sync from MariaDB to TDSQL-C for MySQL
      Sync from TDSQL for MySQL to TDSQL-C for MySQL
    Sync to TDSQL for MySQL
      Sync from TDSQL for MySQL to TDSQL for MySQL
        Supported Capabilities
        Use Instructions
        Sync Operation Guide
      Sync from MySQL/MariaDB/Percona to TDSQL for MySQL
      Sync from MySQL/MariaDB/Percona to TDSQL for TDStore
    Sync to TDSQL TDStore
      Sync from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL TDStore
    MySQL Series Advanced Sync Operations
      Syncing Advanced Object
      Creating Data Consistency Check Task
      Setting SQL Filter Policy
      Database/Table Renaming
      Syncing Online DDL Temp Table
      Column Filtering
    MySQL series Pre Validation Failed Processing
      Database Connection Check
      Peripheral Check
      Version Check

        - Source Instance Permission Check
        - Partial Database Parameter Check
        - Target Database Content Conflict Check
        - Target Database Space Check
        - Binlog Parameter Check
        - Foreign Key Dependency Check
        - View Check
        - Advanced Object Check
        - Warning Item Check
        - TDSQL Proxy Check
        - TDSQL kernel check
        - DLL Ring Sync Check for Single Database/Table Object
        - DLL Sync Conflict Check for Single Database/Table Object in the Same Target
- Synchronize to PostgreSQL
    - Synchronization from PostgreSQL to PostgreSQL
    - PostgreSQL Pre Validation does not Pass Processing
        - Database Connection Check
        - Object Dependency Check
        - Version Check
        - Source or Target Instance Permission Check
        - Incremental Migration Pre Check
        - Structural Compatibility Check
        - Structural Conflict Check
        - Account Conflict Check
        - Parameter Configuration Conflict Check
        - Plugin Compatibility Check
        - Primary Key Dependency Check
        - Check the Key Parameters of the Instance
        - Multi Task Conflict Detection
- Synchronize to Kafka
    - Difference Between Data Sync to Kafka and Data Subscription
    - Data Delivery to Kafka Policy Settings
    - MySQL/MariaDB/Percona/TDSQL-C MySQL Sync to Kafka
        - Supported Capabilities
        - Data Sync Guide
        - Data Consumption Guide
        - Demo Description
            - Avro Demo Description

        - JSON Demo Description
        - Canal Demo Description (Canal ProtoBuf/Canal JSON)
    - TDSQL MySQL Sync to Kafka
        - Supported Capabilities
        - Use Instructions
        - Data Sync Guide
        - Data Consumption Guide
- Task Management
    - Task Status Description
    - Viewing Task
    - Configuring Task
    - Verifying Task
    - Starting Task
    - Retrying Task (MySQL)
    - Speed Limit
    - Stopping Task
    - Modify Sync configuration
    - Pausing Task
    - Resuming Task
    - Resetting Task
    - Deleting Task
    - Changing Configuration
    - Pay-as-You-Go to Monthly Subscription
    - Terminating/Returning Instance
    - Renewing Task
    - Deisolating Task
    - Deleting Instance
    - Creating a Similar Task
    - Viewing Log
    - Deleting Task

Data Sync

Databases Supported by Data Sync

Last updated：2024-04-01 15:42:08

Scenario Overview

Data Sync refers to the real-time sync of data between two data sources. A sync task is performed continuously. After a task is created, the data will be continuously synchronized (almost in real time) to keep consistency between the source and target databases.

Data Sync and **Data Migration** share basic features. **Data Sync** supports simultaneous data writing to both source and target databases and features such as data conflict resolution, dynamic task configuration adjustments, and the construction of complex topologies, making it more convenient for users to manage long-term tasks.

Tencent Cloud DTS supports data sync tasks with self-built databases, TencentDB databases, and third-party cloud databases as the source or target databases. Typical scenarios are described as follows.

Scenario	Description
Cloud-Local Sync	Synchronize databases such as self-built databases in local IDCs, self-built databases on CVM. Reverse synchronization is supported to achieve two-way synchronization between cloud and local databases.
Cross-Cloud Sync	DTS can sync a third-party cloud database to TencentDB instances to achieve dual-cloud sync.
Cross-TencentDB Instance Sync	DTS can perform cross-region and cross-border database sync, and sync between database instances under different Tencent Cloud accounts.
Sync Between IDC Databases	DTS supports two-way sync between self-built databases in local IDCs to achieve data integration and aggregation.

Feature Overview

Sync Data Types

Supports structure sync, full sync, and incremental sync.

Structure sync: In sync task configuration, set **Initialization Type** to Structure Initialization.

Full sync: In sync task configuration, set **Initialization Type** to Structure Initialization and Full data initialization.

Incremental sync: In sync task configuration, do not select any option in **Initialization Type**. Select DML and DDL in **Synchronization Operation Type**.

Access Type

DTS supports the following access types: Public Network, Self-Build on CVM, Direct Connect, VPN Access, Database, CCN, and VPC. Based on the deployment type of the database, different access types can be selected. Each access type requires specific network conditions. For more details, see Overview.

For self-built IDC databases and third-party cloud databases, you can select the following access types: Public Network, Direct Connect, VPN Access, and CCN.

For self-built databases on CVM, select Self-Build on CVM as the access type.

For TencentDB instances, select Database as the access type.

Topology

Data sync tasks of complex topologies are supported, such as one-to-many, many-to-one, two-way, and ring sync. For information about how to construct complex topologies, see the configuration guides: Creating Two-Way Sync Data Structure, Creating Many-to-One Sync Data Structure, and Creating Multi-Site Active-Active IDC Architecture.

Sync to MySQL

Note：

If you need to use the following link for synchronization, you must submit a ticket.

Synchronize data from MySQL/MariaDB/Percona to MariaDB.

Synchronize data from MariaDB/Percona to MySQL.

Synchronize data from or to TDSQL for MySQL.

Synchronize data from MySQL to TDSQL for PostgreSQL.

MySQL > MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases on CVM	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM Third-Party Cloud Providers Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0	Supported	Sync from MySQL/MariaDB/Percona to MySQL

	Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7 TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0		
Third-Party Cloud Providers Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0 Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7	Self-Built Databases ++Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM Third-Party Cloud Providers Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0 Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7 TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0		
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM Third-Party Cloud Providers Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0 Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7 TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0		

MariaDB > MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases	Self-Built Databases	Supported	Sync from MySQL/MariaDB/Percona

Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0		to MySQL
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same Tencent Cloud root account Sync between databases under different Tencent Cloud root accounts	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0		

Percona > MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0	Supported	Sync from MySQL/MariaDB/Percona to MySQL

TDSQL-C MySQL > MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TDSQL-C for MySQL 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0	Supported	Sync from TDSQL-C for MySQL to MySQL

TDSQL MySQL > MySQL

Source Database	Target Database	Two-Way Sync	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1) Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs or on CVM TencentDB for MySQL	Not supported	Sync from TDSQL for MySQL to MySQL

Sync to MariaDB

MySQL > MariaDB

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases on CVM	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)	Supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to MariaDB
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, MariaDB 10.4)		

MariaDB > MariaDB

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)	Supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to MariaDB
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)		

Percona > MariaDB

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)	Supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to MariaDB

TDSQL-C MySQL > MariaDB

Source Database and	Target Database and	Two-Way	Reference Documentation

Version	Version	Sync	
TDSQL-C for MySQL 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)	Supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to MariaDB

TDSQL MySQL > MariaDB

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1) Sync between databases under the same root account Sync between databases under different root accounts	Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4)	Not supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to MariaDB

Sync to TDSQL-C for MySQL

MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases on CVM	TDSQL-C for MySQL 5.7 and 8.0	Supported	Sync from MySQL/MariaDB/Percona to TDSQL-C MySQL
Third-Party Cloud Providers	TDSQL-C for MySQL 5.7 and		

Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0 Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7	8.0		
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	TDSQL-C for MySQL 5.7 and 8.0		

MariaDB > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	TDSQL-C for MySQL 5.7 and 8.0	Supported	Sync from MySQL/MariaDB/Percona to TDSQL-C MySQL
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same root account Sync between databases under different root accounts	TDSQL-C for MySQL 5.7 and 8.0		

Percona > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs	TDSQL-C for MySQL 5.7 and 8.0	Supported	Sync from MySQL/MariaDB/Percona to TDSQL-C MySQL

Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM			

TDSQL-C MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TDSQL-C for MySQL 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	TDSQL-C for MySQL 5.7 and 8.0	Supported	Sync from TDSQL-C for MySQL to TDSQL-C for MySQL

TDSQL MySQL > TDSQL-C MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1) Sync between databases under the same root account Sync between databases under different root accounts	TDSQL-C for MySQL 5.7 and 8.0	Not supported	Sync from TDSQL-C for MySQL to TDSQL-C for MySQL

Sync to TDSQL for MySQL

MySQL > TDSQL MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MySQL 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.6, 5.7 and 8.0 databases on CVM	TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL for MySQL
TencentDB for MySQL 5.6, 5.7 and 8.0	TencentDB for MySQL (Database versions: MySQL		

Sync between databases under the same root account Sync between databases under different root accounts	8.0, Percona 5.7, and MariaDB 10.1)		

MariaDB > TDSQL MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL for MySQL
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same root account Sync between databases under different root accounts	TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)		

Percona > TDSQL MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM	TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL for MySQL

TDSQL-C MySQL > TDSQL MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TDSQL-C for MySQL 5.7 and 8.0	TencentDB for MySQL (Database versions:	Not supported	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to TDSQL for MySQL

Sync between databases under the same root account Sync between databases under different root accounts	MySQL 8.0, Percona 5.7, and MariaDB 10.1)		

TDSQL MySQL > TDSQL MySQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1) Sync between databases under the same root account Sync between databases under different root accounts	Self-built TDSQL for MySQL TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)	Not supported	Sync from TDSQL for MySQL to TDSQL for MySQL
Self-built TDSQL for MySQL	TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1)		

MySQL > TDSQL TDStore

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases on CVM	TDSQL for MySQL TDStore	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL TDStore
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	TDSQL for MySQL TDStore		

MariaDB > TDSQL TDStore

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	TDSQL for TDStore	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL TDStore
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same root account Sync between databases under different root accounts	TDSQL for MySQL TDStore		

Percona > TDSQL TDStore

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM	TDSQL for MySQL TDStore	Not supported	Sync from MySQL/MariaDB/Percona to TDSQL TDStore

Sync to TDSQL for PostgreSQL

MySQL > TDSQL PostgreSQL

Source Database and Version	Target Database and Version	Two-Way Sync	Reference Documentation
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	TencentDB for TDSQL PostgreSQL	Not supported	Sync from TencentDB for MySQL to TDSQL for PostgreSQL

Sync to Kafka

MySQL > Kafka

Source Database and Version	Target Database and Version	Reference Documentation
Self-Built Databases Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built MySQL 5.5, 5.6, 5.7 and 8.0 databases on CVM	Message Queue CKafka 1.1.1 and later	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL databases to Kafka
Third-Party Cloud Providers Alibaba Cloud ApsaraDB for RDS 5.6, 5.7 and 8.0 Alibaba Cloud PolarDB 5.6, 5.7 and 8.0 Amazon RDS for MySQL 5.6, 5.7 and 8.0 Amazon Aurora MySQL 5.6 and 5.7	Message Queue CKafka 1.1.1 and later	
TencentDB for MySQL 5.5, 5.6, 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	Message Queue CKafka 1.1.1 and later	

MariaDB > Kafka

Source Database and Version	Target Database and Version	Reference Documentation
Self-Built Databases Self-built MariaDB 5.5, 10.0-10.6 databases in IDCs Self-built MariaDB 5.5, 10.0-10.6 databases on CVM	Message Queue CKafka 1.1.1 and later	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL databases to Kafka
TencentDB for MariaDB (Database versions: MySQL 8.0, Percona 5.7, MariaDB 10.1, and MariaDB 10.4) Sync between databases under the same root account	Message Queue CKafka 1.1.1 and later	

Sync between databases under different root accounts		

Percona > Kafka

Source Database and Version	Target Database and Version	Reference Documentation
Self-Built Databases Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases in IDCs Self-built Percona 5.5, 5.6, 5.7 and 8.0 databases on CVM	Message Queue CKafka 1.1.1 and later	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL databases to Kafka

TDSQL-C MySQL > Kafka

Source Database and Version	Target Database and Version	Reference Documentation
TDSQL-C for MySQL 5.7 and 8.0 Sync between databases under the same root account Sync between databases under different root accounts	Message Queue CKafka 1.1.1 and later	Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL databases to Kafka

TDSQL MySQL > Kafka

Source Database and Version	Target Database and Version	Reference Documentation
TencentDB for MySQL (Database versions: MySQL 8.0, Percona 5.7, and MariaDB 10.1) Sync between databases under the same root account Sync between databases under different root accounts	Message Queue CKafka 1.1.1 and later	Sync from TDSQL for MySQL to Kafka

Cross-Account TencentDB Instance Sync

Last updated：2023-11-21 20:41:18

Overview

This document describes how to use the DTS data sync feature to sync data between TencentDB instances under different accounts.

Application Scope

Cross-account data sync is supported between TencentDB for MySQL, TDSQL for MySQL, TDSQL-C for MySQL, TencentDB for MariaDB, and TencentDB for PostgreSQL instances. For more information, see the **Cross-Account Sync** column of the **Key features supported by sync** table in Databases Supported by Data Sync.

Prerequisite

You have created the target database instance.

Note

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

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

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

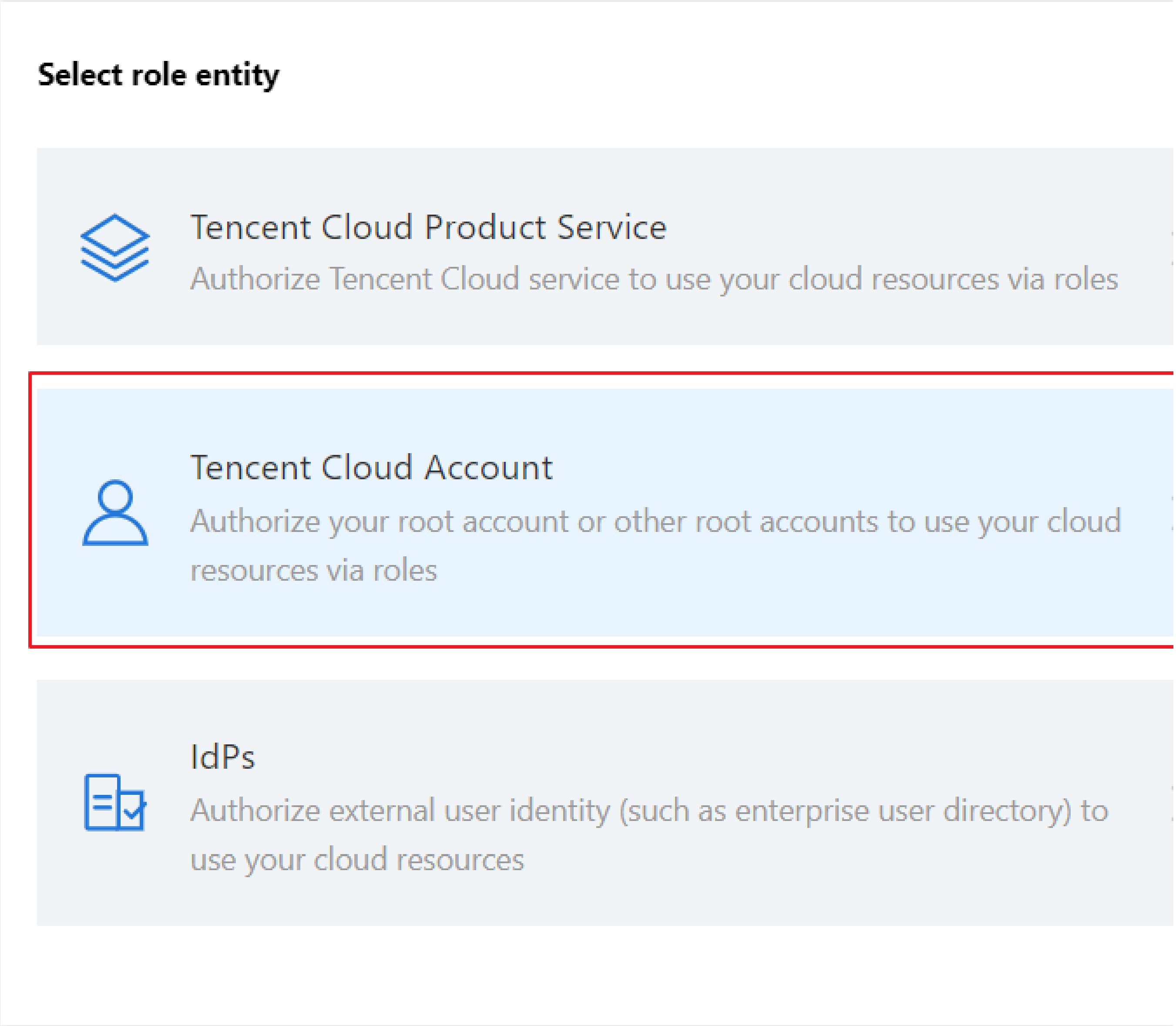
Use the root account to execute the sync task: Before executing the task, ask the source database's root account to grant the target database's root account access to the source database through a role.

Use the sub-account to execute the sync task: Before executing the task, ask the source database's root account to grant the target database's root account access to the source database through a role. Then, ask the target database's root account to grant its sub-account access to the source database through policies.

Authorizing an Account

To execute the sync 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 a sub-account has CAM and role permissions, you can also log in with the sub-account.
2. Click **Roles** 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

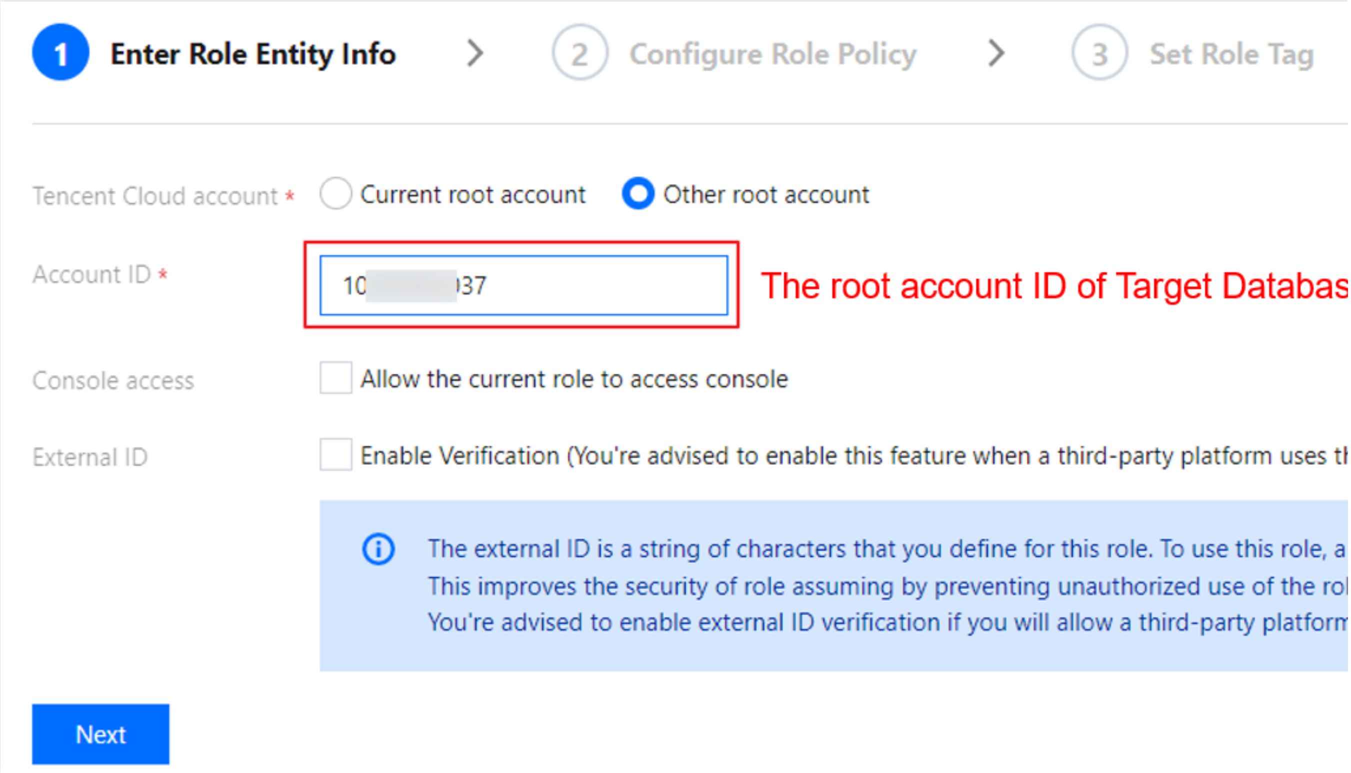
Tencent Cloud Account

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 Databas

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 tl

The external ID is a string of characters that you define for this role. To use this role, a
This improves the security of role assuming by preventing unauthorized use of the rol
You're advised to enable external ID verification if you will allow a third-party platforn

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

Account ID: Enter the Tencent Cloud root account ID of the target database, which can be viewed on the Account Information page. Enter the root account ID here even if the target database instance is under the sub-account.

External ID: You can set it as needed.

Note

If an external ID is used, record and keep it on your own, as it is required for subsequent configurations.

5. On the **Configure Role Policy** page, select the DTS service policy and the source database's corresponding policies and click **Next**.

DTS service policy: Select `QcloudDTSReadOnlyAccess` .

The source database's corresponding policies are detailed below:

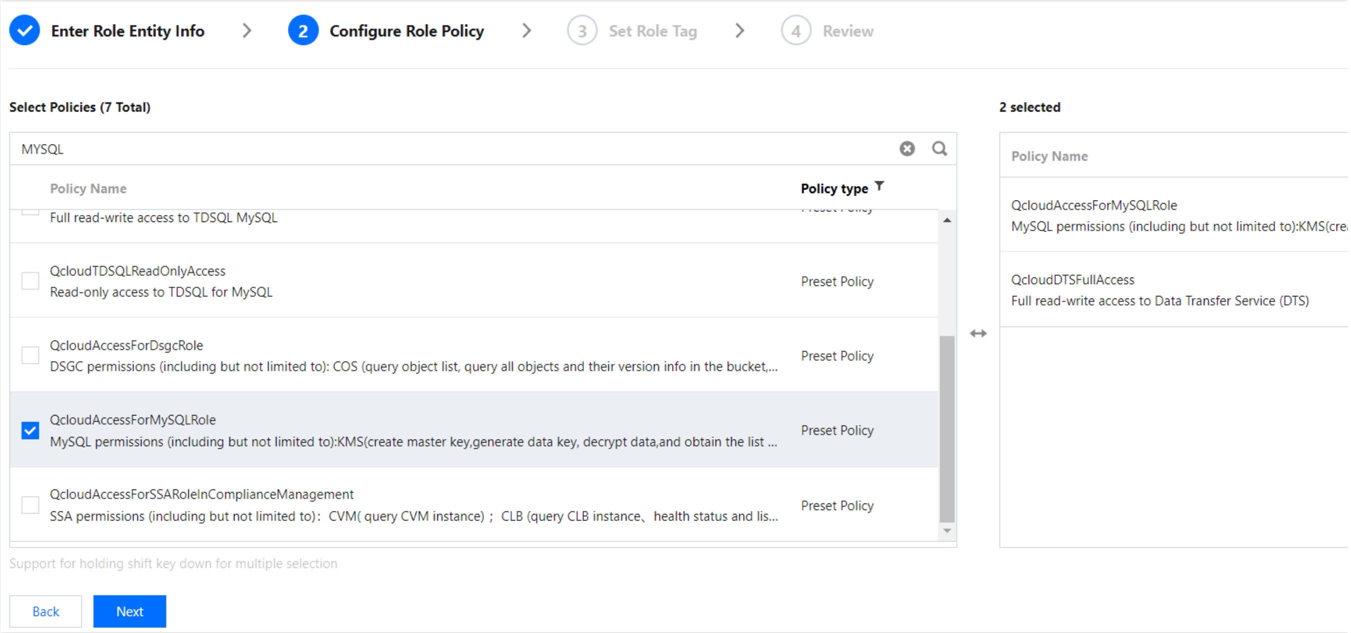
If the source database is TencentDB for MySQL, select `QcloudCDBReadOnlyAccess` (for read-only access to TencentDB for MySQL resources) and `QcloudCDBInnerReadOnlyAccess` (for TencentDB for MySQL list query).

If the source database is TencentDB for MariaDB, select `QcloudMariaDBReadOnlyAccess` (for read-only access to TencentDB for MariaDB).

If the source database is TDSQL for MySQL, select `QcloudTDSQLReadOnlyAccess` (for read-only access to TDSQL for MySQL).

If the source database is TDSQL-C for MySQL, select `QcloudCynosDBReadOnlyAccess` (for read-only access to TDSQL-C for MySQL).

If the source database is TencentDB for PostgreSQL, select `QcloudPostgreSQLReadOnlyAccess` (for read-only access to TencentDB for PostgreSQL).



Select Policies (7 Total)

MYSQL

☐	Policy Name	Policy type
☐	Full read-write access to TDSQL MySQL	Preset Policy
☐	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

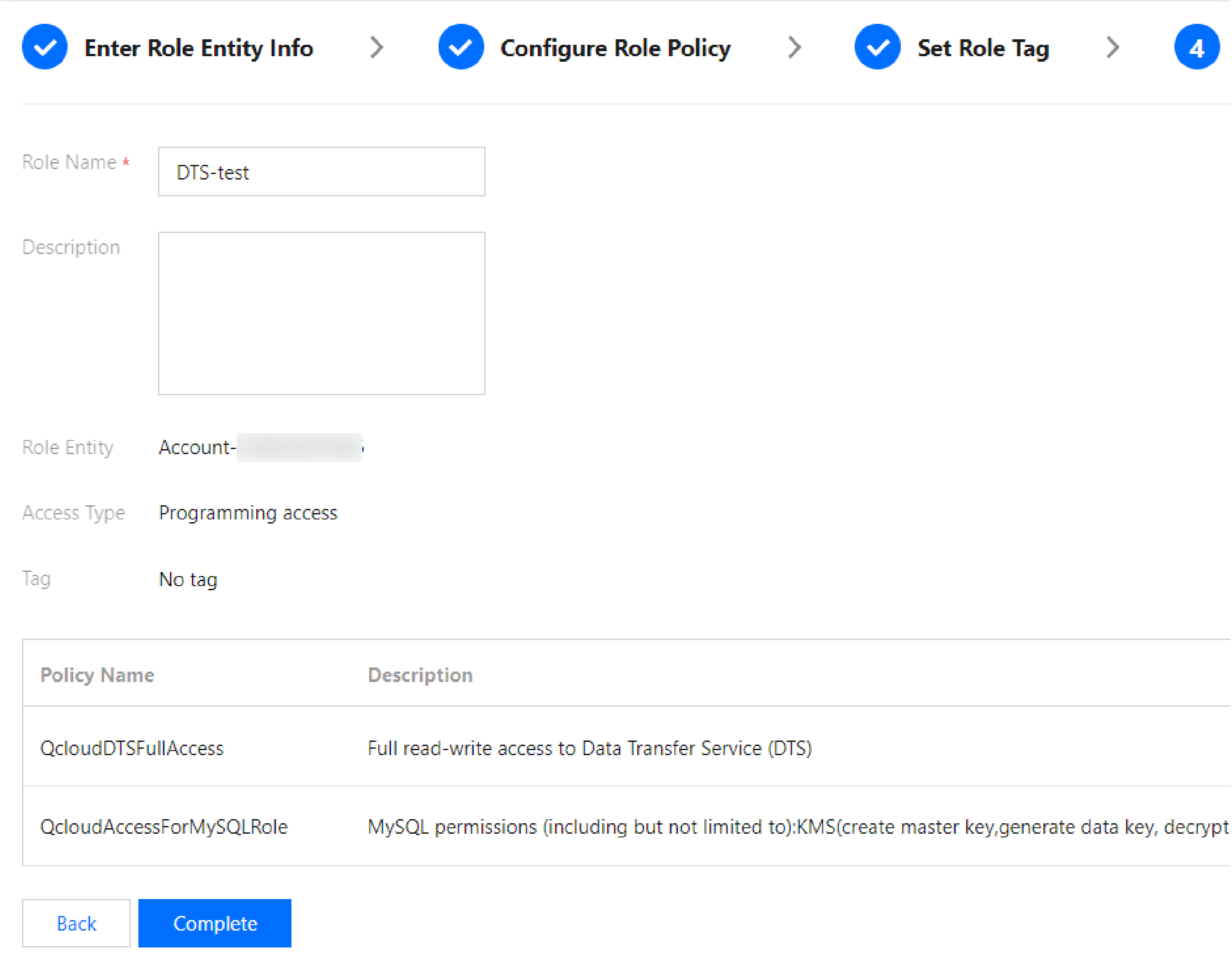
2 selected

Policy Name
QcloudAccessForMySQLRole MySQL permissions (including but not limited to):KMS(cre
QcloudDTSFullAccess Full read-write access to Data Transfer Service (DTS)

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

Note

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



Role Name * DTS-test

Description

Role Entity Account-[redacted]

Access Type Programming access

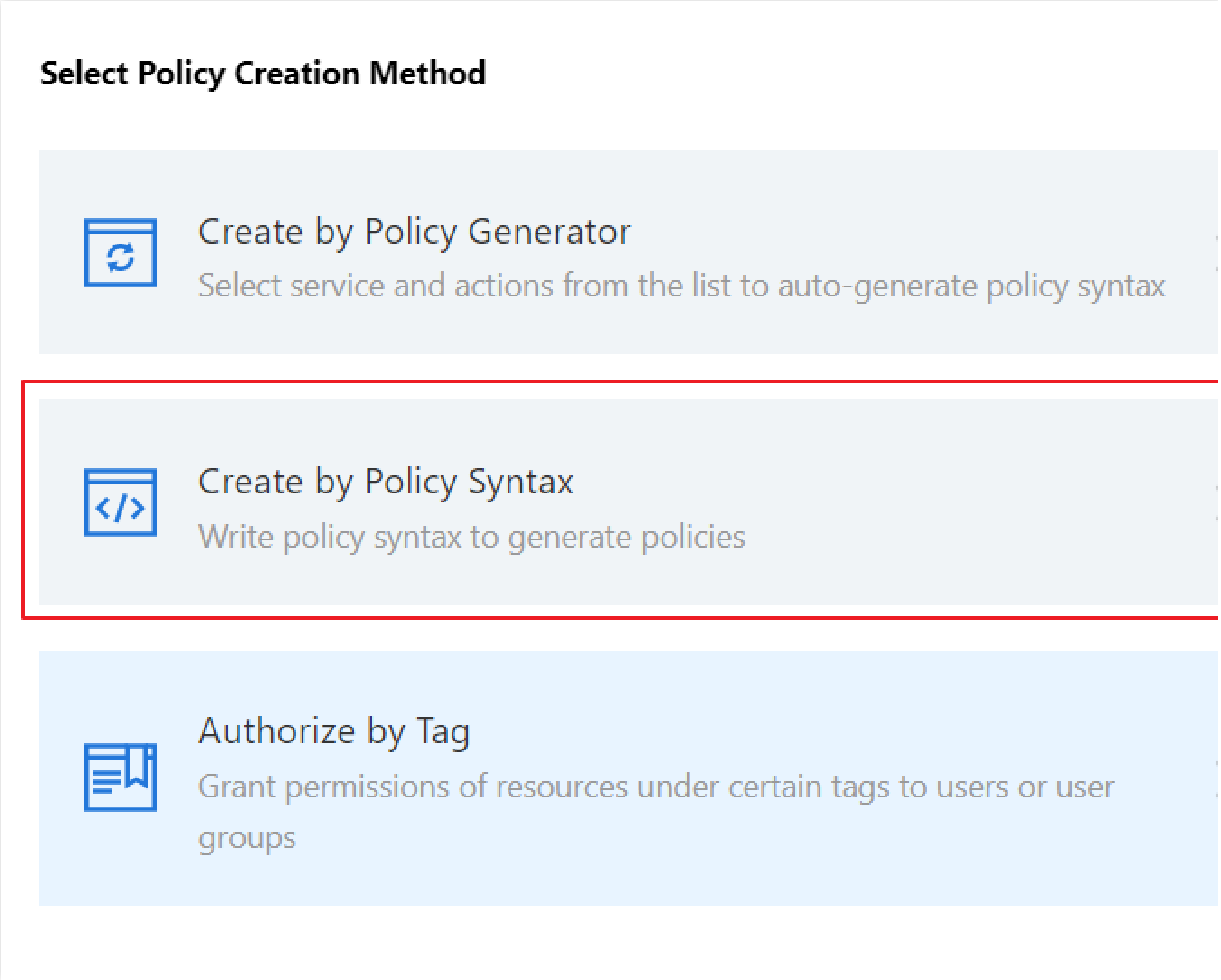
Tag No tag

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

Note

To execute a sync task with the root account, just follow the steps above; to execute a sync task with a sub-account, you also need to complete steps 7–11 below 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**.



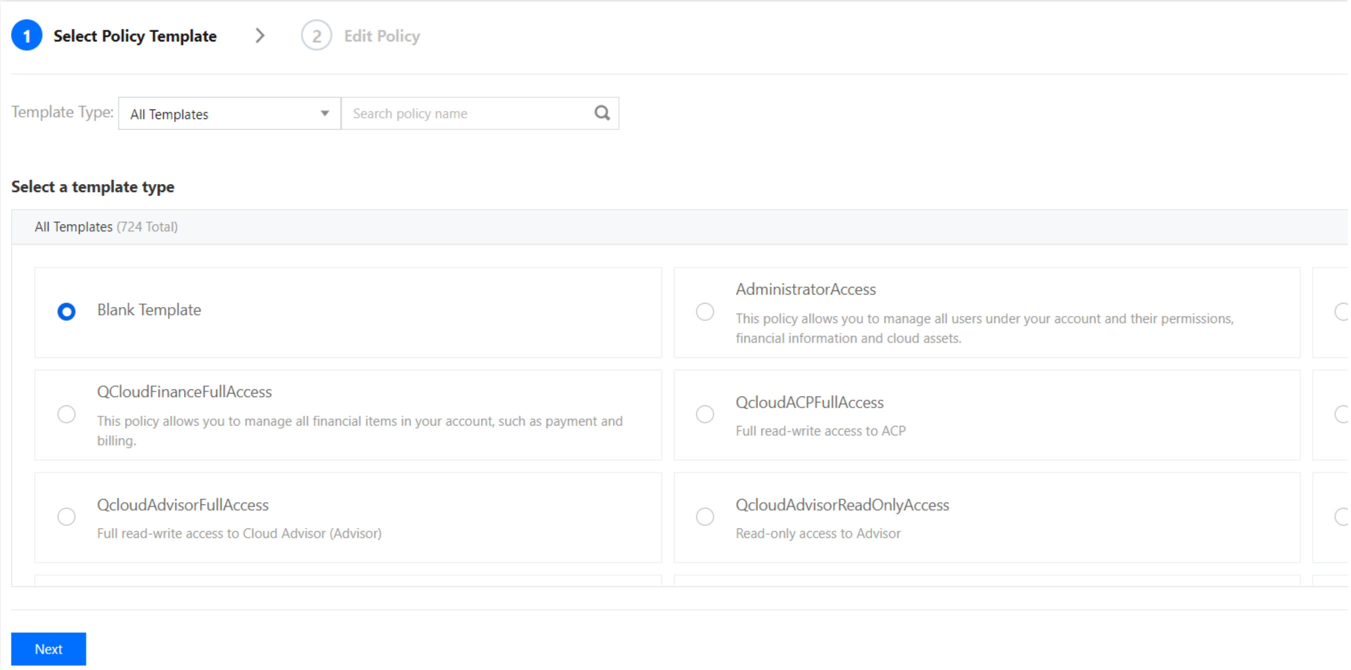
Select Policy Creation Method

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**.



Template Type: All Templates

Search policy name

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.

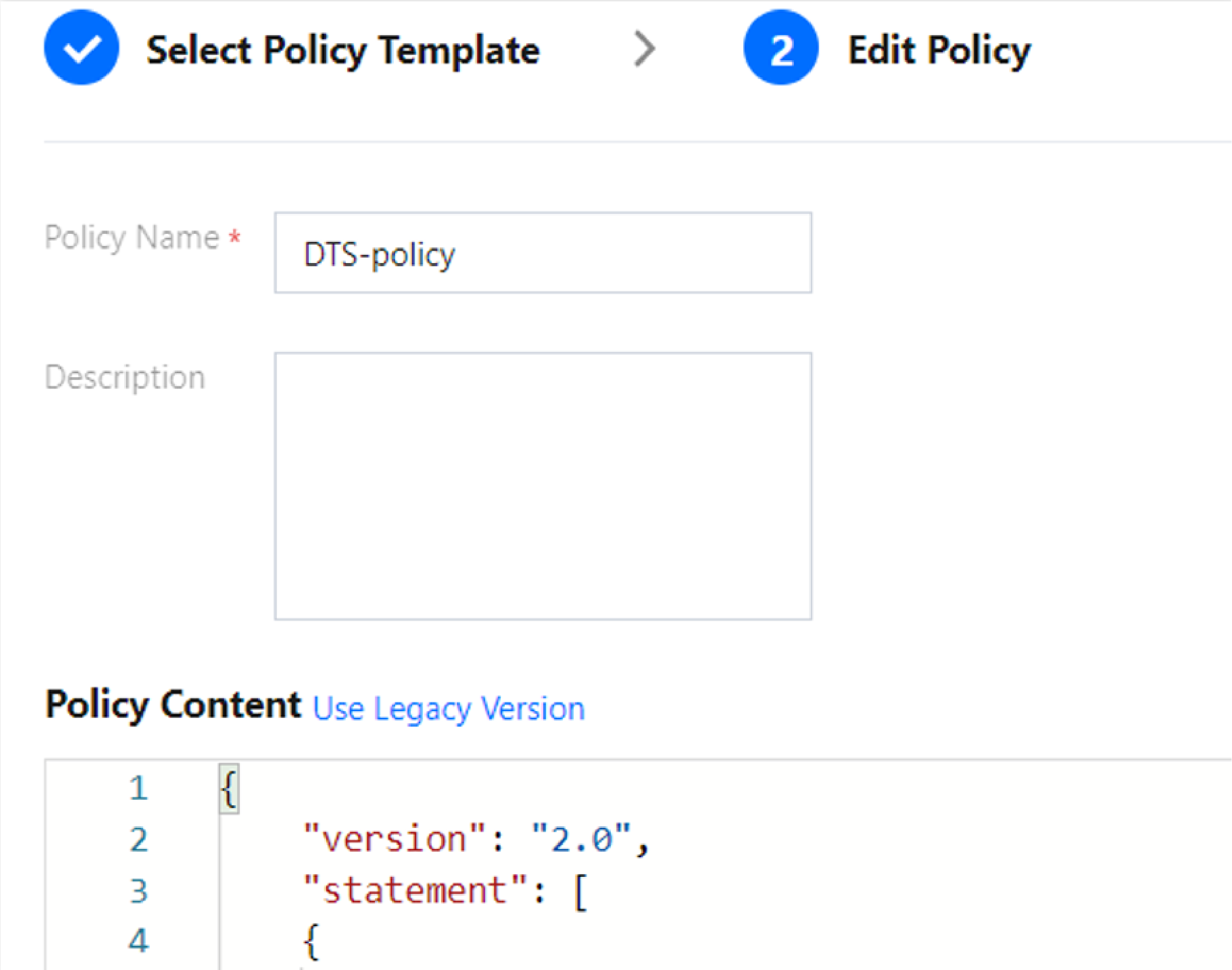
○ QCloudFinanceFullAccess
This policy allows you to manage all financial items in your account, such as payment and billing.

○ QcloudACPFullAccess
Full read-write access to ACP

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

○ QcloudAdvisorReadOnlyAccess
Read-only access to Advisor

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.

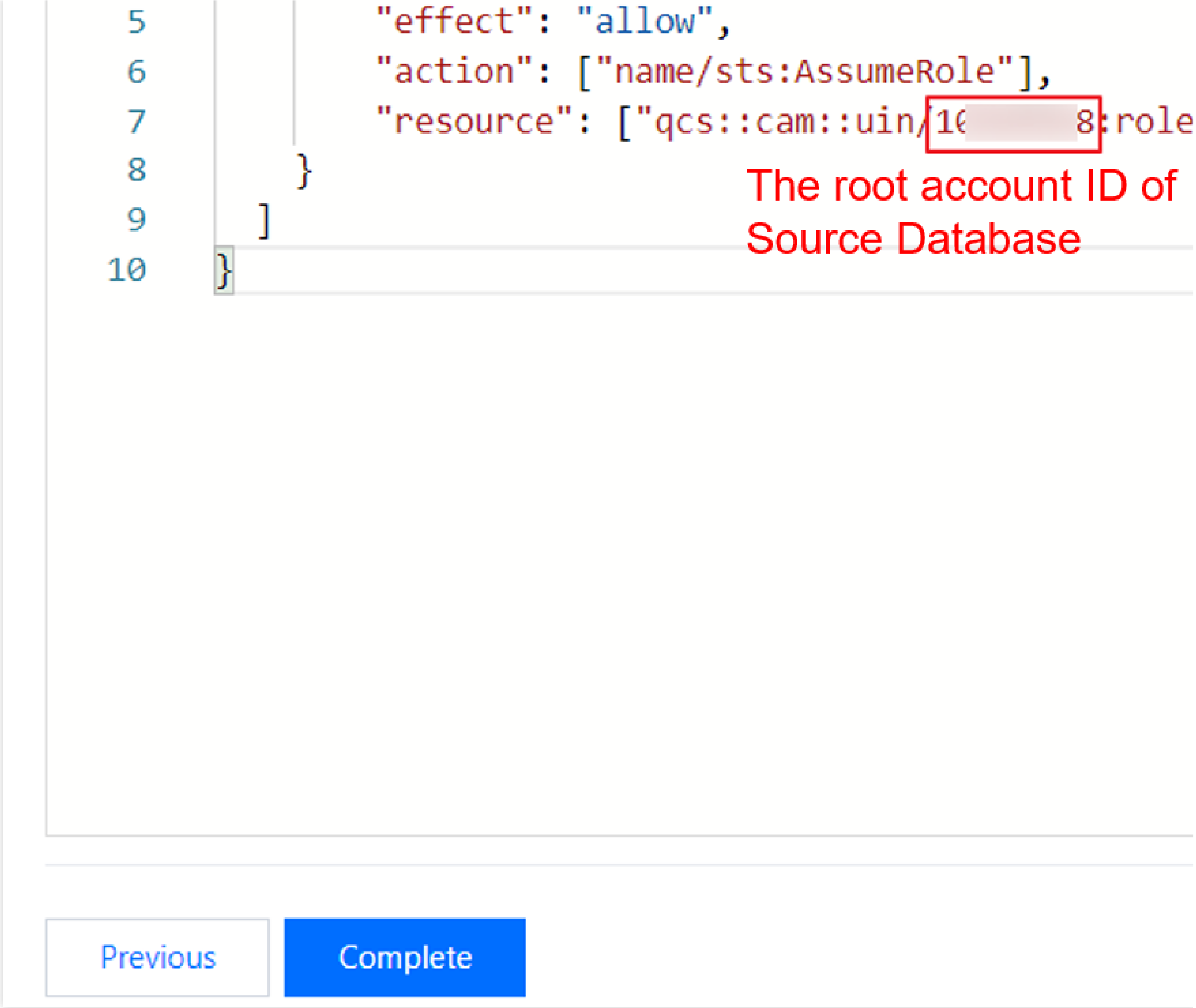


Policy Name * DTS-policy

Description

Policy Content

```
{
    "version": "2.0",
    "statement": [
    {
```



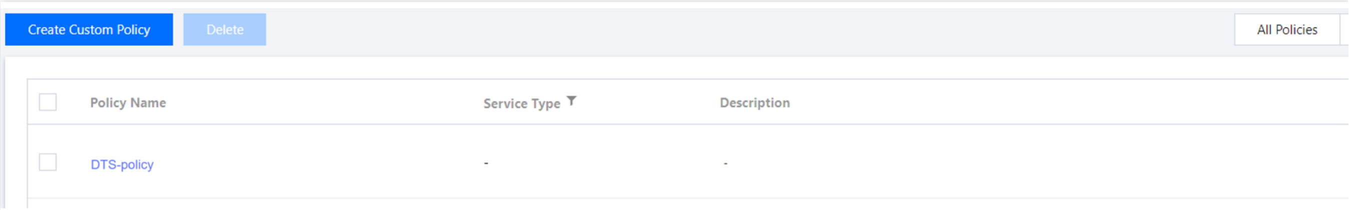
```
            "effect": "allow",
            "action": ["name/sts:AssumeRole"],
            "resource": ["qcs::cam::uin/1[redacted]8:role
        }
    ]
}
```

The root account ID of Source Database

Sample policy syntax:

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

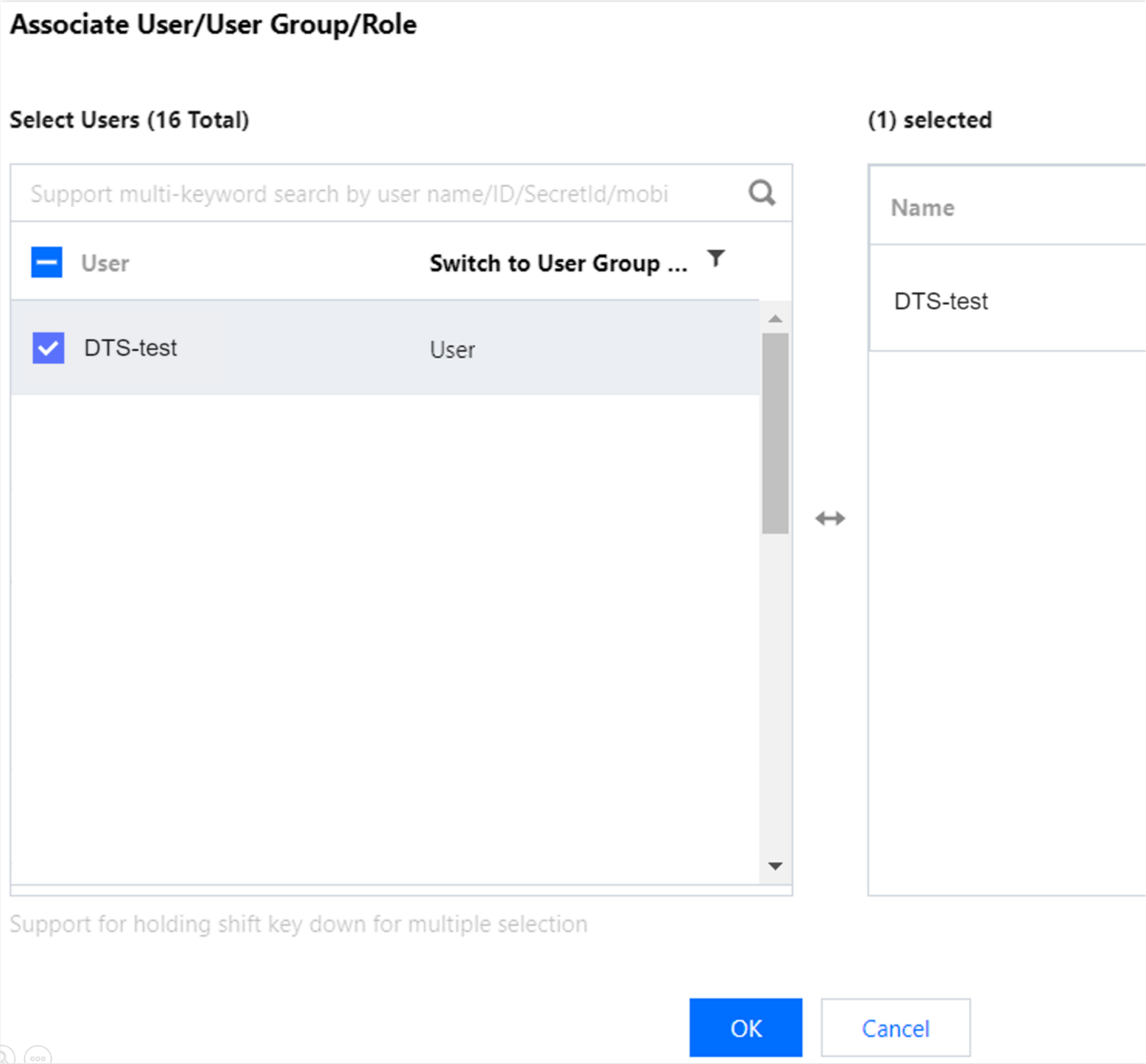
10. (Optional) Click **Complete**, return to the **Policy List** page, and click **Associate Users/Groups**.



Create Custom Policy
Delete
All Policies

☐	Policy Name	Service Type	Description
☐	DTS-policy	-	-

11. (Optional) Select the sub-account of the target database instance (that is, the sub-account executing the sync 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

☐	User	Switch to User Group ...
☑	DTS-test	User

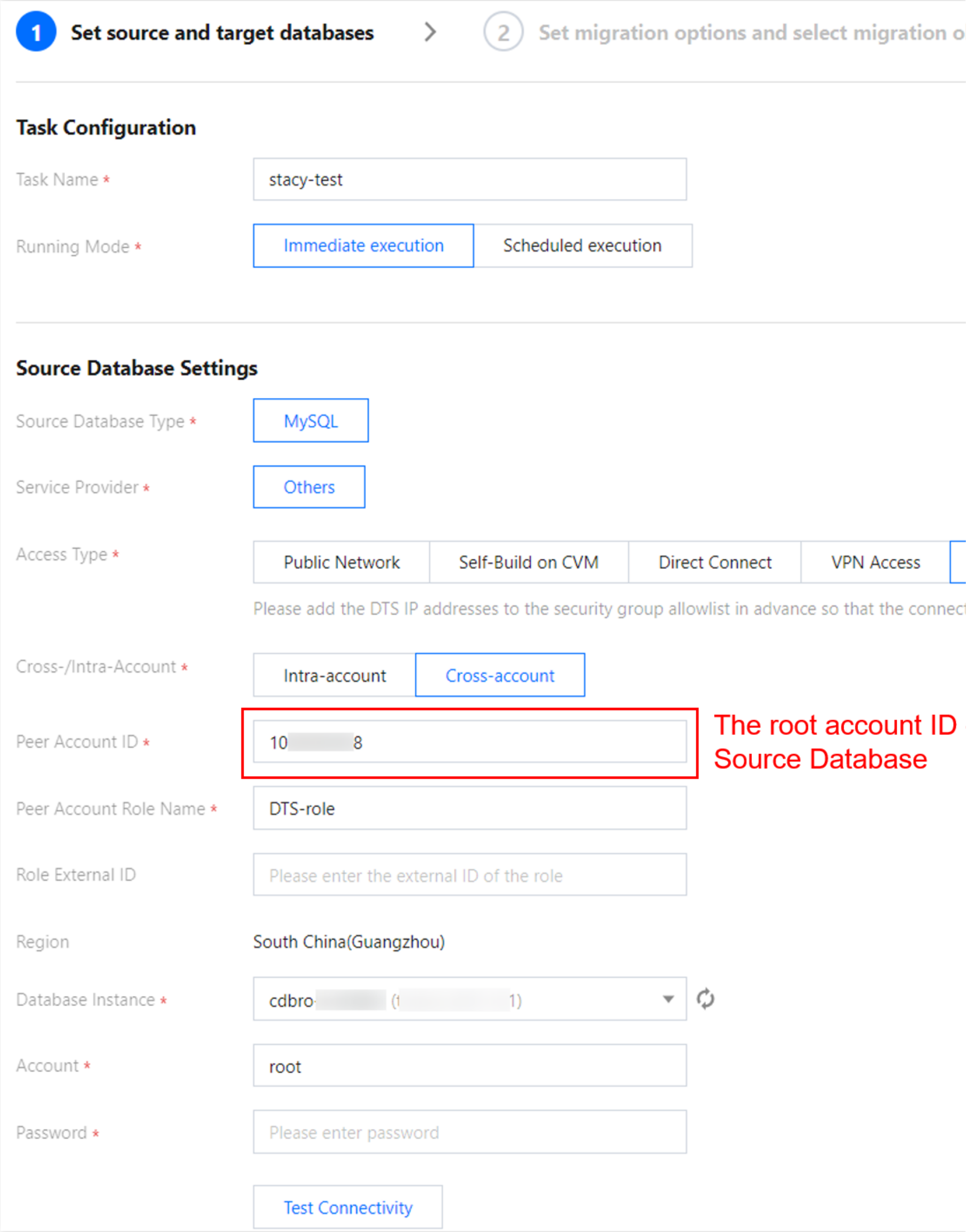
Name
DTS-test

Support for holding shift key down for multiple selection

OK
Cancel

Creating a Sync Task

1. Log in to the DTS console with the Tencent Cloud account of the target database instance.
2. Select **Data Sync** > **Create Sync Task** and purchase a sync task.
3. After making the purchase, return to the data sync task list and click **Configure** in the **Operation** column to enter the sync task configuration page.
4. In the **Set source and target databases** step, configure the source and target database information. The following is an example of data sync between TencentDB for MySQL instances.



1 Set source and target databases
2 Set migration options and select migration o

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

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

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

Peer Account ID * 10[redacted]8

The root account ID
Source Database

Peer Account Role Name * DTS-role

Role External ID: Please enter the external ID of the role

Region: South China(Guangzhou)

Database Instance * cdbro[redacted] ([redacted]1)

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**.

Cross-Account ID: Enter the root account ID of the source database.

Cross-Account Role Name: The **Role Name** configured in step 6 of Authorizing an Account. For more information on roles, see Role Overview and Cross-Account Access Role.

External Role ID: If you have set the external ID in step 4 of Authorizing an Account, here you need to enter the correct external ID to avoid permission errors. If the external role ID is not set, leave this parameter empty.

Note

After completing the above configuration, select the **Region** to obtain the instance list under the source database account. If an error occurs while obtaining the instance list, the configuration may be incorrect, or no authorization has been performed. For more information, see FAQs.

5. On the **Set sync options and objects** page, set the data initialization, data sync, and sync object options and click **Save and Go Next**.

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

If the verification fails, troubleshoot as instructed in Check Item Overview and initiate the verification again.

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

Note

You can click **More** > **Stop** in the **Operation** column to stop a sync task. Before doing so, ensure that data sync has been completed.

Common Issues

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

Check whether the **Cross-Account ID** (the root account ID of the source database) and **Cross-Account Role Name** (the **Role Name** configured in step 6 of Authorizing an Account) have been correctly configured. If the problem persists, try obtaining the source database service permissions as instructed in step 5 of Authorizing an Account.

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

Grant the source database's corresponding policies to the role as instructed in step 5 of 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 while pulling the instance list across accounts?

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

Solution:

Use the root account to create the sync 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.

4. What should I do if a permission error is reported or if I fail to pull the database instance list due to an incorrect or missing external role ID?

Error cause: You have set the external ID in Authorizing an Account, but you configured the DTS task with no external ID or a wrong one. This will cause a permission error or the failure to pull the database instance list.

Solution: If you have set the external ID in the account authorization step, you must enter it correctly when configuring the DTS task. If it is not set, you don't need to enter it here.

Sync to MySQL series

Sync to MySQL

Sync from MySQL/MariaDB/Percona to TencentDB for MySQL

Last updated：2023-06-14 10:15:47

This document describes how to use the data sync feature of DTS to sync data from MySQL/MariaDB/Percona to TencentDB for MySQL.

The following deployment modes of the source database are supported:

Self-built MySQL, third-party cloud MySQL, and TencentDB for MySQL.

Self-built MariaDB and TencentDB for MariaDB.

Self-built Percona.

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

Notes

During full data sync, 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 sync the data during off-peak hours.

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

Sync 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.

During data sync, DTS will use the account that executes the sync task to write the system database `__tencentdb__` in the source database to record the data comparison information during the sync task.

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

The source and target databases must meet the requirements for the sync feature and version as instructed in Databases Supported by Data Sync.

Permissions required of the source database:

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW VIEW,PROCESS,SEL
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'account'@'%';
FLUSH PRIVILEGES;
```

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.

Use limits

Only basic tables, views, procedures, and functions can be synced.

When views, procedures, and functions are synced, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the sync 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 sync, and set the `DEFINER` in the target database to the sync account `user2` (`[DEFINER = sync 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 synced: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during sync by default.

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

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

If the source database is Alibaba Cloud MySQL, then the tables to be synced on v5.6 must have a primary key, while tables on v5.7 and later are unrestricted. If the source database is AWS MySQL, then the tables to be synced must have a primary key.

If the binlog of the source database has a GTID hole, it may compromise the performance of the sync 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 sync.

Operation limits

During the sync, do not perform the following operations; otherwise, the sync 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 delete the system table `__tencentdb__` during incremental sync.

Synchronizable SQL operations

Operation Type	SQL Statements
DML	INSERT, UPDATE, and DELETE
DDL	CREATE DATABASE, DROP DATABASE, ALTER DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE, CREATE VIEW, DROP VIEW, CREATE INDEX, and DROP INDEX **Note：** DDL statements involving partitions cannot be synced.

Environment requirements

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

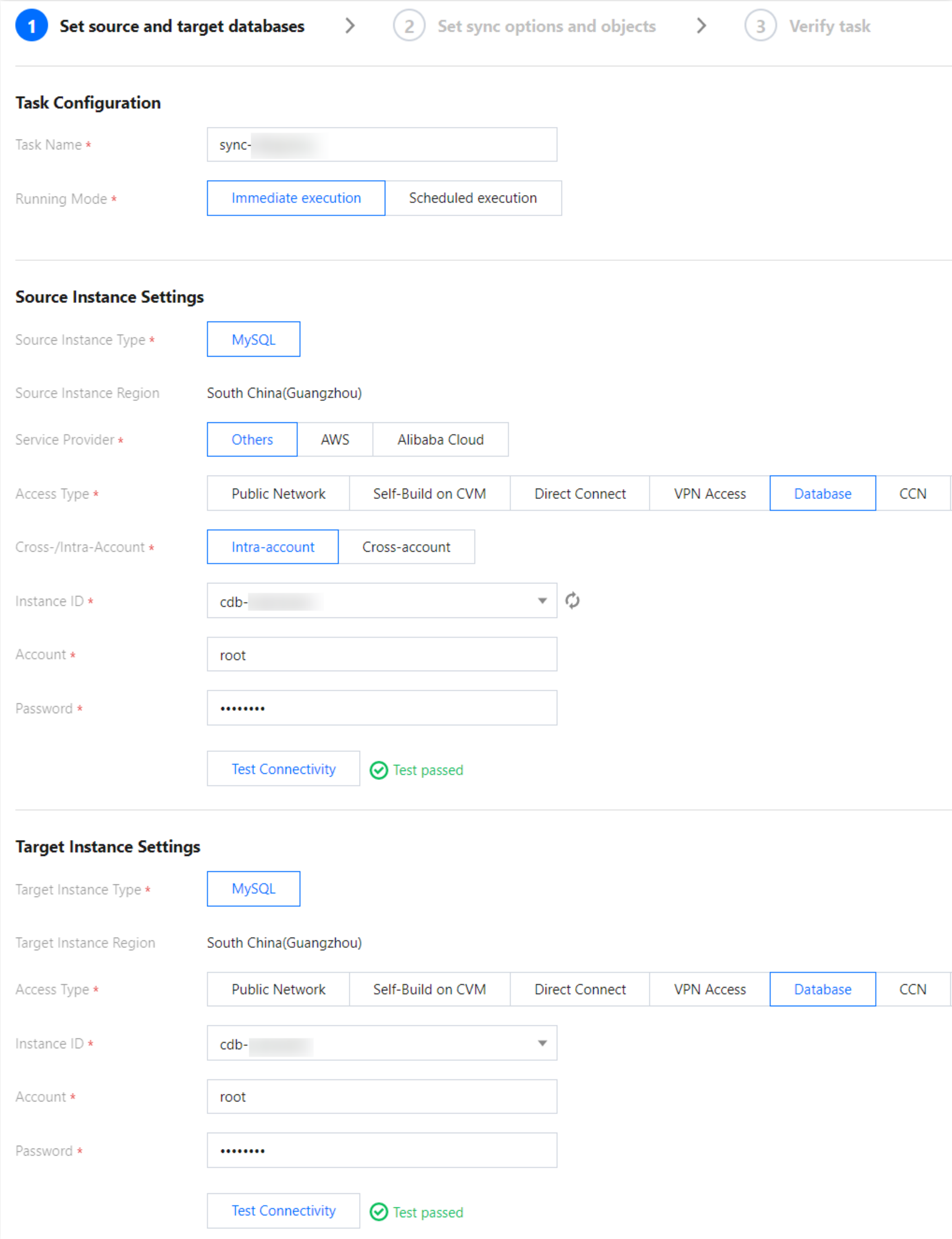
	Foreign key dependency: Foreign key dependency can be set to only one of the following two types: `NO ACTION` and `RESTRICT`. During partial table sync, tables with foreign key dependency must be migrated.
Requirements for the target database	The target database version must be later than or equal to the source database version. The target database must have sufficient storage space. If you select Full data initialization as the initialization type, the target database space must be at least 1.2 times the space of databases/tables to be synced in the source database. The target database cannot have sync 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.
Other requirements	The environment variable `innodb_stats_on_metadata` must be set to `OFF`.

Directions

1. Log in to the data sync purchase page, select appropriate configuration items, and click **Buy Now**.

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing are supported.
Source Instance Type	Select MySQL, which cannot be changed once configured.
Source Instance Region	Select the source instance region, which cannot be changed once configured.
Target Instance Type	Select MySQL, which cannot be changed once configured.
Target Instance Region	Select the target instance region, which cannot be changed once configured.
Specification	Select a specification based on your business needs. The higher the specification, the higher the performance.

2. After successful purchase, return to the data sync list, and you can see the newly created data sync task. You need to configure it before you can use it.
3. In the data sync list, click **Configure** in the **Operation** column to enter the sync task configuration page.
4. On the sync task configuration page, configure the source and target instances and their accounts and passwords, test the connectivity, and click **Next**.


1 Set source and target databases
2 Set sync options and objects
3 Verify task

Task Configuration

Task Name * sync-

Running Mode * Immediate execution | Scheduled execution

Source Instance 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

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

Instance ID * cdb-

Account * root

Password * ••••••••

Test Connectivity ✓ Test passed

Target Instance 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

Instance ID * cdb-

Account * root

Password * ••••••••

Test Connectivity ✓ Test passed

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

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

Category	Parameter	Description

Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	Select the source instance type selected during purchase, which cannot be changed once configured.
	Source Instance Region	Select the source instance region selected during purchase, which cannot be changed once configured.
	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**.
	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: The source and target databases are both deployed in Tencent Cloud VPCs. To use the VPC access type, submit a ticket for application.
	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. VPN Gateway: Select a VPN Gateway instance.
	VPC	Select a VPC and subnet associated with the VPC-based Direct Connect Gateway or VPN Gateway.
	Host Address	IP address or domain name for accessing the source MySQL instance.
	Port	Port for accessing the source MySQL instance.

	Account	Account of the source instance, which must have the required permissions.
	Password	Password of the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **Database**.
	Instance ID	Target instance ID.
	Account	Account of the target instance, which must have the required permissions.
	Password	Password of the target instance account.

Example 2: Syncing data between two TencentDB instances

Category	Parameter	Description
Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	Select the source instance type selected during purchase, which cannot be changed once configured.
	Source Instance Region	Select the source instance region selected during purchase, which cannot be changed once configured.
	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**.
	Access Type	Select a type based on your scenario. In this scenario, select **Database**. For the preparations for different access types, see Overview. Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance.

		Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections. Database: The source database is a TencentDB instance. CCN: The source database can be interconnected with VPCs through CCN. VPC: The source and target databases are both deployed in Tencent Cloud VPCs. To use the VPC access type, submit a ticket for application.
	Instance ID	Source instance ID. You can view the source instance information in the instance list.
	Account	Account of the source instance, which must have the required permissions.
	Password	Password of the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **Database**.
	Instance ID	Target instance ID.
	Account	Account of the target instance, which must have the required permissions.
	Password	Password of the target instance account.

Example 3: Syncing data from Alibaba Cloud RDS to TencentDB over the public network.

Category	Parameter	Description
Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	Select the source instance type selected during purchase, which cannot be changed once configured.

	Source Instance Region	Select the source instance region selected during purchase, which cannot be changed once configured.
	Service Provider	For a self-built database (such as a CVM-based one) or TencentDB database, select **Others**. For a third-party cloud database, select the corresponding service provider. In this scenario, select **Alibaba Cloud**.
	Access Type	For a third-party cloud database, you can select **ublic Network** generally or select **VPN Access**, **Direct Connect**, or **CCN** based on your actual network conditions.In this scenario, select **Public Network**. For the preparations for different access types, see Overview. Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance. Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections. Database: The source database is a TencentDB instance. CCN: The source database can be interconnected with VPCs through CCN. VPC: The source and target databases are both deployed in Tencent Cloud VPCs. To use the VPC access type, submit a ticket for application.
	Host Address	IP address or domain name for accessing the source instance.
	Port	Source instance port.
	Account	Account of the source instance, which must have the required permissions.
	Password	Password of the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **Database**.
	Instance ID	Target instance ID.

	Account	Account of the target instance, which must have the required permissions.
	Password	Password of the target instance account.

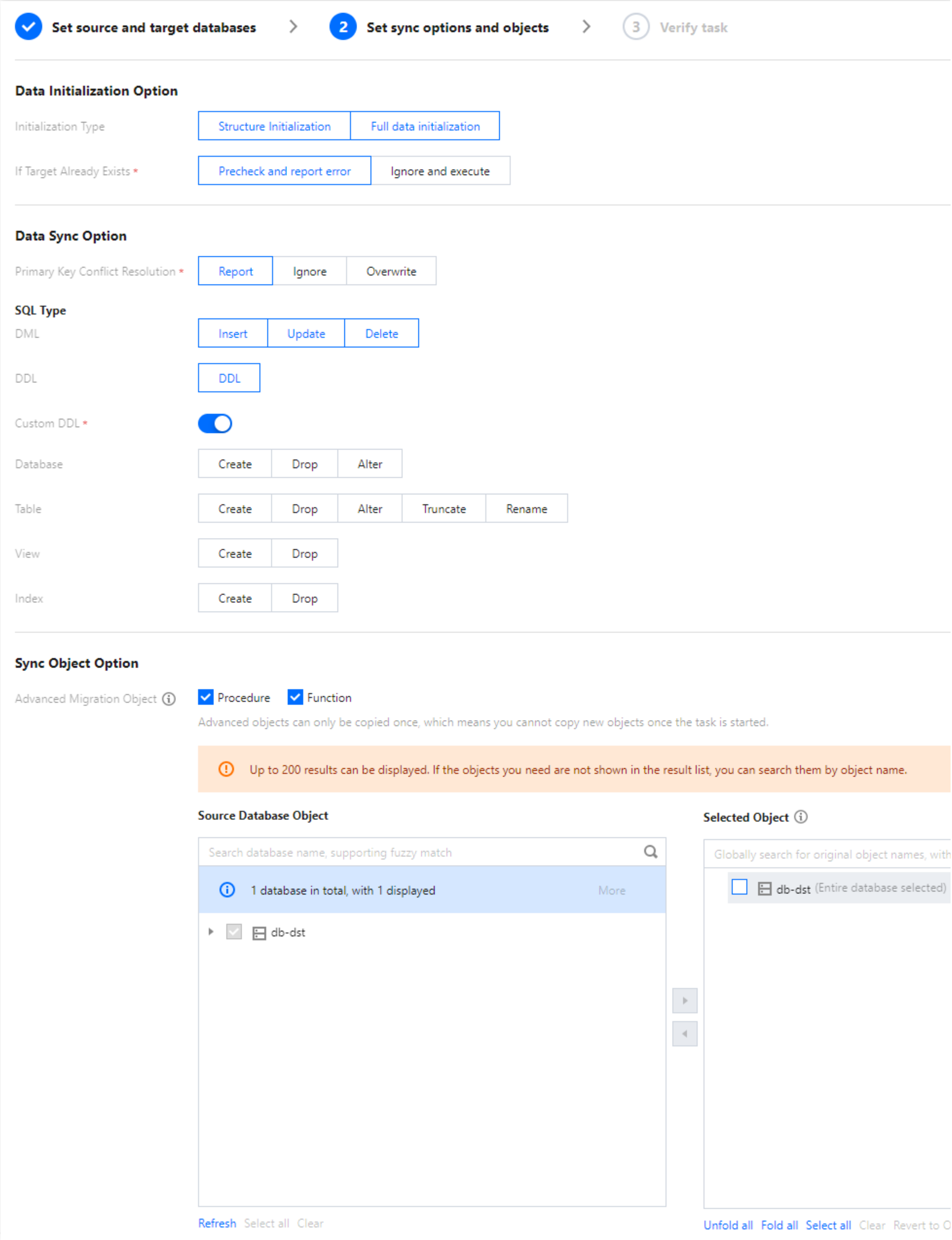
5. On the **Set sync options and objects** page, set the data initialization, data sync, and sync object options and click **Save and Go Next**.

Note:

If you only select **Full data initialization** for **Initialization Type**, the system will assume by default that you have created the table structures in the target database and will neither sync table structures nor check whether the source and target databases have tables with the same name. Therefore, if you select **Precheck and report error** for **If Target Already Exists**, the precheck and error reporting feature won't take effect.

If you select **Full data initialization** only, you need to create the table structures in the target database in advance.

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



Set source and target databases
2 Set sync options and objects
3 Verify task

Data Initialization Option

Initialization Type: Structure Initialization, Full data initialization

If Target Already Exists *: Precheck and report error | Ignore and execute

Data Sync Option

Primary Key Conflict Resolution *: Report | Ignore | Overwrite

SQL Type

DML: Insert, Update, Delete

DDL: DDL

Custom DDL *: on

Database: Create | Drop | Alter

Table: Create | Drop | Alter | Truncate | Rename

View: Create | Drop

Index: Create | Drop

Sync Object Option

Advanced Migration Object: ☑ Procedure ☑ Function

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

Up to 200 results can be displayed. If the objects you need are not shown in the result list, you can search them by object name.

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

☐ db-dst (Entire database selected)

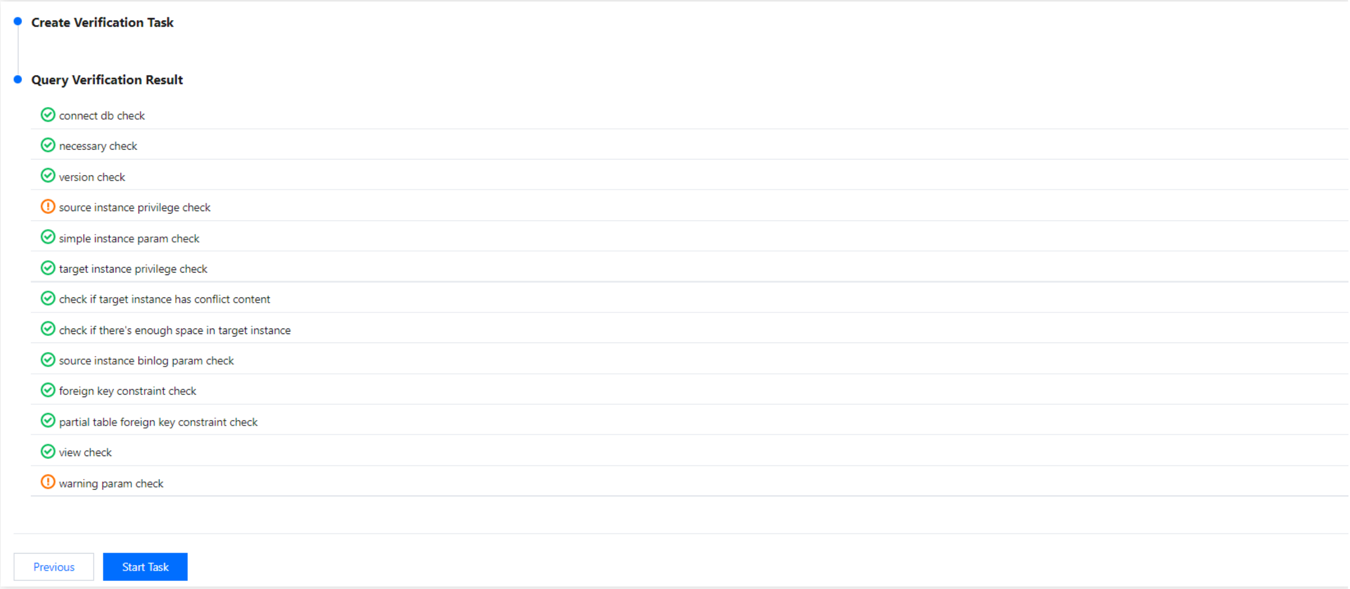
Unfold all Fold all Select all Clear Revert to O

Category	Parameter	Description
Data	Initialization Type	Structure initialization: Table structures in the source

Initialization Option		instance will be initialized into the target instance before the sync task runs. Full data initialization: Data in the source instance will be initialized into the target database before the sync task runs. Both options are selected by default, and you can deselect them as needed.
	If Target Already Exists	Precheck and report error: If a table with the same name exists in both the source and target databases, an error will be reported, and the task will stop. Ignore and execute: Full and incremental data will be directly added to tables in the target instance.
Data Sync Option	Conflict Resolution Method	Report: If a primary key conflict is found during data sync, an error will be reported, and the data sync task will be paused. Ignore: If a primary key conflict is found during data sync, the primary key record in the target database will be retained. Overwrite: If a primary key conflict is found during data sync, the primary key record in the source database will overwrite that in the target database.
	SQL Type	Supported operations include INSERT, UPDATE, DELETE, and DDL. If you select Custom DDL, you can select different DDL statement sync policies as needed. For more information, see Setting SQL Filter Policy.
Sync Object Option	Database and Table Objects of Source Instance	Select the objects to be synced. You can select basic databases, tables, views, procedures, and functions.The sync of advanced objects is a one-time operation: only advanced objects already in the source database before the task start can be synced, while those added to the source database after the task start will not be synced to the target database. For more information, see Syncing 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. When advanced objects are selected for sync, we recommend you not rename databases/tables; otherwise, sync of the advanced objects may fail. Online DDL temp tables can be synced (through tools such as gh-ost or pt-online-schema-change). Click Edit of a table and select a temp table name in the pop-up

		window. For more information, see Syncing Online DDL Temp Table.

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


Create Verification Task

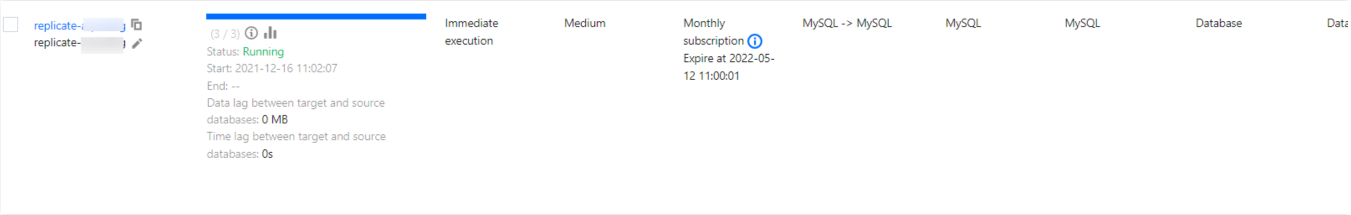
Query Verification Result

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

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

Note:

You can click **More** > **Stop** in the **Operation** column to stop a sync task. You need to ensure that data sync has been completed before stopping the task.


☐ replicate-[redacted]
replicate-[redacted]

(3 / 3)
Status: Running
Start: 2021-12-16 11:02:07
End: --
Data lag between target and source databases: 0 MB
Time lag between target and source databases: 0s

Immediate execution
Medium
Monthly subscription
Expire at 2022-05-12 11:00:01
MySQL -> MySQL
MySQL
MySQL
Database

8. (Optional) You can click a task name to enter the task details page and view the task initialization status and monitoring data.

Sync from TDSQL for MySQL to TencentDB for MySQL

Last updated：2022-08-24 16:37:48

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

Sync from TDSQL-C for MySQL to TencentDB for MySQL

Last updated：2022-09-04 22:19:45

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

Sync to MariaDB

Sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to TencentDB for MariaDB

Last updated：2024-07-08 17:33:46

This document describes how to use the data sync feature of DTS to sync data from MySQL/MariaDB/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.

TencentDB for TDSQL-C MySQL.

This document describes how to sync data from MariaDB to TencentDB for MariaDB. The requirements and steps of data sync from MySQL, Percona and TDSQL-C MySQL to TencentDB for MariaDB are basically the same.

Notes

When DTS performs full data sync, 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 sync the data during off-peak hours.

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

Sync 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.

During data sync, DTS will use the account that executes the sync task to write the system database `__tencentdb__` in the source database to record the data comparison information during the sync task.

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

The source and target databases must meet the requirements for the sync feature and version as instructed in Databases Supported by Data Sync.

Permissions required of the source database:

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW VIEW,PROCESS,SEL
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'account'@'%';
FLUSH PRIVILEGES;
```

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, views, procedures, and functions can be synced.
When views, procedures, and functions are synced, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the sync 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 sync, and set the `DEFINER` in the target database to the sync account `user2` (`[DEFINER = sync 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 synced: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during sync by default. If there is compressed TokuDB engine data in the source database, the target database must support the compression mode before sync can be performed; otherwise, the task will report an error.
Correlated data objects must be synced together; otherwise, sync will fail. Common correlations include table referenced by views, view referenced by views, and tables correlated through primary/foreign keys.
During incremental sync, if the source database has distributed transactions or generates binlog statements in the `STATEMENT` format, sync will fail.
If the source database is Alibaba Cloud ApsaraDB RDS for MySQL, then the tables to be synced on v5.6 must have a primary key, while tables on v5.7 and later are unrestricted. If the source database is Amazon RDS for MySQL, then the tables to be synced must have a primary key.
If the binlog of the source database has a GTID hole, it may compromise the performance of the sync 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 sync.

Operation restrictions

During the sync, do not perform the following operations; otherwise, the sync 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 delete the system table `__tencentdb__` during incremental sync.

Synchronizable SQL Operations

Operation Type	SQL Statement
DML	INSERT, UPDATE, DELETE
DDL	CREATE DATABASE, DROP DATABASE, ALTER DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE, CREATE VIEW, DROP VIEW, CREATE INDEX, DROP INDEX

Notes:

DDL operations involving partitions cannot be synced.

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

Environment requirements

Type	Environment Requirement
Requirements for the source database	The source and target databases can be connected. 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. 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`. 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 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 sync, 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	The target database version must be later than or equal to the source database version. The target database must have sufficient storage space. If you select **Full data initialization** as the initialization type, the target database space must be at least 1.2 times the space of databases/tables to be synced in the source database. The target database cannot have sync 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

You can refer to the directions described in Sync from MySQL/MariaDB/Percona to TencentDB for MySQL.

Sync from TDSQL for MySQL to TencentDB for MariaDB

Last updated：2022-08-24 16:32:09

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

Sync to TDSQL-C for MySQL

Sync from TDSQL-C for MySQL to TDSQL-C for MySQL

Last updated：2023-06-14 10:17:13

This document describes how to use the data sync feature of DTS to sync data from TDSQL-C for MySQL to TDSQL-C for MySQL.

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

- Data sync from MySQL to TDSQL-C for MySQL
- Data sync from TDSQL-C for MySQL to MySQL

Notes

- During full data sync, 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 sync the data during off-peak hours.
- To avoid duplicate data, make sure that the tables to be synced have a primary key or non-null unique key.
- Sync 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.
- During data sync, DTS will use the account that executes the sync task to write the system database `__tencentdb__` in the source database to record the data comparison information during the sync task.
  - To ensure that subsequent data problems can be located, the `__tencentdb__` system database in the source database will not be deleted after the sync 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-C for MySQL cluster as instructed in Creating Cluster.

- Permissions required of the source database:

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW VIEW,PROCESS,SELECT ON *.* TO 'account'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';
FLUSH PRIVILEGES;
```

- 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.

- The source and target databases must meet the requirements for the sync feature and version as instructed in Databases Supported by Data Sync.

Use limits

- Only basic tables and views can be synced, and objects such as functions, triggers, and procedures are not supported.
- When a view is exported, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the sync user `user2` , and if not, DTS will change the `SQL SECURITY` attribute of `user1` in the target database from `DEFINER` to `INVOKER` (`[INVOKER = user1]`), and set the `DEFINER` in the target database to the sync user `user2` (`[DEFINER = user2]`). If the view definition in the source database is too complex, the task may fail.
- If the source TDSQL-C 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 synced: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during sync by default.
- During incremental sync, if the source database has distributed transactions or generates binlog statements in the `STATEMENT` format, sync will fail.
- If the source database is Alibaba Cloud MySQL, then the tables to be synced on v5.6 must have a primary key, while tables on v5.7 and later are unrestricted. If the source database is AWS MySQL, then the tables to be synced must have a primary key.
- If the binlog of the source database has a GTID hole, it may compromise the performance of the sync 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 sync.

Operation limits

During the sync, do not perform the following operations; otherwise, the sync 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 delete the system table `__tencentdb__` during incremental sync.

Supported SQL operations

Operation Type	Synchronizable SQL Operations
DML	INSERT, UPDATE, and DELETE
DDL	CREATE DATABASE, DROP DATABASE, ALTER DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE, CREATE VIEW, DROP VIEW, CREATE INDEX, and DROP INDEX Note DDL statements involving partitions cannot be synced.

Environment requirements

Type	Environment Requirement
Requirements for source database	• The source and target databases can be connected. • Requirements for the 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. • 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 TDSQL-C for MySQL 5.6 or later, if the `gtid_mode` variable is not `ON`, an alarm 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 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 sync, tables with foreign key dependency must be migrated.
Requirements for the target database	• The target database version must be later than or equal to the source database version. • The target database must have sufficient storage space. If you select **Full data initialization** as the initialization type, the target database space must be at least 1.2 times the space of databases/tables to be synced in the source database. • The target database cannot have sync 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.
Other requirements	The environment variable `innodb_stats_on_metadata` must be set to `OFF`.

Directions

1. Log in to the data sync purchase page, select appropriate configuration items, and click **Buy Now**.

Parameter	Description

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing are supported.
Source Instance Type	Select TDSQL-C for MySQL, which cannot be changed once configured.
Source Instance Region	Select the source instance region, which cannot be changed once configured.
Target Instance Type	Select TDSQL-C for MySQL, which cannot be changed once configured.
Target Instance Region	Select the target instance region, which cannot be changed once configured.
Specification	Select a specification based on your business needs. The higher the specification, the higher the performance.

2. After successful purchase, return to the data sync list, and you can see the newly created data sync task. You need to configure it before you can use it.

3. In the data sync list, click **Configure** in the **Operation** column to enter the sync task configuration page.

4. On the sync task configuration page, configure the source and target instances and their accounts and passwords, test the connectivity, and click **Next**.

Category	Parameter	Description
Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	Select the TencentDB instance type selected during purchase, which cannot be changed once configured.
	Source Instance Region	Select the TencentDB instance region selected during purchase, which cannot be changed once configured.

Category	Parameter	Description
	Access Type	Select a type based on your actual conditions. In this scenario, select **Database**. - 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: The source and target databases are both deployed in Tencent Cloud VPCs.. For a third-party cloud database, you can select **Public Network** generally or select **VPN Access**, **Direct Connect**, or **CCN** based on your actual network conditions. For the preparations for different access types, see Overview.
	Instance ID	Instance ID of the source database.
	Account	Account of the source database.
	Password	Password of the source database.
Target Instance Settings	Target Instance Type	Select the target database type, which cannot be changed once configured.
	Target Instance Region	Select the target database region, which cannot be changed once configured.
	Access Type	Select the access type of the target database.
	Instance ID	Instance ID of the target database.
	Account	Account of the target database.
	Password	Password of the target database.

5. On the **Set sync options and objects** page, set the data initialization, data sync, and sync object options and click **Save and Go Next**.

Note：

- If you only select **Full data initialization** for **Initialization Type**, the system will assume by default that you have created the table structures in the target database and will neither migrate table structures nor check whether the source and target databases have tables with the same name. Therefore, if you select **Precheck and report error** for **If Target Already Exists**, the precheck and error reporting feature won't take effect.
- If you select **Full data initialization** only, you need to create the table structures in the target database in advance.
- If you want to rename a table (for example, rename table A table B) during sync, you must select the entire database (or entire instance) where table A resides rather than only table A as the **sync object**; otherwise, the system will report an error.

Category	Parameter	Description
Data Initialization Option	Initialization Type	Structure initialization: Table structures in the source instance will be initialized into the target instance before the sync task runs. Full data initialization: Data in the source instance will be initialized into the target database before the sync task runs. Both options are selected by default, and you can deselect them as needed.
	If Target Already Exists	Precheck and report error: If a table with the same name exists in both the source and target databases, an error will be reported, and the task will stop. Ignore and execute: Full and incremental data will be directly added to tables in the target instance.
Data Sync Option	Conflict Resolution Method	Report: If a primary key conflict is found during data sync, an error will be reported, and the data sync task will be paused. Ignore: If a primary key conflict is found during data sync, the primary key record in the target database will be retained. Overwrite: If a primary key conflict is found during data sync, the primary key record in the source database will overwrite that in the target database.
	SQL Type	Supported operations include INSERT, UPDATE, DELETE, and DDL. If you select **Custom DDL**, you can select different DDL statement sync policies as needed. For more information, see Setting SQL Filter Policy.

Category	Parameter	Description
Sync Object Option	Database and Table Objects of Source Instance	Select the objects to be synced. You can select databases, tables, and views.
	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. ◦ When advanced objects are selected for sync, we recommend you not rename databases/tables; otherwise, sync of the advanced objects may fail. ◦ Online DDL temp tables can be synced (through tools such as gh-ost or pt-online-schema-change). Click **Edit** of a table and select a temp table name in the pop-up window. For more information, see Syncing Online DDL Temp Table.

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.

- 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 sync task list, and you can see that the task has entered the **Running** status.

> Note：
> You can click **More** > **Stop** in the **Operation** column to stop a sync task. You need to ensure that data sync has been completed before stopping the task.

8. (Optional) You can click a task name to enter the task details page and view the task initialization status and monitoring data.

Sync from MariaDB to TDSQL-C for MySQL

Last updated：2023-03-03 15:09:15

The steps of data sync from MariaDB to TDSQL-C for MySQL are basically the same as those described in Sync from TDSQL-C for MySQL to TDSQL-C for MySQL.

Sync from TDSQL for MySQL to TDSQL-C for MySQL

Last updated：2023-09-05 15:10:32

The steps of data sync from TDSQL for MySQL to TDSQL-C for MySQL are basically the same as those described in Sync from TDSQL for MySQL to TDSQL for MySQL.

Sync to TDSQL for MySQL

Sync from TDSQL for MySQL to TDSQL for MySQL

Supported Capabilities

Last updated：2024-07-08 17:34:15

Major Feature Category	Feature Subitem or Description	Supporting Capability
Sync Objects	-	Ordinary object table, index, view
Sync Type	-	Structure sync Full sync (automatically continue with incremental after full completion) Incremental sync
Breakpoint Resume	-	Partial support for full export (structure export, no primary key table export; stage does not support resume); Full import, incremental support.
Task Channel Key Operation	Retry	Supported
	Pause, recover	Supported
	Modify sync configuration (dynamically add or remove table objects)	Not supported
	Create similar task	Supported
	Speed limit	Supported
Sync Direction	Forward/Reverse	Not involved
	Many-to-one (multiple instances merged into one/multiple tables merged into one)	Supported
	One-to-many	Supported
	Two-way sync	Not supported

Structure Sync	Database table filtering	Supported
	Column filtering	Not supported
	Database table name mapping	Supported
	Column name mapping	Not supported
	Custom partitioning (specify the partition type, partition key)	MySQL/MariaDB/Percona sync to TDSQL for MySQL, source's partition table is converted into a single table at the destination
	Skip sync table and correction	Not supported
Full Sync	Where condition filtering	Supported
Incremental Sync	Primary key conflict resolution policy	Conflict overwrite only
	DML sync (INSERT/UPDATE/DELETE)	Supported
	DML filtering	Supported
	DDL sync	CREATE DATABASE, DROP DATABASE, ALTER DATABASE CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE CREATE VIEW, DROP VIEW CREATE INDEX, DROP INDEX CREATE TABLE table name AS SELECT statement is not supported
	DDL filtering	Supported
	Where condition filtering	Supported
	Scheduled sync	Not supported
	Online DDL	Supported
Data Consistency Check	Whole instance verification	Supported
	Row count check	Supported
	Sampling verification	Supported
	Supports consistency check after	Not supported

	Where condition filtering	

Use Instructions

Last updated：2024-07-08 17:34:39

Category	Description
Sync Object	1. Only sync of base tables and views is supported. Sync of functions, triggers, storage processes, and other objects is not supported. 2. Only sync of InnoDB database engines is supported. If there are tables with other database engines, an error will be reported during task check. 3. Interrelated data objects need to be synced at the same time, otherwise sync failure will occur. 4. In the source TDSQL MySQL, there is a limit on the number of tables, with a maximum number of 5,000 for the entire instance. Exceeding this limit will result in DTS task errors; also, having too many tables will increase the access time at the source, leading to performance fluctuation and decline. 5. During the incremental sync stage, if source database table names include TDSQLagent and tdsql_sub characters, they might be filtered out or cause sync anomalies, as these table names are the same as TDSQL system's temporary table names. TDSQLagent is a temporary table for scaling, and tdsql_sub tables are subtables for hash-list and hash-range. Therefore, it is recommended not to set the source's tables to be synced with these names.
Sync Feature	Currently, the primary key conflict resolution policy only supports conflict overwrite. For both full and incremental stages, primary key data conflicts will be handled by conflict overwrite.
Source Database Impact	1. DBbridge occupies certain source database resources while performing full data sync, which may lead to increased load in the source database and add to its stress. If your database configuration is low, it's recommended to perform this during the business off-peak period. 2. During data sync, DBbridge will use the account that executes the sync task to write into the system database `__tencentdb__` in the source database, recording transaction marker ID and other metadata. It's necessary to ensure that the source database has read-write permissions for `__tencentdb__` . To ensure that data comparison issues can be traced, the `__tencentdb__` system database will not be deleted from the source database after the sync task is completed The space occupied by the `__tencentdb__` system database is very small, approximately one thousandth to one ten-thousandth of the source database's storage space (for example, if the source database is 50 GB, then the `__tencentdb__` system database is about 5 MB - 50 MB). Moreover, by using a single thread and a waiting connection mechanism, it has almost no impact on the source database's performance and will not compete for resources. 3. By default, a lock-free sync method is used. The full data export stage does not place a global lock (FTWRL) on the source database; it only places table locks on tables without a primary key.
Operation Restrictions	Please do not perform the following operations during sync, as they will cause the sync task to fail. 1. During the full data export stage, please do not execute any DDL operations that change the database or table structure in the source database.

	2. During the sync task, do not modify or delete user information (including username, password, and permissions) and port numbers in both the source and target databases. 3. Do not clear Binlogs in the source database.
Data Type	1. Modifying the primary key is not supported during incremental sync, including the primary key column, partition table distribution key, comments on primary key columns, adding, deleting, or modifying column fields and their lengths. 2. Geometry-related data types are not supported, and tasks will report an error when encountering such data types. 3. For tables with floating point types, the precision inconsistency between full and incremental sync may cause the sync result's precision to be inconsistent. 4. During the incremental sync process, if the source database generates Binlog statements in the STATEMENT format, it will lead to sync failure. 5. When the source TDSQL MySQL uses the MariaDB 10.1.x kernel, using the timestamp type without specifying precision (e.g., timestamp(3)) is not supported; otherwise, the DTS task will report an error. To resolve it, remove the precision and then recreate the task.
Transaction	1. Scenarios that include both DML and DDL Statements in one transaction are not supported; encountering such a scenario will result in an error. 2. When the source database is TDSQL MySQL (Kernel MariaDB 5.6), XA Transactions are not supported; encountering XA Transactions will result in an error.
HA Switch and Scaling	1. If the source database is a non-GTID Database, DBbridge does not support source HA switchover. Once the source TDSQL MySQL undergoes a switchover, it may cause DBbridge incremental sync interruption. 2. If the source database is a self-built database, and SET connections are used with TDSQL MySQL, when the sync task starts, and the source adds or deletes a SET node, the DBbridge sync task will report an error. Users are required to modify the SET configuration information in DBbridge (to keep it consistent with the actual SET on the source), and then restart the task. Only then can the information about the newly added or deleted SET be synced.
Partition Table Sync	1. The full volume stage supports the sync of primary/secondary partition tables, but the partition syntax needs to comply with the TDSQL MySQL standards. Primary Hash partition tables only support creation through the shardkey method. The key syntax for creating a partition table in TDSQL MySQL is as follows, for detailed syntax please refer to TDSQL MySQL Table Creation Syntax Example. Primary Hash partition: shardkey Primary Range partition: TDSQL_DISTRIBUTED BY RANGE Primary List partition: TDSQL_DISTRIBUTED BY LIST Primary Hash partition + secondary Range/List partition: shardkey + PARTITION BY RANGE/LIST Primary Range partition + secondary Range/List partition: TDSQL_DISTRIBUTED BY RANGE + PARTITION BY RANGE/LIST Primary List partition + secondary Range/List partition: TDSQL_DISTRIBUTED BY LIST + PARTITION BY RANGE/LIST

	2. During the incremental sync stage, it is not supported to intensively create a secondary partition table, then drop a secondary partition table, and then create a secondary partition table again, as this might lead to task exceptions due to table type conflicts. Dropping a non-existent secondary partition table before creating a secondary partition table might cause a deadlock, with no error reported from the task, necessitating manual unlocking. 3. TDSQL MySQL sync to MySQL/MariaDB/Percona link: If the source database to be synced contains secondary partition tables, they will be single tables on the target side after sync.
Consistency Check	1. The scope of data consistency check is limited to comparing the selected database objects in the source database with those synced to the target database. If users write data to the target database during the sync task, this data is not included in the check scope, nor are other advanced objects (such as storage procedures, functions, and views). If Structure Initialization is not selected in the sync task configuration (indicating that table structures are not synced), then table structures are also not checked during the consistency check. 2. The current check task does not recognize DDL operations. If DDL operations are performed in the source database during sync, the result of the check will be inconsistent. Users need to initiate a new check task to obtain accurate comparison results. 3. In the consistency check task, the timeout limit for DTS to query data from the source or the target is 10 minutes per query. This applies to each block check query, row check query, etc. If a single query exceeds 10 minutes (for example, when querying large tables from the source), it will cause the check task to fail. 4. If some DMLs are selected or Where conditions are set for filtering in the sync task configuration, it will cause inconsistencies between the source and target databases. Therefore, consistency checks are not supported. To perform a consistency check, all DMLs and DDLs must be selected. 5. Data check supports unidirectional sync only; it does not support complex topologies like many-to-one or one-to-many. 6. The following configuration in a sync task may lead to inconsistent check results. Please be aware of this when creating a check task. If the 'Full Data Initialization' option is not selected during data initialization, there may be inconsistencies between the source and the target data, potentially leading to discrepancies in the data check results. 7. A data consistency check task may increase the load in the source database instance. Therefore, you need to perform such tasks during the business off-peak period. 8. A data consistency check task can be executed repeatedly, but a DBbridge task can initiate only one consistency check task at any time. 9. Complete comparison and sampling comparison require the tables to have a primary key or unique key; otherwise, they will be skipped and not checked. Row count check does not require a primary key or unique key. 10. If the user chooses to end the sync task before the data consistency check is complete, the consistency check will fail. 11. Consistency checks require the use of the account executing the sync task to create a table CRC_xxx_CMP_xxx in the source database, which is used to record the data comparison information during the sync task.

Sync Operation Guide

Last updated：2024-07-08 17:34:59

Overview

This document describes how to use the data sync feature of DTS to sync data from TDSQL for MySQL to TDSQL for MySQL. The supported scenarios are as follows:

Onto Tencent Cloud: Sync from self-built TDSQL for MySQL to Tencent Cloud TDSQL for MySQL

Onto Tencent Cloud: Sync from Tencent Cloud TDSQL for MySQL to Tencent Cloud TDSQL for MySQL

Off Tencent Cloud: Sync from Tencent Cloud TDSQL for MySQL to self-built TDSQL for MySQL.

Note

During full data sync, DTS consumes certain source database resources, which may increase the load and pressure of the source database. If your database configuration is low, we recommend you sync the data during off-peak hours.

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

During data sync, DTS will use the account that executes the sync task to write the system database `__tencentdb__` in the source database to record the data comparison information during the sync task.

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

The source and target databases must meet the requirements for the sync feature and version as instructed in Databases Supported by Data Sync.

Permissions required of the source database:

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SELECT ON *.* TO 'mig
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'migration account'@'%';
FLUSH PRIVILEGES;
```

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.

Environment Requirements

Type	Environment Requirements
Requirements for the source database	The source and target databases can be connected. 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. 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` . 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 sync, all 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 must have sufficient storage space. If you select **Full data initialization** as the initialization type, the target database space must be at least 1.2 times the space of databases/tables to be synced in the source database. The `max_allowed_packet` parameter of the target database must be set to 4 MB or above.

Directions

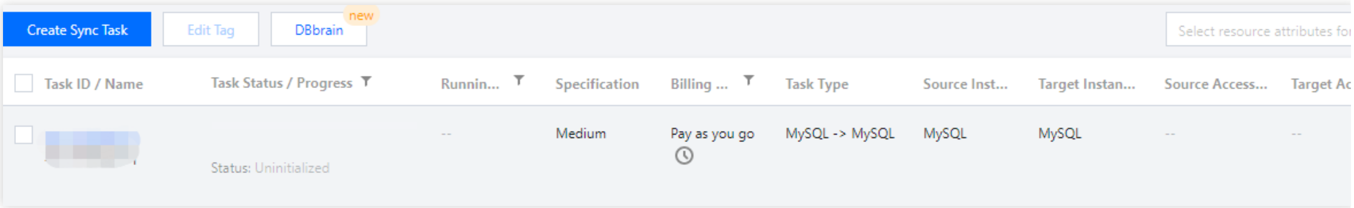
The operations for migrating self-built TDSQL for MySQL to cloud TDSQL for MySQL are basically the same as when going off the cloud. The following is an example of migrating self-built TDSQL MySQL to Tencent Cloud TDSQL MySQL.

1. Log in to the data sync task purchase page, select appropriate configuration items, and click **Buy Now**.

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing are supported.
Source Instance Type	Select TDSQL for MySQL, which cannot be changed after purchase. TDSQL for MySQL is divided into three kernel versions (MySQL/MariaDB/Percona). Here, select **TDSQL for MySQL** as there is no need to distinguish the kernel versions.
Source Instance Region	Select the source region of the DTS task. If the source database is TencentDB, select its region here. If the source database is self-built, select the region closest to it, so that DTS can choose the optimal sycn path to reduce the sycn time.
Target Instance Type	Select TDSQL for MySQL, which cannot be changed after purchase.
Target Instance Region	Select the target instance region, which cannot be changed after purchase.
Specification	Currently, only the Standard Edition is supported.

2. After successful purchase, return to the data sync list, and you can see the newly created data sync task. You need to configure it before you can use it.

3. In the data sync task list, click **Configure** in the **Operation** column to enter the sync task configuration page.



☐ Task ID / Name	Task Status / Progress	Runnin...	Specification	Billing ...	Task Type	Source Inst...	Target Instan...	Source Access...	Target Ac
☐ [redacted]	Status: Uninitialized	--	Medium	Pay as you go	MySQL -> MySQL	MySQL	MySQL	--	--

4. On the sync task configuration page, configure the source and target databases and their accounts and passwords, test the connectivity, and click **Next**.

Setting Items	Parameter	Description
Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.

	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	The source database type selected during purchase, which cannot be changed.
	Source Instance Region	The source instance 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: The source and target databases are both deployed in Tencent Cloud VPCs. To use the VPC access type, submit a ticket for application.
	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 - Proxy:	Enter the proxy address and port.
	Node - SET	Enter the TDSQL for MySQL shard node IP. You need to enter a node IP each shard, and separate multiple nodes by line break. **If the source database is self-built, when the SET node is configured for the first time, the SET node cannot be added or deleted once the connection is saved. If you need to add or delete the SET node, you need to create the sync task again.**
	Account	Account of the source instance, which must have the required

		permissions.
	Password	Password of the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **Database**. If the source instance selects **Database** for access, the target instance selects "Public Network/Self-Build on CVM/Direct Connect/VPN Access/Database/CCN".
	Instance ID	Select the ID of the target instance.
	Account	Account of the target instance, which must have the required permissions.
	Password	Password of the target instance account.

5. On the **Set sync options and objects** page, set the data initialization, data sync, and sync object options and click **Save and Go Next**.

Note：

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

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

Database/Table mapping: Hover over the right side of a selected object, and the **Edit** icon will be displayed. Click it and then enter a mapping name in the pop-up window.

Setting Items	Parameter	Description
Data Initialization Option	Initialization Type	Structure initialization: Table structures in the source instance will be initialized into the target instance before the sync task runs. Full data initialization: Data in the source instance will be initialized into the target instance before the sync task runs. If you select **Full data initialization**, you need to create the table structure in the target database in advance. Both options are selected by default, and you can deselect them as needed. If you select **Full data initialization** only, you need to create the table structures in the target database in advance.

	If Target Already Exists	Precheck and report error: If a table with the same name exists in both the source and target databases, an error will be reported, and the task will stop. Ignore and execute: Full and incremental data will be directly added to tables in the target instance.
Data Sync Option	Conflict Resolution Method	Overwrite: If a primary key conflict is found during data sync, the system will use the primary key record in the source database to overwrite that in the target database.
	SQL Type	The following operations are supported: INSERT, DELETE, UPDATE, and DDL.
Sync Object Option	Database and Table Objects of Source Instance	Select the objects to be synced. You can select databases and tables.
	Selected Object	It displays the selected sync objects, and database/table mapping is supported.

6. On the task verification 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.
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 sync task list, and you can see that the task has entered the **Running** status.

Note：

You can click **More** > **Stop** in the **Operation** column to stop a sync task. Before doing so, ensure that data sync has been completed.

8. (Optional) You can click a task ID to enter the task details page and view the task initialization status and monitoring data.

Sync from MySQL/MariaDB/Percona to TDSQL for MySQL

Last updated：2023-09-05 11:18:25

The steps of data sync from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL for MySQL are basically the same as those described in Sync from TDSQL for MySQL to TDSQL for MySQL.

Sync from MySQL/MariaDB/Percona to TDSQL for TDStore

Last updated：2023-11-16 17:23:25

DTS supports data sync to TDSQL TDStore from the following source database types:

Self-built MySQL, third-party cloud MySQL, and TencentDB for MySQL.

Self-built MariaDB and TencentDB for MariaDB.

Self-built Percona.

The differences between syncing data from MySQL/MariaDB/Percona to TDSQL TDStore and syncing from MySQL/MariaDB/Percona to TencentDB for MySQL are as follows:

Foreign key dependencies are not supported. If the selected sync object contains foreign key dependency data, a verification error will be reported, and you need to deselect such objects.

Other link capabilities are the same. For more information, see Sync from MySQL/MariaDB/Percona to TencentDB for MySQL.

Sync to TDSQL TDStore

Sync from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDSQL TDStore

Last updated：2024-08-13 17:25:14

The steps of data sync from MySQL/MariaDB/Percona/TDSQL-C MySQL to TDSQL TDStore are basically the same as those described in Sync from MySQL/MariaDB/Percona to TencentDB for MySQL.

MySQL Series Advanced Sync Operations

Syncing Advanced Object

Last updated：2023-03-03 15:00:24

Overview

The advanced objects that can be synced with DTS include functions and procedures. The sync of advanced objects is a one-time operation: only advanced objects already in the source database before the task start can be synced, while those added to the source database after the task start will not be synced to the target database.

Note

Currently, advanced objects can be synced among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Notes

We recommend that you do not rename tables when syncing advanced objects; otherwise, the sync may fail.

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

When procedures and functions are synced, DTS will check whether `user1` corresponding to `DEFINER` (`[DEFINER = user1]`) in the source database is the same as the sync 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 sync, and set the `DEFINER` in the target database to the sync account `user2` (`[DEFINER = sync account user2]`).

For cross-version sync, 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 synced.

If `'''` (escaping) exists in the definitions of the selected advanced objects, a MySQL limit will be encountered, which will cause the migration task to fail. We recommend you not select such advanced objects.

Directions

1. On the **Set sync options and objects** page in the data sync task, set the **Advanced Migration Object** feature. Advanced objects are selected by default. You can deselect unwanted advanced object options.

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

Last updated：2023-03-02 11:41:32

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.

Notes

Sync links currently supporting data consistency check are as follows:
MySQL/MariaDB/Percona/TDSQL-C for MySQL > MySQL
MySQL/MariaDB/Percona > MariaDB
MySQL/MariaDB/TDSQL-C for MySQL > TDSQL-C for MySQL

Notes

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

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 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 **stop** a sync 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 sync, the check result will be inconsistent with the actual data, and you need to initiate another check task to get the accurate comparison result.

Data check is supported only for one-way and two-way syncs, but not those with complicated topologies such as many-to-one, one-to-many, ring, and star sync.

If only certain DML and DDL items are selected in the sync task configuration, or the `Where` condition is used for filtering, data in the source and target databases will become inconsistent, and consistency check will not be supported. Therefore, you need to select all DML and DDL items to perform a consistency check.

Note that when a data check task is created, the check result may be inconsistent with the actual data if the sync task is configured as follows.

Full data initialization is not selected for **Initialization Type**. In this case, the data in the source and target databases may be inconsistent, causing the check result to be **Inconsistent**.

Ignore is selected for **Primary Key Conflict Resolution**. In this case, the data in the source and target databases may become inconsistent upon a conflict, causing the check result to be **Inconsistent**.

For existing tasks created before the release of the consistency check feature on January 12, 2023, check tasks cannot be directly created as the DTS version is too earlier. To create a check task, submit a ticket for upgrade.

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 sync 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

1. Log in to the DTS console.

2. On the **Data Sync** page, select the target sync task, select **Operation** > **More**, and click **Create Data Consistency Check Task**.
3. Click **Create Data Consistency Check Task**.

Notes

A data consistency check task can be created only when the **Source-Target Database Data Gap** is smaller than 100 MB. If the button is grayed out, the sync task status does not meet the requirement; for example, only certain DML or DDL items are selected in the task configuration, the `Where` condition is set for filtering, the task fails, the source-target database data gap is greater than 100 MB, or the sync topology is complicated.

4. In the pop-up window, click **OK**.
5. After configuring data consistency check parameters, click **Create and Start Consistency Check Task**.

Parameter	Description
Check Object	All sync objects: Check all the objects selected in the sync task. Custom: Check the selected sync 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.
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 **Data Sync** page, select the target sync task, select **Operation** > **More**, and click **Create Data Consistency Check Task**.
2. In the **Operation** column, click **View**.

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.

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.
```

Setting SQL Filter Policy

Last updated：2023-02-13 11:24:22

Overview

When configuring a sync task, you can set a SQL sync policy, so that only SQL data meeting the set conditions will be synced to the target database. In this way, you can flexibly configure the sync schemes between multiple databases or split data tables as needed.

DTS supports the following SQL filters:

- DML filter: INSERT, UPDATE, and DELETE.
- DDL filter: You can select the specific DDL operation, such as `CREATE TABLE` , `CREATE VIEW` , and `DROP INDEX` .
- WHERE conditional filter: You can customize a filter for a single table.

  Note：
  Currently, you can filter by DDL and WHERE condition for the following data sync links:

  - MySQL > MySQL/MariaDB/TDSQL-C for MySQL
  - MariaDB > MySQL/MariaDB
  - Percona > MySQL/MariaDB
  - TDSQL-C for MySQL > MySQL/TDSQL-C for MySQL

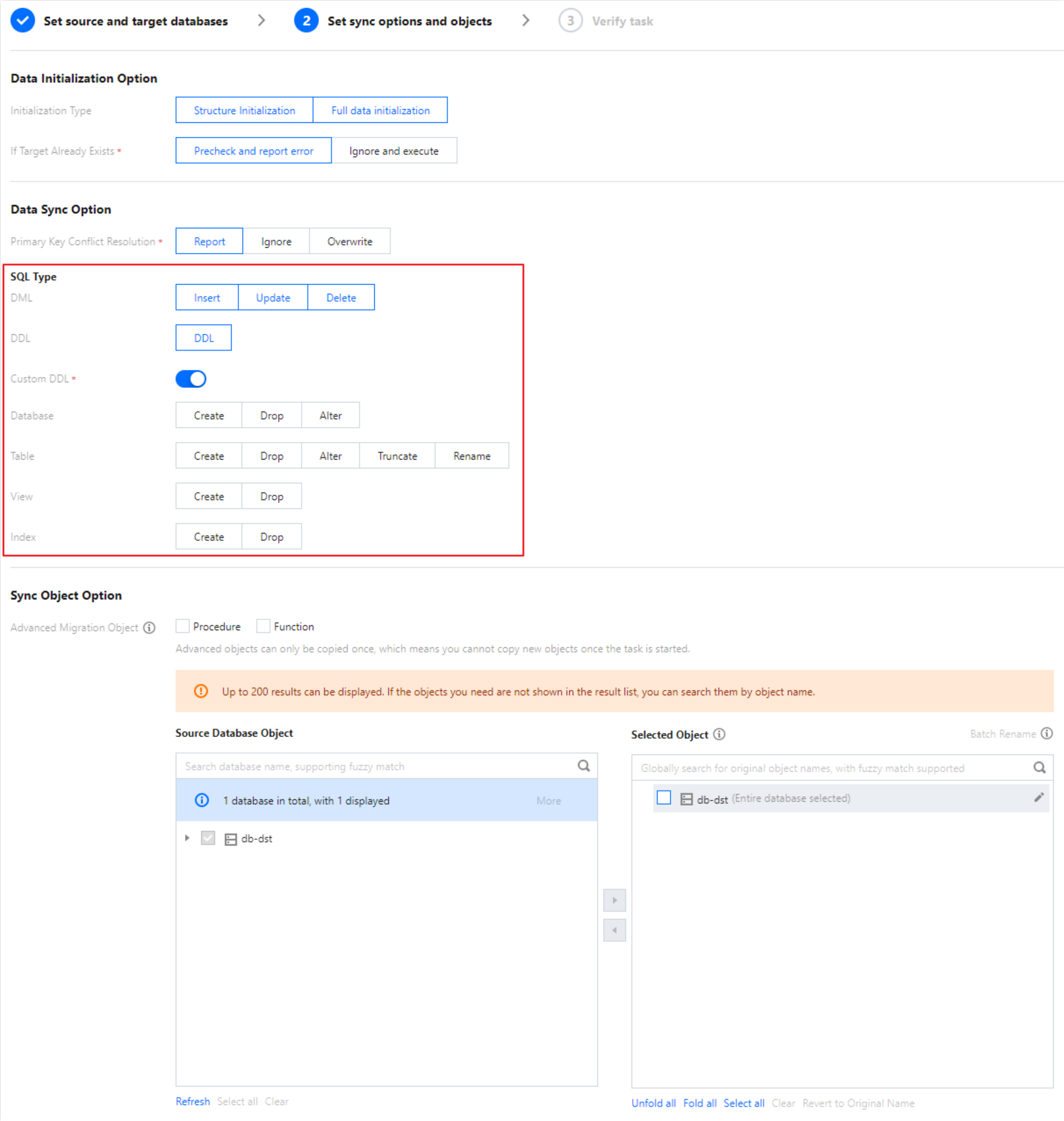
Restrictions

You can set a WHERE conditional filter for a single table. To filter multiple tables, set a filter for each of them.

Directions

1. Select different SQL policies on the **Set sync options and objects** page in Sync Task.
   - DML: INSERT, UPDATE, and DELETE.

- DDL:
    - If you select **DDL** but not **Custom DDL**, all the DDL operations in the source database will be synced to the target database.
    - If you select both **DDL** and **Custom DDL**, you can select specific DDL policies, and only the selected policies will be synced to the target database.
    - If you don't select **DDL**, all the DDL operations will not be synced to the target database.



Set source and target databases
2 Set sync options and objects
3 Verify task

Data Initialization Option

Initialization Type: Structure Initialization, Full data initialization

If Target Already Exists *: Precheck and report error | Ignore and execute

Data Sync Option

Primary Key Conflict Resolution *: Report | Ignore | Overwrite

SQL Type

DML: Insert, Update, Delete

DDL: DDL

Custom DDL *: ●

Database: Create | Drop | Alter

Table: Create | Drop | Alter | Truncate | Rename

View: Create | Drop

Index: Create | Drop

Sync Object Option

Advanced Migration Object ⓘ: ☐ Procedure ☐ Function

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

Up to 200 results can be displayed. If the objects you need are not shown in the result list, you can search them by object name.

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
Batch Rename
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

2. In **Selected Object** on the right of **Sync Object Option**, select the target table and click the edit button on the right to set the WHERE conditional filter.

Note：

Rules for setting a WHERE conditional filter:

- For INSERT operations, the data to be inserted must meet the condition for the filter rule to take effect. For DELETE operations, the data to be deleted must meet the condition for the filter rule to take effect. For UPDATE operations, both the data before and after the update must meet the condition for the filter rule to take effect.
- The rule entered must be a valid Boolean expression, which is more rigorous than MySQL. Some MySQL-supported syntaxes that may trigger a warning (such as comparing a string with a number, and c1 + c2 < "abc") are not supported here. The rules and priority levels of logical, arithmetic, and comparison operations are the same as those in MySQL. The priority levels of operations can be changed through parentheses. Even if there is a NULL, the operation rules are still the same as those in MySQL. DTS will verify the entered conditional filter rule and issue an alert if the rule is invalid.
- Basic operation rules are as follows:
- A column name can be imported as a variable.
- Logical operations (NOT, AND, OR, XOR, &&, and ||) are supported.
- Numerical types (signed/unsigned integer types TINYINT, SMALLINT, MEDIUMINT, INT, and BIGINT, floating point types FLOAT and DOUBLE, and exact type DECIMAL) and their arithmetic operations (+, -, *, /, %, DIV, and MOD) and comparison operations (=, !=, >, <, >=, <=, <>, and <=>) are supported.
- String types (CHAR and VARCHAR) and their comparison operations (binary comparison) are supported.
- Data types (DATE, DATETIME, and TIMESTAMP) and their comparison operations are supported.
- Time type (TIME) and its comparison operations are supported. A date/time-type variable can be compared with a string. In this case, the string will be converted into a date/time-type constant for comparison according to the date/time comparison rules.
- The default time zone of the entered `TIMESTAMP` is UTC+0, and it will be converted to a time in UTC+0 for comparison.

  Example: If the entered filter rule is 'c1 > "2016-10-01 09:00:00"', where `c1` is a TIMESTAMP-type column, then "2016-10-01 09:00:00" will be parsed as a time in the UTC+0 time zone, i.e., "2016-10-01 09:00:00 +00:00" for comparison, and `c1` will also be converted into a time in UTC+0.

3. Click **OK**.
4. Click **Save and Go Next** to proceed with the subsequent sync task process.

Database/Table Renaming

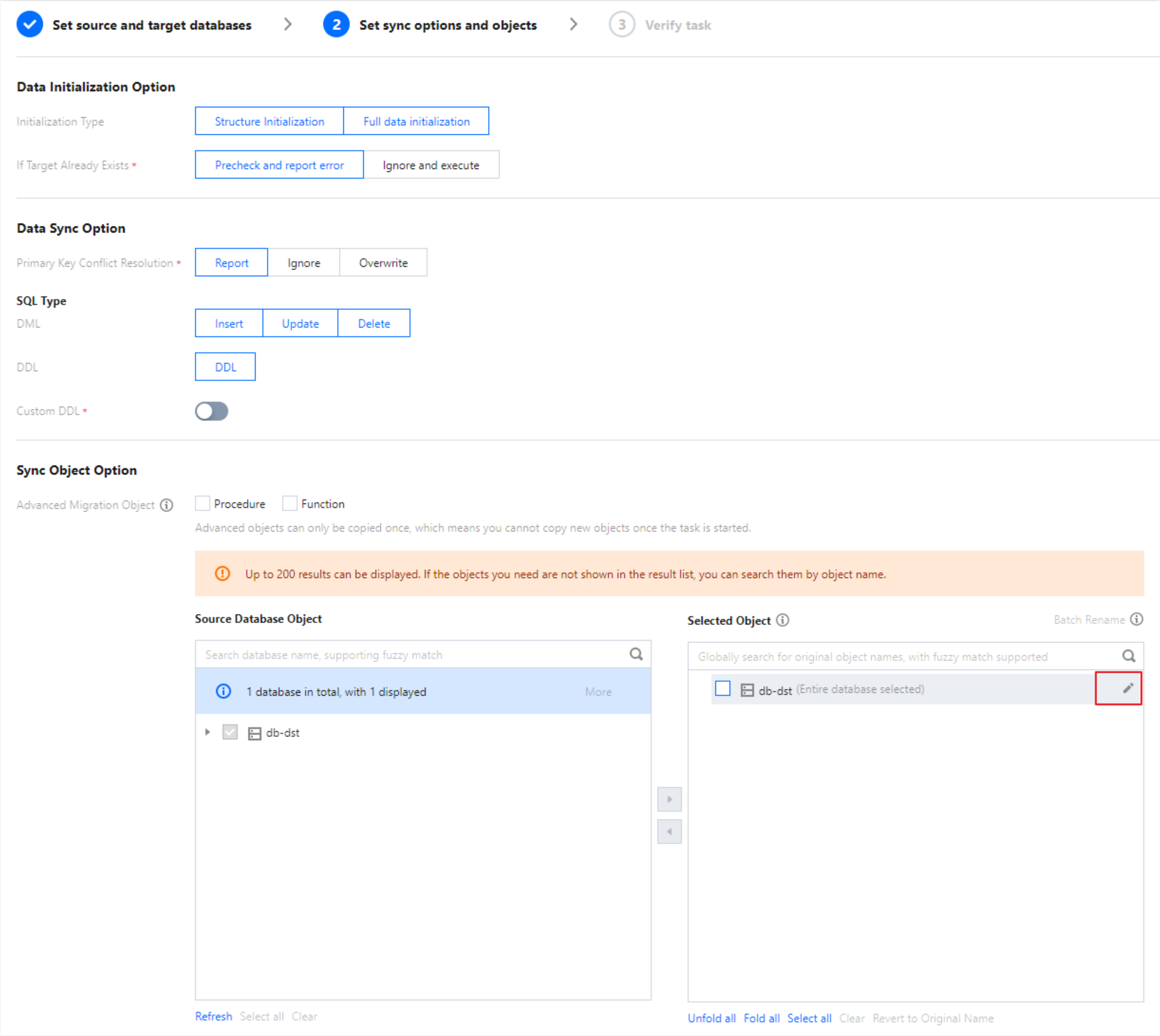
Last updated：2022-12-01 10:46:48

Overview

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

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, and create a sync task.
2. In the **Set source and target databases** step, in **Selected Object** on the right, hover over the object to be modified and click the displayed **Edit** icon.


Set source and target databases
2 Set sync options and objects
3 Verify task
Batch Rename
More
Refresh Select all Clear
Unfold all Fold all Select all Clear Revert to Original Name

Data Initialization Option

Initialization Type: Structure Initialization, Full data initialization

If Target Already Exists *: Precheck and report error | Ignore and execute

Data Sync Option

Primary Key Conflict Resolution *: Report | Ignore | Overwrite

SQL Type

DML: Insert, Update, Delete

DDL: DDL

Custom DDL *: off

Sync Object Option

Advanced Migration Object: ☐ Procedure ☐ Function

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

Up to 200 results can be displayed. If the objects you need are not shown in the result list, you can search them by object name.

Source Database Object

Search database name, supporting fuzzy match

1 database in total, with 1 displayed

☑ db-dst

Selected Object

Globally search for original object names, with fuzzy match supported

☐ db-dst (Entire database selected)

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

Syncing Online DDL Temp Table

Last updated：2023-11-16 17:19:47

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 sync 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 sync objects, so that the object table can be synced 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 syncing 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 syncing temp tables `_table name_new` and `_table name_old` to the target database.

Application Scope

Links supported for syncing online DDL temp tables are as follows:

MySQL/MariaDB/Percona/TDSQL-C for MySQL > MySQL
MySQL/MariaDB/Percona > MariaDB
MySQL/MariaDB/TDSQL-C for MySQL > TDSQL-C for MySQL
MySQL/MariaDB/Percona/TDSQL-C for MySQL > Kafka
MySQL/MariaDB/Percona > TDSQL for TDStore

Restriction

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 Sync** on the left sidebar, and create a sync task.
2. In the **Set sync options and objects** step, enable **Sync Online DDL Temp Table** and select an option for **Online DDL Tool**.

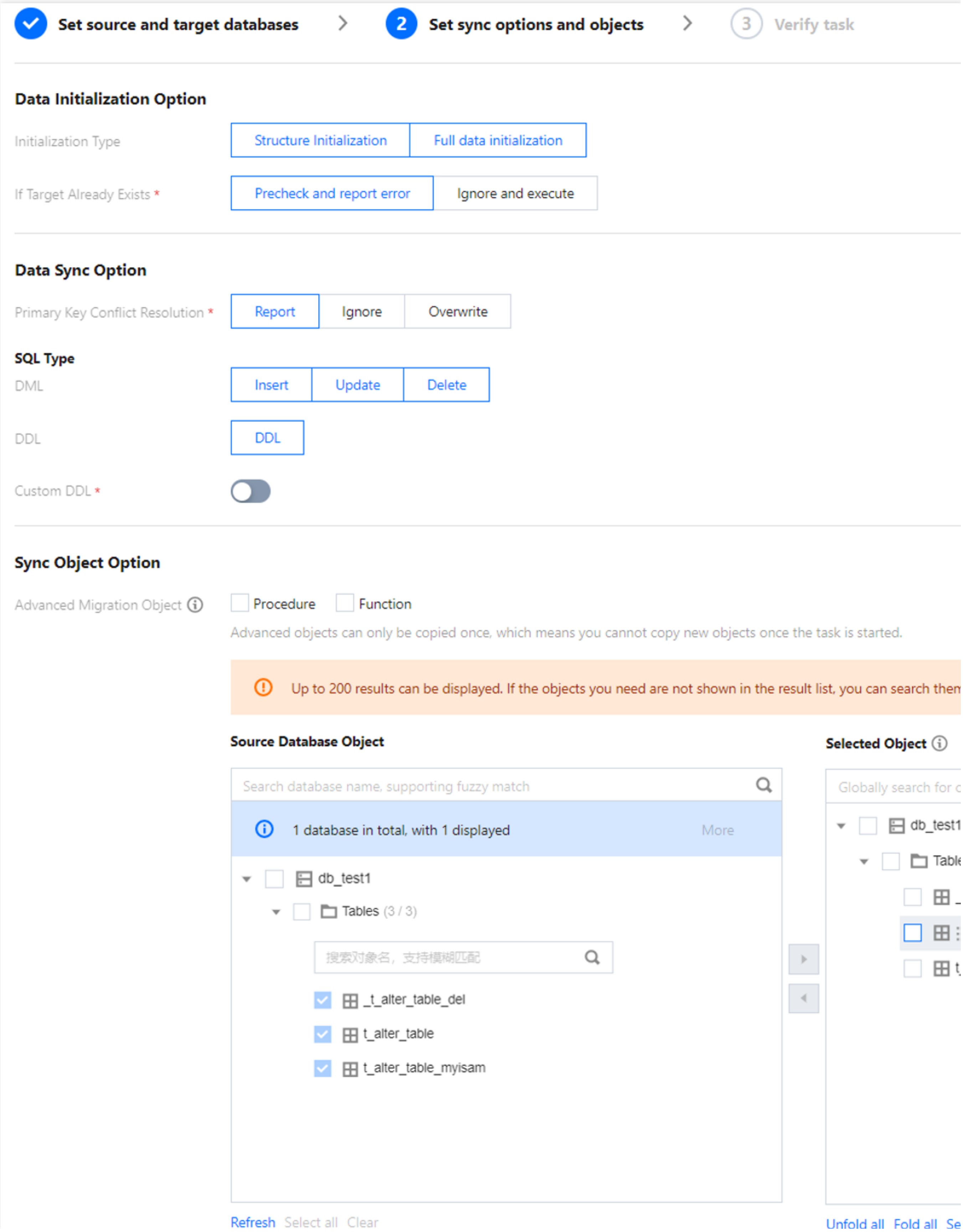
If you select **gh-ost**, DTS will migrate the temp tables (`_table name_ghc` , `_table name_gho` , and `_table name_del`) generated by the gh-ost tool to the target database.

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 select the specified table, and then the button **Sync Online DDL Temp Table** will be displayed below. If you select databases or the "Tables" folder, the button won't 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 sync. In this case, if you just choose the table itself as the **Sync Object**, data in temp tables created by online DDL updates cannot be migrated to the target database. Instead, you must select the entire database or instance where the table to be synced resides.



Set source and target databases > 2 Set sync options and objects > 3 Verify task

Data Initialization Option

Initialization Type: Structure Initialization | Full data initialization

If Target Already Exists *: Precheck and report error | Ignore and execute

Data Sync Option

Primary Key Conflict Resolution *: Report | Ignore | Overwrite

SQL Type

DML: Insert | Update | Delete

DDL: DDL

Custom DDL *

Sync Object Option

Advanced Migration Object: ☐ Procedure ☐ Function

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

Up to 200 results can be displayed. If the objects you need are not shown in the result list, you can search them

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

Refresh Select all Clear

Selected Object

Globally search for c

☐ db_test1

☐ Table

Unfold all Fold all Se

Column Filtering

Last updated：2024-07-08 17:35:43

Overview

During data sync, when users select objects, they can choose the level of repository objects and table objects, or even the column level. This article introduces column-level data sync.

Application Scope

Data sync links between MySQL, MariaDB, Percona, and TDSQL-C for MySQL.

Restrictions

1. Columns of primary keys and unique keys do not support filtering. The system will synchronize columns of primary keys and unique keys to the target by default. In special cases, if primary keys or unique keys are filtered, pre-validation will result in an error.
2. If the filtered column is a composite primary key, foreign key or partition column, or if the filtered column includes an index, then the source does not support DML operation on that column; otherwise, the task may report an error.
3. The interaction explanation between column filtering and view is as follows:
If the view is selected in the sync object, then the related column data of the view needs to be synced as well and cannot be filtered; otherwise, the task will report an error.
If the view is selected in the sync object, and subsequently, the source database associates the created view with the system database table, then column filtering cannot be used in the sync task at the same time; otherwise, the task will report an error.
4. Special explanations for the interaction between the Rename operation and the column filtering feature are as follows:
Table A underwent column filtering, the source then deleted Table A, and recreated another Table A with an identical structure. In this scenario, the created Table A will still undergo column filtering.
Table A underwent column filtering, then the source deleted Table A, and renamed Table B as Table A, with Tables A and B having identical table structures. In this scenario, the sync result for Table B will not undergo column filtering.
5. After column filtering is configured, modifying the sync configuration is not supported. Similarly, when the sync configuration is modified, filtering columns is not supported.
6. The interaction between column filtering and Where condition filtering is as follows:

If a table has undergone column filtering and the user also sets Where rules for this table, the user must ensure that the column names in the Where rules input are included in the table to be synced; otherwise, the Where rules can be set successfully, but subsequent task initiation will result in an error.

7. If the column X of Table A has been filtered, and the user executes the 'CREATE TABLE table_B like table_A' statement on the source, it is recommended to also filter the column X of Table B. In this scenario, DTS's sync of Table B will follow Table A's filtered structure. If the column X of Table B is not filtered, subsequent data updates to the column X of Table B on the source side will cause the DTS task to fail.
8. If the unique key column in the table object includes a function, then when the sync object checks that table, it cannot expand to the column dimension and cannot perform column filtering.
9. Only select scenarios that sync data instead of the table structure. This scenario requires users to create the same table structure on the target before starting the DTS task. If the task also needs column filtering settings, then the table structure created by the users on the target should be after column filtering; otherwise, sync to the target might result in errors.

Directions

1. Log in to the DTS console, choose **Data Sync** Page in the left sidebar to create a sync task.
2. In **Set source and target databases**, you can select table-level and column-level data in **Source Database Object** on the left; after they are checked, click ▸, and the objects to be synced will be displayed on the right.

MySQL series Pre Validation Failed Processing

Database Connection Check

Last updated：2024-07-08 17:48:28

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.

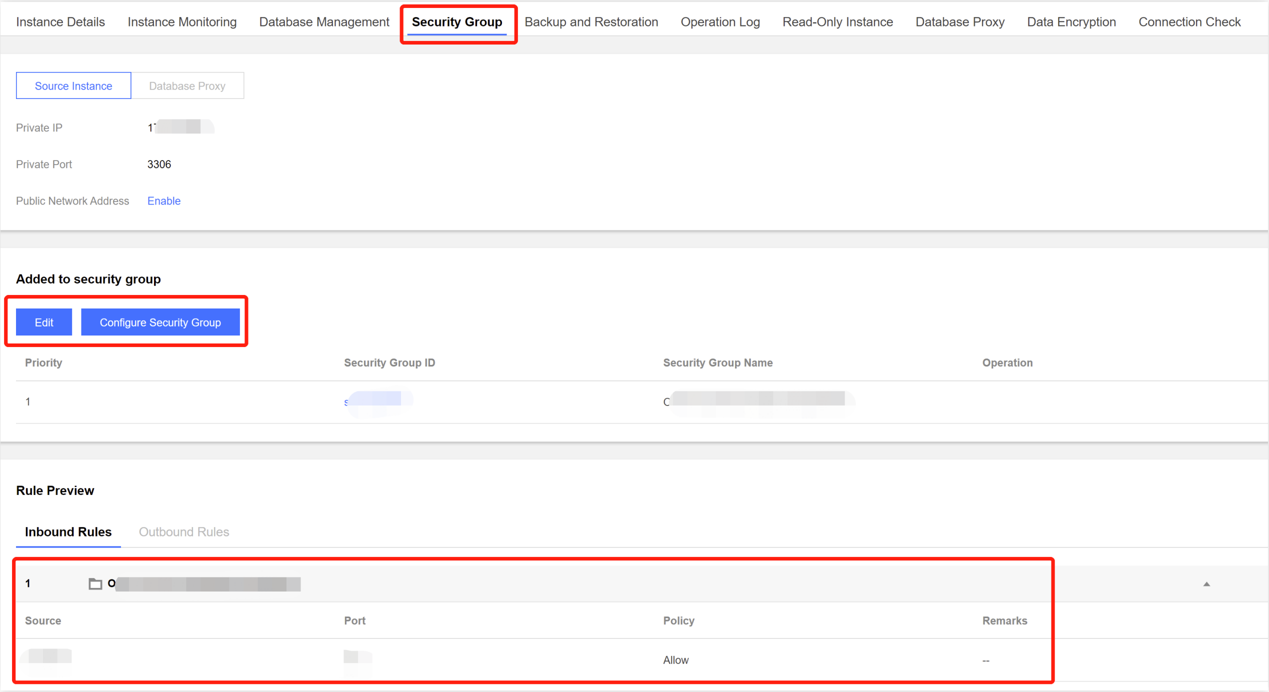
Windows: open Control Panel and find the Windows Defender Firewall and check whether firewall policies are configured.

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.


Instance Details
Instance Monitoring
Database Management
Security Group
Backup and Restoration
Operation Log
Read-Only Instance
Database Proxy
Data Encryption
Connection Check
Source Instance
Database Proxy
Edit
Configure Security Group
Inbound Rules
Outbound Rules

Private IP [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 [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 w
root@%           // Authorize users to access all IPs, which should include the SNA
```

You can verify as follows:

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

SQL Server

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

PostgreSQL

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

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

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

MongoDB

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

Fix

MySQL

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

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

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

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

Note:

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

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

2.2. Restart the database.

```
service mysqld restart
```

2.3. Check whether the configuration takes effect.

```
netstat -tln
```

3. Run the verification task again.

SQL Server

Disable the firewall or trigger.

PostgreSQL

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

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

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

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

3. Run the verification task again.

MongoDB

Configure `bind-address` as instructed in MySQL.

Closed Network Port

Check method

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

MySQL: 3306

SQL Server: 1433

PostgreSQL: 5432

MongoDB: 27017

Redis: 6379

Fix

Open the corresponding database port.

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

Incorrect Database Account or Password

Check method

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

Fix

Modify the data migration task in the DTS console, enter the correct database account and password, and run the verification task again.

Peripheral Check

Last updated：2024-07-08 17:48:28

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 17:48:28

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 17:48:28

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 17:48:28

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 Database Content Conflict Check

Last updated：2024-07-08 17:48:28

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 17:48:28

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 17:48:28

Check Details

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

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

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

`binlog_row_image` must be set to `FULL` .

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

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

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

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

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

Troubleshooting

Enabling binlog

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

If a similar error occurs, fix it as follows:

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

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

Note

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

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

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

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

Note

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

4. Check whether the binlog feature has been enabled.

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

The system will display a result similar to the following:

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

5. Run the verification task again.

Modifying `binlog_format` parameter

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

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

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

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

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

Note

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

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

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

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

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

The system will display a result similar to the following:

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

4. Run the verification task again.

Modifying `binlog_row_image` parameter

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

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

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

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

Therefore, you need to set `binlog_row_image` to `FULL` to make the source database binlog log the full image.

If an error occurs, troubleshoot as follows:

Note

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

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

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

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

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

The system will display a result similar to the following:

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

4. Run the verification task again.

Modifying `gtid_mode` parameter

A global transaction identifier (GTID) uniquely identifies a transaction in the binlog. Using GTIDs can prevent disordered data or source-replica or primary-replica inconsistency due to repeated transaction executions.

GTID is a new feature on MySQL 5.6. Therefore, this problem may only occur on MySQL 5.6 or later versions. DTS only allows you to set `gtid_mode` to `ON` or `OFF` . We recommend that you set it to `ON` ; otherwise, an alarm will be triggered during verification.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The system will display a result similar to the following:

```
mysql> show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
+-------------------------------------+-------+
| Variable_name                       | Value |
+-------------------------------------+-------+
| Ongoing_anonymous_transaction_count | 0     |
+-------------------------------------+-------+
1 row in set (0.00 sec)
```

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

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

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

Note

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

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

7. Run the verification task again.

Modifying `server_id` parameter

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

1. Log in to the source database.

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

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

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

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

The system will display a result similar to the following:

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

4. Run the verification task again.

Deleting `do_db` and `ignore_db` settings

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

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

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

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

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

Note

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

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

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

Note

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

4. Check whether the parameter modification takes effect.

```
show master status;
```

The system will display a result similar to the following:

```
mysql> show master status;
+---------------+----------+--------------+------------------+-------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------+----------+--------------+------------------+-------------------+
| binlog.000011 | 154      |              |                  |                   |
+---------------+----------+--------------+------------------+-------------------+
```

5. Run the verification task again.

Modifying `log_slave_updates` parameter

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

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

Note

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

```
log_slave_updates = ON
```

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

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

Note

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

4. Check whether the configuration takes effect.

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

The system will display a result similar to the following:

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

5. Run the verification task again.

Foreign Key Dependency Check

Last updated：2024-07-08 17:48:28

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.

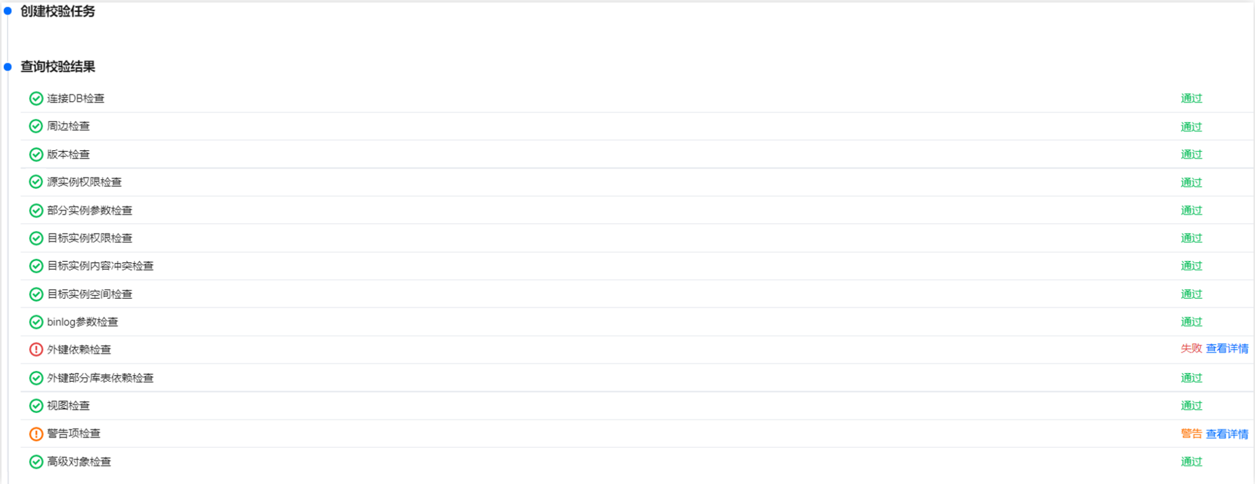
`SET DEFAULT` : When a record is deleted or updated in the parent table, the foreign key column in the child table will be set to the default value, but the InnoDB engine cannot recognize it.

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.



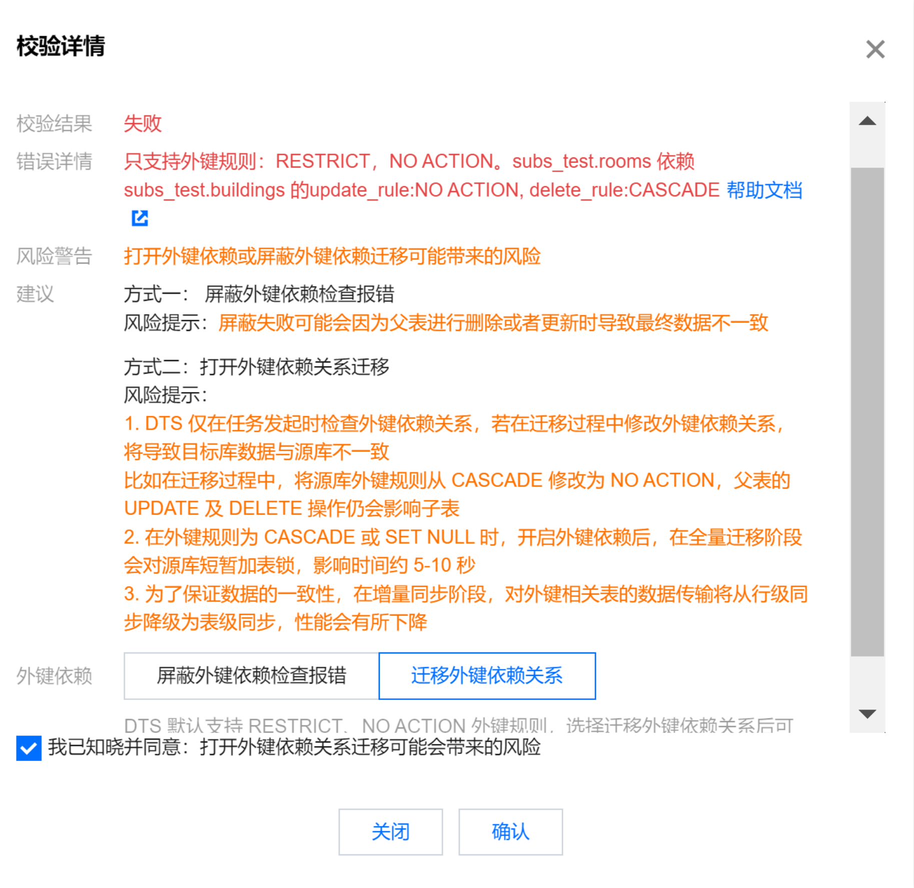
- 创建校验任务
- 查询校验结果

✓ 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:

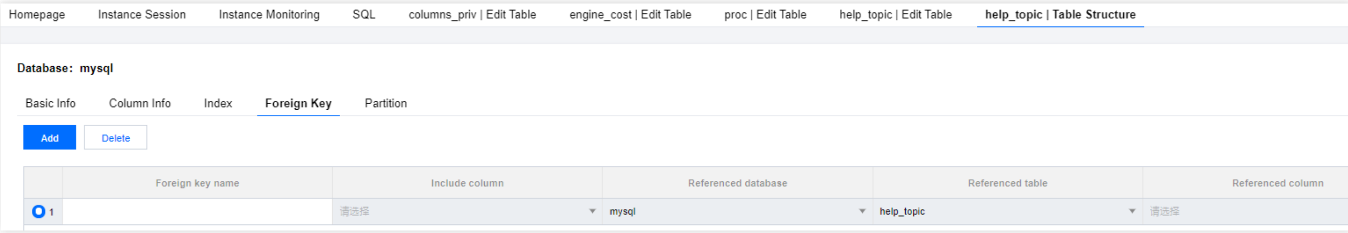
TDSQL for MySQL data migration and sync: Foreign key dependency check can be set only to `NO ACTION` or `RESTRICT` ; otherwise, an error will occur, but you can choose whether to fix the error or ignore it and proceed with the task. You can fix the error as follows:

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.


Homepage
Instance Session
Instance Monitoring
SQL
columns_priv | Edit Table
engine_cost | Edit Table
proc | Edit Table
help_topic | Edit Table
help_topic | Table Structure
Basic Info
Column Info
Index
Foreign Key
Partition
Add
Delete

Database: mysql

	Foreign key name	Include column	Referenced database	Referenced table	Referenced column
○ 1		请选择	mysql	help_topic	请选择

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

Linux

1. Log in to the source database as instructed in Connecting to MySQL Instance.
2. Delete the original foreign key settings.

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

3. Add the foreign key settings again.

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

4. Run the verification task again.

Completing migration objects

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

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

3. Run the verification task again.

View Check

Last updated：2024-07-08 17:48:28

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 17:48:28

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 17:48:28

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 17:48:28

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：2024-07-08 17:48:28

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.

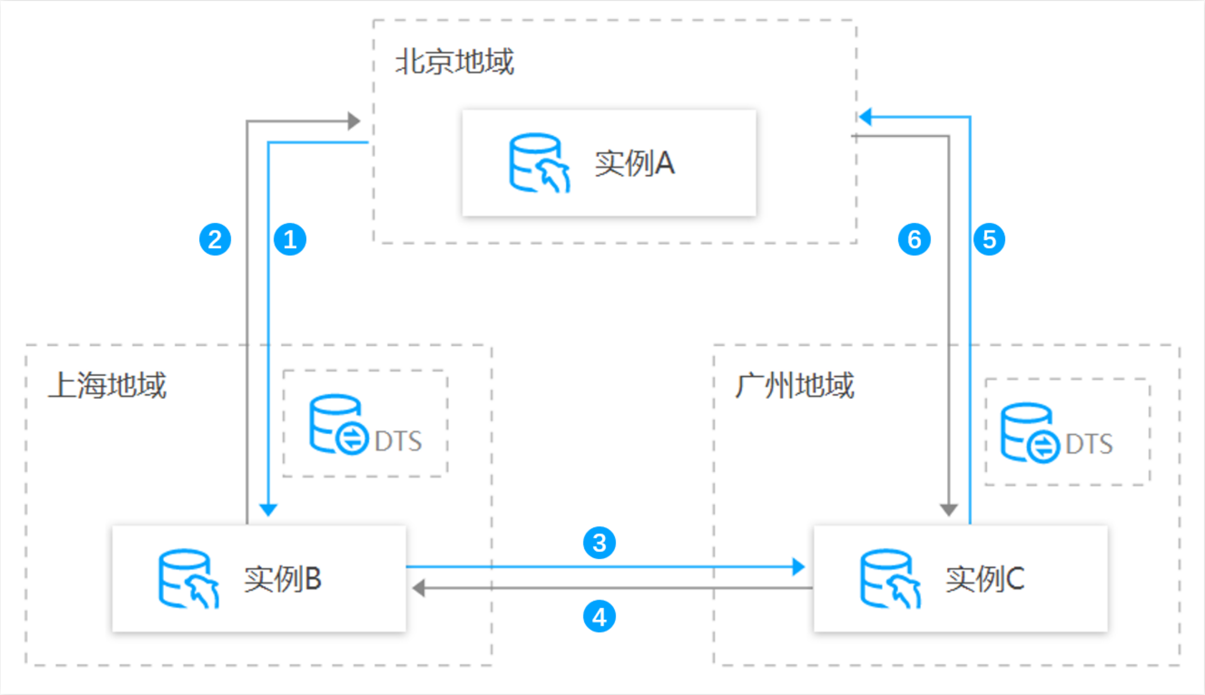
DLL Ring Sync Check for Single Database/Table Object

Last updated：2024-07-08 17:48:28

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.

DLL Sync Conflict Check for Single Database/Table Object in the Same Target

Last updated：2024-07-08 17:48:28

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.

Synchronize to PostgreSQL

Synchronization from PostgreSQL to PostgreSQL

Last updated：2024-04-19 11:05:27

This document introduces how to use Data Transfer Service (DTS) to synchronize data from PostgreSQL databases to TencentDB for PostgreSQL.

The following types of source databases are supported:

Self-built PostgreSQL databases

PostgreSQL databases provided by third-party cloud vendors

TencentDB for PostgreSQL databases

Prerequisites

Ensure the source and target databases meet the sync feature and version requirements. For more information, see Databases Supported by Data Sync.

The account executing the DTS sync task must have the following permissions on the source database: Event Trigger, Replication, CREATE, DROP, ALTER, SELECT, UPDATE, INSERT, and DELETE.

The account executing the DTS sync task must have the following permissions on the target database: CREATE, DROP, ALTER, SELECT, UPDATE, INSERT, and DELETE.

Notes

During full data sync, DTS occupies certain resources of the source database instance. This may increase the load and pressure on the source instance. If your database configuration is low, it is recommended to perform the operation during off-peak hours.

By default, a lock-free method is used, and no global lock is applied to the source database during the sync process.

During data sync, DTS uses the account executing the sync task to create postgres.public.identify and {database}.__tencentdb__sync_*.* in the source database to record task status and configuration information. The target instance uses {database}.__tencentdb__ schema to record sync status information. The space occupied by postgres.public.identify, {database}.__tencentdb__sync_*.*, and {database}.__tencentdb__ schema is only about one thousandth to one ten-thousandth of the storage space of the source database. Therefore, it has no impact on the source database performance and does not compete for resources.

Application Restrictions

1. Sync of the following objects is not supported: ROLE, PROCEDURE, TRIGGER, FUNCTION, RULE, EXTENSION, etc.
2. Sync of public objects and permission changes is not supported.
3. During structure initialization of the full sync stage, foreign keys cannot be imported. The foreign keys are automatically removed before importing.
4. Data sync does not support foreign tables. The full sync stage does not support exporting or importing temporary tables. The incremental sync stage only supports the sync of unlogged tables.
5. When the system checks the plugins during the validation stage before the data sync task starts, if a warning is triggered by the first unmatched plugin, the system does not continue checking the remaining plugins.
6. Sync tasks depend on the max_replication_slots parameter (one replication_slot is occupied by sync task configuration for each database). You can check the used slots by executing the SQL statement select * from pg_replication_slots. If the sum of the number of configured sync databases and replication_slots is larger than max_replication_slots, the attempts to create a new sync task will fail. The user needs to actively end the task so the system can release replication_slots (replication_slots are not released if the failed task is not ended).
7. When the timestamps with time zones are synced from the source instance to the target instance, they will be automatically converted to the corresponding time in the time zone of the target instance.
8. Operations such as source database HA causing the primary node of the source instance to migrate will result in changes to WAL logs. As a result, the sync task will fail, and you need to clear the data already synced to the target database and create a new sync task.

Limits on Operations

Do not perform the following operations during sync, otherwise the sync task will fail.

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

Highly concurrent DDL and DML operations during the incremental sync stage might lead to data conflicts. It is recommended to reduce DDL operations at this stage.

Do not perform operations to clear WAL logs on the source database.

During the incremental sync stage, do not delete system libraries and schemas such as postgres.public.identify, __tencentdb__sync_., and {database}.__tencentdb__ schema.

When both source and target databases are Tencent Cloud databases, do not change the network settings of the database instances. If the original VIPs are deleted due to changes in the source and target instance networks, the sync task will fail or remain in a running state without actually working. In such cases, the sync task cannot be retried and you must manually end the task and create a new task.

Supported SQL Operations for Sync

Operation Type	SQL Operation Statements
DML	INSERT, UPDATE, and DELETE
DDL	CREATE DATABASE, DROP DATABASE, ALTER DATABASE, CREATE SCHEMA, DROP SCHEMA, CREATE TABLE,ALTER TABLE, DROP TABLE, CREATE VIEW, DROP VIEW, CREATE INDEX, DROP INDEX, CREATE SEQUENCE, and DROP SEQUENCE

Note：

The CREATE TABLE table_name AS SELECT statement is not supported.

SQL filters with the WHERE clause are not supported.

Data Types Supported for Sync

Aclitem, Bit, Bool, Box, Bpchar, Bytea, Char, Cid, Cidr, Circle, Date, Float4, Float8, Inet, Int2, Int4, Int8, Interval, Json, Jsonb, Jsonpath, Line, Lseg, Macaddr, Name, Numeric, Oid, Path, Point, Polygon, Record, Text, Tid, Time, Timestamp, Timestamptz, Unknown, Uuid, Varbit, Varchar, XID, Daterange, Int4range, Int8range, Numrange, Tsrange, Tstzrange, Datemultirange, Int4multirange, Int8multirange, Nummultirange, Tsmultirange, Tstzmultirange, XML, Money, Time with time zone, Tsvector, Tsquery, pg_lsn, and txid_snapshot.

Environment Requirements

Type	Environment Requirements
Source Database Requirements	The networks of source and target databases must be interconnected. The requirements for the source database parameters are as follows: Set the wal_level parameter to logical. The source database must meet the following requirement: number of databases to be synchronized + replication_slots <= max_replication_slots. The source database objects must be case-insensitive. Tables in the source database must have a primary key or a composite primary key.
Target Database Requirements	The version of the target database must be no earlier than that of the source database. The target database must have sufficient storage space. If the Initialization Type is set to Full Data Initialization, the space of the target database must be at least 1.2 times the space occupied by the databases and tables to be synchronized.

	The target database cannot have tables, views, or other objects with the same names as those to be synchronized from the source database.

Directions

1. Log in to the data sync purchase page, select the appropriate configuration, and click **Buy Now**.

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing are supported.
Source Instance Type	Select PostgreSQL. It cannot be modified after purchase.
Source Instance Region	Select the region where the source instance is located. It cannot be modified after purchase.
Target Instance Type	Select PostgreSQL. It cannot be modified after purchase.
Target Instance Region	Select the region where the target instance is located. It cannot be modified after purchase.
Specification	Select a specification based on your business needs. The higher the specification, the better the performance. For more details, see Billing Overview.

2. After the purchase is completed, return to the data sync task list. The newly created sync task is displayed. The new task must be configured before it can be used.
3. In the data sync task list, click **Configure** in the **Operations** column to enter the task configuration page.
4. On the sync task configuration page, configure the source instance, target instance, and the accounts and passwords of the instances. After testing connectivity, click **Next**.

Given that the deployment forms and connection types of source databases overlap in multiple scenarios, the sync operations are similar across scenarios. Only typical configuration examples are provided below. Refer to these examples when you configure in other scenarios.

Example 1: Synchronize a local self-built database to TencentDB by using Direct Connect or VPN.

Configuration Item	Parameter	Description
Task Configuration	Task Name	DTS automatically generates a task name. You can configure the name as needed.
	Running Mode	Supports both immediate execution and scheduled execution.

Source Instance Settings	Source Instance Type	The source instance type selected at the time of purchase. You cannot modify this parameter.
	Source Instance Region	The source instance region selected at the time of purchase. You cannot modify this parameter.
	Access Type	Select a type based on your scenario. In this scenario, select Direct Connect or VPN Access. This scenario requires you to select Direct Connect or VPN Access: Configuring VPN-IDC Interconnection. For the preparation required for other connection types, see Overview. Public Network: The source database can be accessed by using the public IP. Self-Build on CVM: The source database is deployed on Cloud Virtual Machine. Direct Connect: The source database can be connected to Tencent Cloud VPC by using Direct Connect. VPN: The source database can be connected to Tencent Cloud VPC through VPN Connections. TencentDB: The source database is a TencentDB instance. CCN: The source database can be connected to Tencent Cloud VPC through Cloud Connect Network.
	VPC-based Direct Connect Gateway/VPN Gateway	If you use Direct Connect, only VPC-based Direct Connect Gateway/VPN Gateway is supported. Confirm the network type of the gateway. To use VPN Gateway, select a VPN Gateway instance.
	VPC	Select a VPC and subnet associated with the VPC-based Direct Connect gateway or VPN gateway.
	Host Address	The IP address or domain name for accessing the source MySQL instance.
	Port	The port for accessing the source MySQL instance.
	Account	The source instance account. It must have the required permissions.
	Password	The password for the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected at the time of purchase, which cannot be modified.
	Target Instance Region	The target instance region selected at the time of purchase, which cannot be modified.
	Access Type	Select a type based on your scenario. For this scenario, select Database.

	Instance ID	Select the target instance ID.
	Account	The target instance account, which must have the required permissions.
	Password	The password for the target instance account.

Example 2: Synchronize a TencentDB database to another TencentDB database

Configuration Item	Parameter	Description
Task Configuration	Task Name	DTS automatically generates a task name. You can configure the name as needed.
	Running Mode	Supports both immediate execution and scheduled execution.
Source Instance Settings	Source Instance Type	The source instance type selected at the time of purchase, which cannot be modifie
	Source Instance Region	The source instance region selected at the time of purchase, which cannot be modi
	Access Type	Select a type based on your scenario. For this scenario, select Database. For prepa for different access types, see the preparation overview. Public Network: The source database can be accessed through the public IP. Self-Build on CVM: The source database is deployed on Cloud Virtual Machine. Direct Connect: The source database can be connected to Tencent Cloud VPC thro Connect](https://www.tencentcloud.com/document/product/216!ac1fc9354c56b77f7143d2b VPN Access: The source database can be connected to Tencent Cloud VPC throu Connections](https://www.tencentcloud.com/document/product/1037!a419dbfe8173788c2b560 TencentDB: The source database is a TencentDB instance. CCN: The source database can be connected to Tencent Cloud VPC through Clou Network.
	Instance ID	The source instance ID. You can view the source instance information on the insta
	Account	The source instance account, which must have the required permissions.
	Password	The password for the source instance account.
Target	Target	The target instance type selected at the time of purchase, which cannot be modifie

Instance Settings	Instance Type	
	Target Instance Region	The target instance region selected at the time of purchase, which cannot be modifi
	Access Type	Select a type based on your scenario. For this scenario, select Database.
	Instance ID	Select the target instance ID.
	Account	The target instance account, which must have the required permissions.
	Password	The password for the target instance account.

Example 3: Synchronize a PostgreSQL database provided by a third-party cloud provider to TencentDB over the public network.

Configuration Item	Parameter	Description
Task Configuration	Task Name	DTS automatically generates a task name. You can configure the name as needed.
	Running Mode	Supports both immediate execution and scheduled execution.
Source Instance Settings	Source Instance Type	The source instance type selected at the time of purchase, which cannot be modified.
	Source Instance Region	The source instance region selected at the time of purchase, which cannot be modified.
	Access Type	For databases from third-party cloud providers, Public Network is used in general cases. You can also choose VPN Access, Direct Connect, or CCN based on the actual network situation. In this scenario, select Public Network. For preparation required for different access types, see Overview. Public Network: The source database can be accessed through the public IP. Self-Build on CVM: The source database is deployed on Cloud Virtual Machine.

		Direct Connect: The source database can be connected to Tencent Cloud VPC through Direct Connect. VPN Access: The source database can be connected to Tencent Cloud VPC through VPN Connections. Database: The source database is a TencentDB instance. CCN: The source database can be connected to Tencent Cloud VPC through Cloud Connect Network.
	Host Address	The IP address or domain name for accessing the source instance.
	Port	The port for accessing the source instance.
	Account	The source instance account, which must have the required permissions.
	Password	The password for the source instance account.
Target Instance Settings	Target Instance Type	The target instance type selected at the time of purchase, which cannot be modified.
	Target Instance Region	The target instance region selected at the time of purchase, which cannot be modified.
	Access Type	Select a type based on your scenario. For this scenario, select Database.
	Instance ID	Select the target instance ID.
	Account	The target instance account, which must have the required permissions.
	Password	The password for the target instance account.

5. On the set sync options and objects page, set Data Initialization Option, Data Sync Option, and Sync Object Option. After completing the settings, click **Save and Go Next**.

Note：

If you select only **Full Data Initialization** for **Initialization Type**, the system assumes that table structures have already been created in the target database. In this case, the system will not synchronize table structures or verify if tables with duplicate names exist in the source database and target database. Therefore, if you also select **Precheck and Report Error** in **If Target Already Exists**, this validation and error-reporting feature will not take effect.

If you plan to rename a table during sync (for example, renaming table A to table B), you must select the entire database (or the entire instance) where table A is located in **Synchronize Object Option**. Otherwise, after the renaming operation, data from table B will not be synchronized to the target database.

Configuration	Parameter	Description

Item		
Data Initialization Option	Initialization Type	Structure Initialization: When you run the sync task, the table structure in the source instance is initialized in the target instance first. Full data initialization: When you run the sync task, data in the source instance will be initialized in the target instance first. In scenarios where only Full data initialization is selected, you need to create the table structure in the target database in advance. By default, both are selected. You can cancel options based on the actual situation.
	If Target Already Exists	Precheck and report error: If a table with the same name exists, an error is reported and the process is not continued. Ignore and execute: Directly synchronize both full and incremental data to the tables in the target instance.
Data Sync Option	Conflict Resolution Method	Report: If a primary key conflict is detected during sync, an error is reported and the data sync task is paused. Ignore: If a primary key conflict is detected during sync, the primary key record in the target database is retained. Overwrite: If a primary key conflict is detected during sync, the primary key record in the source database overwrites that in the target database.
	SQL Type	Supported operations: Insert, Update, Delete, and DDL. Currently, Custom DDL is not supported.
Synchronize Object Option	Database and Table Objects of Source Instance	Select objects to be synchronized. The following objects are supported: databases, tables, and views. Only SEQUENCE is supported as advanced objects.

6. On the Verify task page, complete the verification and click **Start Task** after all verification items pass.

If the verification task fails, you can refer to Check Item Overview
to fix the issue and run the verification task again.

Failed: Indicates that the verification item failed the check and the task is blocked. You need to fix the problem and run the verification task again.

Alarm: Indicates that the verification item does not fully meet the requirements. The task can proceed but affects the business to some extent. You need to decide based on the prompt whether to ignore the alarm or fix the issue and continue.

7. Return to the data sync task list. The task has entered the **Running** status.

Note：

In the **Operation** column, choose **More** > **Stop** to stop a sync task. Make sure data sync is complete before you stop the task.

8. (Optional) You can click the name of a task to enter the task details page and view the initialization state and monitoring data of the task.

PostgreSQL Pre Validation does not Pass Processing

Database Connection Check

Last updated：2024-07-08 18:38:11

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 18:38:11

Version Check

Last updated：2024-07-08 18:38:11

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 18:38:11

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 18:38:11

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：2024-07-12 09:42:31

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 18:38:11

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 18:38:11

Check Details

Check whether the target database user conflicts with the source database user in a data migration scenario.

Troubleshooting

In full database migration, if the target database has the same account as that in the source database, you need to delete it.

If the account in the target database is the initial account, use it to log in to the database and run the following statements:

```
CREATE USER new user WITH PASSWORD 'password';
GRANT pg_tencentdb_superuser TO new username;
ALTER USER new user WITH CREATEDB;
ALTER USER new user WITH CREATEROLE;
```

If the account in the target database is a new user, use it to log in to the database and delete the conflicting user.

```
DROP USER conflicting user;
# If the conflicting user has resource dependencies, run the following statement to
ALTER TABLE table name OWNER TO new user;
```

After the conflicting user is deleted, run the verification task again.

Parameter Configuration Conflict Check

Last updated：2024-07-08 18:38:11

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 18:38:11

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.

Primary Key Dependency Check

Last updated：2024-07-08 18:41:19

Check Details

In PostgreSQL data sync scenarios, you need to check whether the table to be synced has the primary key, and if not, the task verification will report a warning and automatically set `ALTER TABLE XXXREPLICA IDENTITY` .

Troubleshooting

Set a primary key as prompted and execute the verification task again.

Check the Key Parameters of the Instance

Last updated：2024-07-08 18:41:41

Check Details

In PostgreSQL data sync scenarios, the sync task will check whether the replication parameter `wal_level` is set to `logical` . At the same time, the sum of the number of selected databases for sync and the value of `replication_slots` must be smaller than or equal to the value of `max_wal_senders` .

Troubleshooting

Set the value of `wal_level` to `logical` .

Reduce the number of databases to be synced.

Multi Task Conflict Detection

Last updated：2024-07-08 18:42:04

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.

Synchronize to Kafka

Difference Between Data Sync to Kafka and Data Subscription

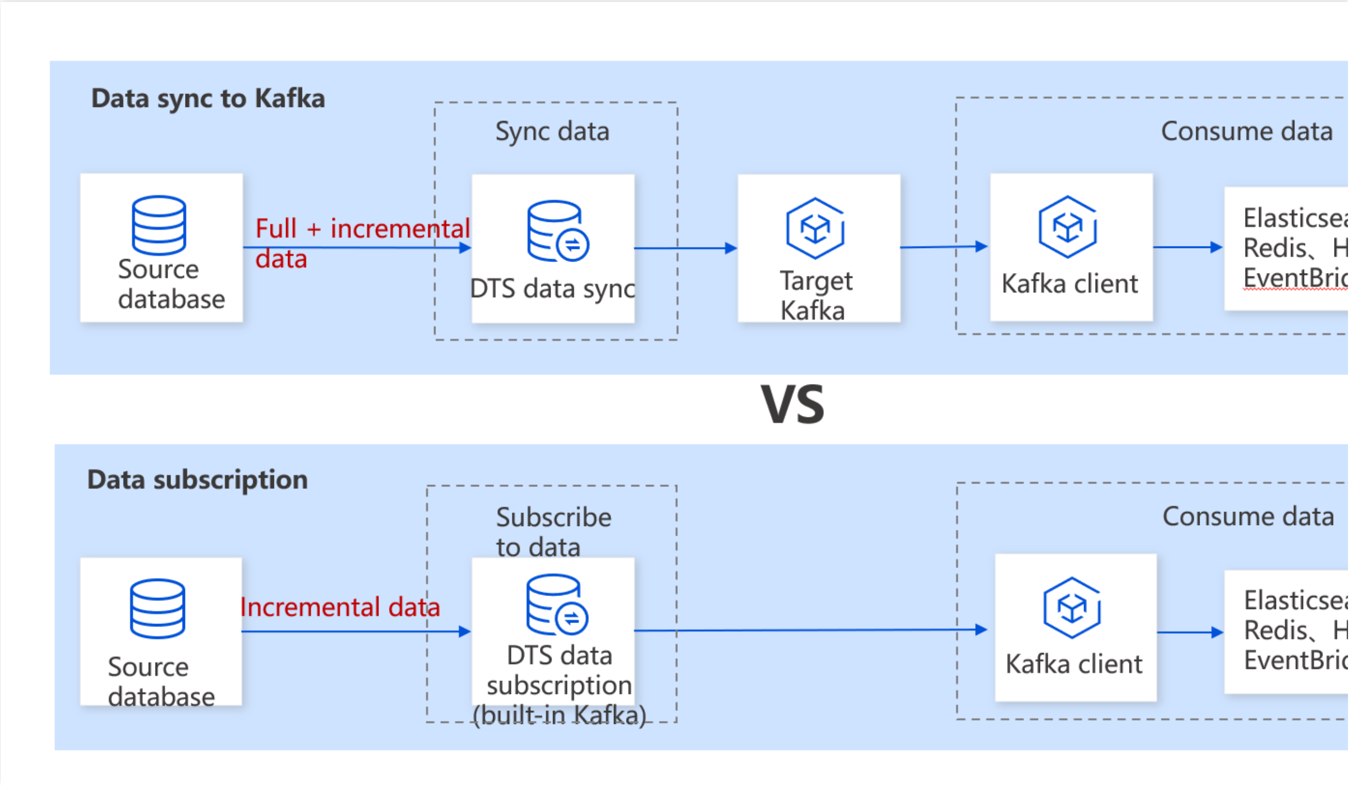
Last updated：2023-12-11 11:31:25

Sync to Kafka is similar to data subscription because both features can be used to obtain real-time data changes in the source database, and therefore can be applied in data archiving, data analysis, and other scenarios.

If you have a CKafka instance, or if you need to sync the full + incremental data of the source database, we recommend that you select the sync to Kafka feature.

If you don't have a CKafka instance, and you only need to obtain the incremental database/table data changes of the source database, we recommend that you select data subscription (Kafka edition).

If both features can meet your business needs, we recommend that you select the most cost-effective solution by referring to their billing details in Billing Overview.


Data sync to Kafka
Sync data
Consume data
Source database
Full + incremental data
DTS data sync
Target Kafka
Kafka client
Elasticsea
Redis、H
EventBric
VS
Data subscription
Subscribe to data
Consume data
Source database
Incremental data
DTS data subscription (built-in Kafka)
Kafka client
Elasticsea
Redis、H
EventBric

Comparison between the two features is as detailed below:

Feature	Sync to Kafka	Data Subscription
Supported links	Sync from MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL to Ckafka	The data source is MySQL/MariaDB/Percona/TDSQL-C for

		MySQL/TDSQL for MySQL/TDSQL for PostgreSQL/MongoDB
Data format	Avro/JSON	ProtoBuf/Avro/JSON
Target Kafka	Your own Kafka (currently, only CKafka is supported). Topic names can be customized.	Built-in Kafka of DTS data subscription. Topic names cannot be customized.
Data type	Full + incremental data	Incremental data
Sync object	Databases, tables, views, functions, and procedures	Databases and tables
SQL filtering	SQL filtering and filtering by WHERE condition are supported. You can filter INSERT, UPDATE, DELETE, and DDL statements.	SQL filtering is not supported. You can only select structure update or data update.

Data Delivery to Kafka Policy Settings

Last updated：2024-07-11 16:18:45

Overview

When syncing the source data to Kafka, it supports flexible delivery policies. It can deliver data from different tables to different Topics, or to a single Topic. The explanations of different policies are as follows:

Policy Category	Single Topic Partitioning Policy	Feature Scenario
Deliver to Custom Topic	Not involved	Data from different libraries and tables can be delivered to different Topics.
Deliver to a Single Topic	Topic partitioning policy (default partitioning policy)	Deliver all to Partition0: deliver all to the first partition of a single Topic. By table name: deliver data from the same table to the same partition. By table name + primary key: deliver data with the same primary key value in the same table to the same partition. This policy is suitable for hot data; after setting, tables with hot data will be scattered and delivered to different partitions.
	Topic partitioning policy (default partitioning policy) + custom partitioning policy	1. First, deliver matched database and table data according to the custom partitioning policy (supports partitioning by table name, table name + primary key, and column). 2. Then, deliver the remaining unmatched database and table data according to the Topic partitioning policy (default partitioning policy).

Custom Topic Name

Feature Description

Users set the Topic name for delivery themselves. DTS writes to the target Kafka according to the filled Topic name:

If the target has the Topic, or does not have the Topic but auto.create.topics.enable is true, DTS will write successfully.

If the target does not have the Topic, and auto.create.topics.enable is false, then DTS will fail to write, and at the same time, the sync task will report an error.

auto.create.topics.enable is a configuration parameter in Kafka, which is used to control whether to allow automatic creation of Topics and typically modified in Kafka's configuration file server.properties. When it is set to true, Kafka will automatically create the Topic upon receiving a request for a non-existing Topic. When it is set to false, it will not automatically create a Topic.

Configuration Rules

When selecting a custom Topic name, if the user has set multiple rules, DTS will match them line by line from top to bottom.

If the set rule (which the source's database name and table name both match with) is matched, then it will be delivered to the Topic corresponding to that rule.

If multiple rules are matched, it will be delivered to the Topic of all matched rules.

The remaining data that does not match any set rule will be delivered to the Topic specified by the last rule.

Matching rules are case-sensitive for database names and table names.

If the source database set lower_case_table_names = 0, then the database and table names in the matching rules must exactly correspond to the case used in the source database.

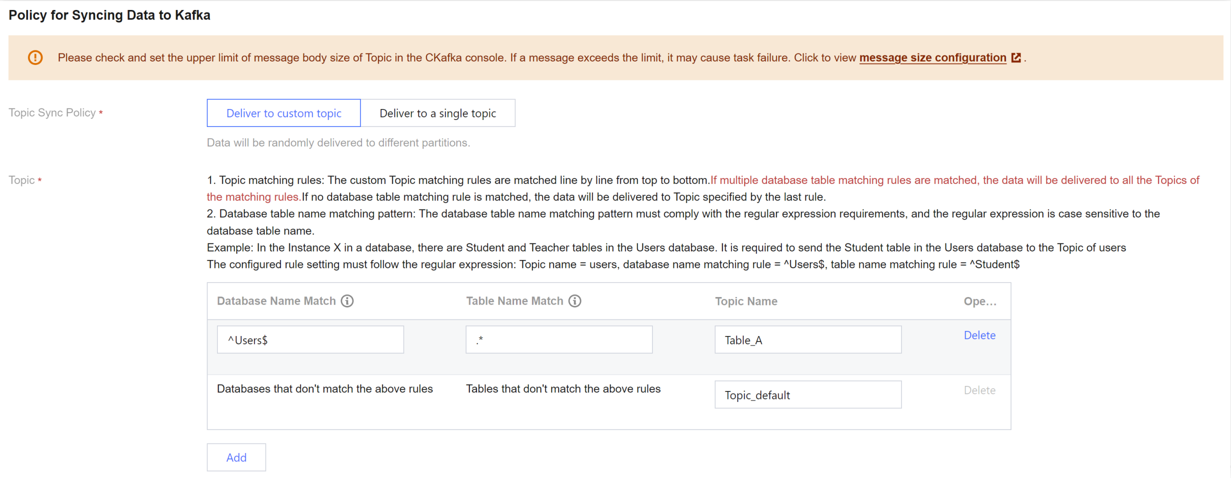
If the source database set lower_case_table_names = 1, then the database and table names are unified to lowercase, and the database and table names in the matching rules must be input in lowercase.

Database, table, and table name matching rules support RE2 regex. For the specific syntax, refer to Syntax Description. To achieve precision matching, start symbol and end symbol $ must be added, e.g., to precisely match the test table, it should be test$.

Configuration Sample

In the Instance X in a database, there are tables Teacher, Student, Student1, Student2, and Student3 in the Users database.

Example 1: Deliver all data from the Users database to Topic_A; deliver the remaining data to Topic_default.



Policy for Syncing Data to Kafka

Please check and set the upper limit of message body size of Topic in the CKafka console. If a message exceeds the limit, it may cause task failure. Click to view **message size configuration** .

Topic Sync Policy * : Deliver to custom topic | Deliver to a single topic

Data will be randomly delivered to different partitions.

Topic *

1. Topic matching rules: The custom Topic matching rules are matched line by line from top to bottom.If multiple database table matching rules are matched, the data will be delivered to all the Topics of the matching rules.If no database table matching rule is matched, the data will be delivered to Topic specified by the last rule.
2. Database table name matching pattern: The database table name matching pattern must comply with the regular expression requirements, and the regular expression is case sensitive to the database table name.

Example: In the Instance X in a database, there are Student and Teacher tables in the Users database. It is required to send the Student table in the Users database to the Topic of users

The configured rule setting must follow the regular expression: Topic name = users, database name matching rule = ^Users$, table name matching rule = ^Student$

Database Name Match	Table Name Match	Topic Name	Ope...
^Users$	.*	Table_A	Delete
Databases that don't match the above rules	Tables that don't match the above rules	Topic_default	Delete

Add

Topic Name	Database Name Match Mode	Table Name Match Mode	Description

Topic_A	^Users$	.*	Users$ indicates a precision match of the Users database, where is the start symbol and $ is the end symbol; entering only Users matches all database names containing Users, such as table X_Users_1. .* matches all table names.
Topic_default	Databases that don't match the above rules	Tables that don't match the above rules	-

Example 2: deliver data from the Teacher table to Topic_A, data from the Student table to Topic_B, and the remaining data to Topic_default.

Topic name	Database Name Match Mode	Table Name Match Mode	Description
Topic_A	^Users$	^Teacher$	Teacher$ indicates a precision match of the table name Teacher, where is the start symbol, and $ is the end symbol; entering only Teacher matches all tables containing Teacher, such as F_Teacher_1.
Topic_A	^Users$	^Student$	-
Topic_default	Databases that don't match the above rules	Tables that don't match the above rules	-

Example 3: deliver data from the Teacher table to Topic_A , tables with the prefix Student (namely tables Student, Student1, Student2, and Student3) all to Topic_B, and the remaining data to Topic_default.

Topic Name	Database Name Match Mode	Table Name Match Mode	Description
Topic_A	^Users$	^Teacher$	-
Topic_A	^Users$	^Student	^Student matches all tables with the prefix Student, namely Student, Student1, Student2, and Student3.
Topic_default	Databases that don't match the above rules	Tables that don't match the above rules	-

Delivering to a Single Topic (Default Partitioning Policy)

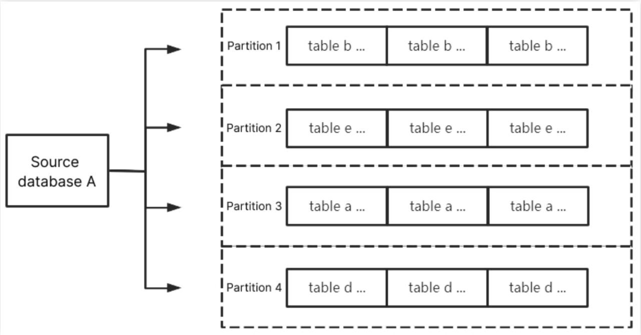
Select an existing Topic on the target, then deliver according to various partitioning policies, supports Single Partition, By Table Name, By Table Name + Primary Key.

Delivering All to Partition0 (i.e., Single Partition)

Deliver all synced data from the source database to the first partition of a single Topic.

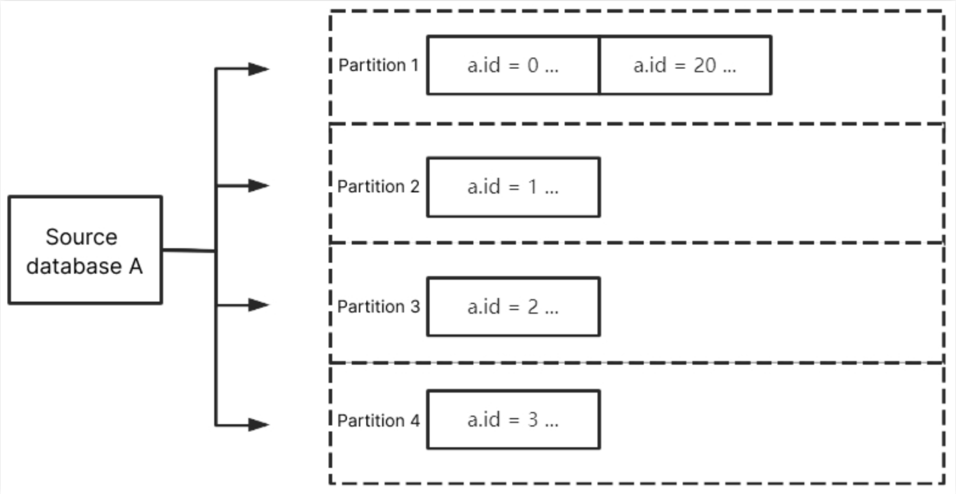
By Table Name

Partition synced data from the source database by table name. After setting, data with the same table name will be delivered to the same partition.


Source database A
Partition 1
table b ...
table b ...
table b ...
Partition 2
table e ...
table e ...
table e ...
Partition 3
table a ...
table a ...
table a ...
Partition 4
table d ...
table d ...
table d ...

By Table Name + Primary Key

Partition synced data from the source database by table name + primary key. After setting, data within the same table with the same primary key value will be delivered to the same partition. This policy is suitable for hot data; after setting, tables with hot data will be scattered into different partitions to enhance the concurrency consumption efficiency.


Source database A
Partition 1
a.id = 0 ...
a.id = 20 ...
Partition 2
a.id = 1 ...
Partition 3
a.id = 2 ...
Partition 4
a.id = 3 ...

Custom Partitioning Policy

When the Topic partitioning policy (default partitioning policy) selects By table name or By table name + primary key, it supports the option to choose a custom partitioning policy.

The custom partitioning policy matches database and table names through regex, partitioning matched data by table name, table name + primary key, or column. The remaining unmatched data is then partitioned according to the Topic partitioning policy (default partitioning policy) settings.

Matching Rules

Database and table name matching rules support RE2 regex. For the specific syntax, refer to Syntax Description.

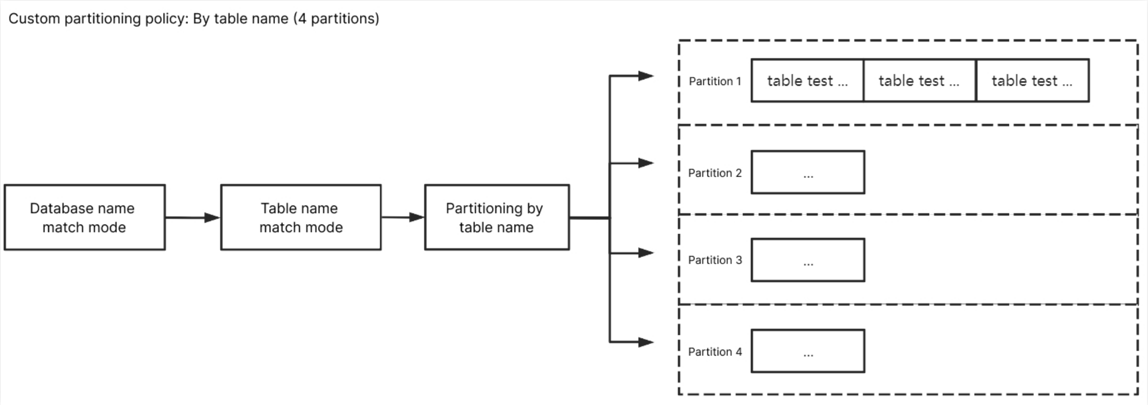
The matching rule for database names is to match through regex **database name**. The matching rule for table names is to match through regex **table name data**. If precise matching is needed for database and table names, start and end symbols should be added, e.g. to match the `test` table, it should be `^test$` .

The matching rule for column names uses equivalence `==` , and is case-insensitive.

Prioritize matching the custom partitioning policy. When there are multiple custom partitioning policies, match them one by one from top to bottom. The remaining data not matched by the custom partitioning policy is then delivered according to the Topic partitioning policy.

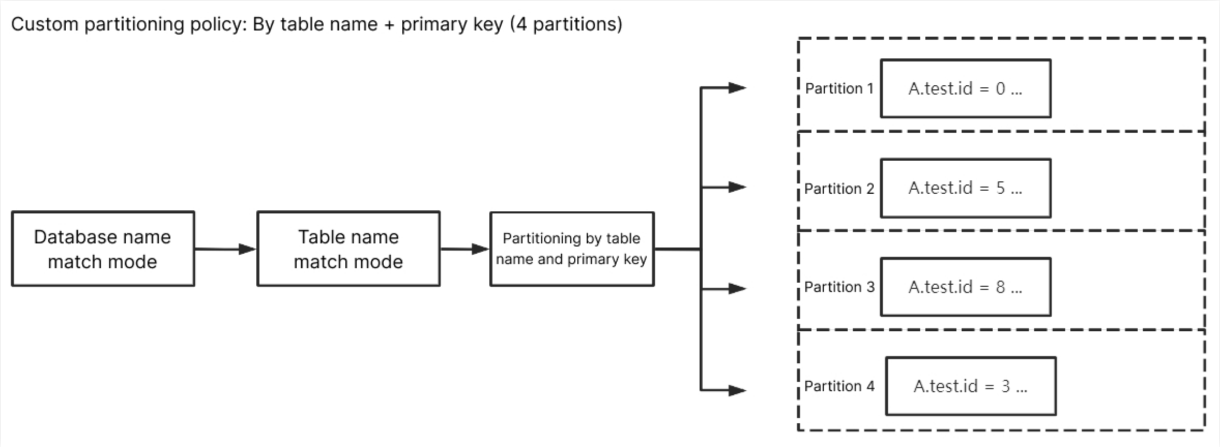
By Table Name

For the database name match mode, enter `^A$` ; for the table name match mode, enter `^test$` . After selecting **By table name**, the data from `A` database's `test` table will be delivered to the same partition. Meanwhile, `test` table's data, except for other unmatched database and table data, will be delivered according to the Topic partitioning policy (default partitioning policy).



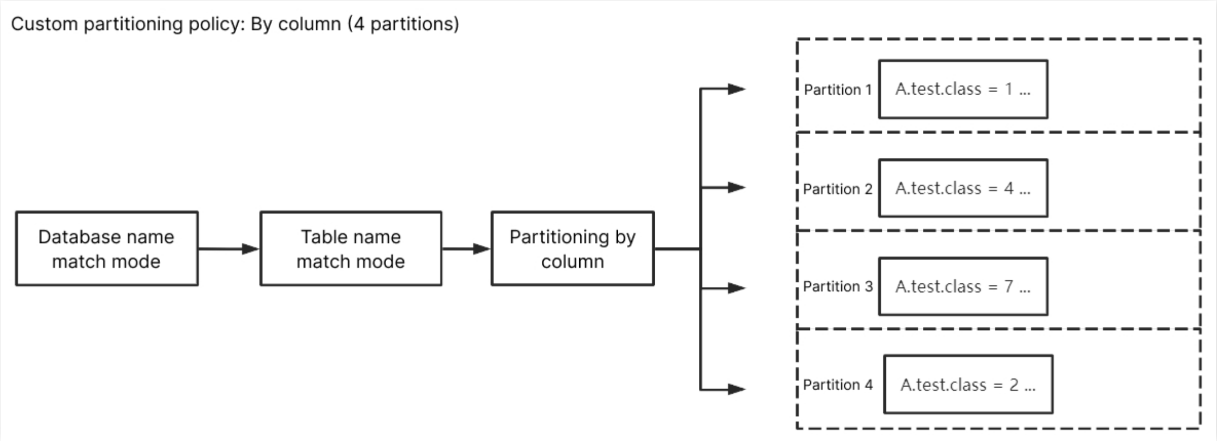
By Table Name + Primary Key

For the database name match mode, enter `^A$` ; for the table name match mode, enter `^test$` . After selecting **By table name + primary key**, data from the `A` database's `test` table will be scattered and delivered to different partitions based on primary key data, eventually delivering data with the same primary key to the same partition. Data from other unmatched databases and tables, except for the `test` table, will be delivered according to the Topic partitioning policy (default partitioning policy).


Custom partitioning policy: By table name + primary key (4 partitions)
Database name match mode
Table name match mode
Partitioning by table name and primary key
Partition 1
A.test.id = 0 ...
Partition 2
A.test.id = 5 ...
Partition 3
A.test.id = 8 ...
Partition 4
A.test.id = 3 ...

By Column

For the database name match mode, enter `^A$` ; for the table name match mode, enter `^test$` . Enter `class` for the custom partition column, and after selecting **By column**, data from `A` database's table `test` with column name `class` will be scattered and delivered to different partitions, eventually delivering updated data from the same column to the same partition. Data from other unmatched databases and tables, except for the `test` table, will be delivered according to the Topic partitioning policy (default partitioning policy).


Custom partitioning policy: By column (4 partitions)
Database name match mode
Table name match mode
Partitioning by column
Partition 1
A.test.class = 1 ...
Partition 2
A.test.class = 4 ...
Partition 3
A.test.class = 7 ...
Partition 4
A.test.class = 2 ...

MySQL/MariaDB/Percona/TDSQL-C MySQL Sync to Kafka

Supported Capabilities

Last updated：2024-07-08 19:02:56

Major Feature Categories	Feature Subitem or Description	Supporting Capability
Sync Object	-	Ordinary object table, index, view
Sync Type	-	Structure sync Full sync (automatically continue with incremental after full completion) Incremental sync
Breakpoint Resume	-	Partial support for full export (structure export, no primary key table export; stage does not support resume); Full import, incremental support.
Task Channel Key Operation	Retry	Supported
	Pause, recover	Supported
	Modify sync configuration (dynamically add or remove table objects)	Supported
	Create similar task	Supported
	Speed limit	Not supported
Sync Direction	Many-to-one (multiple instances merged into one/multiple tables merged into one)	Supported
	One-to-many	Supported
Structure Sync	Database and table filtering	Supported
	Column filtering	Not supported
	Skip sync table and correction	Not supported

Full Sync	Where condition filtering	Supported
Incremental Sync	DML sync (INSERT/UPDATE/DELETE)	Supported
	DML filtering	Supported
	DDL sync	CREATE DATABASE, DROP DATABASE, ALTER DATABASE CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE CREATE VIEW, DROP VIEW CREATE INDEX, DROP INDEX The CREATE TABLE table_name AS SELECT statement is not supported.
	DDL filtering	Supported
	Where condition filtering	Supported
	Scheduled sync	Not supported
	Online DDL sync	Supported
Data Subscription (Synced to Kafka Link-Specific)	Data format	Avro, JSON, Canal Protobuf, Canal JSON
	Designated multi-topic delivery	Supported
	Specified single topic multi-partition delivery	Supported
	Specified single topic + custom partition strategy	Supported
	Filter transaction BEGIN/COMMIT	Supported
	Filter CHECKPOINT messages	Supported

Data Sync Guide

Last updated：2024-07-08 19:02:56

DTS allows you to sync the full and incremental data of the source database to TDMQ for Ckafka, so you can quickly obtain business change data and use it. This document describes how to use DTS to sync data from MySQL/MariaDB/Percona/TDSQL-C for MySQL to TDMQ for Ckafka.

The following deployment modes of the source database are supported:

Self-built MySQL, third-party cloud MySQL, and TencentDB for MySQL.

Self-built MariaDB and TencentDB for MariaDB.

Self-built Percona.

TDSQL-C for MySQL

Note

The same task can be associated with multiple topics, but the same topic cannot be used by multiple sync tasks at the same time, otherwise chaos will occur where data will be consumed from multiple different tasks. This may also result in error reports for abnormal data processing during the task retry.

By default, the locked sync method is adopted. During the full data export, a global lock (FTWRL) will be added to the source database, and the write operation will be blocked for a short time (in seconds) during the table locking process.

If you determine to use the rename operation on a certain table during the sync process and deliver it to the target TDMQ for Ckafka, the partition rules will match the topic and partition based on the new table name.

During the full sync, every time 100,000 entries of data are synced, a checkpoint message will be inserted in the target Kafka to identify the current data sync point.

Before the sync task, if the full data already exists in the source database, the `happenAt` field in the consumer demo will be displayed as 1970-01-01 08:00:00 as there is no accurate time for data writing. You can ignore this time field. When incremental data is consumed, the time field can be displayed correctly.

During the full data sync, DTS exports and imports the source database data to the target TDMQ for Ckafka, and uses the utf8mb4 character set to avoid garbled characters caused by unrecognized character sets.

We recommend that you set the target topic to be empty and not write data to that topic during the sync task, otherwise it may cause message disorder and task error.

There is a performance cap for a single message that DTS syncs to the target Kafka. We recommend that you set the size of a single row of data to be up to 8 MB, otherwise an error may be reported during the data sync.

During data sync, DTS will use the account that executes the sync task to write the system database `__tencentdb__` in the source database to record metadata such as transaction ID. Therefore, you need to make sure the source database has the read/write permission of `__tencentdb__`.

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

The source and target databases must meet the requirements for the sync feature and version as described in Databases Supported by Data Sync.
Source database permissions required for the sync task account:

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW VIEW,PROCESS,SEL
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO 'account'@'%';
FLUSH PRIVILEGES;
```

You need to modify the message retention period and message size limit in target TDMQ for Ckafka.

We recommend that you set the message retention period to 3 days. The data beyond the retention period will be cleared, so you need to consume data in time within the set period. The upper limit for message size refers to the maximum size of a single message that TDMQ for Ckafka can receive. You must set it to be greater than the maximum size of a single row of data in the source database table so that data can be normally delivered to CKafka.

Use Limits

Only tables, views, procedures, and functions can be synced.

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

Only data with the following three database engines can be synced: InnoDB, MyISAM, and TokuDB. Tables with other engines will be skipped during sync by default.

Correlated data objects need to be synced together; otherwise, sync will fail. The correlations include tables correlated through primary/foreign keys.

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 sync.

In order to ensure data consistency, the locked sync is implemented for sync to Kafka by default. However, if the source database is Alibaba Cloud MySQL5.5/5.6, AWS MySQL 5.7.18 and earlier, locks cannot be implemented, which may cause data duplication during the sync to the target TDMQ for Ckafka. If you are more concerned about duplicate data, you can implement deduplication logic when consuming data.

During the process of the sync task, if the task is restarted, which may cause data duplication on the target TDMQ for CKafka.

DTS syncs data based on the smallest data 7unit, which can be a data chunk in the full stage or a checkpoint in the incremental stage. If a data unit has been synced successfully before the task restart, no data duplication will occur; if it is still being synced during the task restart, it will be synced again after the restart, which will lead to data duplication.

A task will be restarted in the following operations: modify the sync object, manually click the retry button in the operation column when the task error occurs, change the configuration, stop the task and resume it, and isolate the task and unisolate it.

If you are more concerned about duplicate data, you can implement deduplication logic when consuming data.

The target Kafka can only be the Pro Edition or Standard Edition on v1.1.1 or later you can purchase on Tencent Cloud. The physical machine of standard edition that has been deactivated is not supported as a target Kafka.

Operation Restrictions

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

Do not modify or delete user information (including username, password, and permissions) in the source database and target TDMQ for CKafka 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 write data to a topic that has been selected for sync to the target database.

Synchronizable SQL Operations

Operation Type	SQL Statement
DML	INSERT, UPDATE, DELETE
DDL	CREATE DATABASE, DROP DATABASE, ALTER DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, CREATE VIEW, DROP VIEW, CREATE INDEX,DROP INDEX

Note

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

Environment Requirements

Type	Environment Requirement
Requirements for the source database	The source and target databases can be connected. 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 `connect_timeout` variable in the source database must be greater than or equal to 10. 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 that 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 only be set to one of the following two types: `NO ACTION` and `RESTRICT` . Other types may affect data consistency results. During partial table sync, tables with foreign key dependency must be migrated. The environment variable `innodb_stats_on_metadata` must be set to `OFF` .
Requirements for the target TDMQ for CKafka	The target TDMQ for CKafka can be connected. The upper limit of message size in target TDMQ for CKafka must be greater than the maximum value of a single row of data in the source database table.

Directions

This section describes how to sync data from MySQL to TDMQ for CKafka. The requirements and steps of data sync from MySQL/MariaDB/Percona/TDSQL-C forMySQL are basically the same.

1. Log in to the data sync task purchase page, select appropriate configuration items, and click **Buy Now**.

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing are supported.
Source Instance Type	Select MySQL, which cannot be changed after purchase.
Source Instance Region	Select the source instance region, which cannot be changed after purchase. If the source database is a self-built one, select a region nearest to it.
Target Instance Type	Select **Kafka**, which cannot be changed after purchase.
Target Instance Region	Select the target instance region, which cannot be changed after purchase.
Specification	Select a specification based on your business needs. The higher the specification, the higher the performance. For more information, see Billing Overview.

2. After making the purchase, return to the data sync task list and click **Configure** in the **Operation** column to enter the sync task configuration page.

3. On the sync task configuration page, configure the source and target databases and their accounts and passwords, test the connectivity, and click **Next**.

Setting Items	Parameter	Description

Task Settings	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	The source database type selected during purchase, which cannot be changed.
	Source Instance Region	The source instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. For the preparations for different access types, see Overview. Public Network: The source database can be accessed through a public IP. Self-Build on CVM: The source database is deployed on a CVM instance. For more information on CVM, see Cloud Virtual Machine Documentation. Direct Connect: The source database can be interconnected with VPCs through Direct Connect. For more information on Direct Connect, see Direct Connect Documentation. VPN Access: The source database can be interconnected with VPCs through VPN Connections. For more information on VPN Connections, see VPN Connections Documentation. Database: The source database is a TencentDB database. CCN: The source database can be interconnected with VPCs through CCN. For more information on CCN, see Cloud Connect Network Documentation.
	Public Network	Host Address: IP address or domain name of the source database. Port: Port used by the source database.
	Self-Build on CVM	CVM Instance: The ID of the CVM instance. Port: Port used by the source database.
	Direct Connect	VPC-Based Direct Connect Gateway: Only VPC-based Direct Connect gateway is supported. Confirm the network type associated with the gateway. VPC: Select a VPC and subnet associated with the VPC-based Direct Connect gateway or VPN gateway. Host Address: IP address of the source database. Port: Port used by the source database.
	VPN Access	VPN Gateway: Select a VPN gateway instance. VPC: Select a VPC and subnet associated with the VPC-based Direct Connect gateway or VPN gateway. Host Address: IP address of the source database. Port: Port used by the source database.

	Database	Instance Name: The ID of the source database instance.
	CCN	Host Address: IP address of the source database server. Port: Port used by the source database. VPC-Based CCN Instance: The name of the CCN instance. Accessed VPC: It refers to the VPC in CCN over which the subscription link is connected. You need to select a VPC other than the VPC to which the source database belongs. For example, if the database in Guangzhou region is used as the source database, select another region, such VPC-Chengdu or VPC-Shanghai, as the accessed VPC. Subnet: Name of the subnet of the selected VPC. Region of Accessed VPC: The region of the source database selected during task purchase must be the same as the region of the accessed VPC; otherwise, DTS will change the former to the latter.
	Account/Password	Account/Password: Enter the source database account and password.
Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **CKafka instance**.
	Instance ID	Select the ID of the target instance.

4. On the **Set sync options and objects** page, set the data initialization, data sync, and sync object options, as well as the policy for syncing data to Kafka. Then, click **Save and Go Next**.

Deliver to custom topic

Deliver to a single topic

Setting Items	Parameter	Description
Data Initialization Option	Initialization Type	Structure initialization: **Table structures** in the source instance will be initialized into the target instance before the sync task runs. Full data initialization: **Data** in the source instance will be initialized into the target instance before the sync task runs. If you select **Full data initialization** only, you need to create the table structures in the target database in advance. Both options are selected by default, and you can deselect them as needed.
	Format of Data	Avro adopts the binary format with a higher consumption efficiency, while

	Delivered to Kafka	JSON adopts the easier-to-use lightweight text format.
Policy for Syncing Data to Kafka	Topic Sync Policy	Deliver to custom topic: Customize the topic name for delivery. After that, the target Kafka will automatically create a topic with the custom name. The synced data is randomly delivered to different partitions under the topic. **If the target Kafka fails to create the topic, the task will report an error.** Deliver to a single topic: Select an existing topic on the target side, and then deliver based on multiple partitioning policies. Data can be delivered to a single partition of the specified topic, or delivered to different partitions by table name or by table name + primary key.
	Rules for delivering to custom topic	If you add multiple rules, the database and table rules are matched one by one from top to bottom. If no rules are matched, data will be delivered to the topic corresponding to the last rule. If multiple rules are matched, data will be delivered to the topics corresponding to all the matched rules. **Note** The matching rule is case-sensitive for the database/table name. If `lower_case_table_names` is set to `0` in the source database, the case of the database/table name in the matching rule must be exactly the same as that in the source database. If `lower_case_table_names` is set to `1` , all the database/table names will be in lower case in the matching rule. Example 1: There are tables named "Student" and "Teacher" in a database named "Users" on database instance X, and you need to deliver the data in the "Users" database to a topic named "Topic_A". Then the rules are configured as follows: Enter `Topic_A` for **Topic Name**, `^Users$` for **Database Name Match**, and `.*` for **Table Name Match**. Enter `Topic_default` for **Topic Name**, `Databases that don't match the above rules` for **Database Name Match**, and `Tables that don't match the above rules` for **Table Name Match**. Example 2: There are tables named "Student" and "Teacher" in a database named "Users" on database instance X, and you need to deliver the data in the "Student" table and "Teacher" tables to topics named "Topic_A" and "Topic_default" respectively. Then the rules are configured as follows: Enter `Topic_A` for **Topic Name**, `^Users$` for **Database Name Match**, and `^Student$` for **Table Name Match**. Enter `Topic_default` for **Topic Name**, `Databases that don't match the above rules` for **Database Name Match**, and `Tables that don't match the above rules` for **Table Name Match**.

	Rules for delivering to a single topic	After selecting a specified topic, the system will perform partitioning based on the specified policy as follows. Deliver all data to partition 0: Deliver all the synced data of the source database to the first partition. By table name: Partition the synced data from the source database by table name. After setting, the data with the same table name will be written into the same partition. By table name + primary key: Partition the synced data from the source database by table name and primary key. This policy is suitable for frequently accessed data. After settings, frequently accessed data is distributed from tables to different partitions by table name and primary key, so as to improve the concurrent consumption efficiency.
	Topic for DDL Storage	(Optional) If you need to deliver the DDL operation of the source database to the specified topic separately, you can select the settings here. After setting, it will be delivered to Partition 0 of the selected topic by default; if not set, it will be delivered based on the topic rules selected above.

Data Sync Option

Setting Items	Parameter	Description
Data Sync Option	SQL Type	The following operations are supported: INSERT, DELETE, UPDATE, and DDL. If **Custom DDL** is selected, you can choose different DDL sync policies as needed. For more information, see Setting SQL Filter Policy.
Sync Object Option	Database and Table Objects of Source Instance	Select objects to be synced, which can be tables, views, procedures, and functions. The sync of advanced objects is a one-time operation. Only advanced objects already in the source database before the task start can be synced, while those added to the source database after the task start will not be synced to the target database. For more information, see Syncing Advanced Object.
	Selected Object	After selecting the sync object on the left, click ◂ to see the selected object on the right.
	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 Syncing Online DDL Temp Table.

5. On the task verification page, complete the verification. After all check items are passed, click **Start Task**.
If the verification fails, troubleshoot 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 sync task list, and you can see that the task has entered the **Running** status.

Note

You can click **More** > **Stop** in the **Operation** column to stop a sync task. Before doing so, ensure that data sync has been completed.

7. (Optional) you can click a task name to enter the task details page and view the task initialization status and monitoring data.

Common Issues

During incremental sync, the synced objects of the source database have not been written for a long time, and the monitoring panel shows that the delay is large

You need to first confirm that the selected synced objects of the source database have no data written to for a long time (while other unsynced database/table objects have data written to). If there is no data written to, ignore this problem as there is an error on the monitoring metric. If not, submit a ticket for processing.

When MySQL databases are synced to the Kafka link, DTS needs to first pull the binlog of the source database, analyze and identify the relevant data of the selected objects, and synced the data to the target Kafka. If the selected synced objects of the source database have no data written to for a long time, while other unsynced database/table objects have data written to, DTS pulls the binlog for analysis and finds that no data needs to be synced to the target side. The data on the target side has not been updated, so the delay is displayed on the monitoring panel. You can try to update the data in the selected synced objects in the source database to refresh and get the real metric data.

Subsequent Operations

After the data is synced to the target Kafka, it can be consumed. We provide you with a consumption demo, so that you can quickly test the process of data consumption and understand the method of data parsing. For more information, see Data Consumption Guide.

Data Consumption Guide

Last updated：2024-07-08 19:02:56

Overview

After data is synced to Kafka, you can consume the subscribed data through Kafka 0.11 or later available at DOWNLOAD. This document provides client consumption demos for Java, Go, and Python for you to quickly test the process of data consumption and understand the method of data parsing.

Note

The demo only prints out the consumed data and does not contain any usage instructions. You need to write your own data processing logic based on the demo. You can also use Kafka clients in other languages to consume and parse data.

The upper limit of message size in target CKafka must be greater than the maximum value of a single row of data in the source database table so that data can be normally synced to CKafka.

In scenarios where only the specified databases and tables (part of the source instance objects) are synced and single-partition Kafka topics are used, only the data of the synced objects will be written to Kafka topics after DTS parses incremental data. The data of non-sync objects will be converted into empty transactions and then written to Kafka topics. Therefore, there are empty transactions during message consumption. The BEGIN and COMMIT messages in the empty transactions contain the GTID information, which can ensure the GTID continuity and integrity. In the consumption demos for MySQL/TDSQL-C for MySQL, multiple empty transactions have been compressed to reduce the number of messages.

To ensure that data can be rewritten from where the task is paused, DTS adopts the checkpoint mechanism for data sync links where Kafka is the target end. Specifically, when messages are written to Kafka topics, a checkpoint message is inserted every 10 seconds to mark the data sync offset. When the task is resumed after being interrupted, data can be rewritten from the checkpoint message. The consumer commits a consumption offset every time it encounters a checkpoint message so that the consumption offset can be updated timely.

When the selected data format is JSON, if you have used or are familiar with the open-source subscription tool Canal, you can choose to convert the consumed JSON data to a Canal-compatible data format for subsequent processing. The demo already supports this feature, and you can implement it by adding the `trans2canal` parameter in the demo startup parameters. Currently, this feature is supported only in Java.

Downloading Consumption Demos

When configuring the sync task, you can select Avro or JSON as the data format. Avro adopt the binary format with a higher consumption efficiency, while JSON adopts the easier-to-use lightweight text format. The reference demo varies by the selected data format.

The following demo already contains the Avro/JSON protocol file, so you don't need to download it separately.

For the description of the logic and key parameters in the demo, see Demo Description.

Demo Language	Avro Format (MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL	JSON Format (MySQL/MariaDB/Percona/TDSQL-C MySQL//TDSQL MySQL
Go	Address	Address
Java	Address	Address
Python	Address	Address

Instructions for the Java Demo

Compiling environment: The package management tool Maven or Gradle, and JDK8. You can choose a desired package management tool. The following takes Maven as an example.

The steps are as follows:

1. Download the Java demo and decompress it.
2. Access the decompressed directory. Maven model and pom.xml files are placed in the directory for your use as needed.

Package with Maven by running `mvn clean package`.

3. Run the demo.

After packaging the project with Maven, go to the target folder `target` and run `java -jar consumerDemo-avro-1.0-SNAPSHOT.jar --brokers xxx --topic xxx --group xxx --trans2sql`.

`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic`, you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.

`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.

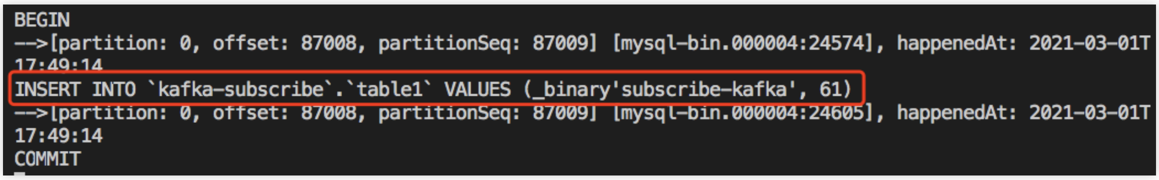
`trans2sql` indicates whether to enable conversion to SQL statement. In Java code, if this parameter is carried, the conversion will be enabled.

`trans2canal` indicates whether to print the data in Canal format. If this parameter is carried, the conversion will be enabled.

Note:

If `trans2sql` is carried, `javax.xml.bind.DatatypeConverter.printHexBinary()` will be used to convert byte values to hex values. You should use JDK 1.8 or later to avoid incompatibility. If you don't need SQL conversion, comment this parameter out.

4. Observe the consumption.



```
BEGIN
-->[partition: 0, offset: 87008, partitionSeq: 87009] [mysql-bin.000004:24574], happenedAt: 2021-03-01T
17:49:14
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 61)
-->[partition: 0, offset: 87008, partitionSeq: 87009] [mysql-bin.000004:24605], happenedAt: 2021-03-01T
17:49:14
COMMIT
```

Instructions for the Go Demo

Compiling environment: Go 1.12 or later, with the Go module environment configured.

The steps are as follows:

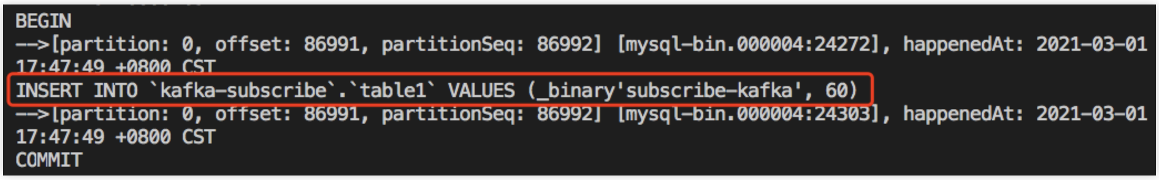
1. Download the Go demo and decompress it.
2. Access the decompressed directory and run `go build -o subscribe ./main/main.go` to generate the executable file `subscribe`.
3. Run `./subscribe --brokers=xxx --topic=xxx --group=xxx --trans2sql=true`.

`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic`, you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.

`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.

`trans2sql` indicates whether to enable conversion to SQL statement.

4. Observe the consumption.



```
BEGIN
-->[partition: 0, offset: 86991, partitionSeq: 86992] [mysql-bin.000004:24272], happenedAt: 2021-03-01
17:47:49 +0800 CST
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 60)
-->[partition: 0, offset: 86991, partitionSeq: 86992] [mysql-bin.000004:24303], happenedAt: 2021-03-01
17:47:49 +0800 CST
COMMIT
```

Instructions for the Python3 Demo

Compiling runtime environment: Install Python3 and pip3 (for dependency package installation).

Use `pip3` to install the dependency package:

```
pip install flag
pip install kafka-python
pip install avro
```

The steps are as follows:

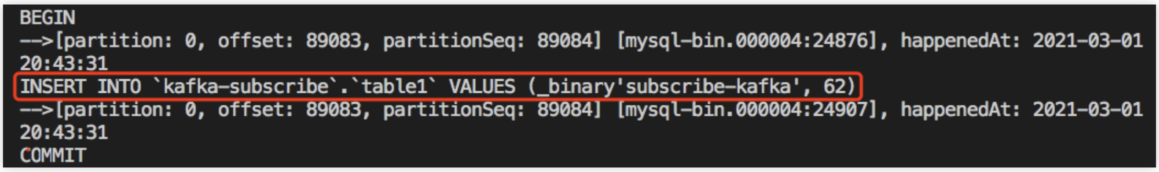
1. Download Python3 demo and decompress it.
2. Run `python main.py --brokers=xxx --topic=xxx --group=xxx --trans2sql=1` .

`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic` , you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.

`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.

`trans2sql` indicates whether to enable conversion to SQL statement.

3. Observe the consumption.



```
BEGIN
-->[partition: 0, offset: 89083, partitionSeq: 89084] [mysql-bin.000004:24876], happenedAt: 2021-03-01
20:43:31
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 62)
-->[partition: 0, offset: 89083, partitionSeq: 89084] [mysql-bin.000004:24907], happenedAt: 2021-03-01
20:43:31
COMMIT
```

Demo Description

Avro Demo Description

Last updated：2024-07-08 19:02:56

Key Logic Description

Files in the demo are as described below, with the Java demo as an example.

`consumerDemo-avro-java\\src\\main\\resources\\avro-tools-1.8.2.jar` : The tool used to generate Avro protocol code.

`consumerDemo-avro-java\\src\\main\\java\\com\\tencent\\subscribe\\avro` : The directory of the Avro tool-generated code.

`consumerDemo-avro-java\\src\\main\\resources\\Record.avsc` : The protocol definition file.

14 structures (also known as schemas in Avro) are defined in `Record.avsc` . The main data structure is `record` , which is used to represent a data record in binlog. The record structure is as follows. Other data structures can be viewed in `Record.avsc` .

```
{
   "namespace": "com.tencent.subscribe.avro",    // The last schema in `Record.avs
   "type": "record",
   "name": "Record",     // `name` is displayed as `Record`, indicating the format
   "fields": [
     {
       "name": "id",     // `id` indicates a globally incremental ID. More record
       "type": "long",
       "doc": "unique id of this record in the whole stream"
     },
     {
          "name": "version",  // `version` indicates the protocol version.
          "type": "int",
          "doc": "protocol version"
        },
        {
          "name": "messageType",   // Message type
          "aliases": [
            "operation"
          ],
          "type": {
            "namespace": "com.tencent.subscribe.avro",
            "name": "MessageType",
            "type": "enum",
            "symbols": [
              "INSERT",
              "UPDATE",
              "DELETE",
              "DDL",
              "BEGIN",
              "COMMIT",
              "HEARTBEAT",
              "CHECKPOINT",
              "ROLLBACK"
              "STATEMENT",
              "INIT_DDL",
              "INIT_INSERT"
            ]
          }
        },
        {
         ......
        },
    }
```

Fields in a record are as explained below:

Field Name in Record	Description
id	The globally incremental ID
version	The protocol version, which is v1 currently.
messageType	The message type. Enumerated values: `INSERT` , `UPDATE` , `DELETE` , `DDL` , `BEGIN` , `COMMIT` , `HEARTBEAT` , `CHECKPOINT` , `STATEMENT` , `INIT_DDL` , `INIT_INSERT` . `INIT_DDL` : DDL operation on full data; `DDL` : DDL operation on incremental data; `INIT_INSERT` : INSERT operation on full data; `INSERT` : INSERT

	operation on incremental data.
fileName	The name of the binlog file where the current record is located.
position	The end offset of the current record in the binlog in the format of `End_log_pos@binlog file number` . For example, if the current record is in file `mysql-bin.000004` and the end offset is 2196, then the value of this parameter will be `2196@4` .
safePosition	The start offset of the current transaction in the binlog, which is in the same format as described above.
timestamp	The time when the data was written to the binlog, which is a UNIX timestamp in seconds.
gtid	The current GTID, such as c7c98333-6006-11ed-bfc9-b8cef6e1a231:9.
transactionId	The transaction ID, which is generated only for COMMIT events.
serverId	The server ID of the source database, which can be viewed by running `SHOW VARIABLES LIKE 'server_id'` .
threadId	The ID of the session that committed the current transaction, which can be viewed by running `SHOW processlist;` .
sourceType	The source database type, which currently can only be MySQL.
sourceVersion	The source database version, the value of which is the same as that of `select version();`
schemaName	Database name
tableName	Table name
objectName	Format: Database name.table name
columns	The definitions of columns in the table
oldColumns	The data of the row before DML execution. If the message is an INSERT message, the array will be null. There are 12 element types in the array: Integer, Character, Decimal, Float, Timestamp, DateTime, TimestampWithTimeZone, inaryGeometry, TextGeometry, BinaryObject, TextObject, and EmptyObject. For more information, see the demo.
newColumns	The data of the row after DML execution. If the message is a DELETE message, the array will be null. There are 12 element types in the array: Integer, Character, Decimal, Float, Timestamp, DateTime, TimestampWithTimeZone, inaryGeometry,

	TextGeometry, BinaryObject, TextObject, and EmptyObject. For more information, see the demo.
sql	The DDL SQL statement.
executionTime	The DDL execution duration in seconds.
heartbeatTimestamp	The timestamp of the heartbeat message in seconds. This field is present only for heartbeat messages.
syncedGtid	The collection of GTIDs parsed by DTS in the format of `c7c98333-6006-11ed-bfc9-b8cef6e1a231:1-13` .
fakeGtid	Whether the current GTID is forged. If `gtid_mode` is not enabled, DTS will forge a GTID.
pkNames	If the table in the source database has a primary key, this parameter will be carried in the DML message; otherwise, it will not be carried.
readerTimestamp	The time when DTS processed the current data record, which is a UNIX timestamp in milliseconds.
tags	The `status_vars` in `QueryEvent` . For more information, see QueryEvent.
total	The total number of message segments if the message is segmented. This field is invalid on the current version (version=1) and is reserved for extension.
index	The index of message segments if the message is segmented. This field is invalid on the current version (version=1) and is reserved for extension.

The field describing column attributes in a record is `Field` , including the following four attributes:

name: The column name.

dataTypeNumber: The type of the data recorded in the binlog. For values, see MySQL source code documentation.

isKey: Whether the current key is the primary key.

originalType: The type defined in DDL.

Database field mappings

The following lists the mappings between database (such as MySQL) field types and data types defined in the Avro protocol.

Type in MySQL	Corresponding Type in Avro
MYSQL_TYPE_NULL	EmptyObject

MYSQL_TYPE_INT8	Integer
MYSQL_TYPE_INT16	Integer
MYSQL_TYPE_INT24	Integer
MYSQL_TYPE_INT32	Integer
MYSQL_TYPE_INT64	Integer
MYSQL_TYPE_BIT	Integer
MYSQL_TYPE_YEAR	DateTime
MYSQL_TYPE_FLOAT	Float
MYSQL_TYPE_DOUBLE	Float
MYSQL_TYPE_VARCHAR	Character
MYSQL_TYPE_STRING	Character. If the original type is binary, this type will correspond to BinaryObject.
MYSQL_TYPE_VAR_STRING	Character. If the original type is varbinary, this type will correspond to BinaryObject.
MYSQL_TYPE_TIMESTAMP	Timestamp
MYSQL_TYPE_DATE	DateTime
MYSQL_TYPE_TIME	DateTime
MYSQL_TYPE_DATETIME	DateTime
MYSQL_TYPE_TIMESTAMP_NEW	Timestamp
MYSQL_TYPE_DATE_NEW	DateTime
MYSQL_TYPE_TIME_NEW	DateTime
MYSQL_TYPE_DATETIME_NEW	DateTime
MYSQL_TYPE_ENUM	TextObject
MYSQL_TYPE_SET	TextObject
MYSQL_TYPE_DECIMAL	Decimal
MYSQL_TYPE_DECIMAL_NEW	Decimal

MYSQL_TYPE_JSON	TextObject
MYSQL_TYPE_BLOB	BinaryObject
MYSQL_TYPE_TINY_BLOB	BinaryObject
MYSQL_TYPE_MEDIUM_BLOB	BinaryObject
MYSQL_TYPE_LONG_BLOB	BinaryObject
MYSQL_TYPE_GEOMETRY	BinaryObject

JSON Demo Description

Last updated：2024-07-08 19:02:56

The demo for each programming language uses JSON for serialization and contains a `Record` definition file. In the demo for Java, the path of the definition file is `consumerDemo-json-java\\src\\main\\java\\json\\FlatRecord.java`.

Type of Field in Record

Field Name in Record	Description
id	The globally incremental ID.
version	The protocol version, which is v1 currently.
messageType	The message type. Enumerated values: `INSERT`, `UPDATE`, `DELETE`, `DDL`, `BEGIN`, `COMMIT`, `HEARTBEAT`, `CHECKPOINT`.
fileName	The name of the binlog file where the current record is located.
position	The end offset of the current record in the binlog in the format of `End_log_pos@binlog file number`. For example, if the current record is in file `mysql-bin.000004` and the end offset is 2196, then the value of this parameter will be `2196@4`.
safePosition	The start offset of the current transaction in the binlog, which is in the same format as described above.
timestamp	The time when the data was written to the binlog, which is a UNIX timestamp in seconds.
gtid	The current GTID, such as c7c98333-6006-11ed-bfc9-b8cef6e1a231:9.
transactionId	The transaction ID, which is generated only for COMMIT events.
serverId	The server ID of the source database, which can be viewed by running `SHOW VARIABLES LIKE 'server_id'`.
threadId	The ID of the session that committed the current transaction, which can be viewed by running `SHOW processlist;`.
sourceType	The source database type, which currently can only be MySQL.

sourceVersion	The source database version, which can be viewed by running: `select version();` .
schemaName	Database name.
tableName	Table name.
objectName	Format: Database name.table name.
columns	The definitions of columns in the table.
oldColumns	The data of the row before DML execution. If the message is an INSERT message, the array will be null.
newColumns	The data of the row after DML execution. If the message is a DELETE message, the array will be null.
sql	The DDL SQL statement.
executionTime	The DDL execution duration in seconds.
heartbeatTimestamp	The timestamp of the heartbeat message in seconds. This field is present only for heartbeat messages.
syncedGtid	The collection of GTIDs parsed by DTS in the format of `c7c98333-6006-11ed-bfc9-b8cef6e1a231:1-13` .
fakeGtid	Whether the current GTID is forged. If `gtid_mode` is not enabled, DTS will forge a GTID.
pkNames	If the table in the source database has a primary key, this parameter will be carried in the DML message; otherwise, it will not be carried.
readerTimestamp	The time when DTS processed the current data record, which is a UNIX timestamp in milliseconds.
tags	The `status_vars` in `QueryEvent` . For more information, see QueryEvent.
total	The total number of message segments if the message is segmented. This field is invalid on the current version (`version=1`) and is reserved for extension.
index	The index of a message segment if the message is segmented. This field is invalid on the current version (`version=1`) and is reserved for extension.

MySQL Column Attribute in Record

name: The column name.

dataTypeNumber: The type of the data recorded in the binlog. For values, see MySQL source code documentation.

isKey: Whether the current key is the primary key.

originalType: The type defined in DDL.

MySQL Data Type Conversion Logic

In the JSON protocol, all MySQL data types are converted to strings.

String types such as `varchar` are all converted to UTF-8 encoding.

Numeric types are all converted to strings equal to the value, such as "3.0".

Time types are output in the format of `yyyy-dd-mm hh:MM:ss.milli` .

Timestamp types are output as the number of milliseconds.

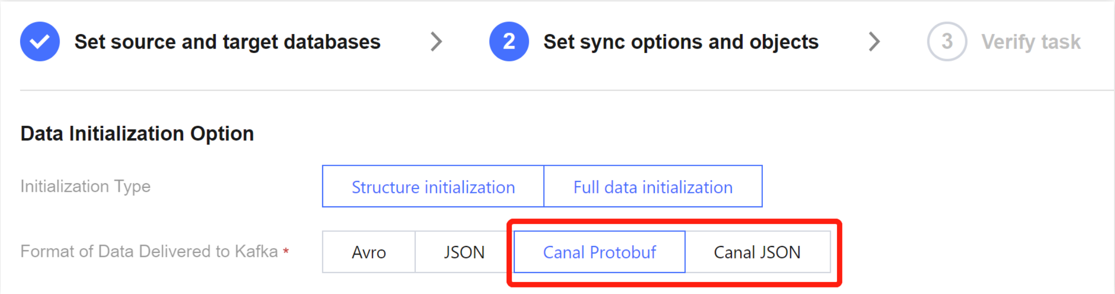
Binary types such as `binary` and `blob` are output as strings equal to their hex values, such as "0xfff".

Canal Demo Description (Canal ProtoBuf/Canal JSON)

Last updated：2024-07-11 15:57:00

Feature Description

The sync data written to Kafka via DTS supports compatibility with the open-source tool in the Canal format, using the ProtoBuf or JSON serialization protocol. During the configuration of DTS sync tasks, you can choose the data format Canal ProtoBuf or Canal JSON, and then use the Consumer Demo for business adaptation, to connect the consumer data.


Set source and target databases
2 Set sync options and objects
3 Verify task

Data Initialization Option

Initialization Type: Structure initialization, Full data initialization

Format of Data Delivered to Kafka *: Avro, JSON, Canal Protobuf, Canal JSON

If you want to learn more about Canal, see Canal details.

Scheme Comparison

Feature	DTS Sync to Kafka Scheme	Canal Sync Scheme
Data Type	Full + increment	Increment only
Data Format	Canal ProtoBuf, Canal JSON	ProtoBuf, JSON
Cost	Purchase cloud resources, which basically require no subsequent maintenance once being configured initially.	Customers shall deploy and maintain by themselves.

Canal JSON Format Compatibility Statement

Users can consume data using the consumption program from the previous Canal scheme. When consuming data in the Canal JSON format in the DTS scheme, the field names are consistent with those in the Canal scheme's JSON format, and only the following differences need to be noted.

1. In the source database, fields of binary-related types (including binary, varbinary, blob, tinyblob, mediumblob, longblob and geometry) will be converted into HexString after being synced to the target. Users should be aware of this when consuming data.
2. Fields of the Timestamp type in the source database will be converted to the 0 timezone (e.g., 2021-05-17 07:22:42 +00:00) when they are synced to the target. Users need to consider the timezone information when parsing and converting.
3. The JSON format of the Canal scheme defines the sqlType field, which is used in Java Database Connectivity (JDBC) to represent the SQL data type. Since Canal uses Java at the bottom layer, and DTS is implemented in Golang at the bottom layer, this field is left empty in the Canal JSON format provided by DTS.

Canal ProtoBuf Format Compatibility Statement

For consuming data in the Canal ProtoBuf format, it is necessary to use the protocol document provided by DTS, because this protocol document incorporates features such as full sync logic, which is included in the Consumer Demo. Therefore, users need to use the Consumer Demo provided by DTS, and adapt their own business logic based on this Demo in order to connect the consumer data.

When data is consumed in the Canal ProtoBuf format provided by DTS, the field names are consistent with the ProtoBuf format provided by the Canal scheme, and only the following differences need to be noted.

1. In the source database, fields of binary-related types (including binary, varbinary, blob, tinyblob, mediumblob, longblob and geometry) will be converted into HexString after being synced to the target. Users should be aware of this when consuming data.
2. Fields of the Timestamp type in the source database will be converted to the 0 timezone (e.g., 2021-05-17 07:22:42 +00:00) when being synced to the target. Users need to consider the timezone information when parsing and converting.

TDSQL MySQL Sync to Kafka Supported Capabilities

Last updated：2024-07-08 19:02:56

Major Feature Categories	Feature Subitem or Description	Supporting Capability
Sync Object	-	Ordinary object table, index, view
Sync Type	-	Structure sync Full sync (automatically continue with incremental after full completion) Incremental sync
Breakpoint Resume	-	Partial support for full export (structure export, no primary key table export; stage does not support resume); Full import, incremental support.
Task Channel Key Operation	Retry	Supported
	Pause, recover	Supported
	Modify sync configuration (dynamically add or remove table objects)	Not supported
	Create similar task	Supported
	Speed limit	Supported
Sync Direction	Many-to-one (multiple instances merged into one/multiple tables merged into one)	Supported
	One-to-many	Supported
Structure Sync	Database and table filtering	Supported
	Column filtering	Not supported
	Skip sync table and correction	Not supported
Full Sync	Where condition filtering	Not supported

Incremental Sync	DML sync (INSERT/UPDATE/DELETE)	Supported
	DML filtering	Supported
	DDL sync	CREATE DATABASE, DROP DATABASE, ALTER DATABASE CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE, RENAME TABLE CREATE VIEW, DROP VIEW CREATE INDEX, DROP INDEX The CREATE TABLE table_name AS SELECT statement is not supported.
	DDL filtering	Supported
	Where condition filtering	Not supported
	Scheduled sync	Not supported
	Online DDL sync	Supported
Data Subscription (Synced to Kafka Link-Specific)	Data format	Avro, JSON
	Designated multi-topic delivery	Supported
	Specified single topic multi-partition delivery	Supported
	Specified single topic + custom partition strategy	Not supported
	Filter transaction BEGIN/COMMIT	Not supported
	Filter CHECKPOINT messages	Not supported

Use Instructions

Last updated：2024-07-08 19:02:56

Category	Description
Sync Object	1. Only sync of base tables and views is supported. Sync of functions, triggers, storage processes, and other objects is not supported. 2. Only sync of InnoDB, MyISAM, and TokuDB database engines is supported. Tables with any other data engines will be skipped by default. 3. Interrelated data objects need to be synced at the same time, otherwise sync failure will occur. 4. During the incremental sync stage, if source database table names include TDSQLagent and tdsql_sub characters, they might be filtered out or cause sync anomalies, as these table names are the same as TDSQL system's temporary table names. TDSQLagent is a temporary table for scaling, and tdsql_sub tables are subtables for hash-list and hash-range. Therefore, it is recommended not to set the source's tables to be synced with these names.
Sync Feature	1. A single task can associate with multiple Topics, but the same Topic cannot be used by multiple sync tasks simultaneously, as it would lead to data corruption. The data consumed would be from different tasks, and could also lead to exceptions in data processing or errors during task retries. 2. During the full sync stage, for every 100,000 rows of data synced, a checkpoint message is inserted into the target Kafka to mark the current data sync position. 3. The single message synced to the target Kafka by DBbridge has a performance limit. It is recommended that single row data in the source database not exceed 8 MB to avoid errors during the sync process. 4. If a user is sure to rename a table during the sync process, when delivered to the target Kafka, the partitioning rule will match the Topic and partition according to the new table name.
Source Database Impact	1. During data sync, DBbridge will use the account that executes the sync task to write the system database `__tencentdb__` in the source database, recording transaction marker ID and other metadata. It is necessary to ensure that the source database has read-write permissions for `__tencentdb__` . To ensure that data comparison issues can be traced, the `__tencentdb__` system database will not be deleted from the source database after the sync task is completed The space occupied by the `__tencentdb__` system database is very small, approximately one thousandth to one ten-thousandth of the source database's storage space (for example, if the source database is 50 GB, then the `__tencentdb__` system database is about 5 MB - 50 MB). Moreover, by using a single thread and a waiting connection mechanism, it has almost no impact on the source database's performance and will not compete for resources. 2. By default, a lock-free sync method is used. The full data export stage does not place a global lock (FTWRL) on the source database; it only places table locks on tables without a primary key.

Target Kafka requirements	1. It is necessary to modify the message retention time and message size limit in the target Kafka. The message retention time is recommended to be set for 3 days. Data exceeding the retention time will be cleared. Users are advised to consume the data in a timely manner within the set time. The message size limit refers to the maximum size of a single message's memory that Kafka can receive. When setting, it needs to be larger than the maximum size of a single row of data in the source database table, to ensure that data from the source database can be properly delivered to Kafka. 2. It is recommended that the target Topic is empty. Also, during the sync task process, do not write data into the Topic selected for sync at the target, as it may cause message disorder and task errors.
Restart impact	During the sync task, if a task restart occurs (e.g., a restart due to HA switchover at the source database or a restart caused by back-end task anomaly detection), **it may** lead to duplicated data in the target Kafka. DBbridge syncs based on the smallest data cell (in the full volume phase, a chunk of data from a single table object is the smallest data cell; in the incremental phase, each marked location is a data cell). If a data cell has finished syncing and then a restart occurs, it will not cause data duplication; however, if a restart occurs while a data cell is still syncing, the data cell will need to be resynced upon restart, leading to data duplication. If users are particularly concerned about duplicated data, please set deduplication logic when consuming data.
Operation Restrictions	1. During the full data export stage, please do not execute any DDL operations that change the database or table structure in the source database. 2. Do not modify or delete user information (including username, password, and permissions) and port numbers in the source database and the target. 3. Do not clear Binlogs in the source database.
Data Type	1. When TDSQL MySQL is used as the source database and connected via the proxy method, if the source database employs the float data type for floating numbers, it will cause data precision issues in the full volume phase. To maintain data precision, it is recommended to use the double data type. The impact of precision issues in the full volume phase is as follows: The precision of data sync between the full volume stage and the incremental stage is inconsistent. Using float as the key value, the primary key data operated in the full volume stage and the incremental stage are inconsistent. 2. During the full data sync stage, DBbridge exports data from the source database and imports it into the target Kafka, both using the utf8mb4 character set to avoid garbled character issues due to unrecognized character sets. 3. Geometry-related data types are not supported, and tasks will report an error when encountering such data types. 4. During the incremental sync process, if the source database generates Binlog statements in the STATEMENT format, it will lead to sync failure. 5. When the data type delivered to Kafka is selected as OGG JSON (ROW), the source database's tables cannot contain column names like gtid, current_ts, op_ts, op_type, pos,

	primary_keys, and set_id, table. These column names conflict with the field names in the parsed JSON header, leading to data being overwritten and abnormal consumption data if these column names exist in the source database's tables. 6. Before the sync task, for the full volume of data already existing in the source database, since there's no accurate time of data insertion, the happenAt field in the Consumer Demo will display as 1970-01-01 08:00:00, which needs no attention. When the incremental data is consumed, the time field can be correctly displayed.
Transaction	Scenarios that include both DML and DDL statements in one transaction are not supported; encountering such a scenario will result in an error.
HA Switch and Scaling	1. If the source database is a non-GTID database, DBbridge does not support source HA switchover. Once the source TDSQL MySQL undergoes a switchover, it may cause DBbridge incremental sync interruption. 2. DBbridge uses SET for direct connection to TDSQL MySQL, and does not support scaling of TDSQL MySQL. When Proxy is used to connect to TDSQL MySQL, logical scaling and scale-out of TDSQL MySQL are supported. However, DBbridge may report errors during scale-out.
Partition Table Sync	1. The syntax of primary/secondary partition tables must comply with standards. Primary Hash partition tables only support creation through the shardkey method. The key syntax for creating partition tables in TDSQL MySQL is as follows. For detailed syntax, refer to TDSQL MySQL Create Table Syntax Example. Primary Hash partition: shardkey Primary Range partition: TDSQL_DISTRIBUTED BY RANGE Primary List partition: TDSQL_DISTRIBUTED BY LIST Primary Hash partition + secondary Range/List partition: shardkey + PARTITION BY RANGE/LIST Primary Range partition + secondary Range/List partition: TDSQL_DISTRIBUTED BY RANGE + PARTITION BY RANGE/LIST Primary List partition + secondary Range/List partition: TDSQL_DISTRIBUTED BY LIST + PARTITION BY RANGE/LIST 2. During the incremental sync stage, concurrent DDL operations are not supported. The next DDL operation should be executed after the previous one takes effect, otherwise errors may occur due to out-of-order DDL operations. Also, rapid create, drop and create operations on tables with the same name are not supported, as they may cause incorrect table type errors.
Designated Position Sync	1. If full sync and incremental sync are executed separately, please note that during full sync, there cannot be DDL changes between the location of the fully synced data by DBbridge (position 1) and the start location (position 2) set for incremental sync; otherwise, the task errors will occur. 2. From initiating synced data to before the start of the incremental task (i.e., before the task step changes from finding the specified location to syncing the increment), it is recommended not to carry out master-slave switch, increase sharding, or redo standby machine operations on the source library, as these may affect DBbridge's access to the GTID location of the source library. Also, the source cannot perform DDL operations; otherwise, the task errors will occur.

	3. Because designated position sync is determined based on the time in the Binlog context (SET TIMESTAMP = XXXX) to judge its GTID, to ensure the accuracy of synced data, it is advised not to modify this context. 4. Please ensure that the database's setting time zone is consistent with the current console time zone (i.e., the browser's time zone), or convert it to the corresponding time of the database setting time zone; otherwise, the results of the designated position sync may not meet expectations. 5. Please make sure that the time is consistent across all cluster nodes of TDSQL MySQL; otherwise, the results of the designated position sync may not meet expectations. 6. If the first event at the specified sync point is a two-stage XA transaction, then that XA transaction will not be synced.

Data Sync Guide

Last updated：2024-07-08 19:02:56

DTS allows you to sync the full and incremental data of the source database to Ckafka, so that you can quickly obtain business change data and use it. This document describes how to use DTS to sync data from TDSQL for MySQL to Ckafka.

Currently, TDSQL for MySQL is the only supported source database type.

Prerequisites

The source and target databases must meet the requirements for the sync feature and version as instructed in Databases Supported by Data Sync.

Source database permissions required for the sync task account:

```
GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT, REPLICATION SLAVE, SELECT ON *.* TO
FLUSH PRIVILEGES;
```

You need to modify the message retention period and message size limit in target Ckafka.

We recommend that you set the message retention period to 3 days. The data beyond the retention period will be cleared, so you need to consume data in time within the set period. The upper limit for message size refers to the maximum size of a single message that Ckafka can receive. You must set it to be greater than the maximum size of a single row of data in the source database table so that data can be normally delivered to CKafka.

Directions

1. Log in to the data sync task purchase page, select appropriate configuration items, and click **Buy Now**.

Parameter	Description
Billing Mode	Monthly subscription and pay-as-you-go billing modes are supported.
Source Instance Type	Select TDSQL for MySQL, which cannot be changed after purchase.
Source Instance Region	Select the source instance region, which cannot be changed after purchase.
Target Instance Type	Select Kafka, which cannot be changed after purchase.
Target Instance Region	Select the target instance region, which cannot be changed after purchase.
Specification	Select a specification based on your business needs. The higher the specification, the higher the performance. For more information, see Billing Overview.

2. After making the purchase, return to the data sync task list to view the task you just created.Then, click **Configure** in the **Operation** column to enter the **Configure Sync Task** page.
3. On the **Configure Sync Task** page, configure **Instance ID**, **Account**, and **Password** for the source instance, configure **Instance ID** for the target instance, **test connectivity**, and click **Next**.

Setting Items	Parameter	Description
Task Configuration	Task Name	DTS will automatically generate a task name, which is customizable.
	Running Mode	Immediate execution and scheduled execution are supported.
Source Instance Settings	Source Instance Type	The source database type selected during purchase, which cannot be changed.
	Source Instance Region	The source instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, you can only select **Database**.
	Account/Password	Account/Password: Enter the source database account and password.

Target Instance Settings	Target Instance Type	The target instance type selected during purchase, which cannot be changed.
	Target Instance Region	The target instance region selected during purchase, which cannot be changed.
	Access Type	Select a type based on your scenario. In this scenario, select **CKafka instance**.
	Instance ID	Select the instance ID of the target instance.

4. On the **Set sync options and objects** page, set the following items: **Data Initialization Option**, **Policy for Syncing Data to Kafka**, **Data Sync Option**, and **Sync Object Option**. Then click **Save and Go Next**.

Deliver to custom topic

Deliver to a single topic

Setting Items	Parameter	Description
Data Initialization Option	Initialization Type	Structure initialization: **Table structures** in the source instance will be initialized into the target instance before the sync task runs. Full data initialization: **Data** in the source instance will be initialized into the target instance before the sync task runs. If you select **Full data initialization** only, you need to create the table structures in the target database in advance. Both options are selected by default, and you can deselect them as needed.
	Format of Data Delivered to Kafka	Avro adopts the binary format with a higher consumption efficiency, while JSON adopts the easier-to-use lightweight text format.
Policy for Syncing Data to Kafka	Topic Sync Policy	Deliver to custom topic: Customize the topic name for delivery. After that, the target Kafka will automatically create a topic with the custom name. The synced data is randomly delivered to different partitions under the topic. **If the target Kafka fails to create the topic, the task will report an error.** Deliver to a single topic: Select an existing topic on the target side, and then deliver data based on multiple partitioning policies. Data can be delivered to a single partition of the specified topic, or delivered to different partitions by table name or by table name + primary key.
	Rules for delivering to custom topic	If you add multiple rules, the database and table rules are matched one by one from top to bottom. If no rules are matched, data will be delivered to the topic corresponding to the last rule. If multiple rules are matched, data will be delivered to the topics corresponding to all the matched rules. Example 1: There are tables named "Student" and "Teacher" in a database named "Users" on database instance X. If you need to deliver the data in the

		"Users" database to a topic named "Topic_A". The rules are configured as follows: Enter `Topic_A` for **Topic Name**, `^Users$` for **Database Name Match**, and `.*` for **Table Name Match**. Enter `Topic_default` for **Topic Name**, `Databases that don't match the above rules` for **Database Name Match**, and `Tables that don't match the above rules` for **Table Name Match**. Example 1: There are tables named "Student" and "Teacher" in a database named "Users" on database instance X. If you need to deliver the data in the "Student" table and "Teacher" tables to topics named "Topic_A" and "Topic_default" respectively. The rules are configured as follows: Enter `Topic_A` for **Topic Name**, `^Users$` for **Database Name Match**, and `^Student$` for **Table Name Match**. Enter `Topic_default` for **Topic Name**, `Databases that don't match the above rules` for **Database Name Match**, and `Tables that don't match the above rules` for **Table Name Match**.
	Rules for delivering to a single topic	After selecting a specified topic, the system will perform partitioning based on the specified policy as follows. Deliver all data to partition 0: Deliver all the synced data of the source database to the first partition. By table name: Partition the synced data from the source database by table name. After setting, the data with the same table name will be written into the same partition. By table name + primary key: Partition the synced data from the source database by table name and primary key. This policy is suitable for frequently accessed data. After settings, frequently accessed data is distributed from tables to different partitions by table name and primary key, so as to improve the concurrent consumption efficiency.
	Topic for DDL Storage	(Optional) If you need to deliver the DDL operation of the source database to the specified topic separately, you can select the settings here. After setting, it will be delivered to Partition 0 of the selected topic by default; if not set, it will be delivered based on the topic rules selected above.

Data Sync Option

Setting Items	Parameter	Description
Data Sync Option	SQL Type	The following operations are supported: INSERT, DELETE, UPDATE, and DDL.
Sync Object	Database and Table	Only the database/table objects can be synced.

Option	Objects of Source Instance	

5. On the task verification 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.
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 sync task list, and you can see that the task has entered the **Running** status.

Note

You can click **More** > **Stop** in the **Operation** column to stop a sync task. Before doing so, ensure that data sync has been completed.

7. (Optional) you can click a task name to enter the task details page and view the task initialization status and monitoring data.

Subsequent Operations

After the data is synced to the target Kafka, the data can be consumed. We provide you with a c

Data Consumption Guide

Last updated：2024-07-08 19:02:56

Overview

After data is synced to Kafka, you can consume the subscribed data through Kafka 0.11 or later available at DOWNLOAD. This document provides client consumption demos for Java, Go, and Python for you to quickly test the process of data consumption and understand the method of data parsing.

Note

The demo only prints out the consumed data and does not contain any usage instructions. You need to write your own data processing logic based on the demo. You can also use Kafka clients in other languages to consume and parse data.

The upper limit of message size in target CKafka must be greater than the maximum value of a single row of data in the source database table so that data can be normally synced to CKafka.

In scenarios where only the specified databases and tables (part of the source instance objects) are synced and single-partition Kafka topics are used, only the data of the synced objects will be written to Kafka topics after DTS parses incremental data. The data of non-sync objects will be converted into empty transactions and then written to Kafka topics. Therefore, there are empty transactions during message consumption. The BEGIN and COMMIT messages in the empty transactions contain the GTID information, which can ensure the GTID continuity and integrity. In the consumption demos for MySQL/TDSQL-C for MySQL, multiple empty transactions have been compressed to reduce the number of messages.

To ensure that data can be rewritten from where the task is paused, DTS adopts the checkpoint mechanism for data sync links where Kafka is the target end. Specifically, when messages are written to Kafka topics, a checkpoint message is inserted every 10 seconds to mark the data sync offset. When the task is resumed after being interrupted, data can be rewritten from the checkpoint message. The consumer commits a consumption offset every time it encounters a checkpoint message so that the consumption offset can be updated timely.

When the selected data format is JSON, if you have used or are familiar with the open-source subscription tool Canal, you can choose to convert the consumed JSON data to a Canal-compatible data format for subsequent processing. The demo already supports this feature, and you can implement it by adding the `trans2canal` parameter in the demo startup parameters. Currently, this feature is supported only in Java.

Downloading Consumption Demos

When configuring the sync task, you can select Avro or JSON as the data format. Avro adopt the binary format with a higher consumption efficiency, while JSON adopts the easier-to-use lightweight text format. The reference demo varies by the selected data format.

The following demo already contains the Avro/JSON protocol file, so you don't need to download it separately.

For the description of the logic and key parameters in the demo, see Demo Description.

Demo Language	Avro Format	JSON Format
Go	Address	Address
Java	Address	Address
Python	Address	Address

Instructions for the Java Demo

Compiling environment: The package management tool Maven or Gradle, and JDK8. You can choose a desired package management tool. The following takes Maven as an example.

The steps are as follows:

1. Download the Java demo and decompress it.
2. Access the decompressed directory. Maven model and pom.xml files are placed in the directory for your use as needed.

Package with Maven by running `mvn clean package` .

3. Run the demo.

After packaging the project with Maven, go to the target folder `target` and run `java -jar consumerDemo-avro-1.0-SNAPSHOT.jar --brokers xxx --topic xxx --group xxx --trans2sql` .

`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic` , you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.

`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.

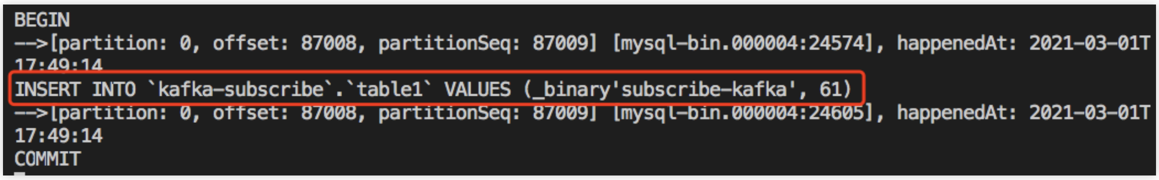
`trans2sql` indicates whether to enable conversion to SQL statement. In Java code, if this parameter is carried, the conversion will be enabled.

`trans2canal` indicates whether to print the data in Canal format. If this parameter is carried, the conversion will be enabled.

Note：

If `trans2sql` is carried, `javax.xml.bind.DatatypeConverter.printHexBinary()` will be used to convert byte values to hex values. You should use JDK 1.8 or later to avoid incompatibility. If you don't need SQL conversion, comment this parameter out.

4. Observe the consumption.



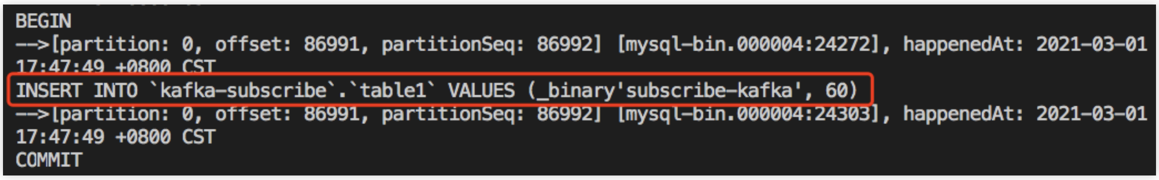
```
BEGIN
-->[partition: 0, offset: 87008, partitionSeq: 87009] [mysql-bin.000004:24574], happenedAt: 2021-03-01T
17:49:14
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 61)
-->[partition: 0, offset: 87008, partitionSeq: 87009] [mysql-bin.000004:24605], happenedAt: 2021-03-01T
17:49:14
COMMIT
```

Instructions for the Go Demo

Compiling environment: Go 1.12 or later, with the Go module environment configured.

The steps are as follows:

1. Download the Go demo and decompress it.
2. Access the decompressed directory and run `go build -o subscribe ./main/main.go` to generate the executable file `subscribe`.
3. Run `./subscribe --brokers=xxx --topic=xxx --group=xxx --trans2sql=true`.
`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic`, you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.
`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.
`trans2sql` indicates whether to enable conversion to SQL statement.
4. Observe the consumption.



```
BEGIN
-->[partition: 0, offset: 86991, partitionSeq: 86992] [mysql-bin.000004:24272], happenedAt: 2021-03-01
17:47:49 +0800 CST
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 60)
-->[partition: 0, offset: 86991, partitionSeq: 86992] [mysql-bin.000004:24303], happenedAt: 2021-03-01
17:47:49 +0800 CST
COMMIT
```

Instructions for the Python3 Demo

Compiling runtime environment: Install Python3 and pip3 (for dependency package installation).

Use `pip3` to install the dependency package:

```
pip install flag
pip install kafka-python
pip install avro
```

The steps are as follows:

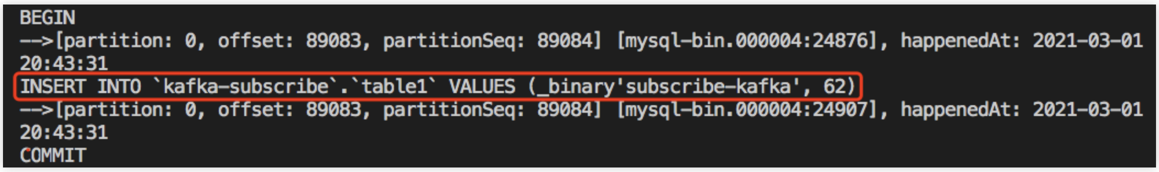
1. Download Python3 demo and decompress it.
2. Run `python main.py --brokers=xxx --topic=xxx --group=xxx --trans2sql=1` .

`brokers` is the CKafka access address, and `topic` is the topic name configured for the data sync task. If there are multiple topics, data in these topics need to be consumed separately. To obtain the values of `brokers` and `topic` , you can go to the **Data Sync** page and click **View** in the **Operation** column of the sync task list.

`group` is the consumer group name. You can create a consumer in CKafka in advance or enter a custom group name here.

`trans2sql` indicates whether to enable conversion to SQL statement.

3. Observe the consumption.



```
BEGIN
-->[partition: 0, offset: 89083, partitionSeq: 89084] [mysql-bin.000004:24876], happenedAt: 2021-03-01
20:43:31
INSERT INTO `kafka-subscribe`.`table1` VALUES (_binary'subscribe-kafka', 62)
-->[partition: 0, offset: 89083, partitionSeq: 89084] [mysql-bin.000004:24907], happenedAt: 2021-03-01
20:43:31
COMMIT
```

Task Management

Task Status Description

Last updated：2024-07-08 19:09:49

Billable task status description

Note

A monthly subscribed task is billed upon purchase. A pay-as-you-go task is free of charge during configuration and will incur fees only when it runs normally, and billing will stop when it is paused.

The **Renew** button is displayed only for monthly subscribed task instances.

The **Postpaid-to-Prepaid** button is displayed only for pay-as-you-go task instances.

Status	Description	Supported Operations
Uninitialized	The purchase has been completed, but no sync task has been configured.	View, Configure, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Initialized	The sync task has been configured.	View, Configure, Verify, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Checking	The sync task is being checked.	View, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Verification passed	The sync task passed the verification.	View, Configure, Start, Verify, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Verification failed	The sync task failed the verification.	View, Configure, Verify, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Preparing	The sync task is ready to start after passing the verification.	View, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Running	The sync task is running.	View, Pause, Stop, Modify Sync Configuration, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return

Task error	The sync task was interrupted during sync due to an exception. You can click **Retry** to resume the task in the console.	View, Retry, Stop, Adjust Specification, Postpaid-to-Prepaid, Renew, Renew, Terminate/Return
Retrying	The task is being retried after you click **Retry**.	View, Stop, Adjust Specification, Postpaid-to-Prepaid, Renew, Terminate/Return
Pausing	The sync task is being paused after you click **Pause**.	View, Terminate/Return
Paused	The task is paused. If it is paused for more than seven days, it will fail.	View, Resume, Stop, Terminate/Return
Resuming	The paused task is being resumed. After it is resumed, its status will become **Running**.	View, Terminate/Return
Stopping	The task is being stopped after you click **Stop**. Once stopped, it cannot be resumed.	View, Terminate/Return
Stopped	The sync task has ended.	View, Terminate/Return, Reset
Resetting	The stopped task is being reset. You can reset it to clear the old task configuration. After the task status becomes **Uninitialized**, you can configure a new sync task.	View, Terminate/Return
Task failed	The sync task failed.	View, Terminate/Return
Adjusting specification	The task specification is being adjusted after you click **Adjust Specification**.	View, Renew
Isolating	The task instance is being isolated after you click **Terminate/Return**. The task instance is being isolated by the system automatically upon expiration or one day after your account has overdue payments.	View, Renew, Delete Instantly
Isolated	The task is isolated after you click **Terminate/Return** or it is moved to the recycle bin due to expiration or overdue payments. The task is isolated due to expiration or overdue payments. It can be restored after it is renewed or your account balance is topped up.	View, Deisolate/Resume, Renew, Delete Instantly
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.	View
Topping up	The account balance is being topped up, and the pay-as-you-go task moved to the recycle bin due to expiration or overdue payments can be restored.	View, Renew
Renewing	The monthly subscribed task is being renewed.	View

Free task status description

Status	Description	Supported Operations
Uninitialized	The purchase has been completed, but no sync task has been configured.	View, Configure, Delete
Initialized	The sync task has been configured.	View, Configure, Verify, Delete
Checking	The sync task is being checked.	View, Delete
Verification passed	The sync task passed the verification.	View, Configure, Start, Verify, Delete
Verification failed	The sync task failed the verification.	View, Configure, Verify, Delete
Preparing	The sync task is ready to start after passing the verification.	View, Delete
Running	The sync task is running.	View, Pause, Stop, Delete
Task error	The sync task was interrupted during sync due to an exception. You can click **Retry** to resume the task in the console.	View, Retry, Stop, Delete
Retrying	The task is being resumed after you click **Retry**.	View, Stop, Delete
Stopping	The task is being stopped after you click **Stop**.	View, Delete
Stopped	The task is stopped.	View, Delete
Task failed	The sync task failed.	View, Delete
Deleting	The task is being deleted after you click **Delete**. Once deleted, it cannot be restored, and its resources will be	View

	cleared.	

Viewing Task

Last updated：2024-07-08 19:09:49

Overview

In task details, you can view various information, such as sync task, sync configuration, structure initialization, data initialization, and monitoring data.

Prerequisites

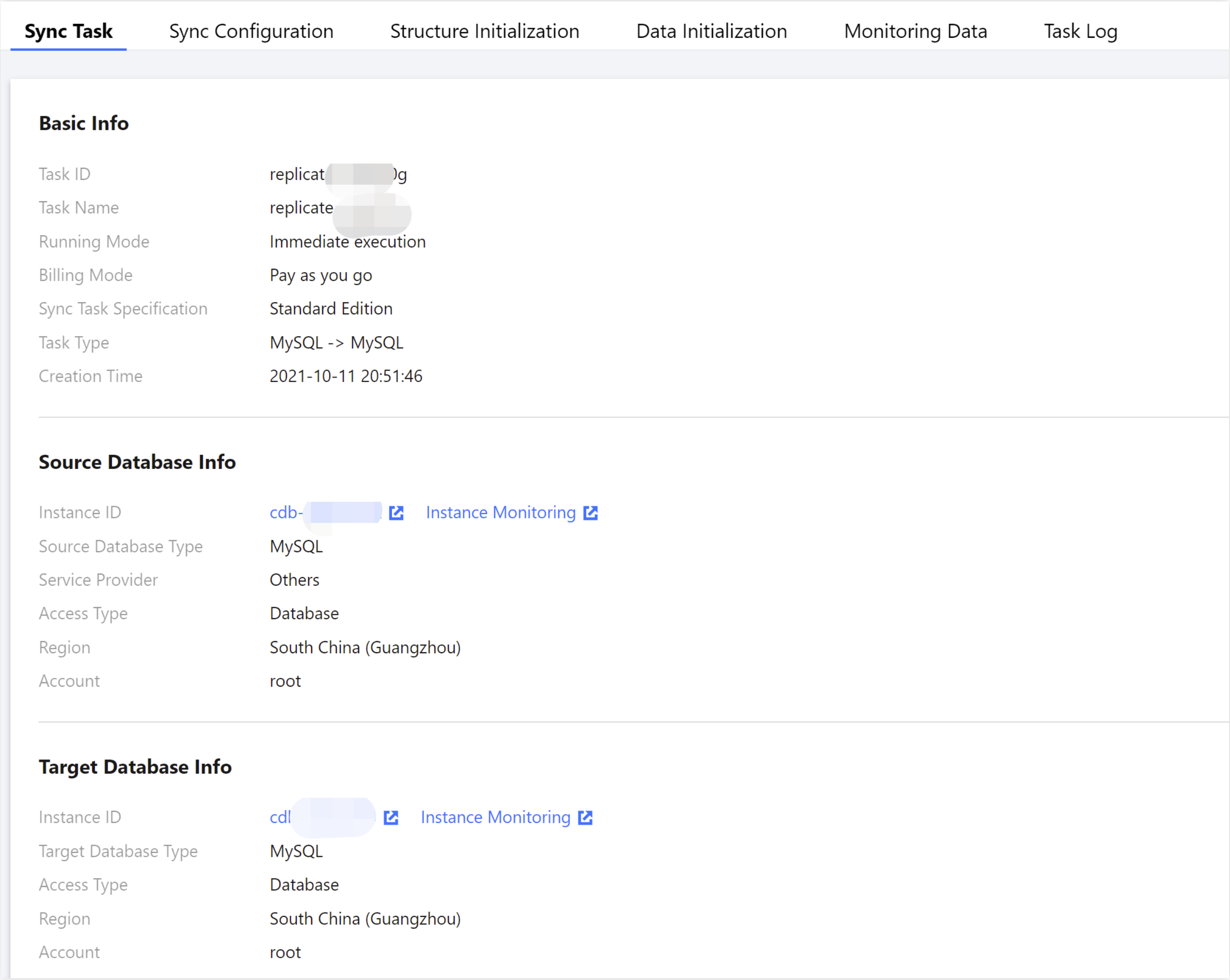
You have successfully created a data sync task and logged in to the DTS console.

Directions

Method 1: in the data sync task list, select the target sync task and click its name.
Method 2: in the data sync task list, select the target sync task and click **View** in the **Operation** column.

Sync task page

It displays the information of the task, source database, and target database.


Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log
Instance Monitoring
Instance Monitoring

Basic Info

Task ID	replicat[redacted]g
Task Name	replicate[redacted]
Running Mode	Immediate execution
Billing Mode	Pay as you go
Sync Task Specification	Standard Edition
Task Type	MySQL -> MySQL
Creation Time	2021-10-11 20:51:46

Source Database Info

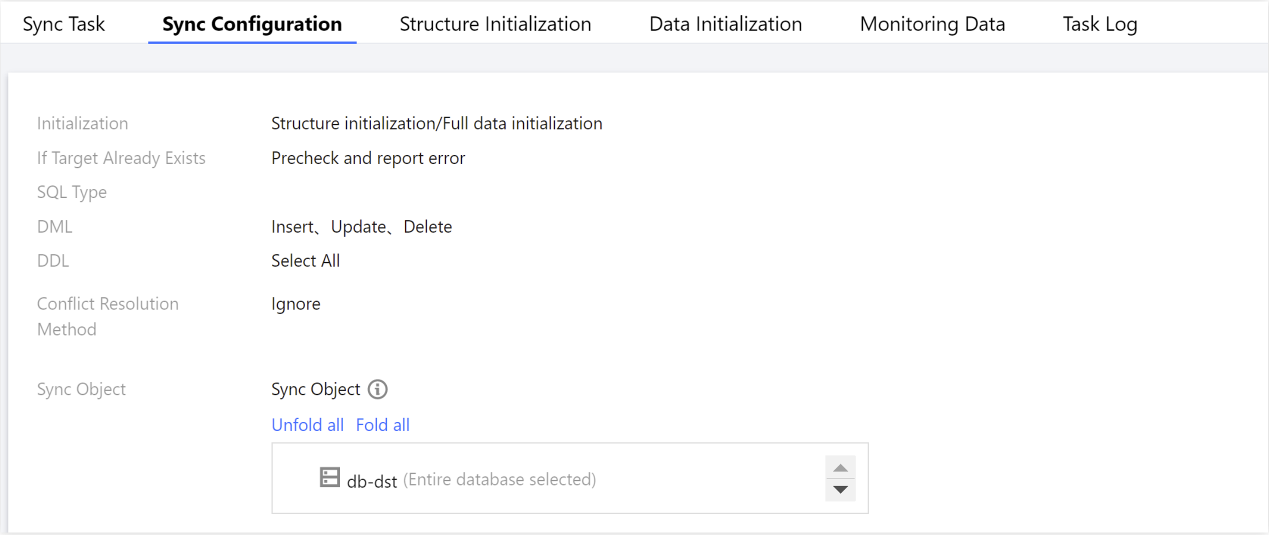
Instance ID	cdb-[redacted]
Source Database Type	MySQL
Service Provider	Others
Access Type	Database
Region	South China (Guangzhou)
Account	root

Target Database Info

Instance ID	cdl[redacted]
Target Database Type	MySQL
Access Type	Database
Region	South China (Guangzhou)
Account	root

Sync configuration page

It displays the sync task configuration.

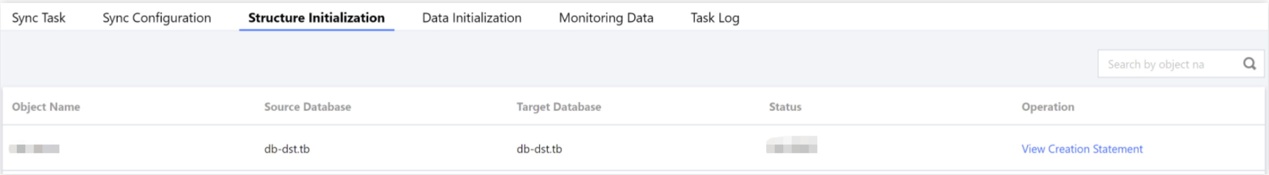

Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log
Unfold all
Fold all

Initialization	Structure initialization/Full data initialization
If Target Already Exists	Precheck and report error
SQL Type	
DML	Insert、Update、Delete
DDL	Select All
Conflict Resolution Method	Ignore
Sync Object	Sync Object ⓘ db-dst (Entire database selected)

Structure initialization page

If you select **Structure Initialization** in the sync configuration, relevant information will be displayed on this page.

Parameter	Description
Object Name	Table name in the target database.
Source Database	Name of the source database object, such as database and table.
Target Database	Name of the target database object, such as database and table.
Status	Current status. If the task fails, the failure reason will be displayed.
Operation	Click **View Creation Statement** to view the creation syntax executed in the target database.


Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log
Search by object na

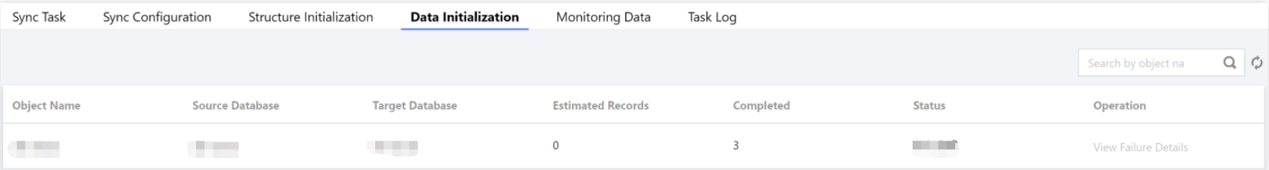
Object Name	Source Database	Target Database	Status	Operation
[redacted]	db-dst.tb	db-dst.tb	[redacted]	View Creation Statement

Task initialization page

If you select **Data Initialization** in the sync configuration, relevant information will be displayed on this page.

Note:

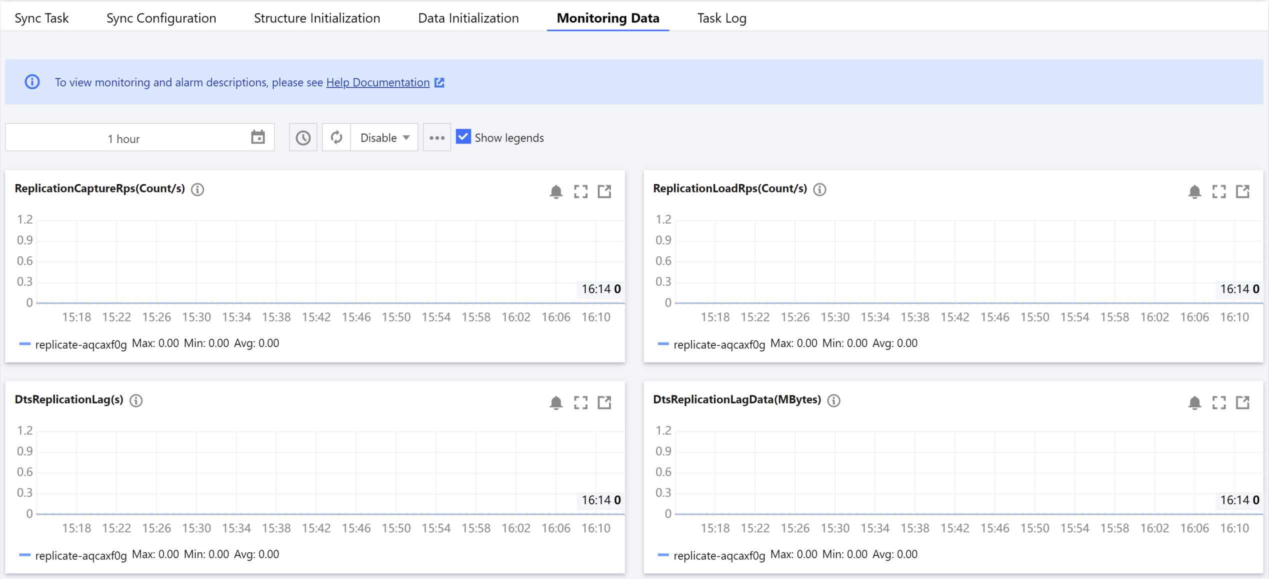
If the task fails, you can click **View Failure Details** to troubleshoot.


Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log
Search by object na

Object Name	Source Database	Target Database	Estimated Records	Completed	Status	Operation
[redacted]	[redacted]	[redacted]	0	3	[redacted]	View Failure Details

Monitoring data display

"-" will be displayed by default if no monitoring data is collected.



Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log

To view monitoring and alarm descriptions, please see Help Documentation

1 hour
Disable
☑ Show legends

ReplicationCaptureRps(Count/s)

1.2
0.9
0.6
0.3
0
15:18 15:22 15:26 15:30 15:34 15:38 15:42 15:46 15:50 15:54 15:58 16:02 16:06 16:10
16:14 0

replicate-aqcaxf0g Max: 0.00 Min: 0.00 Avg: 0.00

ReplicationLoadRps(Count/s)

1.2
0.9
0.6
0.3
0
15:18 15:22 15:26 15:30 15:34 15:38 15:42 15:46 15:50 15:54 15:58 16:02 16:06 16:10
16:14 0

replicate-aqcaxf0g Max: 0.00 Min: 0.00 Avg: 0.00

DtsReplicationLag(s)

1.2
0.9
0.6
0.3
0
15:18 15:22 15:26 15:30 15:34 15:38 15:42 15:46 15:50 15:54 15:58 16:02 16:06 16:10
16:14 0

replicate-aqcaxf0g Max: 0.00 Min: 0.00 Avg: 0.00

DtsReplicationLagData(MBytes)

1.2
0.9
0.6
0.3
0
15:18 15:22 15:26 15:30 15:34 15:38 15:42 15:46 15:50 15:54 15:58 16:02 16:06 16:10
16:14 0

replicate-aqcaxf0g Max: 0.00 Min: 0.00 Avg: 0.00

Configuring Task

Last updated：2024-07-08 19:09:49

Overview

After purchasing sync resources, you need to configure the detailed parameters of a new sync task. You can also modify the parameters of a completed configuration task, such as sync object and type.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **Configure** in the **Operation** column.
2. On the sync task page, configure or modify the relevant parameters and save your editing.

For the requirements and instructions on configuring different links, see Data Sync.

Verifying Task

Last updated：2024-07-08 19:09:49

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 Sync** on the left sidebar, select the target sync task, and click **More** > **Verify** in the **Operation** column.
2. In the pop-up window, you can create a verification task and query the verification result.

Starting Task

Last updated：2024-07-08 19:09:49

Overview

If an exception occurs during task verification before a sync task starts, the task will be verified again after you fix it. In this case, you need to start the task after successful verification.

Prerequisites

The task has passed data verification.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **More** > **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 19:09:49

Overview

After a sync task is started, if it is interrupted due to an exception, you can manually retry it.

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 sync:

Full export: Retry is only supported for exporting unlocked tables with primary keys. It doesn't work when exporting locked tables.

Full import: Retry is supported.

Incremental sync: Retry is supported.

Application Scope

Retry is supported data sync links among MySQL, TDSQL-C or MySQL, MariaDB, Percona, and Kafka.

Directions

Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **More** > **Retry** in the **Operation** column.

Speed Limit

Last updated：2024-07-08 19:09:49

Overview

In the DTS data sync 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. For details, see DTS Performance Tuning Best Practices.

Application Scope

Data sync 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 sync 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.

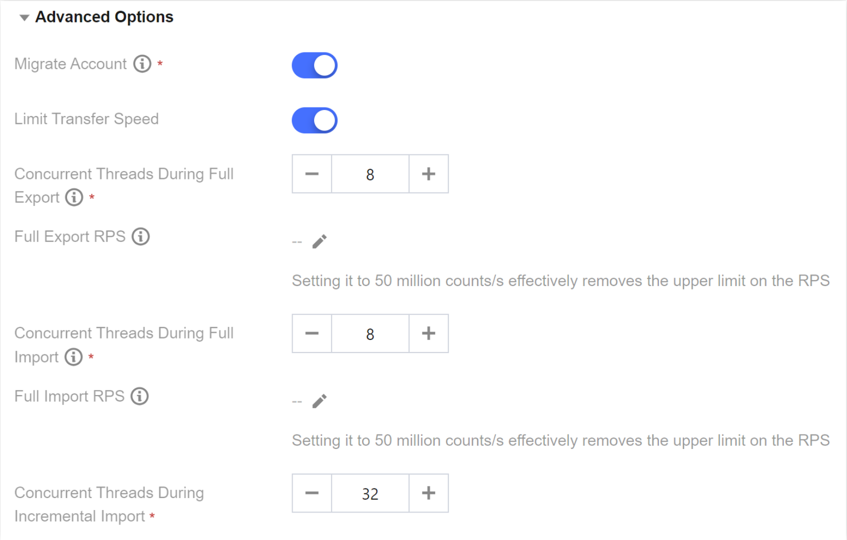
Task Stage	Adjust Scenario	Adjustment Plan	Adjustment Result
Task Configuration Stage	Source database configuration is too low to handle the increased load from DTS, resulting in the source database crashing after the DTS task starts.	Reduce the number of export threads for the source database to lower 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.	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.

		The default RPS is very large in the full volume stage, and generally, there's no need to 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	Reduce the number of threads for importing into the target database to minimize the lock competition.	Valid.

	database's own writes (e.g., lock wait timeout error)		

Speed Adjustment in Task Configuration

During the initial configuration stage of the sync task, in the Advanced Options on the **Set sync options and objects** page, you can set the transfer speed parameters.



Advanced Options

Migrate Account ⓘ *

Limit Transfer Speed

Concurrent Threads During Full Export ⓘ * 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

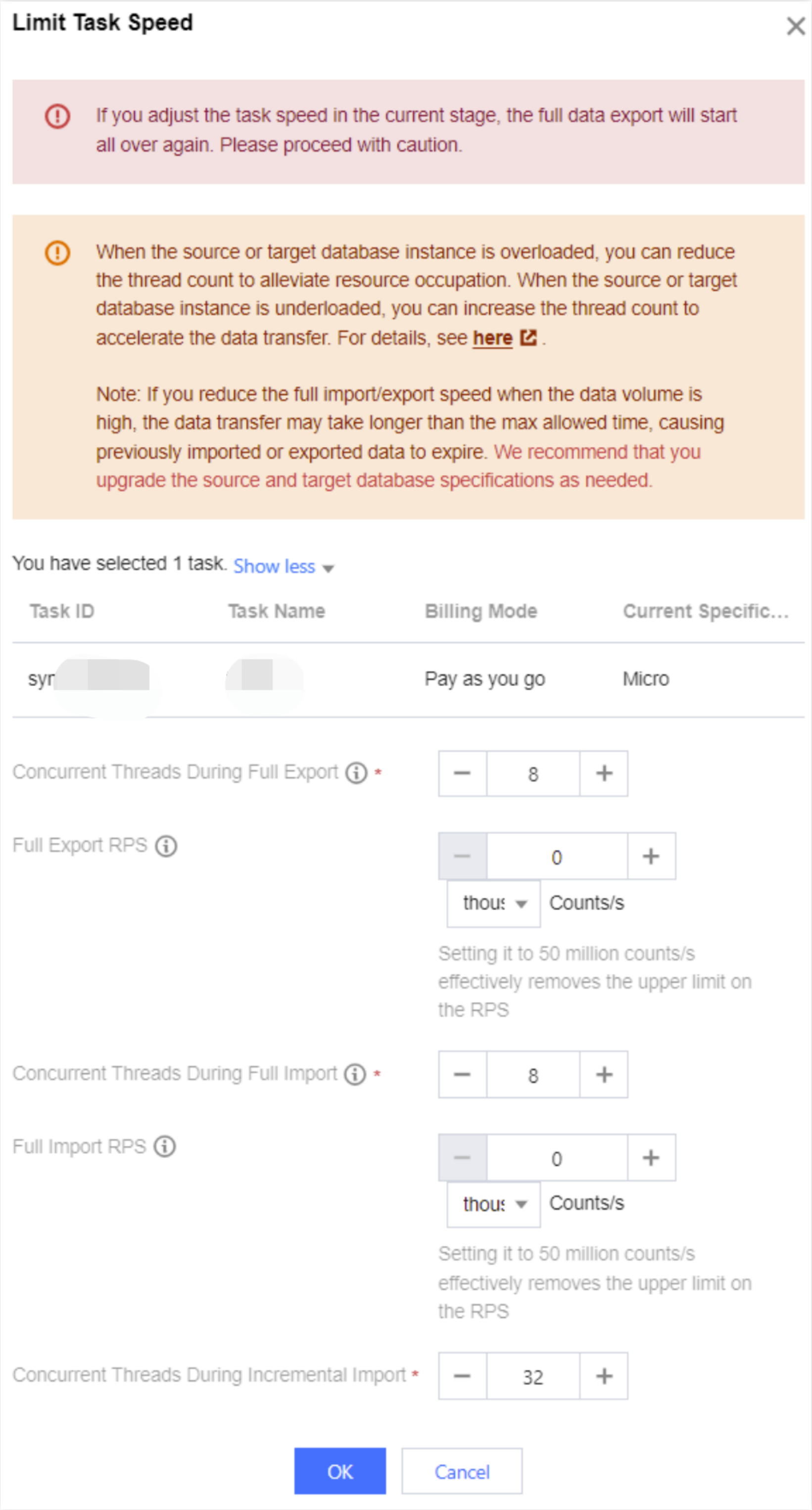
Speed Adjustment When the Task Is Running

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

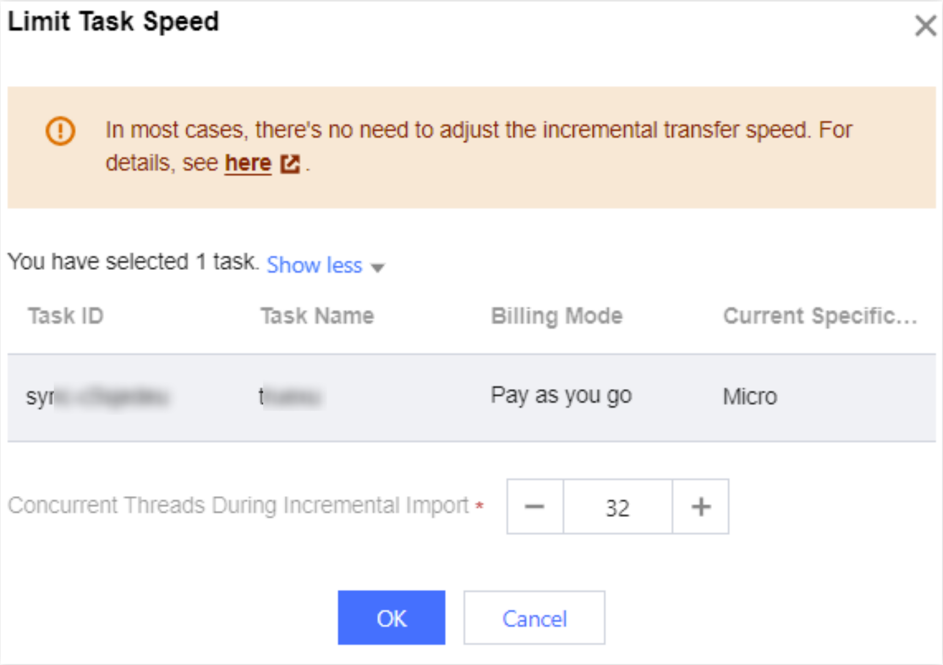
Full Import RPS: 0 thous Counts/s

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...
syr[illegible]	t[illegible]	Pay as you go	Micro

Concurrent Threads During Incremental Import * 32

Stopping Task

Last updated：2024-07-08 19:09:49

Overview

Data sync is a continuous task that will not stop on its own. You need to manually stop it at an appropriate time.

Note

For a pay-as-you-go task, the billing will stop after the task is stopped.

For a monthly subscribed task, fees will not be automatically refunded after the task is stopped. You need to click **Terminate**/**Return** before you can be refunded. For a task that can be automatically renewed, if it is stopped but not terminated, it will still be billed when each billing cycle starts.

Directions

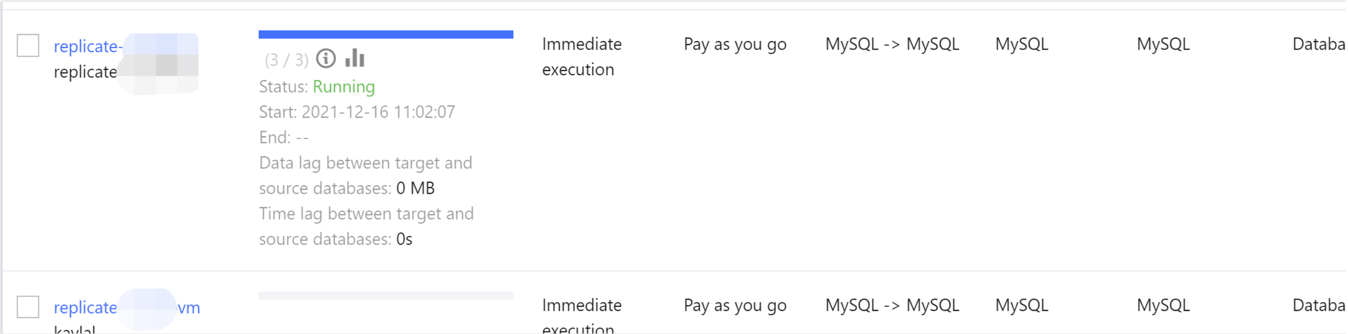
1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **More** > **Stop** in the **Operation** column.

Note

We recommend that you check the task status and ensure the following conditions are met before stopping a 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.



☐ replicate-[redacted] replicate[redacted]

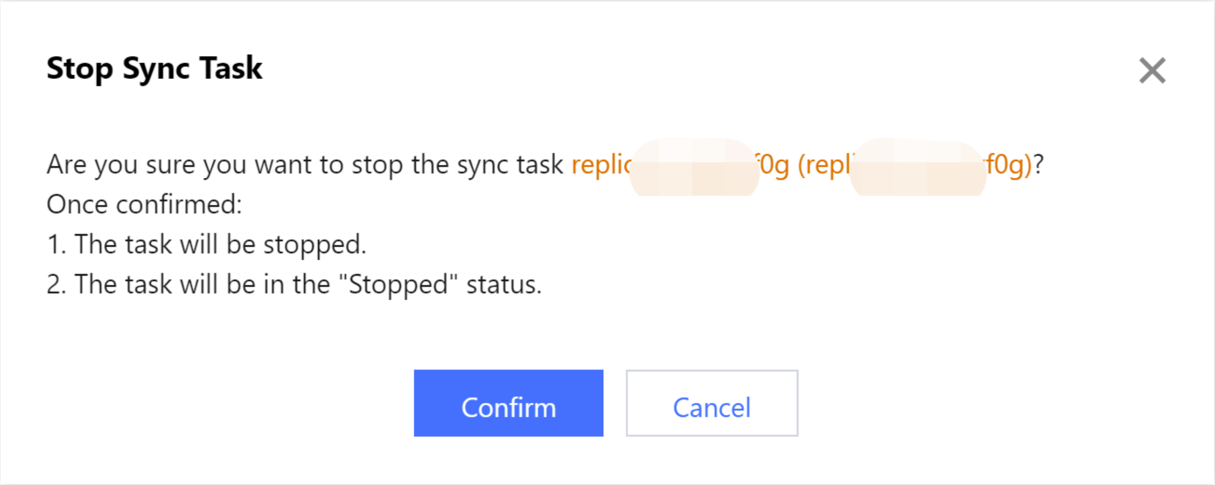
(3 / 3)
Status: Running
Start: 2021-12-16 11:02:07
End: --
Data lag between target and source databases: 0 MB
Time lag between target and source databases: 0s

Immediate execution | Pay as you go | MySQL -> MySQL | MySQL | MySQL | Databa

☐ replicate[redacted]vm

Immediate execution | Pay as you go | MySQL -> MySQL | MySQL | MySQL | Databa

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



Stop Sync Task

Are you sure you want to stop the sync task replic[redacted]0g (repl[redacted]f0g)?
Once confirmed:
1. The task will be stopped.
2. The task will be in the "Stopped" status.

Modify Sync configuration

Last updated：2024-07-08 19:09:49

Overview

After a newly configured sync task has started, you can modify the sync configuration to add or delete sync objects and modify sync parameters as follows.

When you modify the sync configuration, the existing sync task won't be paused or affected. Except data initialization options, the primary key conflict resolution policy, SQL sync policy, databases, tables, and advanced objects can all be modified.

Adding an object: When the sync task type is set to full + incremental sync, adding a new sync object will result in DTS syncing both the existing and incremental data of the added object to the target database. Conversely, when the sync task type is set to incremental sync only, adding a new sync object will result in DTS syncing only the incremental data of the added object to the target database.

Deleting an object: After the configuration is modified, the incremental data of the deleted object will no longer be synced to the target database, but the data already synced before the modification will be retained.

Modifying the primary key conflict resolution policy: The new policy will be applied only to the data synced after the configuration is modified.

Application Scope

Only the following links support the modification of synchronization configuration operations.

MySQL/MariaDB/Percona/TDSQL-C MySQL -> MySQL

MySQL/MariaDB/Percona -> MariaDB

MySQL/MariaDB/TDSQL-C MySQL -> TDSQL-C MySQL

MySQL/MariaDB/Percona/TDSQL-C MySQL -> Kafka

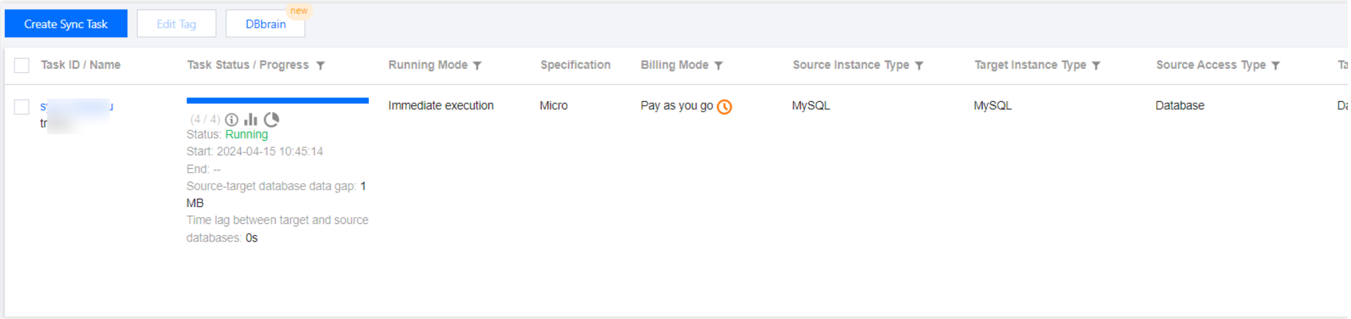
Notes

1. After the sync configuration modification feature is launched, you can modify only newly initiated tasks.
2. The configuration of a sync task can be modified only if the task is in the **Incremental Sync** stage and in **Running** status and the data gap between the source and target databases is less than 1 GB.
3. When you add or delete objects, be sure to select associated objects together; otherwise, the sync task will fail.

4. When modifying the sync configuration, you can rename only newly added database/table objects. If you modify previously selected objects, the system will report an error during verification, and you need to change them back to the original names.
5. Upon initial modification of the full-speed parameters within the speed limit settings (including full export thread data and RPS, as well as the full import thread count and RPS), subsequent modifications to the sync objects will first redirect users to the speed limit parameter adjustment interface. It is recommended that users increase the full-speed parameters as much as possible (based on a comprehensive assessment of the load conditions of the source and target databases) to prevent prolonged duration of full sync for added objects.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **Modify Sync Configuration** in the **Operation** column.
The configuration of a task can be modified only if the task is in the **Incremental Sync** stage and in **Running** status and the data gap between the source and target databases is less than 1 GB. It cannot be modified in other statuses such as **Task failed** or **Task error**.

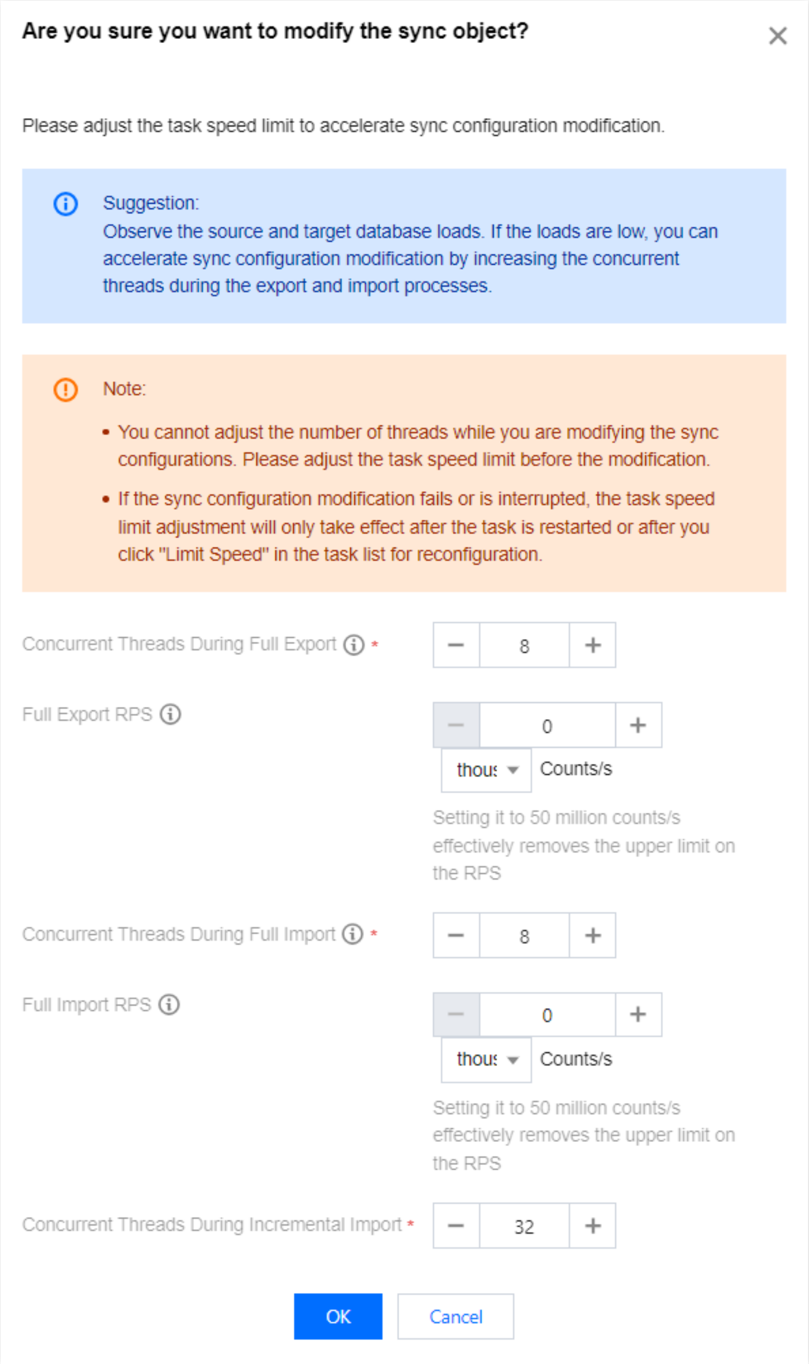


☐	Task ID / Name	Task Status / Progress	Running Mode	Specification	Billing Mode	Source Instance Type	Target Instance Type	Source Access Type	Ta
☐	s[redacted]u tr	(4 / 4) Status: Running Start: 2024-04-15 10:45:14 End: -- Source-target database data gap: 1 MB Time lag between target and source databases: 0s	Immediate execution	Micro	Pay as you go	MySQL	MySQL	Database	Da

2. (Optional) Verify if there is a need to adjust the speed parameters, and upon confirmation, click **OK**.

Note：

Upon initial modification of the full-speed parameters within the speed limit settings (including full export thread data and RPS, as well as the full import thread count and RPS), subsequent modifications to the sync objects will first redirect users to the speed limit parameter adjustment interface. It is recommended that users increase the full-speed parameters as much as possible (based on a comprehensive assessment of the load conditions of the source and target databases) to prevent prolonged duration of full sync for newly added objects.



Are you sure you want to modify the sync object?

Please adjust the task speed limit to accelerate sync configuration modification.

Suggestion:
Observe the source and target database loads. If the loads are low, you can accelerate sync configuration modification by increasing the concurrent threads during the export and import processes.

Note:

- You cannot adjust the number of threads while you are modifying the sync configurations. Please adjust the task speed limit before the modification.
- If the sync configuration modification fails or is interrupted, the task speed limit adjustment will only take effect after the task is restarted or after you click "Limit Speed" in the task list for reconfiguration.

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

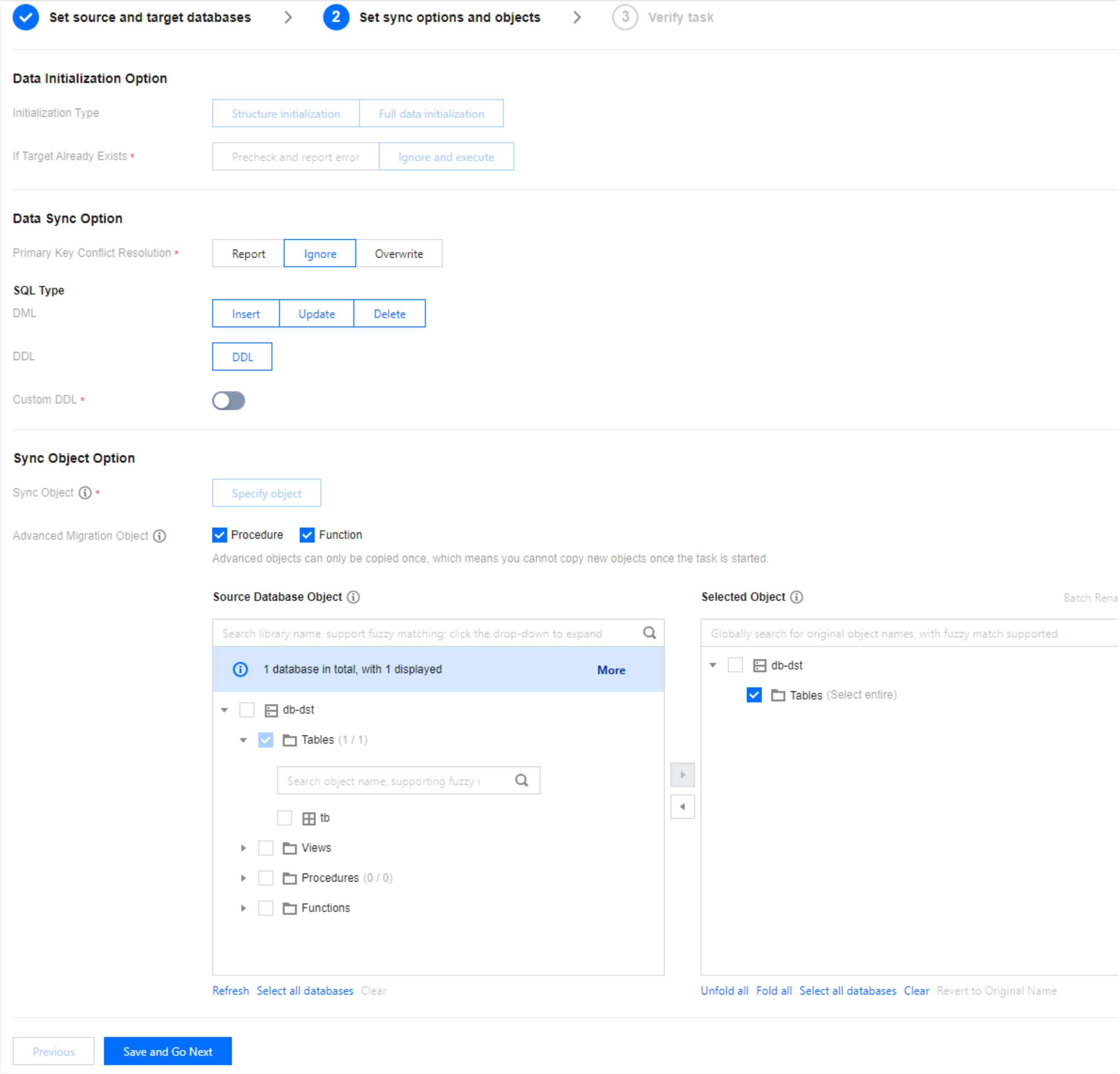
Concurrent Threads During Full Import * 8

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. On the sync task configuration page, in **Set sync options and objects**, modify relevant parameters such as objects and the primary key conflict resolution policy, and click **Save and Go Next**. Except data initialization options, all options can be modified.



Set source and target databases > 2 Set sync options and objects > 3 Verify task

Data Initialization Option

Initialization Type: Structure initialization | Full data initialization

If Target Already Exists *: Precheck and report error | Ignore and execute

Data Sync Option

Primary Key Conflict Resolution *: Report | Ignore | Overwrite

SQL Type

DML: Insert | Update | Delete

DDL: DDL

Custom DDL *

Sync Object Option

Sync Object *: Specify object

Advanced Migration Object: ☑ Procedure ☑ Function

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

Source Database Object

Search library name, support fuzzy matching; click the drop-down to expand

1 database in total, with 1 displayed More

- ☐ db-dst
  - ☑ Tables (1 / 1)
    - ☐ tb
  - ☐ Views
  - ☐ Procedures (0 / 0)
  - ☐ Functions

Refresh Select all databases Clear

Selected Object

Batch Rena

Globally search for original object names, with fuzzy match supported

- ☐ db-dst
  - ☑ Tables (Select entire)

Unfold all Fold all Select all databases Clear Revert to Original Name

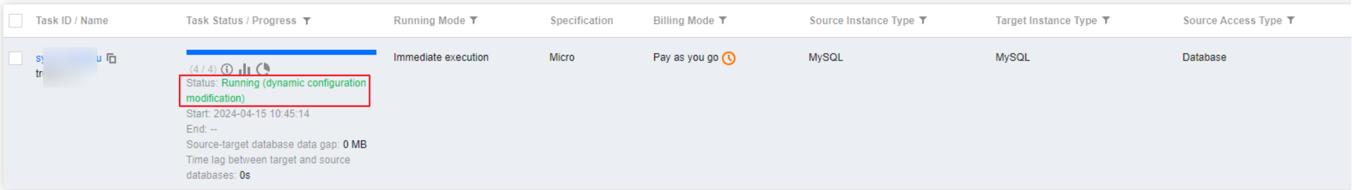
4. Verify the task again. After the verification is passed, click **Start Task**.

5. Return to the task list, and you can see that the task status is **Running (dynamically modifying configuration)**, meaning that DTS is processing the data after modification.

Notes:

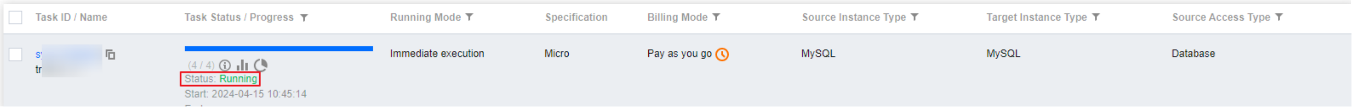
After the sync configuration is modified, when the task is in **Running (dynamically modifying configuration)** status, do not perform DDL operations on the source database; otherwise, the task may fail.

When you add a sync object, DTS will first sync the existing data of the object to the target database. If the volume of the existing data is high, the task will remain in **Running (dynamically modifying configuration)** status for a long time.



☐	Task ID / Name	Task Status / Progress	Running Mode	Specification	Billing Mode	Source Instance Type	Target Instance Type	Source Access Type
☐	sy[redacted]u tr	(4 / 4) Status: Running (dynamic configuration modification) Start: 2024-04-15 10:45:14 End: -- Source-target database data gap: 0 MB Time lag between target and source databases: 0s	Immediate execution	Micro	Pay as you go	MySQL	MySQL	Database

When the task status becomes **Running**, the configuration has been modified, and you can modify the configuration again if needed.



☐	Task ID / Name	Task Status / Progress	Running Mode	Specification	Billing Mode	Source Instance Type	Target Instance Type	Source Access Type
☐	s[redacted] tr	(4 / 4) Status: Running Start: 2024-04-15 10:45:14	Immediate execution	Micro	Pay as you go	MySQL	MySQL	Database

Pausing Task

Last updated：2024-07-08 19:09:49

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 sync links among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Note

Only sync tasks in **Running** 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.

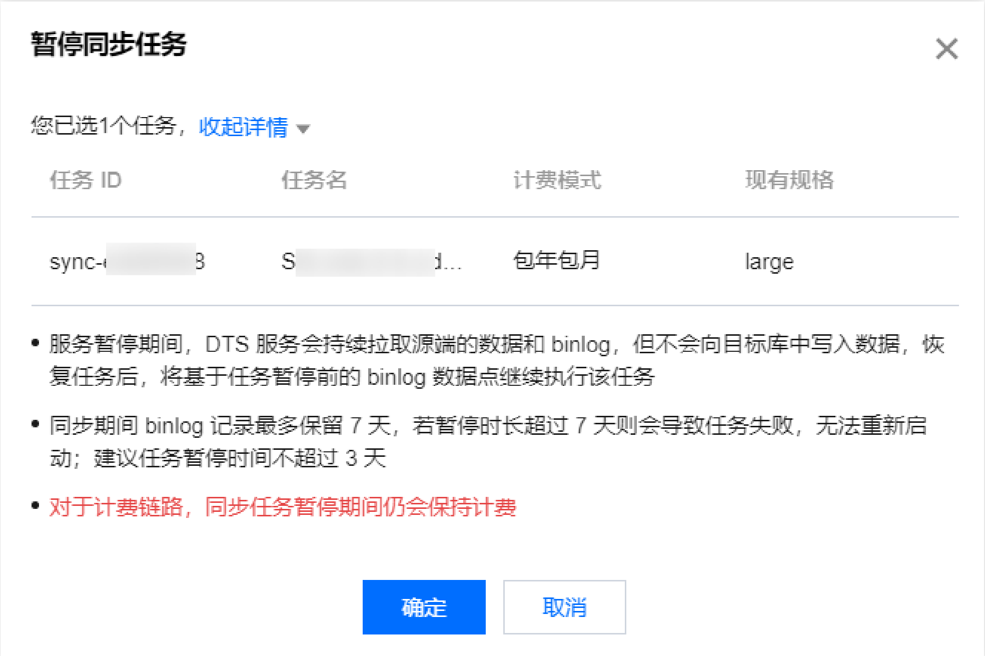
In the full import and incremental sync stages, after a task is resumed, DTS will continue with the data transfer based on the offset of the binlog synced before the task is paused.

During sync, 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 of the source database during the task pause, billable links will still be billed when the sync task is paused.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **Pause** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.



暂停同步任务

您已选1个任务，收起详情 ▾

任务 ID	任务名	计费模式	现有规格
sync-[redacted]	S[redacted]…	包年包月	large

- 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
- 同步期间 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，同步任务暂停期间仍会保持计费

Resuming Task

Last updated：2024-07-08 19:09:49

Overview

This document describes how to resume a paused sync task.

Application Scope

Data sync links among MySQL, TDSQL-C for MySQL, MariaDB, and Percona.

Note

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 sync, 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 sync task is paused.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **Resume** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.

Resetting Task

Last updated：2024-07-08 19:09:49

Overview

If your monthly subscribed sync task stops and you want to start a new one, you can clear the old task configuration data and configure the new task data through the reset operation. In this way, the same task ID can be used for billing, eliminating your need to return the old task and purchase a new one.

After the reset, the task ID will remain unchanged, and the task name will be the same as the task ID. The old task will be reset to **Uninitialized** status, with its data being cleared permanently.

Note

Only tasks in **Stopped** status can be reset.

Make sure that the old task is stopped or no longer needed before resetting it.

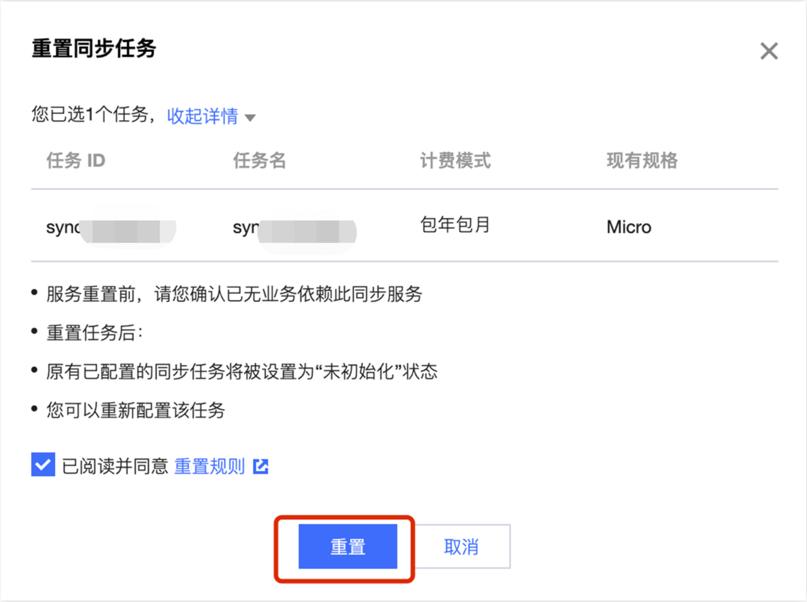
Application Scope

The reset feature is available to monthly subscribed sync tasks among MySQL, MariaDB, Percona, and TDSQL-C for MySQL.

The task reset feature was released in December 2022 and is only supported for new sync tasks created after that. If you need to reset the tasks created before, submit a ticket to upgrade them.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task instance, and click **Reset** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Reset**.



重置同步任务

您已选1个任务，收起详情

任务 ID	任务名	计费模式	现有规格
sync[redacted]	syn[redacted]	包年包月	Micro

- 服务重置前，请您确认已无业务依赖此同步服务
- 重置任务后：
- 原有已配置的同步任务将被设置为“未初始化”状态
- 您可以重新配置该任务

☑ 已阅读并同意 重置规则

Deleting Task

Last updated：2024-07-08 19:09:49

Overview

You can delete the instance tasks that have stopped or failed if they are no longer needed. Once the tasks are deleted, the relevant resources will also be cleared.

Application Scope

This operation is available for data sync tasks that haven’t been billed.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Delete** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Delete**.

Changing Configuration

Last updated：2024-07-08 19:09:49

Overview

When your business requirements change, you can modify the selected sync link specification through the operation below.

For monthly subscribed tasks, the link configuration fees incurred after the configuration change in that month will be calculated based on the new price. You may be refunded or charged extra based on the price difference. The previous billing cycle remains unchanged.

Note

The configuration change takes 3-5 minutes to take effect, during which the task will be paused.

Application Scope

This operation is available for data sync tasks that have been billed.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Adjust Specification** in the **Operation** column.
2. In the pop-up window, select the specification to modify and click **OK**.

Pay-as-You-Go to Monthly Subscription

Last updated：2024-07-08 19:32:57

Overview

DTS allows you to change pay-as-you-go tasks to monthly subscribed ones. After you perform the billing mode change operation, DTS will generate a monthly subscription renewal order, and you need to pay for the order for the change to take effect.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **More** > **Postpaid-to-Prepaid** in the **Operation** column.
2. In the pop-up window, select the renewal period and configure auto-renewal, confirm that everything is correct, and click **OK**.

Terminating/Returning Instance

Last updated：2024-07-08 19:09:49

Overview

Terminate: You can terminate the instance tasks that have stopped or failed if they are no longer needed.
Return: You can return a monthly subscribed task if it is no longer needed. For refund description, see Refund.
A terminated or returned task will be automatically deleted after seven days of isolation. Proceed with caution.
If you accidentally terminate a pay-as-you-go task, you can resume it as instructed in Deisolating Task.
You can resume a monthly subscribed task that has been terminated or returned by renewing it.
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, a pay-as-you-go sync task is purchased but not started.
We recommend that you terminate completed or unwanted tasks promptly to unfreeze fees in time.

Application Scope

This operation is available for data sync tasks that have been billed.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Terminate**/**Return** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct, select **I have read and agree to Termination Rules**, and click **Terminate Now**.

Renewing Task

Last updated：2024-07-08 19:09:49

Overview

Monthly subscribed data sync tasks can be renewed by month. You can resume a task that is isolated due to overdue payments by renewing it.

Generally, you can renew a sync task in the DTS console or billing center. To renew multiple tasks, go to the billing center.

Application Scope

This operation is available for data sync tasks that have been billed.

Renewal in the DTS Console

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Renew** in the **Operation** column.
2. In the pop-up window, select the renewal period and click **OK**.

Renewal in the Billing Center

1. In the top right corner of the DTS console, select **Fees** > **Billing Center**.
2. Select **Renewal Management** in the left sidebar and select a time range and the service you want to renew.

To renew a single task, select the target task and click **Renew** in the **Operation** column.

To renew multiple tasks, select the target tasks and click **Batch Renewal**.

Deisolating Task

Last updated：2024-07-08 19:09:49

Overview

This document describes how to resume an accidentally terminated pay-as-you-go sync task.

Application Scope

This operation is available for pay-as-you-go data sync tasks.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Deisolate**/**Resume** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Resume**.

Deleting Instance

Last updated：2024-07-08 19:09:49

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 sync tasks that have been billed.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target task, and click **More** > **Delete Instantly** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **Delete Instantly**.

Creating a Similar Task

Last updated：2024-07-08 19:09:49

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, Conflict Strategy, and DDL Sync Policy, will all be consistent with those of the historical tasks. Users can also modify these options as needed.

In scenarios such as task failure, many-to-one sync, or one-to-many sync, a similar task can be created quickly to configure a new task.

Directions

1. Log in to the DTS console, choose **Data Sync** on the left sidebar, select the specified task, 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 19:09:49

Overview

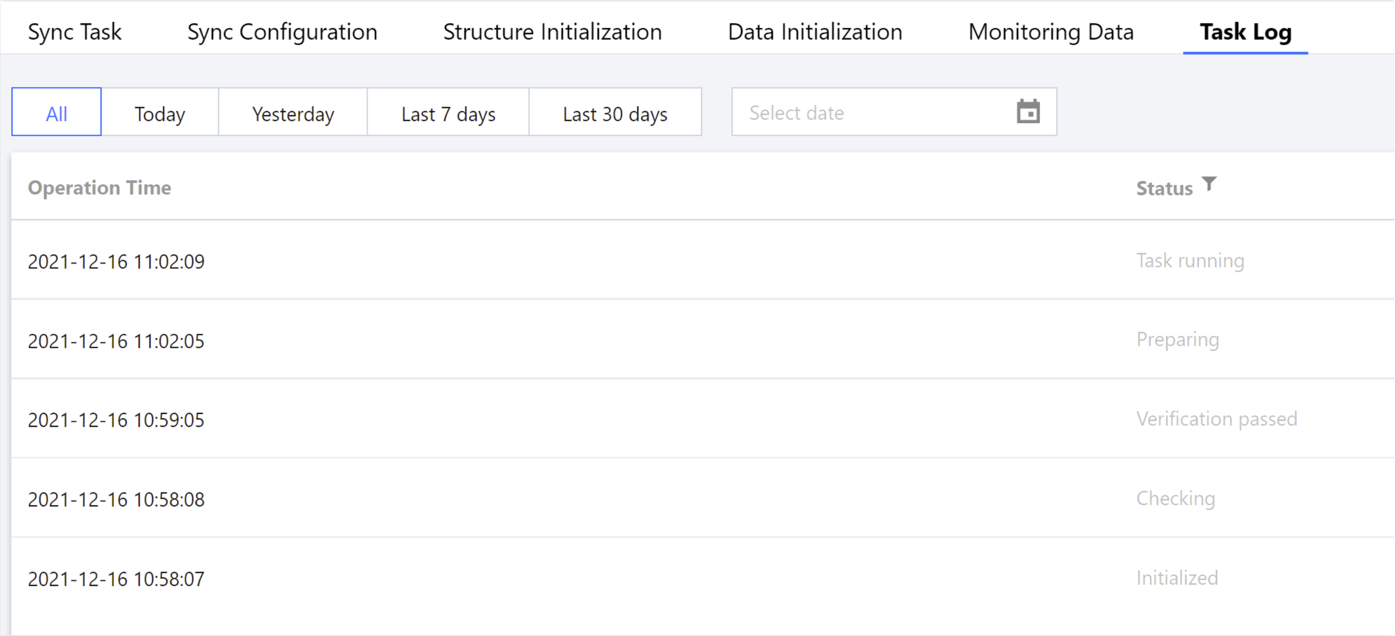
During a sync task, you can view the sync task logs to know the task progress.

Directions

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

- Method 1: on the **Data Sync** page, select the target sync task and click the task name.
- Method 2: on the **Data Sync** page, select the target sync task and click **More** > **View** in the **Operation** column.

2. View the sync logs.
   Switch the tab to view the task logs.



Sync Task
Sync Configuration
Structure Initialization
Data Initialization
Monitoring Data
Task Log
All
Today
Yesterday
Last 7 days
Last 30 days
Select date

Operation Time	Status
2021-12-16 11:02:09	Task running
2021-12-16 11:02:05	Preparing
2021-12-16 10:59:05	Verification passed
2021-12-16 10:58:08	Checking
2021-12-16 10:58:07	Initialized

Task Status Description

Status	Description

Status	Description
Uninitialized	The purchase has been completed, but no sync task has been configured.
Initialized	The sync task has been configured.
Checking	The sync task is being checked.
Verification passed	The sync task passed the verification.
Verification failed	The sync task failed the verification.
Preparing	The sync task is ready to start.
Task running	The sync task is running.
Retryable error occurred	The sync task was interrupted during sync due to an exception. You can retry and resume the task in the console.
Stopping	The sync task is being manually stopped during execution.
Completed	The task is stopped.
Task failed	The sync task failed.
Deleting	The task is being deleted. You can manually delete a task that is completed, failed, or no longer needed.
Deleted	The task is deleted. Once deleted, it will no longer exist or occupy any resources.

Deleting Task

Last updated：2024-07-08 19:09:49

Overview

You can delete a task that is completed, no longer needed, or failed. Once deleted, the task will no longer exist or occupy any resources. Therefore, you should proceed with caution.

Prerequisites

The sync task is completed, no longer needed, or failed.

Directions

1. Log in to the DTS console, select **Data Sync** on the left sidebar, select the target sync task, and click **More** > **Delete** in the **Operation** column.
2. In the pop-up window, confirm that everything is correct and click **OK**.