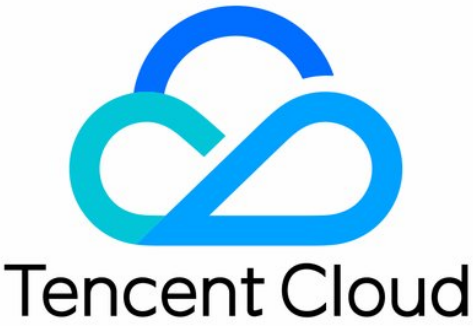
Data Transmission Service

Operation Guide

Product Documentation



Contents

Operation Guide
MySQL Data Migration
Online Import of MySQL Data

Last updated : 2020-12-01 16:12:12

Operation Scenarios

DTS provides data migration and continuous data replication from self-created MySQL databases to TencentDB, allowing you to migrate hot data without interrupting your business. MySQL with public IP/port, Direct Connect-based MySQL in local IDC, and CVM-based self-created MySQL can be migrated. Currently, DTS is supported for MySQL 5.7.

Preparations

Precautions

- A DTS data migration task involves two steps: cold backup data export and incremental data sync. As cold backup data export and data comparison after migration have certain impact on the load of the source database, you are recommended to perform database migration during off-hours or in the replica database.
- Migration of specified table
  If `lower_case_table_name` is specified for migration, a migration verification task will be performed to check whether the source and target databases have the same configuration; and if not, an error will be displayed, helping you prevent any restart issues caused by `lower_case_table_name`
- Migration of entire database
  - Before migrating configurations, if the source database has some parameters requiring database restart for them to take effect (e.g., `lower_case_table_name`), and their values are different from those in the target database, you need to configure restart of the target database.
  - After the full backup is imported, you need to restart the target database.
- SUPER privilege for the source database is required in some scenarios.

SUPER privilege for source database

SUPER privilege for the source database is not required in most scenarios and required only in the following scenarios:

- You have selected "Full Detection" as "Data Consistency Detection".

- During binlog sync, if you create an event in the source database, and an account that is not used for DTS data migration is specified as the `DEFINER` for this event, an error will occur if the SUPER privilege is unavailable.

Migratable databases

- CVM-based self-created MySQL databases in the classic network or VPCs can be migrated to TencentDB.
- MySQL databases with public IP/port can be migrated to TencentDB.
- Direct Connect-based or VPN-based MySQL databases can be migrated to TencentDB.

Precheck items

1. To avoid conflict, check whether the target database has a table with the same name as a table in the source database.
2. Check the database version. Data migration is supported for MySQL 5.1, 5.5, 5.6, and 5.7. As TencentDB no longer supports MySQL 5.1, you are recommended to upgrade MySQL 5.1 to MySQL 5.5 first and then migrate data to TencentDB for MySQL 5.5. You can also use DTS to directly migrate from local MySQL 5.1 to TencentDB for MySQL 5.5.
3. Check whether the capacity of the target database is greater than that of the source database.
4. Create a migration account on the source MySQL database (this is not required if you already have an authorized account for data migration).

```
GRANT ALL PRIVILEGES ON *.* TO "migration account"@"%" IDENTIFIED BY "migration password";
FLUSH PRIVILEGES;
```

5. Confirm the source database's MySQL variables.
   Run `SHOW GLOBAL VARIABLES LIKE 'XXX';` to view the global MySQL variables so as to check whether the database in the current state is suitable for migration:

```
server_id > 1
log_bin = ON;
binlog_format = ROW/MIXED
binlog_row_image = FULL
innodb_stats_on_metadata = 0
You are recommended to set `wait_timeout` to above or equal to 3,600 seconds and mandatorily belo
w 7,200 seconds.
Set `interactive_timeout` and `wait_timeout` to the same value.
If the source database is replica, confirm the following parameters in it:
log_slave_updates = 1
```

6. Modify the variables.
   a. For a self-created MySQL database, modify the source database's MySQL configuration file `my.cnf` and restart the database:

```
log-bin=[custom binlog filename]
```

b. Modify the configuration dynamically:

```
set global server_id = 99;
set global binlog_format=ROW;
set global binlog_row_image=FULL;
set global innodb_stats_on_metadata = 0;
```

Directions

1. Create a migration task

Log in to the DTS Console, enter the data migration page, and click **Create Migration Task**.

2. Set the source and target databases

Configure the task, source database, and target database. After the network connectivity test is successful, click **Create**.

a. Task settings

- Task Name: name the task.
- Scheduled Execution: specify the start time of the migration task.
- Tag: categorize and manage resources by various metrics.

b. Source database settings

Source Database Type: MySQL with public IP, CVM-based self-created MySQL, Direct Connect-based MySQL, and VPN-based MySQL are supported.

Source Database Type	Description
MySQL with public IP	It refers to a MySQL database that can be accessed through a public IP. The following information is required: • MySQL server address • MySQL port • MySQL account • MySQL password
CVM-based self-	You can migrate a MySQL database deployed on a CVM instance in the classic network or VPC by specifying the CVM instance ID. The following information is required:

created MySQL	• Region: all CVM-based self-created MySQL databases can be migrated to TencentDB for MySQL over the private network • CVM instance ID • MySQL port • MySQL account • MySQL password
Direct Connect-based MySQL	After connecting a self-created MySQL database in your local IDC to Tencent Cloud through Direct Connect, you can use DTS to migrate data to Tencent Cloud. The following information is required: • Direct Connect gateway: it is the Direct Connect gateway used by the database server to connect to Tencent Cloud. For more information, please see Direct Connect Gateway • VPC: it is the VPC where the Direct Connect gateway resides • MySQL server address: it is the address of your MySQL server in the IDC. The IP mapped by the Direct Connect gateway will be accessed during DTS data migration • MySQL port • MySQL account • MySQL password
VPN-based MySQL	After connecting a self-created MySQL database in your local IDC to Tencent Cloud through VPN Connections or a CVM-based self-created VPN service, you can use DTS to migrate data to Tencent Cloud. The following information is required: • Region: currently, only VPN services in the same region as that of the target TencentDB instance are supported • VPN type: you can select VPN Connections or CVM-based self-created VPN • VPN gateway: you need to provide the VPN gateway information only if VPN Connections is selected. For more information, please see VPN • VPC: it is the VPC where the VPN Connections instance resides • MySQL server address: it is the address of your MySQL server in the IDC. The IP mapped by the Direct Connect gateway will be accessed during DTS data migration • MySQL port • MySQL account • MySQL password

c. Target database settings

Select the target database instance and enter its account and password.

3. Set migration options and select migration objects

Select the database to be migrated (you can choose to migrate the entire database or only certain tables), create a migration task, and check the task information.

⚠ Note :

1. The `character_set_server` and `lower_case_table_names` configuration items will be migrated only if the entire database is migrated.
2. If the character set configuration of the migrated tables in the source database is different from that of the target database, the former will be retained.

- **Migration Type**: structure migration, full migration, and full + incremental migration are supported.
- **Migration Object**: you can migrate the whole instance or specific objects.
- **Overwrite Target Database with Source Database Root Account**: the root account is used for TencentDB security verification. If the source database root account does not exist, issues may occur when you use TencentDB after migration. Therefore, to migrate the whole instance, you need to specify whether the target database root account should be overwritten by the source database root account. If you want to use the source database root account or the target database does not have any root account, choose **Yes**; if you want to keep the target database root account, choose **No**.
- **Read-only Target Database**: if the read-only option is enabled, data migrated to the target database will remain read-only during the migration process and cannot be modified until you complete the migration.
- **Data Consistency Detection**: you can perform a full detection or do not perform any detection.

4. Verify the migration task

After the migration task is created, you need to click **Next step: verify task** to verify the task information. You cannot start the migration task until all the check items are passed. Then, click **Start Task**.
There are 3 statuses for task verification:

- Passed: the verification is fully successful.
- Warning: the verification failed. Database operation may be affected during or after data migration, but the migration task can still be executed.
- Failed: the verification failed, and the migration task could not be executed. In this case, please check and modify the migration task information according to the error and then verify the task again. For the cause of failure, click **View Details** to view the "Verification Details".

5. Start migration

After the verification is passed, you can click **Immediate start** in the migration task list to start data migration. Please note that if you have configured scheduled execution, the migration task will begin queuing and be executed at the specified time; otherwise, it will be executed immediately.
After the migration is started, you can view the corresponding migration progress under the migration task. The subsequent steps required for migration and the current stage will be displayed if you hover over the exclamation mark after the current step.

> ⚠ **Note :**
>
> Due to system design limitations, multiple migration tasks committed or queued at the same time will be performed in sequence by queuing time.

6. Perform incremental sync

When a migration task is created, the incremental sync option is selected by default. When data migration is completed, the target database will be set as the replica of the source database, and new data in the source database during migration will be synced to the target database through source/replica sync. In this way, any modification of the source database will be synced to the target database.

> ⚠ **Note :**
>
> Before closing the sync, do not write data into the target database; otherwise, data comparison may fail due to inconsistency between the source and target databases, resulting in migration failure.

7. Cancel migration (optional)

> ⚠ **Note :**
>
> 1. Data that has been synced to the target database will not be cleared if you click **Cancel**.
> 2. Restarting the task may cause the verification or task to fail. You may have to manually clear all databases or tables that may cause conflicts in the target database before you can start the migration task again.
> 3. When migrating a single table, make sure that all tables depended on by its foreign keys are also migrated.

To cancel an ongoing migration task, click **Cancel**.

8. Complete migration

> ⚠ **Note :**
>
> If the migration is in "not completed" status, the migration task will continue, so will data sync.

After the migration progress is 100%, the data lag between target and source databases is 0 MB, and the time lag between them is 0 seconds, you can click **Complete** on the right to complete the migration task.

Offline Migration of MySQL Data

Last updated : 2021-01-06 10:04:08

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

Data Migration Through Console

There are two modes for migrating data through the console: physical backup and logical backup. For more information, please see:

- Restoring Database from Physical Backup
- Restoring Database from Logical Backup

Data Migration Through Command Line Tool

1. Generate the SQL file to import with the MySQL command line tool "mysqldump" in the following way:

   ⚠ Note :

   - The data files exported through mysqldump must be compatible with the SQL specification of your purchased TencentDB for MySQL version. You can log in to the database and get the MySQL version information by running the `select version();` command. The name of the generated SQL file can contain letters, digits, and underscores but not "test".
   - Make sure that the same source and target database version, source and target database character set, and mysqldump tool version are used. You can specify the character set using the parameter `--default-character-set` .

   ```
   shell > mysqldump [options] db_name [tbl_name ...] > bak_pathname
   ```

   Here, `options` is the export option, `db_name` is the database name, `tbl_name` is the table name, and `bak_pathname` is the export path.
   For more information on how to export data with mysqldump, see mysqldump - A Database Backup Program.
2. Import data to the target database with the MySQL command line tool as follows:

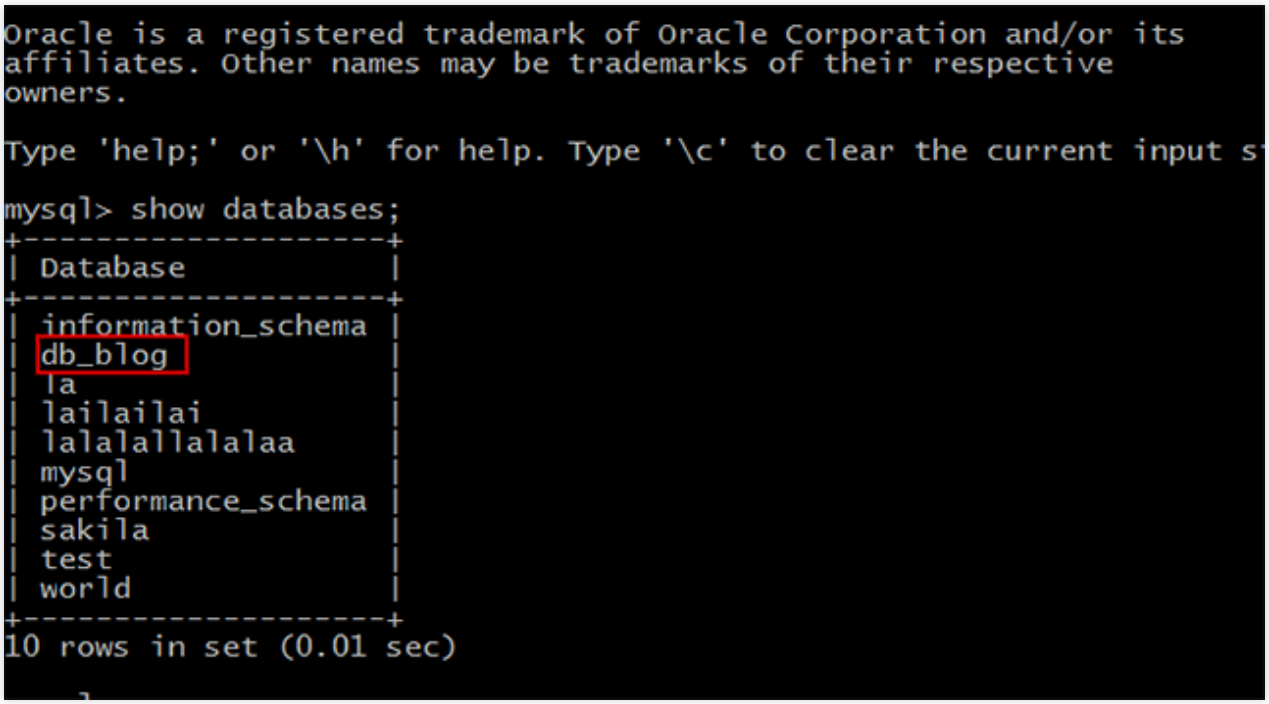
   ```
   shell > mysql -h hostname -P port -u username -p < bak_pathname
   ```

Here, `hostname` is the target server for data restoration, `port` is the port of target server, `username` is the username of the database on the target server, and `bak_pathname` is the full path to the backup file.

Migrating data (Windows)

1. Use the Windows version of mysqldump to generate the SQL file to import. For more information, please see the description in Data Migration Through Command Line Tool.
2. Enter the command prompt and import the data into the target database with the MySQL command line tool.

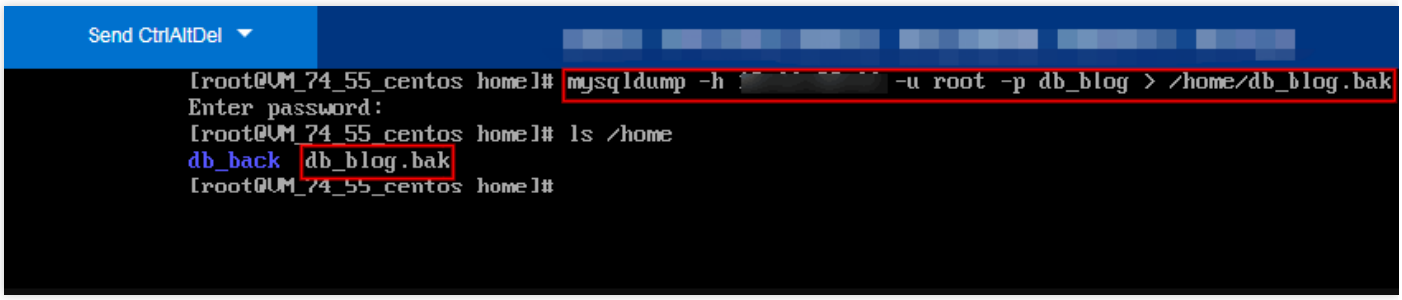
3. Log in to the target MySQL database, run the `show databases;` command, and you can see that the backup database has been imported into the target database.



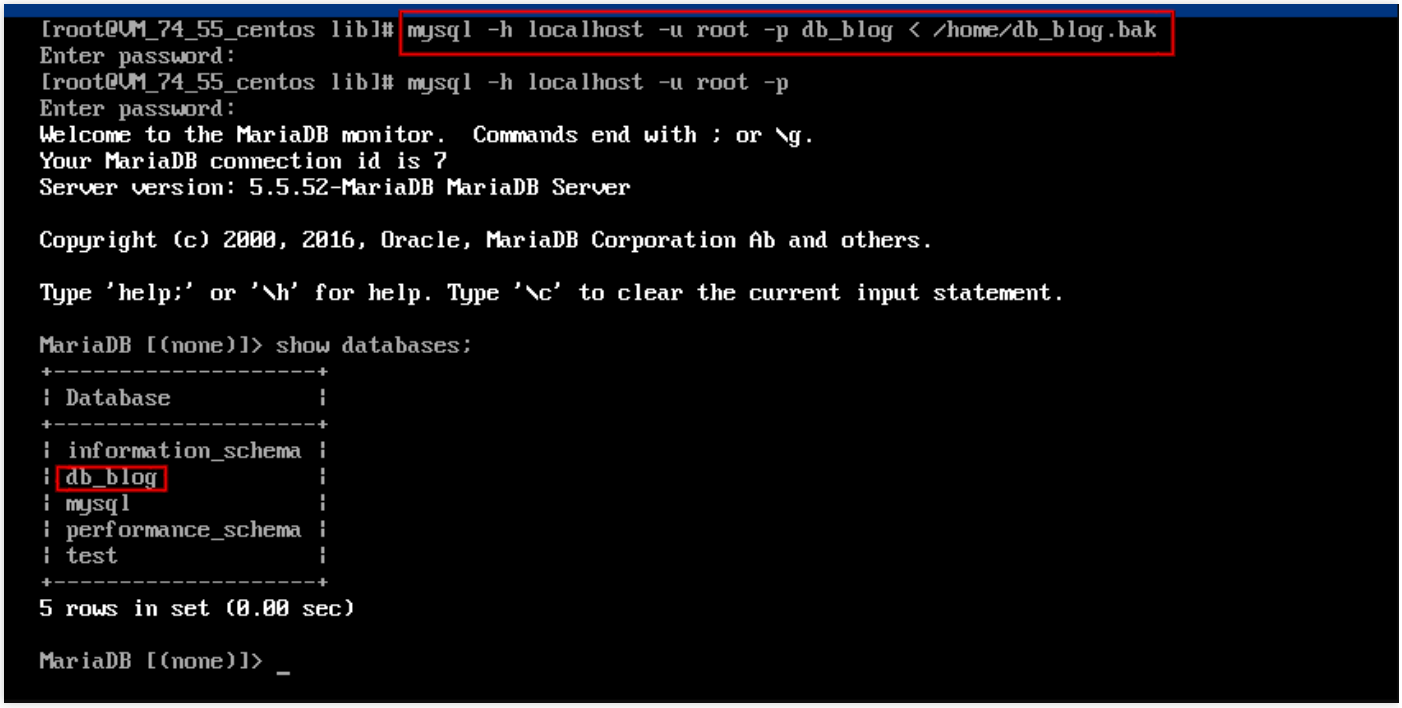
Migrating data (Linux)

This document takes Linux CVM instance as an example. For more information on how to access a database on a CVM instance, see Accessing a MySQL Database.

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



2. Use the MySQL command line tool to restore the data to the target database.
3. Log in to the target MySQL database, run the `show databases;` command, and you can see that the backup database has been imported into the target database.



Issues with Character Set Encoding of Imported Data Files

1. If no character set encoding is specified during data file import into TencentDB, the one set by the database will be used.
2. Otherwise, the specified character set will be used.
3. If the specified character set encoding is different from that of TencentDB, garbled text will be displayed.

For more information on character set encoding, please see "Notes on Character Set" in Use Limits.

Migrating from Alibaba Cloud ApsaraDB for RDS to TencentDB

Last updated : 2021-02-09 18:30:11

This document describes how to migrate data from Alibaba Cloud ApsaraDB for RDS to TencentDB using DTS, Tencent Cloud's data migration tool.

Environment Requirements

ApsaraDB RDS for MySQL 5.6 or 5.7.

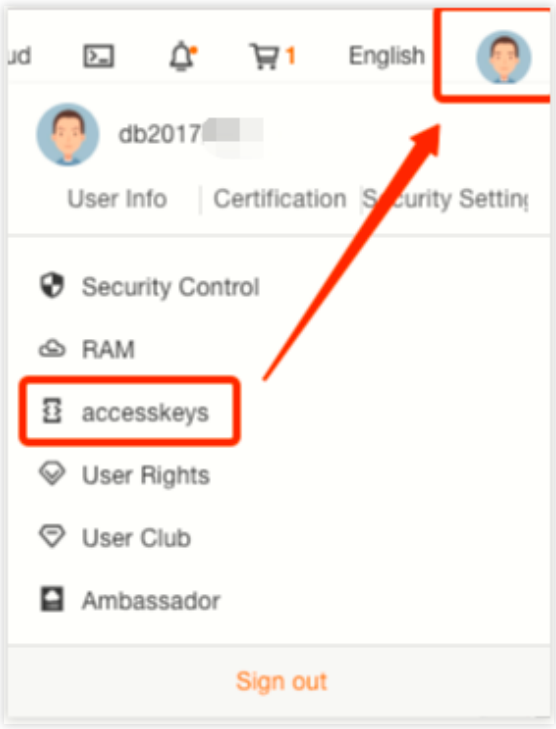
> ⚠ **Note :**
>
> During data transfer, the data replication method of TencentDB must be async replication, which can only be changed after data transfer is completed.

Obtaining the Basic Information and AccessKey of the Source Database

1. Log in to the RDS console and select a target instance.
2. You can obtain the necessary information on the target instance's basic information page as shown below:

> ⚠ **Note :**
>
> The public network address provided by Alibaba Cloud needs to be converted into IP format. You can query IP/server addresses here.

3. Hover over the profile photo in the top-right corner and select **accesskeys** in the drop-down menu to get the required Accesskey.



Creating a DTS Task for TencentDB

1. Creating a migration task

1) Log in to the DTS console and click **Create Migration Task** on the data migration page.
2) Select the corresponding region in **Linkage Region** and click **Buy at 0 USD**.

2. Configuring the migration task

Configure the task, source database, and target database. After the network connectivity test is successful, click **Create**.

a. Task settings

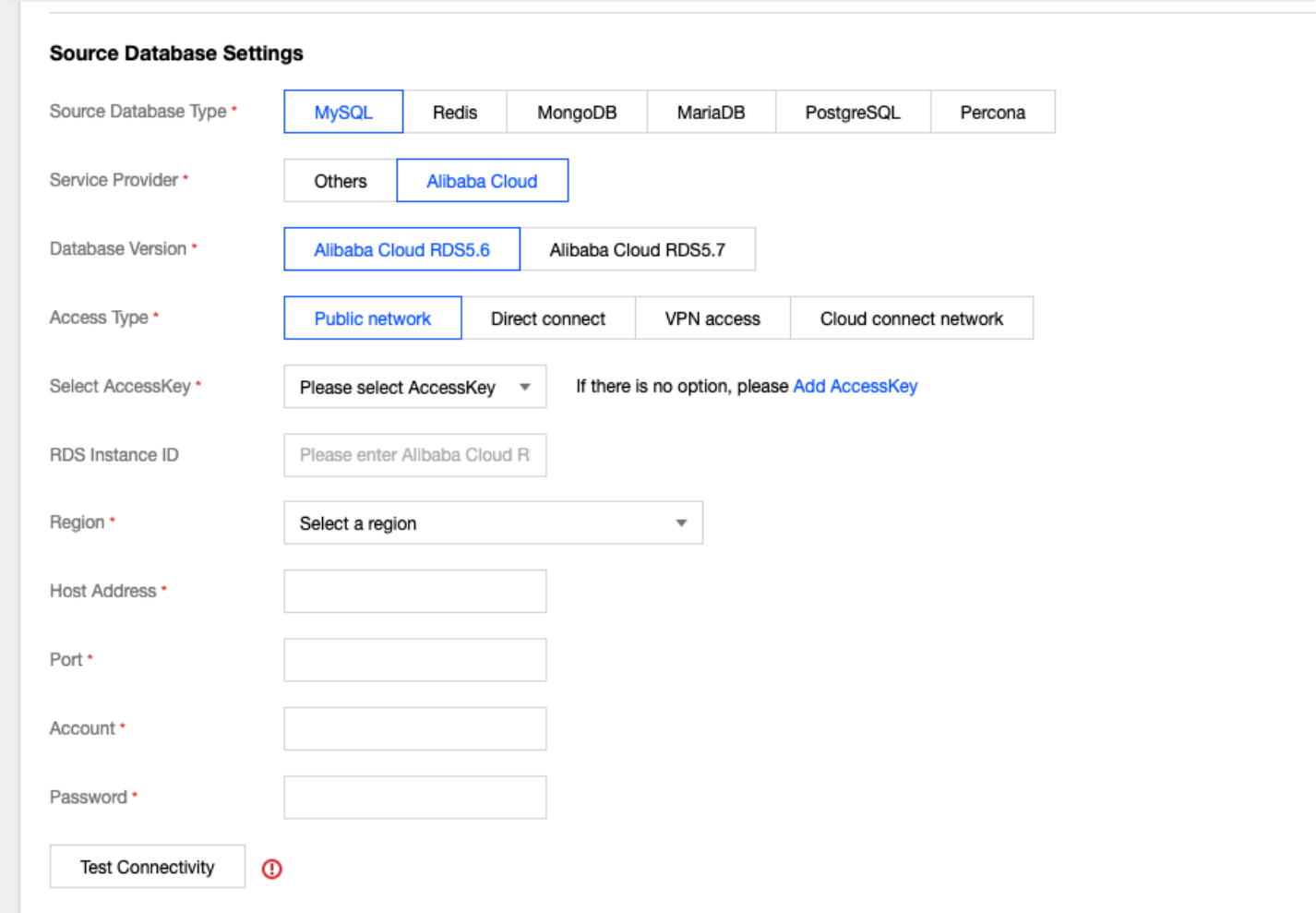
- Task name: specify a name for the task.
- Scheduled Execution: specify the start time of the migration task.

b. Source database information

Select a connection type as needed, enter the connection information of the source database, and click **Test Connectivity**.

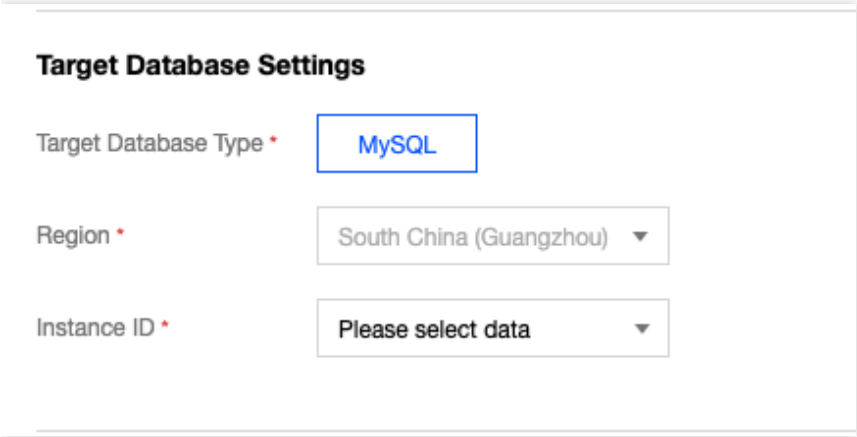
⚠ Note :

- You need to add the IP of your TencentDB instance to Alibaba Cloud's allowlist for IP mapping; otherwise, the connectivity test would fail.
  - For mapping a TencentDB for MySQL instance with a public IP, you need to add the public IP of the corresponding region to Alibaba Cloud's allowlist.
  - If the source database type is configured as "Direct Connect" or "VPN" during DTS configuration, an IP for external mapping will appear after the task is generated. You need to add it to Alibaba Cloud's allowlist.
- You can submit a ticket to query the DTS IP allowlist of corresponding region, or click **Test Connectivity**, and the DTS IP allowlist will be displayed in the detection result when the connectivity test fails.



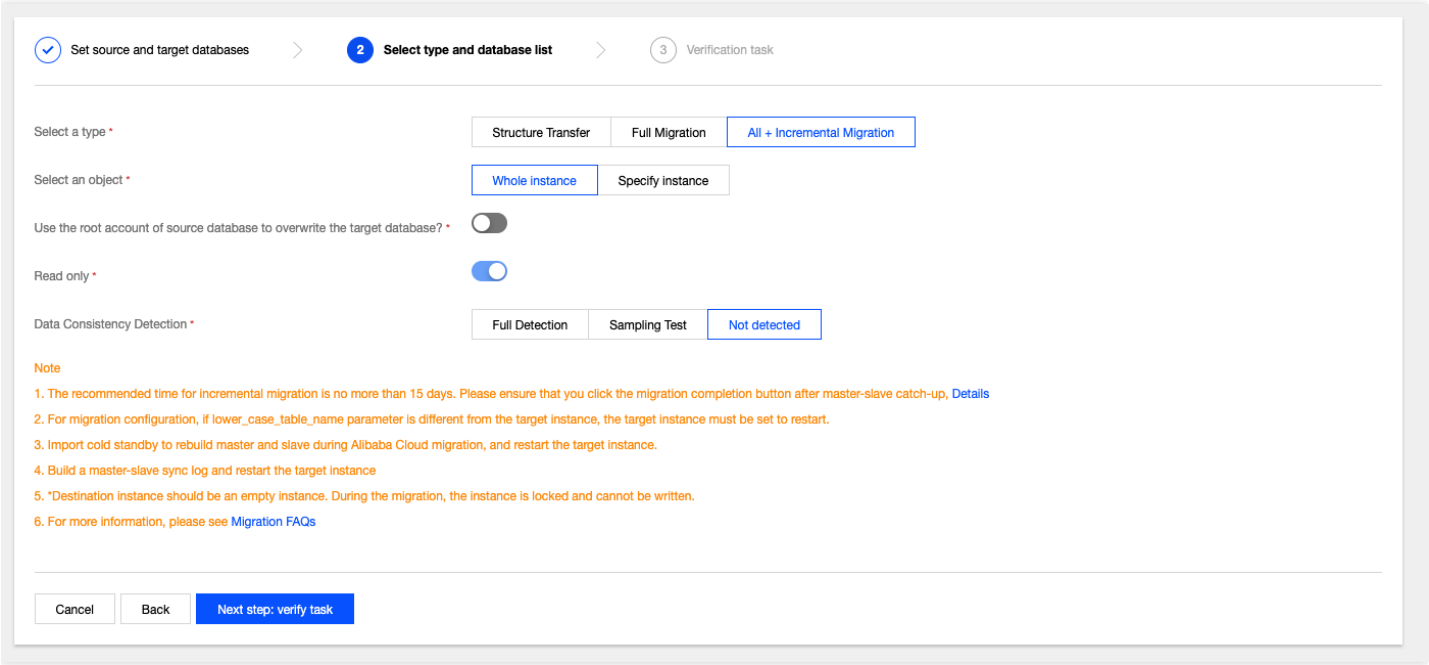
c. Target database information

Select TencentDB instance as target instance type and enter the link information of the target instance.



3. Select the database to migrate

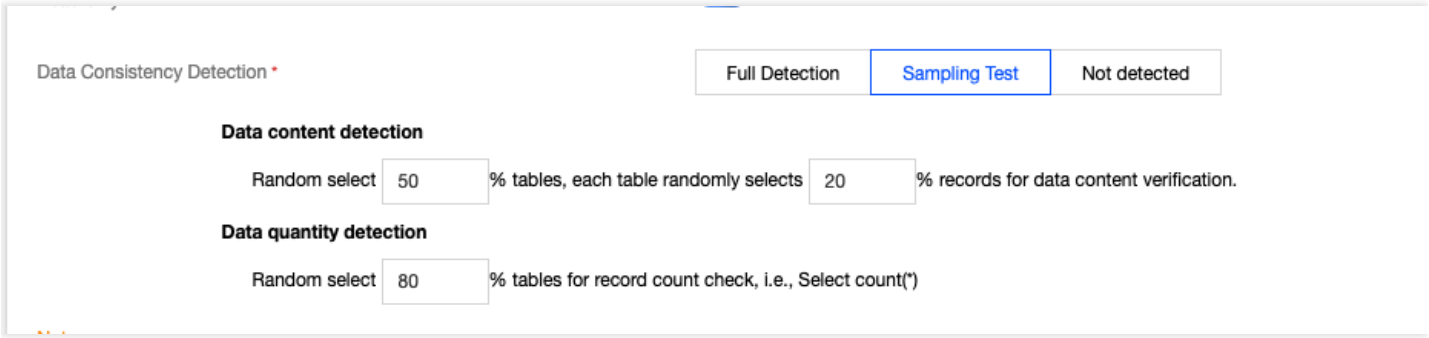
Select the database to migrate, create a migration task information, and check the task information.



4. Performing data consistency check

Select a data consistency check type as needed (e.g., full check or no check).

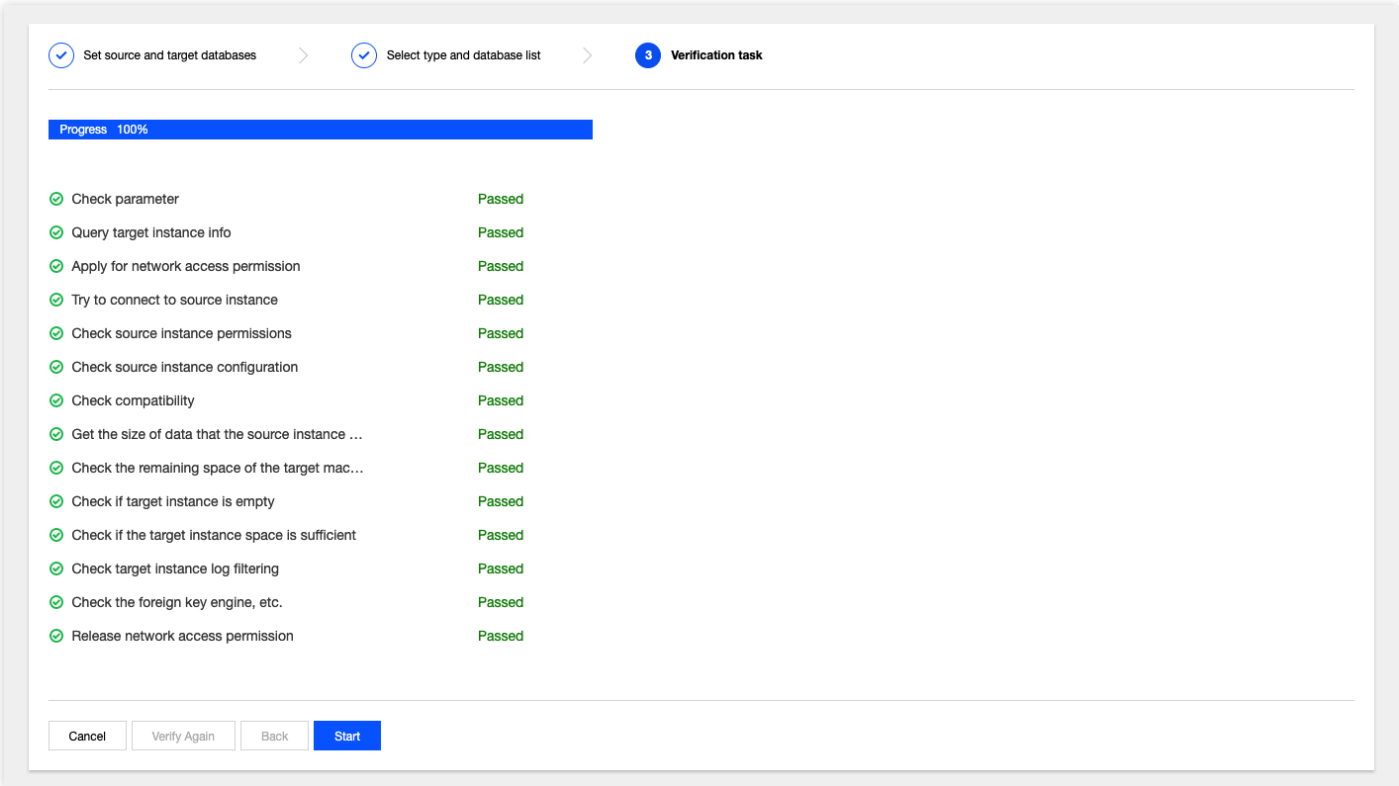
> **⚠ Note :**
>
> The detection ratio fields are required if sampling detection is selected.



5. Checking the migration task information

After the migration task is created, you need to click **Next: Check Task** to verify the task information. You cannot start the migration task until all the check items are passed.
There are 3 statuses for task check:

- Passed: the check is fully successful.
- Warning: the check fails. Database operation may be affected during or after data migration, but the migration task can still be executed.
- Failed: the check fails and the migration task cannot be executed. In this case, please check and modify the migration task information according to the error and then check the task again.



6. Starting migration

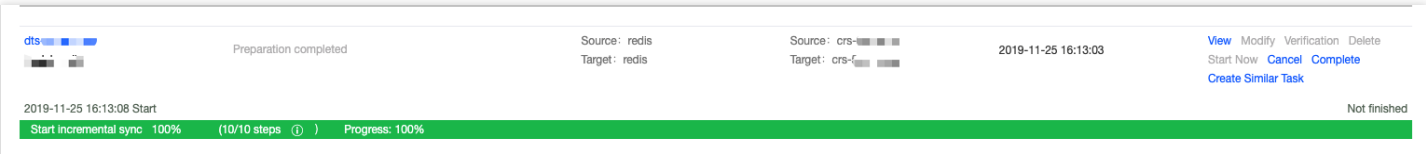
> **Note :**
>
> Due to system design limitations, multiple migration tasks submitted or queued at the same time will be performed in sequence by queuing time.

After the check is passed, you can click **Start** to start data migration. Please note that if you have configured scheduled execution, the migration task will begin queuing and be executed at the specified time; otherwise, it will be executed immediately.
After the migration is started, you can view the corresponding migration progress under the migration task. The subsequent steps required for migration and the current stage will be displayed if you hover over the exclamation mark after the current step.

> **Note :**
>
> If an error occurs: "errMsg": "Failed to start backup task. SDK.ServerUnreachable Unable to connect server:HTTP Error 403: Forbidden":
> You can troubleshoot the problem in the following steps:
>
> - Check whether your Alibaba Cloud key has permission to initiate cold backup on the ApsaraDB for RDS instance. If an Alibaba Cloud root account is used, which has all permissions, this cause can be ruled out.
> - Log in to the Alibaba Cloud Console, check whether the ApsaraDB for RDS instance is executing a conflicting task such as cold backup or upgrading; if automatic backup is enabled, you need to disable it.
> - If the problem persists after the above two steps, please contact Alibaba Cloud to find out the cause of failure in initiating a cold backup task with an API.

7. Canceling migration (optional)

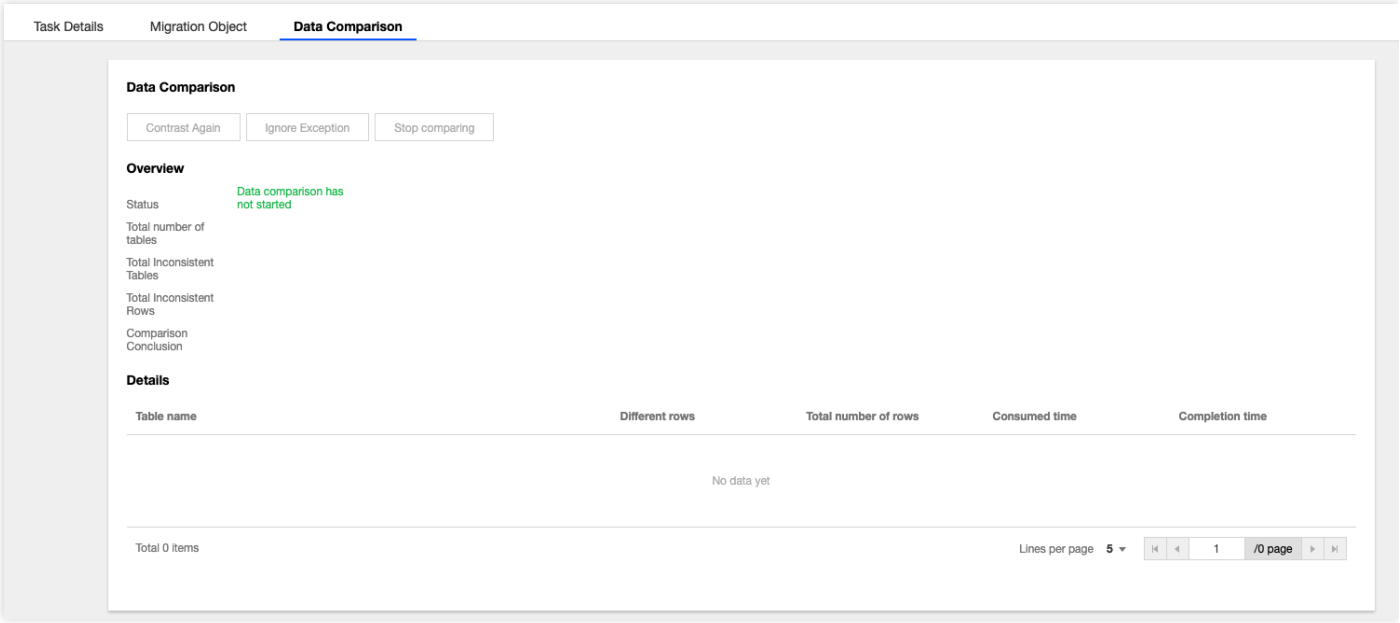
To cancel an in-progress migration task, click **Cancel**.



8. Disconnecting the migration task

- Make sure that there are no writes into the ApsaraDB for RDS instance at the business side. You can change the account password for business connection or adjust the permission granted to the account, while ensuring that the account used for DTS sync allows reads and writes.

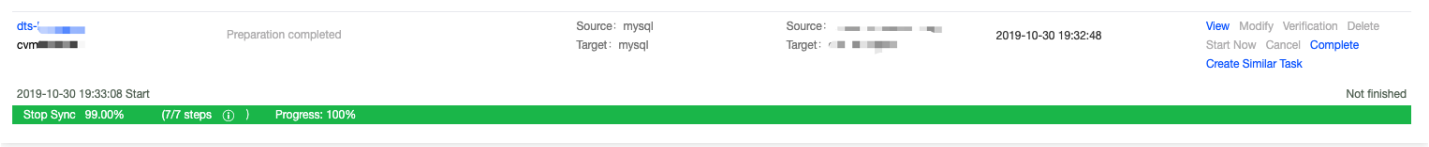
- Stop business connection to ApsaraDB for RDS and run the `show processlist` command to confirm that there is no business connection.
- Run the `show master status` command to get the latest GTID of ApsaraDB for RDS and compare it with that of TencentDB obtained via `show slave status`, so as to ensure that there is no synchronization delay between TencentDB and ApsaraDB for RDS instances.
- Check whether your original ApsaraDB for RDS account can be used to log in to TencentDB; if not, try to escalate it to a privileged account in the following steps:
  While the DTS task continues syncing, add a privileged account on ApsaraDB for RDS (the password encryption method can be changed to general encryption), which will be synced to TencentDB via DTS.
- You can use the console (as shown below) or extract core table contents to check data consistency.



- Run the `show slave status` command to record the sync time point of TencentDB.

9. Completing the migration

After the migration is 100% complete, you can click **Complete** on the right. You can also call the DTS TencentCloud API to stop the sync.



> **Note :**
>
> If the migration is in **uncompleted** status, the migration task will continue, so will data sync.

10. Restarting your business application

Disable the read-only feature of TencentDB, restart the application, and observe the status of TencentDB to make sure that it is running normally.

Redis Data Migration
Migrating with DTS

Last updated : 2020-08-31 11:19:43

DTS Overview

Tencent Cloud Data Transmission Service (DTS) is a data transfer service that integrates such features as data migration, sync, and subscription, helping you migrate your databases without interrupting your business and build a high-availability database architecture for remote disaster recovery through real-time sync channels. Its data subscription feature grants you real-time access to incrementally updated data in your TencentDB instance, so that you can consume such data based on your business needs. Currently, DTS for Redis supports data migration on different versions of Redis and in various network scenarios.

Term	Description
Source instance	Source instance of the migration.
Target instance	Target instance to be migrated to, i.e., user-purchased TencentDB for Redis.
Self-built instance on CVM	Redis service deployed on a CVM instance.
Self-built instance on public network	Redis service deployed on a public network.

Migration Support Description

Note :

For more information on compatibility requirements for migration from standalone edition to memory edition (cluster architecture), please see Notes on Migration from Standalone Edition to Cluster Edition.

Supported features

- Data migration: DTS supports one-time migration of all data to the cloud.
- Data sync: DTS supports real-time data sync with the cloud by combining full migration and incremental sync.

Supported versions

- DTS supports Redis 2.8, 3.0, 3.2, 4.0, and 5.0.
- DTS supports single-node, Redis cluster, Codis, and twemproxy architectures.
- Migration permission requirements: to migrate data via DTS, the source instance must support SYNC or PSYNC commands.

Supported networks

DTS supports data migration and data sync across a public network, CVM-created instances, Direct Connect gateways, VPN gateways, and CCN.

Supported scenarios

- Cloudification migration: DTS supports migrating your Redis instance in a traditional IDC to TencentDB for Redis, moving your businesses to the cloud in an efficient and convenient manner.
- Self-built service migration: DTS supports migrating your Redis service created with a virtual machine in Tencent Cloud or other clouds to TencentDB for Redis.
- Migration of Redis data from other cloud vendors: DTS supports migrating Redis data from other cloud vendors to Tencent Cloud provided that the SYNC or PSYNC command permission has been granted.
- Migration between instances on Tencent Cloud: DTS supports data migration or real-time sync between instances on Tencent Cloud. The supported versions are as follows:

Target Instance / Source Instance	2.8 Memory Edition (standard architecture)	4.0 Memory Edition (standard architecture)	4.0 Memory Edition (cluster architecture)	5.0 Memory Edition (standard architecture)	5.0 Memory Edition (cluster architecture)
2.8 Memory Edition (standard architecture)	✓	✓	✓	✓	✓
4.0 Memory Edition (standard architecture)	x	✓	✓	✓	✓
4.0 Memory Edition (cluster architecture)	x	✓	✓	✓	✓
5.0 Memory Edition	x	✓	✓	✓	✓

(standard architecture)					
5.0 Memory Edition (cluster architecture)	x	✓	✓	✓	✓

Migration limitations

- To ensure migration efficiency, cross-region migration is not supported for self-built instances on CVM.
- To migrate instances over a public network, make sure that the source instance is accessible from the public network.
- Only instances that are running normally can be migrated, while instances with no password initialized or with ongoing tasks cannot be migrated.
- The target instance must be empty with no data. During the migration process, the target instance will be read-only and cannot be written to.
- After the successfully migrated data is verified by your business, the connection to the source instance can be closed and then switched to the destination instance.

Migration Process

1. Create a migration task

1) Log in to the DTS Console, go to the **Data Migration** page, and click **Create Migration Task**.
2) Select the corresponding region in the "Linkage Region" section and click **Buy at 0 USD**.

2. Configure the task

- Task name: specify a name for the task.
- Scheduled execution: specify the start time of the migration task.

> **Note :**
>
> - To modify the scheduled task, you must click **Scheduled start** again after the verification is passed, so as to make the task start at the specified time.
> - If the specified time has passed, the task will start immediately. You can also click **Immediate start** to start the task immediately.

3. Set the source instance and target instance

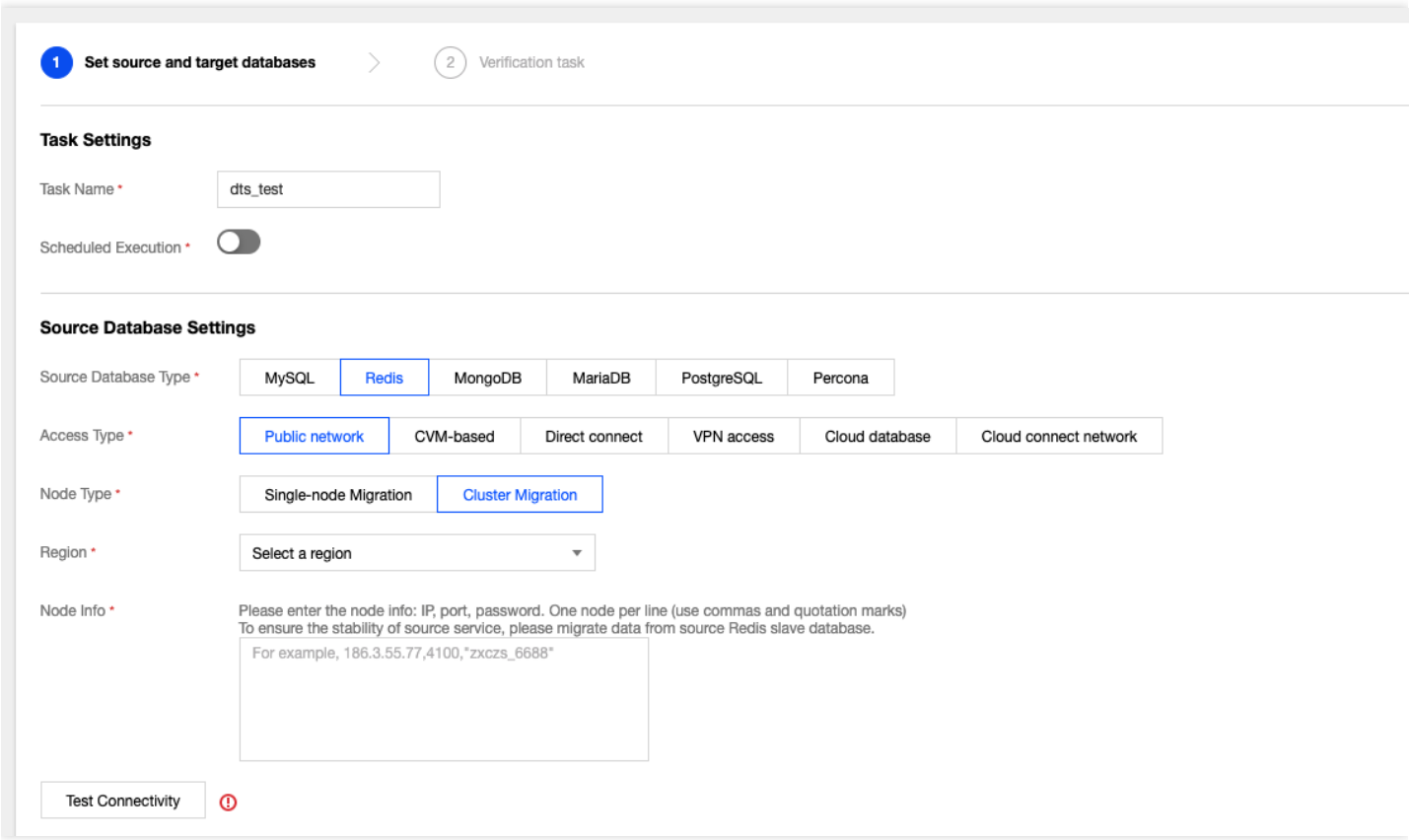
Redis instances on CVM are used here as an example, and the same is true for migration of instances over a public network.

Field	Description	Remarks	Required
Task name	Name of the migration task	Used by users for their management of tasks	Yes
CVM instance ID	ID of the CVM instance where the source Redis instance resides	The migration task checks the CVM running conditions based on the CVM instance ID	Yes
CVM private IP	Private IP of the CVM instance where the source Redis instance resides	The migration task checks the CVM private IP	Yes
Port	Port number of source instance	The migration task will access the source instance service	Yes
Password	Source instance password	The password is used for the authentication for accessing the source instance	No
Instance ID	Target instance ID	Data is synchronized to the target instance	Yes

Notes on migration in the cluster edition

DTS supports migration in the Redis Cluster Edition. For cluster schemes with the Redis Cluster, Codis, or twemproxy architecture, simply enter the addresses and passwords of all shard nodes of the source cluster as the node information when creating the task. It is strongly recommended to perform data migration from a replica node (slave) of the source instance to avoid any impact on business access to the source instance. DTS supports password-free migration. The following is an example for entering

relevant information for migration:



4. Start the migration task

1) After the network connectivity test is successful, click **Save**.
2) DTS begins to verify the migration task, and once the migration requirements are met, the migration task will be started.
3) Upon task start, the task status will change to **Verifying**, indicating that another round of parameter verification is underway. During this process, you are only able to cancel or view the task or check the verification progress.
4) After parameter verification succeeds, data migration will start.
During data sync, changes in data offset, source instance, and target instance key will be displayed.

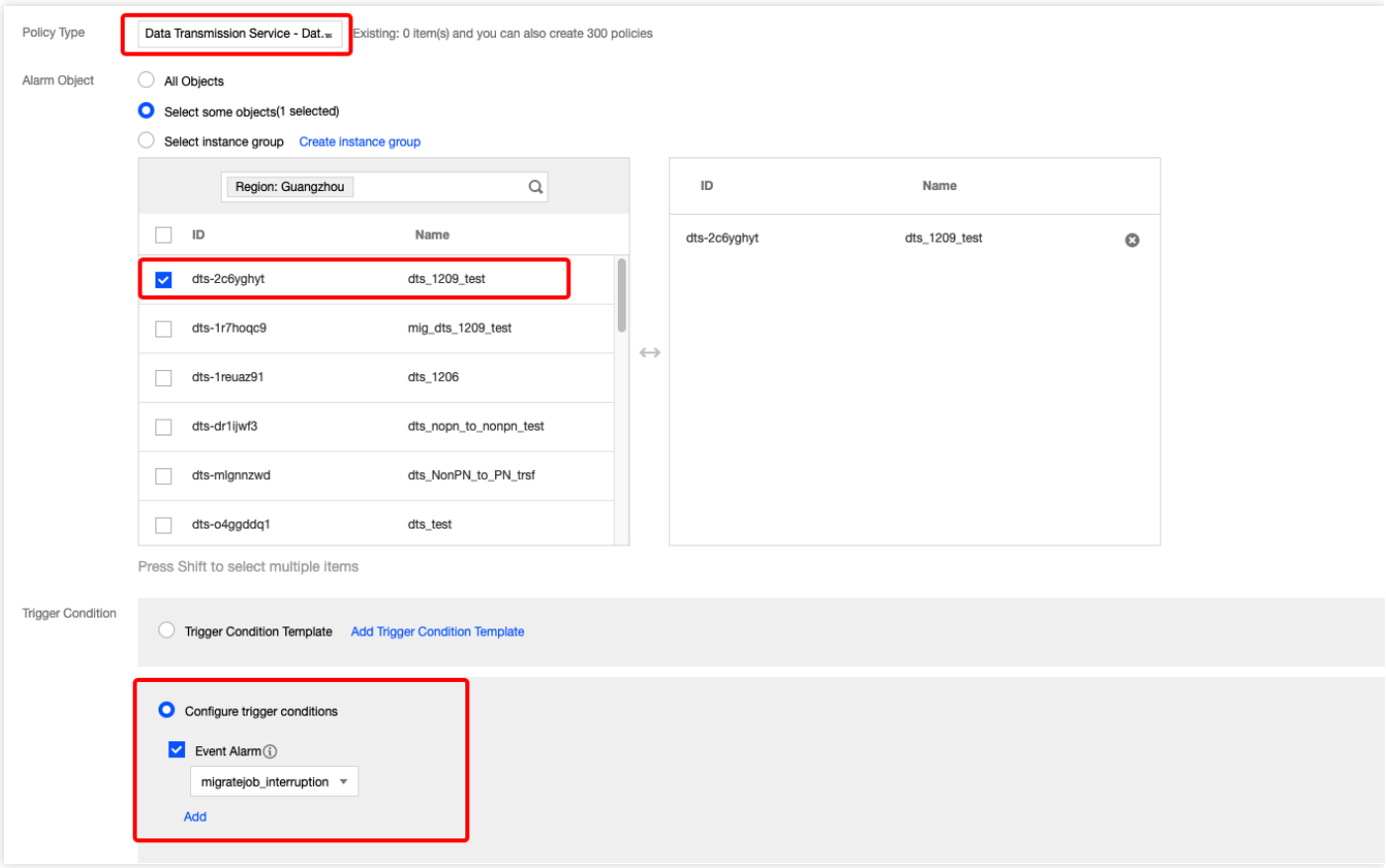
5. Configure a migration alarm

DTS supports migration interruption alarming to keep you informed of any exceptions. A migration alarm can be configured as follows:

1) Log in to the Cloud Monitor Console and select **Alarm Configuration** > **Alarm Policy** on the left sidebar.
2) Click **Add** to create an alarm policy.

- Policy type: select **Data Transmission Service** > **Self-built Migration**.

- Alarm object: select the DTS task to be monitored and configure the **Trigger Condition** and **Alarm Object** to finish alarm configuration.



6. Complete the migration task

Before disabling data sync, the data can be verified on the target instance, and if everything is correct, the migration task can be completed.

If the keys of the source instance and the target instance are identical, click **Complete** to finish data sync.

Migration with redis-port

Last updated : 2021-02-02 09:48:51

Overview

Download redis-port (Linux 64-bit).

redis-port is a collection of open-source tools mainly used for database sync, data import, and data export between Redis nodes and supports cross-version Redis data migration. The toolkit contains the following tools:

- redis-sync: it is used for data migration between Redis instances.
- redis-restore: it supports importing Redis backup files (in RDB format) to specified Redis instance.
- redis-dump: it supports backing up Redis data in RDB format.
- redis-decode: it supports decoding Redis RDB backup files into readable files.

Compatible Versions

- Source instances on Redis 2.8, 3.0, 3.2, and 4.0 are supported.
- Target instances on Redis 2.8, 3.0, 3.2, and 4.0 and in all editions of TencentDB are supported, including Redis Memory Edition and CKV Edition.

Online Migration Through redis-sync

How it works

- redis-sync has two modules which are simulated as replication nodes to sync data from the source instance and translate the replicated data into write commands to update the target instance.
- Data replication is done in two phases: full sync and incremental sync.

Parameter description:

- -n: number of concurrent write tasks. You are recommended to leave it empty or set it to CPU core quantity * 2.
- -m: source instance address in the format of `”password”@ip:port` or `ip:port` (in password-free mode).
- -t: target instance address in the format of `”password”@ip:port` or `ip:port` (in password-free mode).

- --tmpfile=FILE: temporary filename.
- --tmpfile-size=SIZW: maximum size of the temporary file.
- --help: help command.

Sample:

```
./redis-sync -m 127.0.0.1:6379 -t "xxx2018"@10.0.5.8:6379
```

Output log:

```
[root@VM_5_16_centos bin]# ./redis-sync -m 127.0.0.1:6379 -t "xxx2018"@10.0.5.8:6379
2019/02/21 09:56:00 sync.go:76: [INFO] sync: master = "127.0.0.1:6379", target = "xxx2018@10.0.5.8:6379"
2019/02/21 09:56:01 sync.go:103: [INFO] +
2019/02/21 09:56:01 sync.go:109: [INFO] sync: runid = "f63e2ad58e2fcc15c8cc122f15778389a012c1a4", offset = 18576271
2019/02/21 09:56:01 sync.go:110: [INFO] sync: rdb file = 9063349 (8.64mb)
2019/02/21 09:56:01 sync.go:208: [INFO] sync: (r/f,s/f,s) = (read,rdb.forward,rdb.skip/rdb.forward,rdb.skip)
2019/02/21 09:56:02 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=(1703936/71754,0/0,0) ~ (1.62mb/-,-/-,-) ~ speed=(1.62mb/71754,0/0,0)
2019/02/21 09:56:03 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=(3407872/153850,0/0,0) ~ (3.25mb/-,-/-,-) ~ speed=(1.62mb/82096,0/0,0)
2019/02/21 09:57:54 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=(80487526/411969,0/1587212,0) ~ (76.76mb/-,-/-,-) ~ speed=(0/0,0/0,0)
```

Use instructions:

- The database capacity of the target instance should be greater than that of the source instance; otherwise, the migration will fail.
- If migration is interrupted for reasons such as network failure, you need to empty the target instance first and then perform migration again; otherwise, there may be dirty data.
- The progress of migration is displayed in the log, where "sync: rdb = 9063349 - [100.00%]" indicates that full data has been synced and incremental data sync is in progress, while "speed=(0/0,0/0,0)" indicates that incremental data has been synced.
- You can stop data sync and migration by pressing Ctrl + C or through other means.

Importing Data Through redis-restore

redis-restore supports importing Redis backup files (in RDB format) on Redis 2.8, 3.0, 3.2, and 4.0 as well as AOF files into the specified Redis instance.

Parameter description:

- -n: number of concurrent write tasks. You are recommended to leave it empty or set it to CPU core quantity * 2.
- -i: RDB file path.
- -t: target instance address in the format of `”password”@ip:port` or `ip:port` (in password-free mode).
- -a: AOF file path.
- --db=DB: database ID of the target Redis instance for backup file import, which should be the same as that of the source instance.
- --unixtime-in-milliseconds=EXPR: the Key expiration time value is updated in the process of data import.
- --help: help command.

Sample:

```
./redis-restore dump.rdb -t 127.0.0.1:6379
```

Backing up Data Through redis-dump

redis-dump supports backing up Redis data into RDB files and incremental data into AOF files.

> **Note：**
>
> TencentDB for Redis currently does not support backing up data through redis-dump. You can back up and download data in the TencentDB for Redis console or through APIs. However, you can use redis-dump to back up your self-built Redis instances.

Parameter description:

- -n: number of concurrent write tasks. You are recommended to leave it empty or set it to CPU core quantity * 2.
- -m: Redis instance address in the format of `”password”@ip:port` or `ip:port` (in password-free mode).
- -o: path of the output RDB file.
- -a: Path to the outputted AOF file.
- --help: Help command.

Sample:

```
./redis-dump 127.0.0.1:6379 -o dump.rdb
```

MongoDB Data Migration

Last updated : 2020-08-12 17:31:37

Tencent Cloud Data Transmission Service (DTS) is a data transfer service that integrates such features as data migration, sync, and subscription, helping you migrate your databases without interrupting your business and build a high-availability database architecture for remote disaster recovery through real-time sync channels. Its data subscription feature grants you real-time access to incrementally updated data in your TencentDB instance, so that you can consume such data based on your business needs. Currently, DTS for MongoDB supports data migration on different versions of MongoDB and in various network scenarios.

Term	Description
Source instance	Source instance to be migrated from.
Target instance	Target instance to be migrated to, i.e., user-purchased TencentDB for MongoDB.
CVM-created instance	MongoDB service deployed on a CVM instance.
Public network-created instance	MongoDB service deployed on the public network.

Migration Support Description

Supported features

- Data migration: DTS supports one-time migration of all data to the cloud.
- Data sync: DTS supports real-time data sync by combining full migration and incremental sync.
- Table migration: DTS supports table migration.
- Migration availability: DTS supports auto-retry and checkpoint restart.
- Latency monitoring: DTS displays the latency of incremental sync in the console.

Supported versions

- DTS supports MongoDB 3.0, 3.2, 3.4, 3.6, and 4.0.
- Currently, DTS supports only MongoDB replica sets but not sharded clusters.

Supported networks

DTS supports data migration and data sync across a public network, CVM-created instances, Direct Connect gateways, VPN gateways, and CCN.

Supported scenarios

- Migration to cloud: DTS supports migrating your MongoDB instance in a traditional IDC to TencentDB for MongoDB, moving your businesses to the cloud in an efficient and convenient manner.
- Self-built service migration: DTS supports migrating your MongoDB service created with virtual machines in Tencent Cloud or other clouds to TencentDB for MongoDB.
- Migration of MongoDB data from other cloud vendors: DTS supports migrating MongoDB data from other cloud vendors to TencentDB for MongoDB.
- Migration between instances on Tencent Cloud: DTS supports data migration or real-time sync between instances on Tencent Cloud.

Notes on migration

- To ensure migration efficiency, cross-region migration is not supported for self-built instances on CVM.
- To migrate instances over a public network, make sure that the source instance is accessible from the public network.
- Before migration, please create a read-only account for the source instance; otherwise, the migration verification will fail.
- After the successfully migrated data is verified by your business, the connection to the source instance can be closed and then switched to the target instance.
- Online migration is not supported for self-built single-node instances as they have no oplog.

Migration Directions

1. Log in to the DTS Console and click **Create Migration Task** on the data migration page, select a **Linkage Region**, and click **Buy at 0 USD**.

> **Note :**
>
> - Currently, MongoDB database migration is free of charge.
> - Please select the region with caution as it cannot be changed once the migration task is created.

2. On the "Set source and target databases" page, select MongoDB as the source database type, and specify instance and account information.

> **Note :**

> Before migration, please create a read-only account for the source instance; otherwise, the migration verification will fail.

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

4. On the "Set migration options and select migration objects" page, set migration options and select migration objects (you can select some tables).

5. On the "Verify task" page, verify the migration task and click **Start Task**.

6. After the verification is passed, return to the data migration list. After the incremental sync is 100% complete, click **Complete** in the "Operation" column to finish the migration task.

PostgreSQL Data Migration

Last updated : 2021-01-06 10:04:08

Overview

DTS provides data migration and continuous data replication from self-created PostgreSQL databases to TencentDB, allowing you to migrate hot data without interrupting your business. PostgreSQL with public IP/port, Direct Connect-based PostgreSQL in local IDC, and CVM-based self-created PostgreSQL can be migrated.

> ⚠ **Note :**
>
> Currently, data migration is only supported for PostgreSQL databases on 9.3.x or 9.5.x. Incremental sync is not supported for 9.3.x and can be used in 9.5.x only through an online sync plugin. For more information on how to configure and use the plugin, please see Sync Plugin Configuration.

Directions

Creating DTS data migration service

Log in to the DTS console and click **Create Migration Task** on the data migration page.

Setting source and target databases

On the page you are redirected to, enter the settings of the task, source database, and target database.

Task setting

- Task Name: specify a name for the task.
- Scheduled Execution: specify the start time of the migration task.

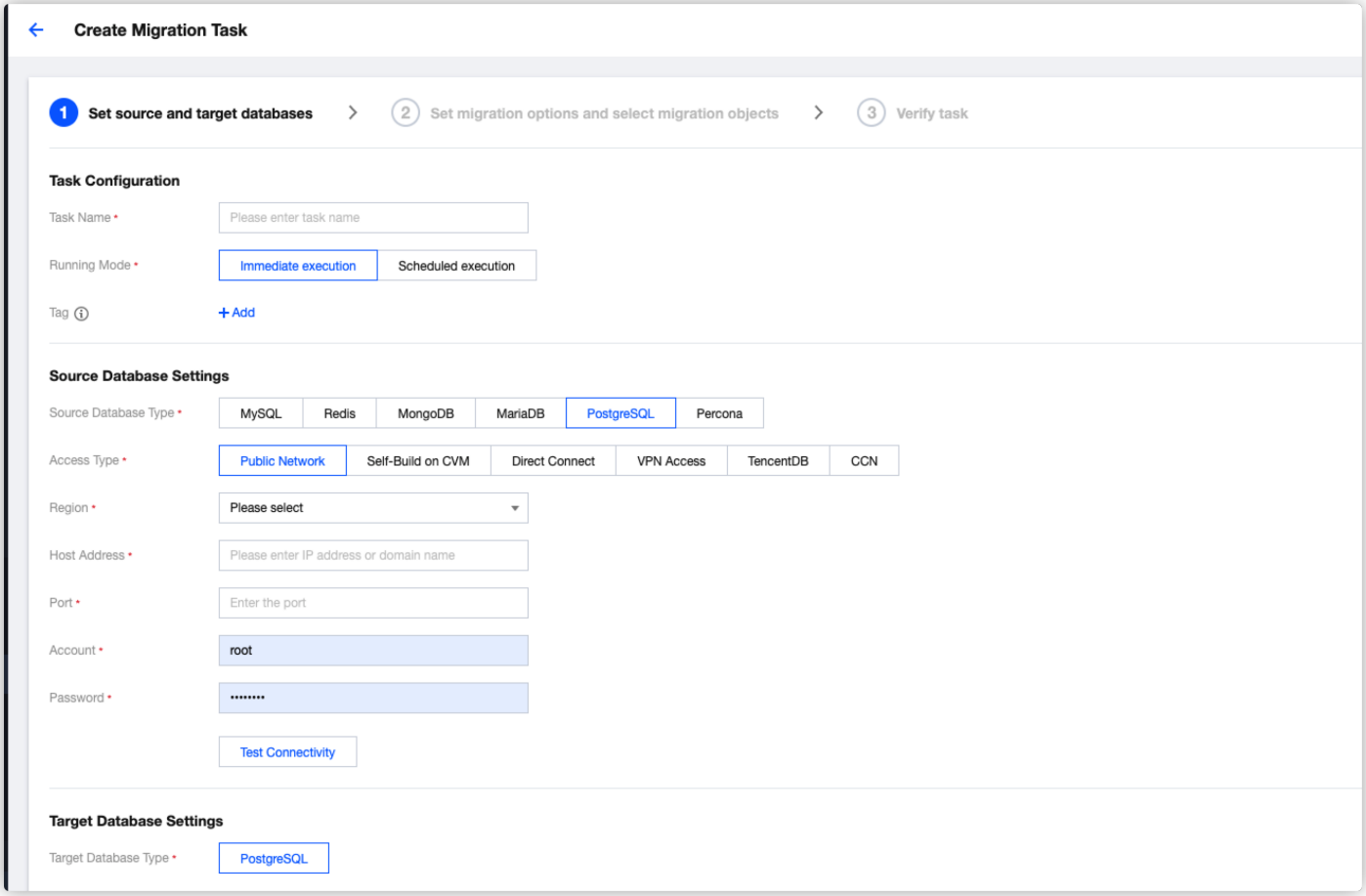
Source and target database settings

Source Database Type: PostgreSQL with public IP, CVM-based self-created PostgreSQL, Direct Connect-based PostgreSQL, VPN-based PostgreSQL, and TencentDB for PostgreSQL are supported.

Source Database Type	Description

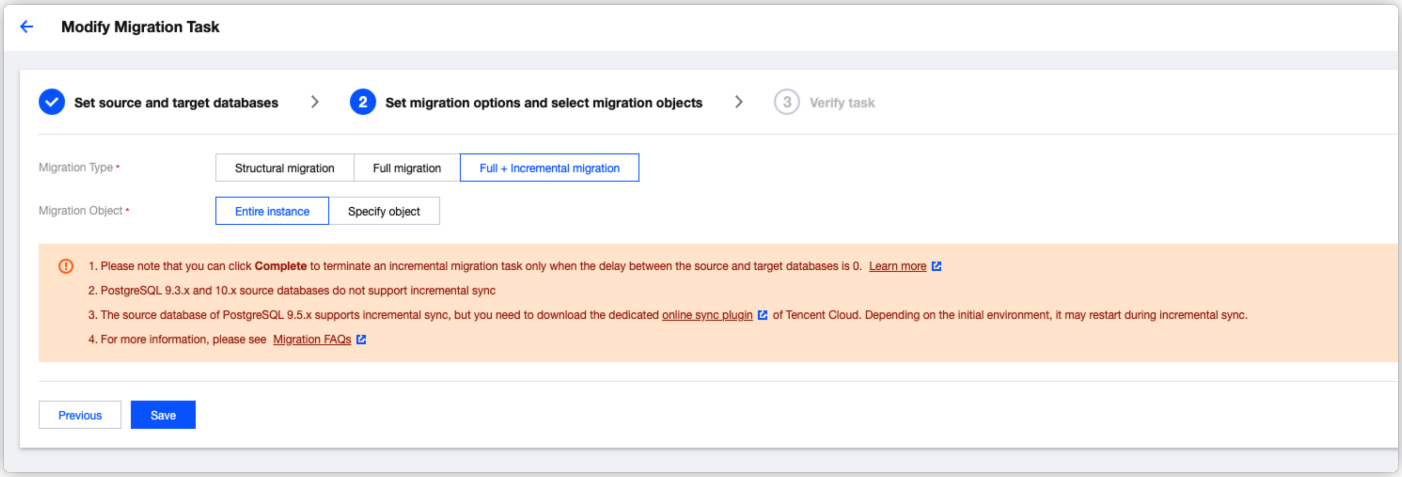
PostgreSQL with public IP	It refers to a PostgreSQL database that can be accessed through a public IP. The following information is required: • PostgreSQL server address • PostgreSQL port • PostgreSQL account • PostgreSQL password
CVM-based self-created PostgreSQL	You can migrate a PostgreSQL database deployed on a CVM instance in the classic network or VPC by specifying the CVM instance ID and network environment. The following information is required: • Region: currently, a CVM-based self-created PostgreSQL database can be migrated to a TencentDB instance only if they are in the same region; otherwise, you should use the CVM instance's public network access by selecting **PostgreSQL with Public IP** for migration • CVM network: you can select the classic network or VPC • VPC: if a VPC is selected, you need to specify the VPC and subnet where the CVM instance resides • CVM instance ID • PostgreSQL port • PostgreSQL account • PostgreSQL password
Direct Connect-based PostgreSQL	After connecting a self-created PostgreSQL database in your local IDC to Tencent Cloud through Direct Connect, you can use DTS to migrate data to Tencent Cloud. The following information is required: • Direct Connect gateway: it is the Direct Connect gateway used by the database server to connect to Tencent Cloud. For more information, please see Direct Connect Gateway • VPC: it is the VPC where the Direct Connect gateway resides • PostgreSQL server address: it is the address of your PostgreSQL server in the IDC. The IP mapped by the Direct Connect gateway will be accessed during DTS data migration • PostgreSQL port • PostgreSQL account • PostgreSQL password
VPN-based PostgreSQL	After connecting a self-created PostgreSQL database in your local IDC to Tencent Cloud through VPN Connections or a CVM-based self-created VPN service, you can use DTS to migrate data to Tencent Cloud. The following information is required: • Region: currently, only VPN services in the same region as that of the target TencentDB instance are supported • VPN type: you can select VPN Connections or CVM-based self-created VPN • VPN gateway: you need to provide the VPN gateway information only if VPN Connections is selected. For more information, please see VPN • VPC: it is the VPC where the VPN Connections instance resides • PostgreSQL server address: it is the address of your PostgreSQL server in the IDC. The IP mapped by the Direct Connect gateway will be accessed during DTS

	data migration • PostgreSQL port • PostgreSQL account • PostgreSQL password
TencentDB for PostgreSQL	It refers to a TencentDB for PostgreSQL instance. The following information is required: • PostgreSQL instance ID • PostgreSQL account • PostgreSQL password



Selecting database to be migrated

Select the database to be migrated (you can choose to migrate the entire database or only certain tables).

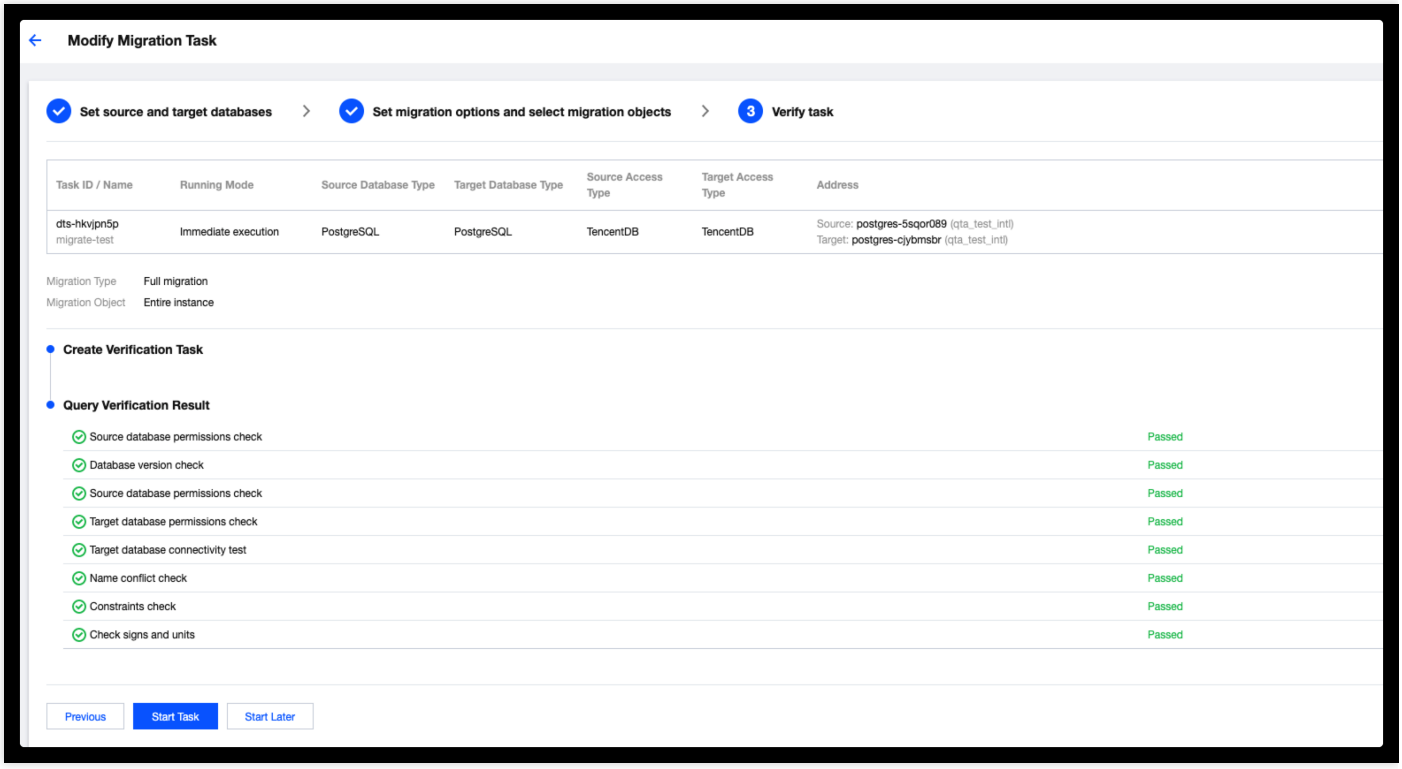


Verifying migration task information

Click **Next step: verify task** to verify the migration task information. You cannot start the migration task until all the check items are passed. After the verification is completed, click **Start**.

There are 3 statuses for task verification:

- Passed: the verification is fully successful.
- Warning: the verification fails. Database operation may be affected during or after data migration, but the migration task can still be executed.
- Failed: the verification fails and the migration task cannot be executed. In this case, please check and modify the migration task information according to the error and then verify the task again.



Starting migration

After the verification is passed, return to the data migration list and click **Start** in the **Operation** column to start data migration. Please note that if you have configured scheduled execution, the migration task will begin queuing and be executed at the specified time; otherwise, it will be executed immediately.

After the migration is started, you can view the corresponding migration progress under the migration task. The subsequent steps required for migration and the current stage will be displayed if you hover over the exclamation mark after the current step.

> ⚠ **Note :**
>
> Due to system design limitations, multiple migration tasks committed or queued at the same time will be performed in sequence by queuing time.

Incremental sync

When a migration task is created, the incremental sync option is selected by default. When data migration is completed, the target database will be set as the slave of the source database, and new data in the source database during migration will be synced to the target database through master/slave sync. In this way, any modification of the source database will be synced to the target database.

After migration, you need to click **Complete** to close the sync between the source and target databases so as to complete the migration.

> ⚠ **Note :**
>
> Before closing the sync, do not write data into the target database; otherwise, data comparison may fail due to inconsistency between the source and target databases, resulting in migration failure.

Stopping migration

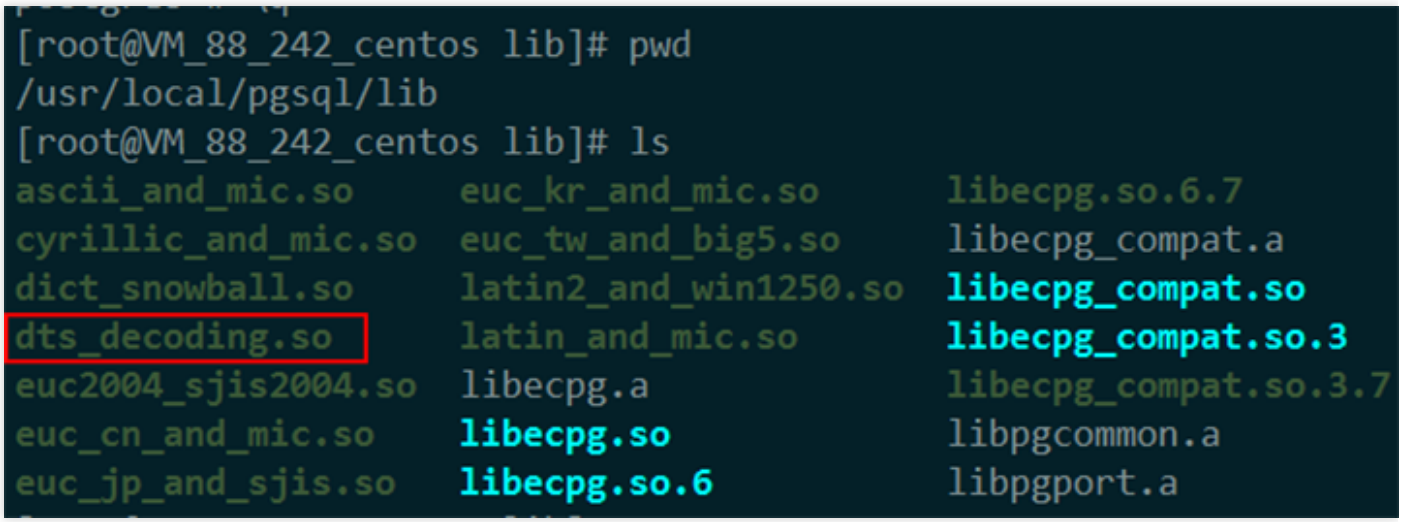
To stop an ongoing migration task, click **Cancel** in the **Operation** column of the task.

> ⚠ **Note :**
>
> - Restarting the task may cause the verification or task to fail. You may have to manually clear all databases or tables that may cause conflicts in the target database before you can start the migration task again.
> - When migrating a single table, make sure that all tables depended on by its foreign keys are also migrated.

Completing migration

Sync Plugin Configuration

1. Download and copy dts_decoding to the `lib` directory in the PostgreSQL installation path.



2. Modify the configuration file `postgresql.conf` in the `data` directory.

```
wal_level >= logical
 Available max_replication_slots >= number of databases to be migrated
 Available max_wal_senders >= number of databases to be migrated
```

```
# - Settings -

wal_level = logical                     # minimal, archive, hot_standby, or logical
```

```
# Set these on the master and on any standby that will send replication data.

max_wal_senders = 10            # max number of walsender processes
                                # (change requires restart)
#wal_keep_segments = 0          # in logfile segments, 16MB each; 0 disables
#wal_sender_timeout = 60s       # in milliseconds; 0 disables

max_replication_slots = 10      # max number of replication slots
```

3. Modify the configuration file `pg_hba.conf` in the `data` directory.
   Replication connection needs to be configured.

```
# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
host    all             all             0.0.0.0/0               md5
host    replication     all             0.0.0.0/0               md5
# IPv6 local connections:
host    all             all             ::1/128                 trust
```

4. Restart the source database.

> ⚠ **Note :**
>
> When specified tables are migrated, if tables use rules or are associated with other tables, data insertion during incremental migration may fail, as some SQL operations are not supported by the migration feature. In this case, you can migrate schemas or all databases.

MariaDB Data Migration

Last updated : 2021-01-06 10:16:26

1. Creating a migration task

1) Log in to the DTS console and click **Create Migration Task** on the data migration page.
2) Select the corresponding region in **Linkage Region** and click **Buy at 0 CNY**.

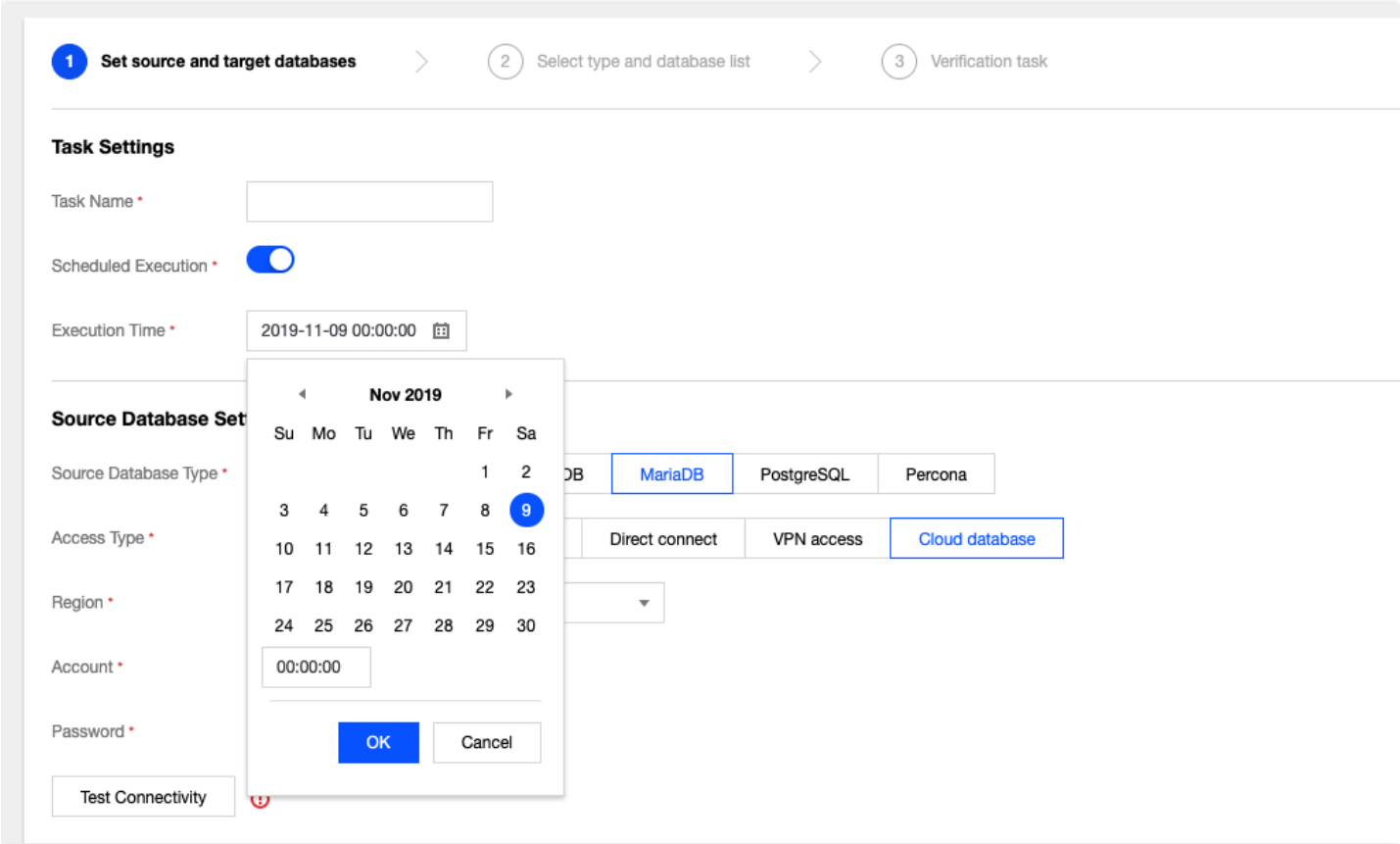
> **Note :**
>
> Please select the region with caution as it cannot be changed once the migration task is created.

2. Setting the source and target instances

Enter relevant information to set the task, source database, and target database.

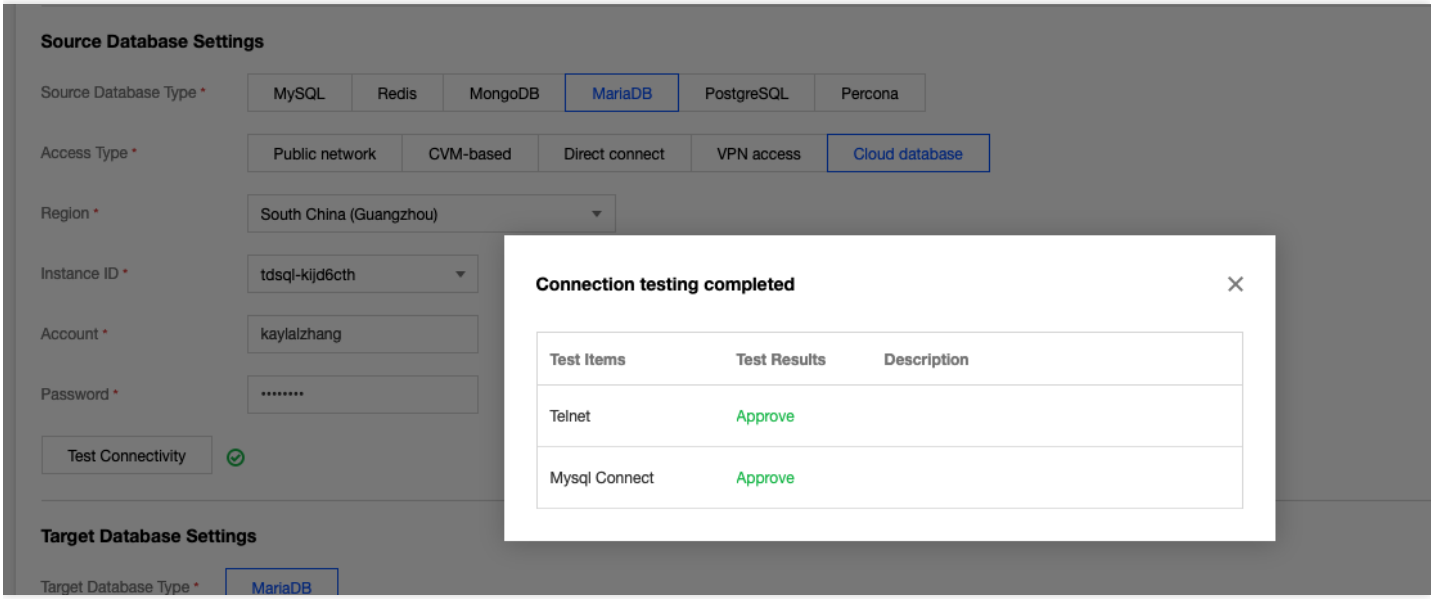
Task setting

Enter the name of the migration task. If you want the migration task to be executed at a later time, you can set scheduled execution.



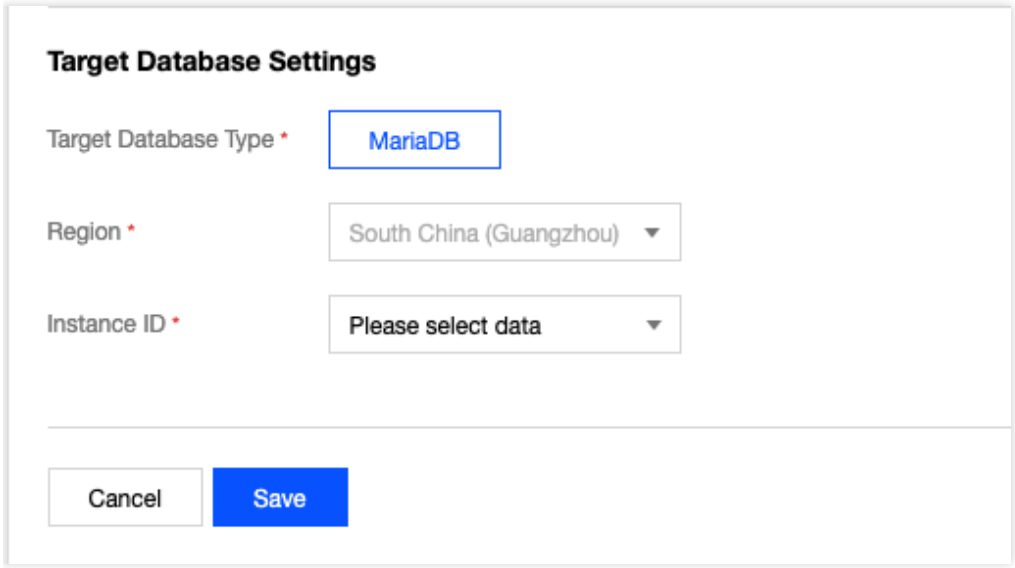
Source database setting

Enter the information of the source database and then click **Test Connectivity** to check whether your source database is connected.



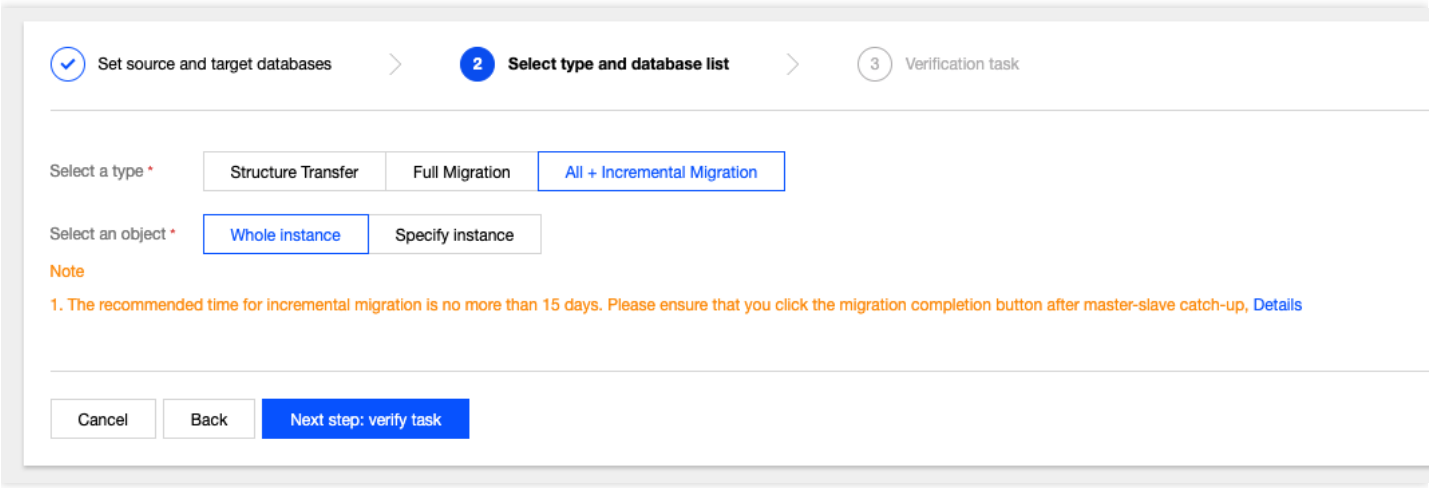
Target database setting

Enter the information of the target database and click **Save**.



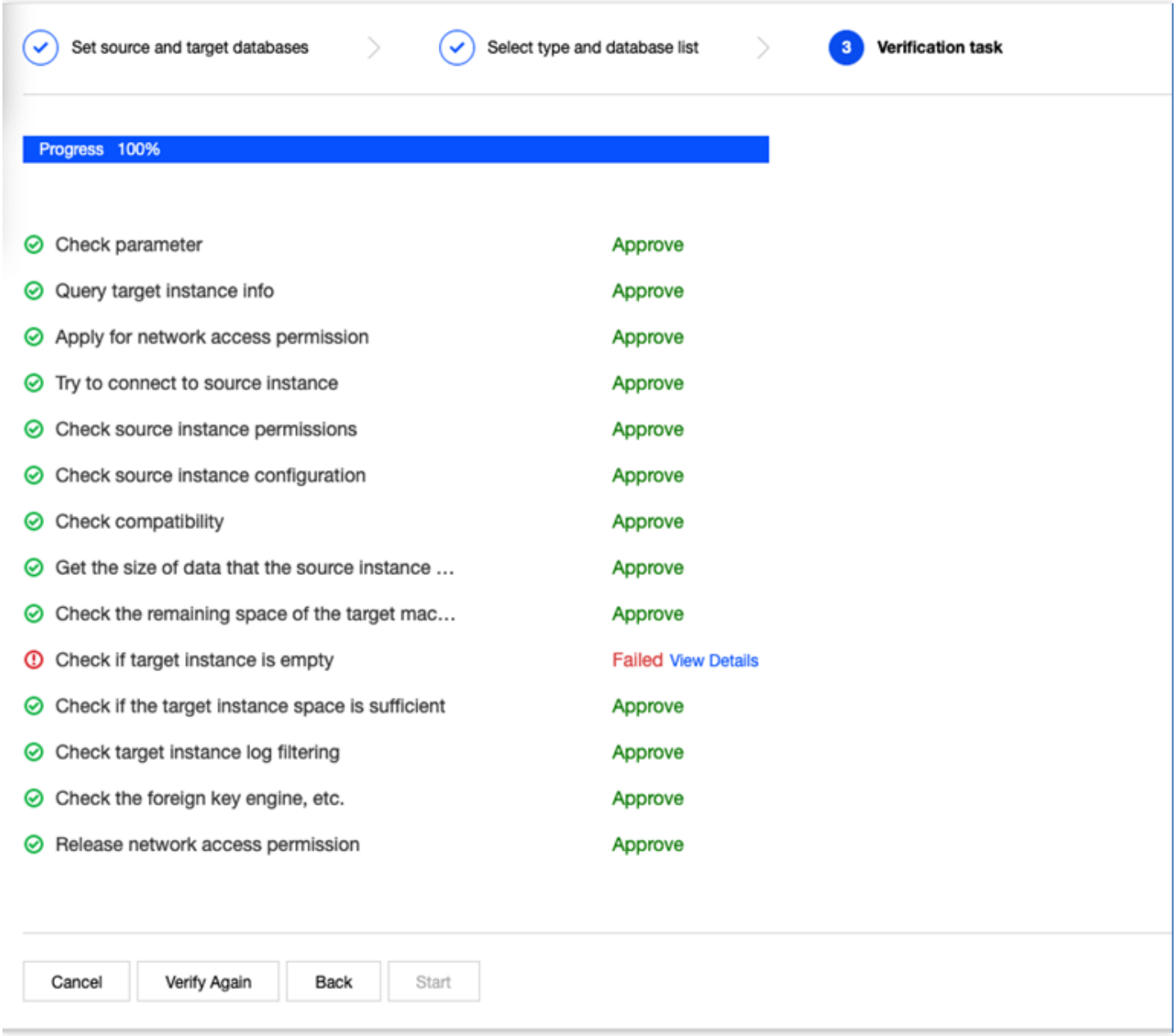
3. Selecting type and table

Select the type and table list and click **Next step: verify task**.



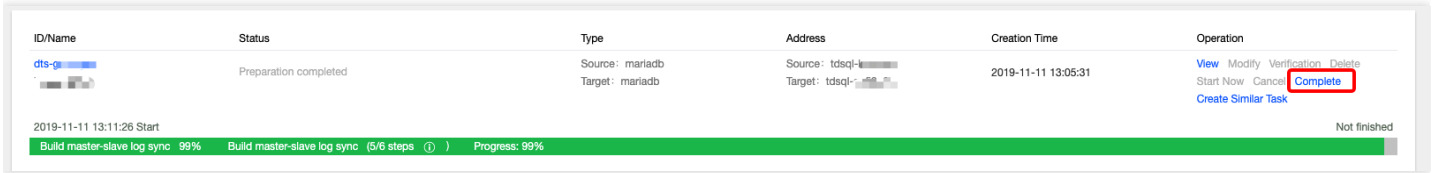
4. Checking the task

Check whether the source instance runs normally and whether the sets to be migrated to the target instance are in conflicts.

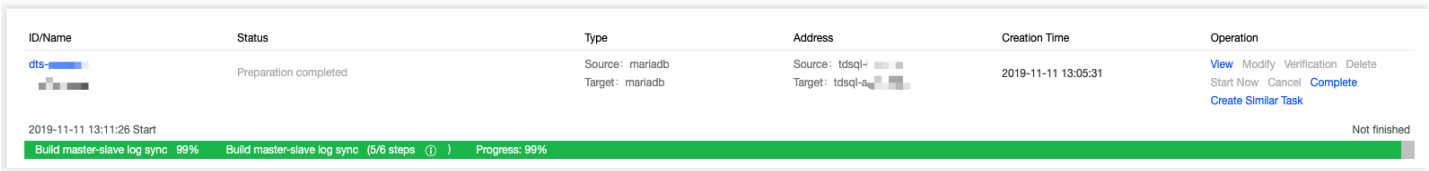


5. Completing migration

After the check is passed, return to the migration task list. After the incremental sync is 90% complete, click **Complete** on the right of the migration task.



The migration task is completed.



Data Subscription (Legacy) Database/Table Data Subscription to Local File System

Last updated : 2021-01-06 10:04:08

This document provides a simple example that walks you through how to pull a table from data subscription to a local file system as well as a simple Local Demo. The following operations are performed on CentOS.

Configuring Environment

- Java environment configuration

```
yum install java-1.8.0-openjdk-devel
```

- Download the data subscription SDK

Getting Key

Log in to the CAM console to get a key.

Selecting Data Subscription

1. Log in to the DTS console, and select **Data Subscription** on the left to go to the data subscription page.

2. Select the name of the TencentDB instance to be synced, click **Start**, return to the data subscription page, and click the data subscription you created. For detailed directions, please see How to Get a Data Subscription.

3. Check the corresponding DTS channel, IP, and port and enter them together with the obtained key into the corresponding `LocalDemo.java` file.

```
// Enter the key obtained from TencentCloud API here
context.setSecretId("AKIDfdsfdsfsdt1331431sdfds"); Enter the `secretID` obtained from TencentClou
```

```
d API
context.setSecretKey("test111usdfsdfsddsfRkeT"); Enter
the `secretKey` obtained from TencentCloud API
// Enter the IP and port obtained through data subscription in the DTS service here
context.setServiceIp("10.66.112.181"); Enter the IP obtained from the data subscription configura
tion
context.setServicePort(7507);
Enter the port obtained from the data subscription configuration
final DefaultSubscribeClient client = new DefaultSubscribeClient(context);
// Enter the names of both the database and table to sync and modify the name of the file where t
hey will be stored
final String targetDatabase = "test"; Enter the name of the database to subscribe to
final String targetTable = "alantest"; Enter the name of the table to subscribe to

final String fileName = "test.alan.txt"; Enter the name of the local file for storage

client.addClusterListener(listener);
// Enter the `dts-channel` configuration information obtained from the data subscription configur
ation here
client.askForGUID("dts-channel-e4FQxtYV3It4test"); Enter the DTS channel name obtained from the d
ata subscription
client.start();
```

Compiling and Testing

1.
```
javac -classpath binlogsdk-2.6.0-release.jar -encoding UTF-8 LocalDemo.java
```

2. Launch the program. If no errors are reported, the program works properly. Check the previously configured local file.

```
java -XX:-UseGCOverheadLimit -Xms2g -Xmx2g -classpath .:binlogsdk-2.6.0-release.jar LocalDemo
```

3. Check the previously configured `test.alan.txt` file and you can see that data has been pulled into it.

```
[root@VM_71_10_centos ~]# cat test.alan.txt
checkpoint:144251@3@3575890@317713
record_id:000001000000000004D9110000000000000001
record_encoding:utf8
fields_enc:latin1,utf8
gtid:4f21864b-3bed-11e8-a44c-5cb901896188:1562
source_category:full_recorded
source_type:mysql
table_name:alantest
record_type:INSERT
```

```
db:test
timestamp:1523356039
primary:
Field name: id
Field type: 3
Field length: 2
Field value: 26
Field name: name
Field type: 253
Field length: 4
Field value: alan
```

Database/Table Data Subscription to Redis

Last updated : 2020-04-27 17:04:09

This document provides a simple example that walks you through how to pull a table from data subscription to Redis as well as a simple Redis Demo. The following operations are performed on CentOS.

Configuring Environment

- Java environment configuration

```
yum install java-1.8.0-openjdk-devel
```

- Download the data subscription SDK
- Download jedis-2.9.0.jar

Getting Key

Log in to the CAM Console to get a key.

Selecting Data Subscription

1. Log in to the DTS Console and select **Data Subscription** on the left sidebar to enter the data subscription page.
2. Select the name of the TencentDB instance to be synced, click "Start", return to the data subscription page, and click the data subscription you created. For detailed directions, please see How to Get a Data Subscription.
3. Check the corresponding DTS channel, IP, and port and enter them together with the obtained key into the corresponding `RedisDemo.java` .

```
context.setSecretId("AKIDfdsfdsfsdt1331431sdfds"); Enter the `secretID` obtained from TencentCloud API.
context.setSecretKey("test111usdfsdfsddsfRkeT"); Enter the `secretKey` obtained from TencentCloud API.
// Enter the IP and port obtained through data subscription in the DTS service here
```

```
context.setServiceIp("10.66.112.181"); Enter the IP obtained from the data subscription configurati
on
context.setServicePort(7507); Enter the port obtained from the data subscription configuration

// Create a consumer
//SubscribeClient client=new DefaultSubscribeClient(context,true);
final DefaultSubscribeClient client = new DefaultSubscribeClient(context);

final Jedis jedis = new Jedis("127.0.0.1", 6379); Enter your corresponding Redis host and port

final String targetDatabase = "test"; Enter the name of the database to subscribe to
final String targetTable = "alantest"; Enter the name of the table to subscribe to. The table has t
wo fields, namely, `id` and `name`, of which `id` is the primary key

// Create a subscription listener
ClusterListener listener = new ClusterListener() {
@Override
public void notify(List<ClusterMessage> messages) throws Exception {
// System.out.println("--------------------:" + messages.size());
for(ClusterMessage m:messages){
DataMessage.Record record = m.getRecord();
// Filter out uninteresting subscription messages
if(!record.getDbName().equalsIgnoreCase(targetDatabase) || !record.getTablename().equalsIgnoreCase
(targetTable)){
// Note: Ack must be performed for uninteresting messages too
m.ackAsConsumed();
continue;
}

if(record.getOpt() != DataMessage.Record.Type.BEGIN && record.getOpt() != DataMessage.Record.Type.C
OMMIT){
List<DataMessage.Record.Field> fields = record.getFieldList();

//INSERT RECORD
//String pk = record.getPrimaryKeys();

if(record.getOpt() == DataMessage.Record.Type.INSERT){

String keyid="";
String value="";
for (DataMessage.Record.Field field : fields) {

// Get the `id` value as the primary key first, find the `name` column, and insert the correspondin
g values of `key` and `name` in Redis
if(field.getFieldname().equalsIgnoreCase("id")){
keyid=field.getValue();
continue;
}
if(field.getFieldname().equalsIgnoreCase("name")){
```

```
value=field.getValue();

}
jedis.set(keyid, value);
}

}
```

Compiling and Testing

1. ```
   [root@VM_71_10_centos ~]# javac -classpath binlogsdk-2.6.0-release.jar:jedis-2.9.0.jar -encoding UTF-8 RedisDemo.java
   ```
2. Launch the program. If no errors are reported, the program works properly. Check the previously configured local file.
   ```
   java -XX:-UseGCOverheadLimit -Xms2g -Xmx2g -classpath .:binlogsdk-2.6.0-release.jar:jedis-2.9.0.jar RedisDemo
   ```
3. Perform `INSERT` and `UPDATE` operations. If you find that the data has been indeed inserted and modified successfully in Redis, you can perform `DELETE` operations to delete the corresponding data from Redis.

```
MySQL [test]> insert into alantest values(1001,'alan1');
Query OK, 1 row affected (0.00 sec)

MySQL [test]> update alantest set name='alan2' where id=1001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

-----------------------
127.0.0.1:6379> get 1001
"alan2"


MySQL [test]> update alantest set name='alan3' where id=1001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

-----------------------
127.0.0.1:6379> get 1001
"alan3"

MySQL [test]> delete from alantest where id=1001;
```
```

```
Query OK, 1 row affected (0.00 sec)

----------------------

127.0.0.1:6379> get 1001
(nil)
```

Database/Table Data Subscription to Kafka

Last updated : 2021-01-06 10:04:08

This document provides a simple example that walks you through how to pull a table from data subscription to Kafka as well as a simple Kaflka Demo.

Configuring Environment

- OS: CentOS
- Download relevant resources
  - Data subscription SDK
  - SLF4J components
  - Kafka-clients
- Java environment configuration

```
yum install java-1.8.0-openjdk-devel
```

Installing Kafka

1. Please install Kafka as instructed in Kafka Quick Start.
2. After Kafka is launched, create a "testtop" topic.

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --t
opic testtop
Created topic ”testtop”.
```

Getting Key

Log in to the CAM console to get a key.

Selecting Data Subscription

1. Log in to the DTS console, and select **Data Subscription** on the left to go to the data subscription page.
2. In the subscription list, click a subscription name to enter the subscription details page and view the corresponding channel ID, service IP, and service port.
3. Enter them together with the obtained key into the corresponding `KafkaDemo.java` .

```
import com.qcloud.dts.context.SubscribeContext;
import com.qcloud.dts.message.ClusterMessage;
import com.qcloud.dts.message.DataMessage;
import com.qcloud.dts.subscribe.ClusterListener;
import com.qcloud.dts.subscribe.DefaultSubscribeClient;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.Producer;
import org.apache.kafka.clients.producer.ProducerConfig;
import org.apache.kafka.clients.producer.ProducerRecord;
import org.apache.kafka.common.serialization.StringSerializer;
import org.apache.log4j.Logger;

import java.util.List;
import java.util.Properties;

public class KafkaDemo {
public static void main(String[] args) throws Exception {
//Initialize a kafka producer
final String TOPIC = "testtop";
Properties props = new Properties();
props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, "10.168.1.6:9092");
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);

final Producer<String, String> producer = new KafkaProducer<String, String>(props);

//Create a context
SubscribeContext context = new SubscribeContext();
context.setSecretId("AKIDPko5fVtvTDE0WffffkCwd4NzKcdePt79uauy");
context.setSecretKey("ECtY8F5e2QqtdXAe18yX0EBqK");
// Subscription channel region
context.setRegion("ap-beijing");
final DefaultSubscribeClient client = new DefaultSubscribeClient(context);

// Create a subscription listener
ClusterListener listener = new ClusterListener() {
@Override
public void notify(List<ClusterMessage> messages) throws Exception {
System.out.println("--------------------:" + messages.size());
for(ClusterMessage m:messages){
```

```
DataMessage.Record record = m.getRecord();


if(record.getOpt() != DataMessage.Record.Type.BEGIN && record.getOpt() != DataMessage.Record.Type.C
OMMIT){
List<DataMessage.Record.Field> fields = record.getFieldList();

//Print the information of each column
for (int i = 0; i < fields.size(); i++) {
DataMessage.Record.Field field = fields.get(i);
System.out.println("Database Name:" + record.getDbName());
System.out.println("Table Name:" + record.getTablename());
System.out.println("Field Value:" + field.getValue());
System.out.println("Field Value:" + field.getValue().length());
System.out.println("Field Encoding:" + field.getFieldEnc());
}

//Send the entire record to the specified Kafka topic
System.out.println("Record++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");
producer.send(new ProducerRecord<String, String>(TOPIC, record.toString()));
}

m.ackAsConsumed();
}
}
@Override
public void onException(Exception e){
System.out.println("listen exception" + e);
}
};
// Add a listener
client.addClusterListener(listener);
client.askForGUID("dts-channel-p15e9eW9rn8hA68K");
client.start();
}

}
```

Compiling and Testing

1. Compile the client program `KafkaDemo.java`.

```
javac -classpath binlogsdk-2.9.1-jar-with-dependencies.jar:slf4j-api-1.7.25.jar:slf4j-log4j12-1.7
.2.jar:kafka-clients-1.1.0.jar -encoding UTF-8 KafkaDemo.java
```

2. Launch the program. If no errors are reported, the program works properly.

```
java -XX:-UseGCOverheadLimit -Xms2g -Xmx2g -classpath .:binlogsdk-2.9.1-jar-with-dependencies.ja
r:kafka-clients-1.1.0.jar:slf4j-api-1.7.25.jar:slf4j-log4j12-1.7.2.jar KafkaDemo
```

3. Insert a data entry into the `alantest` table, and you will find that the data has been stored in the `testtop` subscribed to by Kafka.

```
MySQL [test]> insert into alantest values(123456,'alan');
Query OK, 1 row affected (0.02 sec)
[root@VM_71_10_centos kafka_2.11-1.1.0]# bin/kafka-console-consumer.sh --bootstrap-server localho
st:9092 --topic testtop --from-beginning
checkpoint:144251@3@1275254@1153089
record_id:000001000000000011984100000000000000001
record_encoding:utf8
fields_enc:latin1,utf8
gtid:4f21864b-3bed-11e8-a44c-5cb901896188:5552
source_category:full_recorded
source_type:mysql
table_name:alantest
record_type:INSERT
db:test
timestamp:1524649133
primary:id
Field name: id
Field type: 3
Field length: 6
Field value: 123456
Field name: name
Field type: 253
Field length: 4
Field value: alan
```

Data Subscription Kafka Edition Adding Consumer Group

Last updated : 2021-02-22 10:39:56

Data subscription Kafka Edition allows users to create multiple consumer groups for multi-point consumption. By creating multiple consumption groups, you can consume for multiple times and increase consumption channels to reduce usage costs and improve the data consumption speed.

Prerequisites

- You have created the Data Subscription Kafka Edition.

> **Note :**
>
> In the data subscription list, the subscription with the "Kafka Edition" tag is the data subscription in Kafka edition.

Notes

- The data subscription task must be in the running status.
- A data subscription task can contain up to 10 consumer groups.
- A consumer group can only have one consumer for consumption.

Directions

1. Log in to the DTS console, select **Data Subscription** on the left sidebar to go to the **Data Subscription** page.
2. In the data subscription list, select a data subscription and click its name or **View Subscription Details** in the **Operation** column to go to the subscription management page.
3. In the subscription management page, select the **Consumption Management** tab and click **Create Consumer Group**.
4. In the pop-up window, set the consumer group information and click **Create**.
   - Consumer Group Name: set the consumer group name as needed.
   - Account: set the consumer group account.
   - Password: set the password of the consumer group account.

- Confirm Password: enter the same password again.
- Remarks: set the remarks to record the information.

Managing Consumer Group

Last updated : 2021-02-22 10:05:17

Data subscription Kafka Edition supports the management of consumer groups, including password modification and consumer group deletion. This document describes how to modify the consumer group password and delete the consumer group in the console.

Prerequisites

- You have created a consumer group.

Modifying the Password of Consumer Group

1. Log in to the DTS console, select **Data Subscription** on the left sidebar to go to the **Data Subscription** page.
2. In the data subscription list, select a data subscription and click its name or **View Subscription Details** in the **Operation** column to go to the subscription management page.
3. Select the **Consumption Management** tab and click **Change Password** in the **Operation** column.
4. In the pop-up window, modify the password and click **Modify**.

Deleting the Consumer Group

1. Log in to the DTS console, select **Data Subscription** on the left sidebar to go to the **Data Subscription** page.
2. In the data subscription list, select a data subscription and click its name or **View Subscription Details** in the **Operation** column to go to the subscription management page.
3. Select the **Consumption Management** tab and click **Delete** in the **Operation** column.
4. In the pop-up dialog box, confirm that everything is correct and click **Confirm**.

> ⚠ **Note :**
>
> After the consumer group is deleted, the consumption offset in the consumer group will be deleted, but the data in the data subscription will not be deleted.

Modifying Subscribed Object

Last updated : 2021-02-22 10:05:17

Data subscription Kafka edition supports dynamic increase or decrease of the subscribed objects during data consumption. This document describes how to modify the subscribed object of the data subscription Kafka edition in the console.

Prerequisites

- You have created the Data Subscription Kafka Edition.

Directions

1. Log in to the DTS console, select **Data Subscription** on the left sidebar to go to the **Data Subscription** page.
2. In the data subscription list, select a data subscription and select **More** > **Modify subscribed object** in the **Operation** column to go to the **Configure data subscription** page.
3. In the **Configure data subscription** page, select the subscription type, edit the subscribed object, and click **Save**.
4. Return to the subscription list, the subscription instance enters the “enabling” state, and the task is pre-checked and initialized. After being enabled, the subscription instance enters the running state, and the Kafka client can be used to consume the subscription data.

Resetting Data Subscription

Last updated : 2021-02-22 10:05:17

Data subscription Kafka edition supports resetting the subscription task to remove the information and data of the current subscription instance, and reconfiguring the data subscription task. This document describes how to reset the data subscription in the console.

Prerequisites

- You have created the Data Subscription Kafka Edition.
- The state of the data subscription is “running” or “abnormal”.

Directions

1. Log in to the DTS console, select **Data Subscription** on the left sidebar to go to the **Data Subscription** page.
2. In the data subscription list, select a data subscription and select **More** > **Reset subscription** in the **Operation** column.
3. In the pop-up dialog box, confirm that everything is correct and click **Confirm**.

> ⚠ **Note :**
>
> - After the subscription is reset, the binding relationship between the subscription instance and the source instance will be unassociated, and the state of the instance will become “not started”. You can perform the initialization configuration again.
> - The subscription to the incremental data of the source database will be stopped once the subscription is reset, and the incremental data stored in the subscription will be deleted.

4. In the data subscription list, click **Configure Subscription** in the **Operation** column to reconfigure the subscription task.

Data Consumption Demo

Last updated : 2021-02-22 10:05:17

In data subscription Kafka edition, you can directly consume subscription data through Kafka client version 0.11 and above. This document provides two client consumption demos in Java and Go languages.

Downloading Consumption Demo

Refer to the following table to download the consumption Demo code of data subscription Kafka edition client:

Demo Language	Download Address
Go	Address
Java	Address

Configuration Parameters

Parameter	Description
brokerlist	The private network access address of Data Subscription Kafka Edition
topic	The subscription topic of the data subscription channel
group	Consumer group name
user	Consumer group account name
password	Consumer group password