

Data Transfer Service Fix for Verification Failure 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

STencent 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

Fix for Verification Failure

Check Item Overview

Check Items of MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL

- Version Check
- Source Instance Permission Check
- Partial Database Parameter Check
- Target Database Content Conflict Check
- Target Database Space Check
- **Binlog Parameter Check**

MongoDB Check Items

- MongoDB Connection Check
- Database/Table Content Conflict Check
- Source/Target Database Account Permission Check
- Database Version Check
- Database Capacity Check

PostgreSQL Check Items

- Source/Target Database Connectivity Check
- Source Instance Version Check
- Structure Conflict Check
- Target Database Existence Check
- Primary Key Dependency Check
- Key Instance Parameter Check
- Read-Only Database Check for Target Instance
- Multi-Task Conflict Check

TDSQL for PostgreSQL

- Database Connection Check
- Version Check
- **Peripheral Check**
- Source Instance Permission Check
- Constraint Check
- Redis/Tendis
 - Network Connectivity
 - Source-Target Instance Version Compatibility
 - Source Instance Parameter Check
 - Target Instance Capacity Check

Whether Target Instance Is Empty

Target Instance Status Check

SQL Server

Migration Parameter Check

Source/Target Database Connectivity Check

Database Version Check

User Permission Check

Database Connection Check

Source Database Existence Check

Target Database Existence Check

Peripheral Check

Migration Network Check

Version Check

Source Instance Permission Check

Account Conflict Check

Partial Database Parameter Check

Source Instance Parameter Check

Source Instance Type (Master or Replica)

Parameter Configuration Conflict Check

Target Database Content Conflict Check

Target Database Space Check

Target Database Load Check

Local Disk Space Check

Binlog Parameter Check

Oplog Check

Foreign Key Dependency Check

View Check

Advanced Object Check

Incremental Migration Precondition Check

Extension/Plugin Compatibility Check

Source Database Balancer Check

Source Database Node Role Check

ShardKey Check

Warning Item Check

DLL Ring Sync Check for Single Database/Table Object

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

Level-2 Subpartitioned Table Check

Primary Key Check

DDL Check for Tables to Be Migrated

System Database Conflict Check

Table Structure Check for Source and Target Instances

InnoDB Table Check

Migration Object Dependency Check

Constraint Check

Fix for Verification Failure Check Item Overview

Last updated : 2021-11-16 18:14:36

This document lists the check items for database migration, sync, and subscription tasks. If an error is reported for a check item, fix it as instructed in the corresponding document.

MySQL/TDSQL-C

- Database connection check
- Peripheral check
- Version check
- Partial database parameter check
- Target database content conflict check
- Target database space check
- Binlog parameter check
- Foreign key dependency check
- View check
- Warning item check
- RDS check

TDSQL for MySQL

- Database connection check
- Peripheral check
- Version check
- Partial database parameter check
- Target database content conflict check
- Target database space check
- Binlog parameter check
- Foreign key dependency check
- Warning item check

PostgreSQL

- Source database connectivity check
- Source database version check
- Target database connectivity check
- Account conflict check
- Parameter configuration conflict check
- Structure conflict check
- Extension/Plugin compatibility check
- Incremental migration precondition check

MongoDB

- MongoDB connection check
- Database/Table content conflict check
- Source database node role check
- Oplog check
- Database version check
- Database capacity check
- Target database load check
- Shardkey check
- Source database balancer check

SQL Server

- Migration parameter check
- Migration network check
- Source database connectivity check
- Target database connectivity check
- Disk space check
- Database version check
- Source database existence check
- Target database existence check

Check Items of MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL Version Check

Last updated : 2023-08-25 16:55:44

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 : 2023-08-25 16:58:15

Check Details

Check whether you have the operation permissions of the database by referring to the following: Permission requirements for data migration Migration from MySQL to TencentDB for MySQL Migration from TDSQL for MySQL to TDSQL for MySQL Permission requirements for data sync Sync from MySQL/MariaDB/Percona to TencentDB for MySQL Sync from TDSQL for MySQL to TDSQL for MySQL Permission requirements for data subscription Creating MySQL or TDSQL for MySQL 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 : 2023-08-25 16:57:08

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 <u>row_format</u> in a database table is <u>FIXED</u>, 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 <u>row_format</u> to <u>DYNAMIC</u> 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:





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 <code>lower_case_table_names</code> 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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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 : 2023-08-25 17:00:01

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 : 2023-08-25 16:21:23

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 : 2023-08-25 16:23:03

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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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:





<pre>mysql> show variables like '%ONGOING_#</pre>	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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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 maste	er status;	+		
File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
+	+ 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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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.

MongoDB Check Items MongoDB Connection Check

Last updated : 2023-08-25 16:26:32

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database.

The source IP addresses are blocked by the source database. For more information, see Failed Connectivity Test > Source IP Addresses Blocked in Source Database.

The network port is closed. For more information, see Failed Connectivity Test > Closed Network Port. The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Database/Table Content Conflict Check

Last updated : 2023-08-25 16:29:29

Check Details

In MongoDB data migration scenarios, the target database can contain collections with the same name as those in the source database, but the collections must be empty.

Troubleshooting

If a conflict causes an error, you can delete collections with the same name in the target database or clear their data.

Source/Target Database Account Permission Check

Last updated : 2023-08-25 16:28:36

Check Details

Check whether you have the operation permissions of the database as described below: Permission requirements for data migration: Migration from MongoDB to TencentDB for MongoDB.

Troubleshooting

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

Database Version Check

Last updated : 2023-08-25 16:30:12

The source and target database versions must be supported by MongoDB.

Database Capacity Check

Last updated : 2023-08-25 16:25:05

Check Details

In MongoDB data migration scenarios, the storage space of the target database needs to be at least 1.3 times the size of the collections to be migrated in the source database.

Troubleshooting

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

Upgrade the storage specification of the target database to use an instance with a larger capacity for migration.

PostgreSQL Check Items Source/Target Database Connectivity Check

Last updated : 2023-08-25 16:33:48

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database.

The source IP addresses are blocked by the source database. For more information, see Failed Connectivity Test > Source IP Addresses Blocked in Source Database.

The network port is closed. For more information, see Failed Connectivity Test > Closed Network Port. The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Source Instance Version Check

Last updated : 2023-08-25 16:34:36

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.

Structure Conflict Check

Last updated : 2023-08-25 16:40:54

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.

Target Database Existence Check

Last updated : 2023-08-25 16:37:41

Check Details

Check whether there is a target database, and if not, an error will be reported.

Troubleshooting

Do not delete the target database during migration; otherwise, you need to create a migration task again.

Primary Key Dependency Check

Last updated : 2023-08-25 16:38:25

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.

Key Instance Parameter Check

Last updated : 2023-08-25 16:39:14

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.

Read-Only Database Check for Target Instance

Last updated : 2023-08-25 16:39:58

Check Details

When configuring a PostgreSQL sync task, you need to check whether the target instance is read-only, and if so, data cannot be synced and the task will report an error.

Troubleshooting

Do not select a read-only instance as the target instance when configuring a sync task.

Multi-Task Conflict Check

Last updated : 2023-08-25 16:41:29

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.
TDSQL for PostgreSQL Database Connection Check

Last updated : 2023-08-25 16:51:50

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.

Version Check

Last updated : 2023-08-25 16:53:12

Check Details

Log in to the database, run select tbase_version(); , and Tbase_V2.xx must be returned.

Troubleshooting

Select the ID of the TDSQL for PostgreSQL instance on the supported version.

Peripheral Check

Last updated : 2023-08-25 16:45:59

Check Details

The wal_level of the DN node must be logical .

Troubleshooting

1. Log in to the source database.





`wal_level` = `logical`

3. After the modification is completed, restart the database.

Source Instance Permission Check

Last updated : 2023-07-19 17:04:58

Check Details

You must have the REPLICATION permission, that is, pg_roles.rolreplication .

You must have the SELECT permission of the subscribed table. For entire database subscription, you must have the SELECT permission of all the tables under the schema.

For more information, see Creating TDSQL for PostgreSQL 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.

Constraint Check

Last updated : 2023-08-25 16:46:52

Check Details

The subscribed table must have a primary key or have REPLICA IDENTITY as FULL.

Troubleshooting

If the verification fails, modify the corresponding table.

Redis/Tendis Network Connectivity

Last updated : 2023-07-06 15:42:34

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database.

The source IP addresses are blocked by the source database. For more information, see Failed Connectivity Test > Source IP Addresses Blocked in Source Database.

The network port is closed. For more information, see Failed Connectivity Test > Closed Network Port. The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Source-Target Instance Version Compatibility

Last updated : 2023-07-06 15:41:28

Check Details

The migration from Redis to TencentDB for Redis is supported for source Redis 2.2.6 or later. The migration from Redis to TencentDB for KeeWiDB is supported for source Redis 4.0. To migrate to other versions, submit a ticket for application.

The target database version must be later than or equal to the source database version; otherwise, an alarm will be triggered for compatibility problems during verification.

The target database must have the latest proxy.

Troubleshooting

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

Source Instance Parameter Check

Last updated : 2023-08-25 16:54:11

Check Details

In Redis migration scenarios, if the target database is TencentDB for Redis, the number of databases in the source instance must be not greater than that in the target instance. Check whether the status of the source instance is normal.

Troubleshooting

Modify the number of databases of the source and target instances and restart the verification task. Confirm the status of the source instance.

Target Instance Capacity Check

Last updated : 2023-07-06 15:39:54

Check Details

For migration from Redis to TencentDB for Redis/KeeWiDB, the space of the target database must be at least 1.5 or 2 times the volume of the data to be migrated in the source database, respectively.

Troubleshooting

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

Upgrade the storage specification of the target database to use an instance with a larger capacity for migration.

Whether Target Instance Is Empty

Last updated : 2023-07-06 15:39:01

Check Details

For Redis migration, the target instance must be empty; otherwise, an error will be reported.

Troubleshooting

If an error is reported, delete the content in the target instance and restart the verification task.

Target Instance Status Check

Last updated : 2023-07-06 15:38:05

Check Details

Check whether the target database exists, and if not, an error will be reported.

Troubleshooting

Do not delete the target database during migration; otherwise, you need to create a migration task again.

SQL Server Migration Parameter Check

Last updated : 2023-08-25 16:42:46

Check Details

The target instance cannot have databases with the same name as those in the source database; otherwise, an error will be reported.

Troubleshooting

If an error is reported, you need to delete the database with the same name from the target instance or rename it.

Source/Target Database Connectivity Check

Last updated : 2023-07-06 15:32:33

Check Details

The source and target databases need to be normally connected, and if not, a connection failure will be reported.

Causes

The network or server where the source database resides has a security group or firewall configured. For more information, see Failed Connectivity Test > Security Group or Firewall Configured in Network or Server of Source Database

The source IP addresses are blocked by the source database. For more information, see Failed Connectivity Test > Source IP Addresses Blocked in Source Database.

The network port is closed. For more information, see Failed Connectivity Test > Closed Network Port. The database account or password is incorrect.

Troubleshooting

Refer to the causes above based on the actual scenario and troubleshoot as instructed.

Database Version Check

Last updated : 2023-07-06 15:33:15

Only migration from Basic Edition to High Availability Edition (including Dual-Server High Availability Edition and Cluster Edition) is supported, and the version number of the target database must be later than that of the source database. Cross-major version migration is not supported.

User Permission Check

Last updated : 2023-07-06 15:32:00

Check Details

Check whether you have the operation permissions of the database as instructed in Migration from SQL Server to TencentDB for SQL Server.

Troubleshooting

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

Database Connection Check

Last updated : 2021-11-15 15:02:07

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.
 - i. Log in to the corresponding database and click an instance ID in the instance list to enter the instance management page.
 - ii. 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								
Private IP	1								
i invato ii									
Private Port	3306								
Public Network Add	ess Enable								
Public Network Add									
Added to securi	ty group								
Edit Co	nfigure Security Group								
Priority		Security G			Security Group Name		Operat	ion	
Priority		Security G	roup ID		Security Group Name	2	Operat	ion	
Priority 1		Security G	roup ID		Security Group Name		Operat	ion	
Priority 1		Security G	roup ID		Security Group Name	2	Operat	ion	
Priority 1		Security G	roup ID		Security Group Name	2	Operat	ion	
Priority 1		Security G	roup ID		Security Group Name		Operat	ion	
Priority 1 Rule Preview		Security G	roup ID		Security Group Name		Operat	ion	
Priority 1 Rule Preview		Security G	roup ID		Security Group Name		Operat	ion	
Priority 1 Rule Preview Inbound Rules	Outbound Rules	Security G	roup ID		Security Group Name		Operat	ion	
Priority 1 Rule Preview Inbound Rules	Outbound Rules	Security G	roup ID		Security Group Name		Operat	ion	
Priority 1 Rule Preview Inbound Rules 1	Outbound Rules	Security G	roup ID		Security Group Name	5	Operat	ion	*
Priority 1 Rule Preview Inbound Rules 1	Outbound Rules	Security G	roup ID		Security Group Name		Operat	ion	•
Priority 1 Rule Preview Inbound Rules 1 Source	Outbound Rules	Security G	roup ID		Security Group Name		Operat	ks	•
Priority 1 Rule Preview Inbound Rules Source	Outbound Rules	Security G	roup ID		Security Group Name		Operat	ks	
Priority 1 Rule Preview Inbound Rules Source	Outbound Rules	Security G	roup ID		Security Group Name		Operat Remar 	ks	•

Fix

Fix it as follows based on the actual conditions:

- The firewall is enabled on the server:
 - i. Disable the server firewall, log in to DTS, and run the verification task again.

Note :

This method is applicable to both Windows and Linux.

- ii. Set the DTS IP range policy to Allow.
- The SNAT IP range of DTS is blocked in the security group:
 - i. 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@10.0.0/8 // Authorize users to access through `10.0.0.0/8`, and other IP
s will be blocked (incorrect configuration)
root@% // Authorize users to access all IPs, which should include the SNAT IPs
(correct configuration)

You can verify as follows:

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

SQL Server

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

PostgreSQL

- If the source database is another database in the cloud, check whether the secure access policies in the source database have restrictions. Check as follows according to the specific cloud vendor:
- If the source database is a self-built PostgreSQL database, enter the data directory in the \$PGDATA directory, find the pg_hba.conf file, and check whether the file contains a deny policy or only allows access from certain IP addresses over the network.

```
# cat pg_hba.conf
local replication all trust
host replication all 127.x.x.1/32 trust
host replication all ::1/128 trust
```

```
host all all 0.0.0/0 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 th
e database account entered in the data migration task
mysql> flush privileges;
```

- 2. For a self-built database, if the bind-address configuration is incorrect, modify it as instructed below.
 - 2.1. Add the following content to the /etc/my.cnf file:

Note :

The default path of the <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code> , 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.

Source Database Existence Check

Last updated : 2021-11-15 16:08:05

Check Details

Check whether the source database to be migrated exists, and if not, an error will be reported.

Fix

Do not delete the source database during migration; otherwise, you need to create a migration task again.

Target Database Existence Check

Last updated : 2021-11-15 16:08:05

Check Details

Check whether the target database exists, and if not, an error will be reported.

Fix

Do not delete the target database during migration; otherwise, you need to create a migration task again.

Peripheral Check

Last updated : 2022-08-24 16:28:42

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.

TDSQL for PostgreSQL check details

The wal_level of the DN node must be logical.

Troubleshooting

- 1. Log in to the source database.
- 2. Change the value of <code>innodb_stats_on_metadata</code> to <code>OFF</code> .

```
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.

Migration Network Check

Last updated : 2021-11-15 16:08:05

Check Details

Check whether the internal migration network is connected.

Fix

If an error is reported during the migration network check, submit a ticket for assistance.

Version Check

Last updated : 2023-02-13 10:44:39

MySQL/TDSQL for MySQL 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. 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.

PostgreSQL check details

- 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 versions are different, some special compatibility issues may arise, and a warning will appear during
 migration. Read the compatibility report for each version to check whether your business is using any incompatible
 features.

TDSQL for PostgreSQL check details

Log in to the database, run select tbase_version(); , and Tbase_V2.xx must be returned.

MongoDB check details

The source and target database versions must be supported by MongoDB.

SQL Server check details

Only migration from Basic Edition to High Availability Edition (including Dual-Server High Availability Edition and Cluster Edition) is supported, and the version number of the target instance must be later than that of the source instance.

Redis check details

- The source Redis instance must be on v2.2.6 or later. Earlier versions don't support migration through DTS.
- The target database version must be later than or equal to the source database version; otherwise, an alarm will be triggered for compatibility problems during verification.
- The proxy of the target Redis database must be on the latest version.

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 instance version or use a database instance on a later version.

Source Instance Permission Check

Last updated : 2022-09-16 10:07:13

Check details

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

- Permission requirements for data migration
 - Migration from MySQL to TencentDB for MySQL
 - Migration from PostgreSQL to TencentDB for PostgreSQL
 - Migration from MongoDB to TencentDB for MongoDB
 - Migration from SQL Server to TencentDB for SQL Server
- Permission requirements for data sync
 - Sync from TDSQL for MySQL to TDSQL for MySQL

TDSQL for PostgreSQL check details

You must have the REPLICATION permission, i.e., $pg_roles.rolreplication$.

You must have the SELECT permission of the subscribed table. For entire database subscription, you must have the SELECT permission of all the tables under the schema.

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.

Account Conflict Check

Last updated : 2021-11-15 15:55:34

Check Details

Check whether the target database user conflicts with the source database user.

Fix

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 statemen
t to modify the owner of the resources first (with a table as an example belo
w):
alter table table name owner to new user;
```

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

Partial Database Parameter Check

Last updated : 2021-11-15 15:15:23

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

- row_format in the source database table cannot be FIXED .
- The lower_case_table_names variable in 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, the warning "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" will be reported.

Fix

Modifying row_format parameter in source database

If the value of row_format in the database table is FIXED, data overflow will occur if the storage length of each row in the table exceeds the limit, and an error will be reported. Therefore, you need to modify it to another format such as DYNAMIC to make the storage length of each row vary by 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 '%row_format%';

The system should display a result similar to the following:

```
mysql> show table status like '%row_format%';
```

©2013-2022 Tencent Cloud. All rights reserved.



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 lower_case_table_names parameter.

```
alter global lower_case_table_names = 1
```

4. Run the following command to restart the databases:

```
[\$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 should 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)

I IOW III SEL (0.00 SEC)

6. Run the verification task again.

Modifying max_allowed_packet parameter in 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 inability to capture the SQL statements of large exception event packets. If its value is too small, program errors mat occur, causing backup failure 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 = 4M

3. Check whether the configuration takes effect.

show global variables like '%max_allowed_packet%';

The system should 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 connect_timeout variable in source database

connect_timeout is the database connection time. Connection requests after connect_timeout elapses will be rejected. If this value is too small, database connections will be closed frequently, affecting the processing efficiency. Therefore, we recommend you set a value above 10.

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 parameter is successfully modified.

show global variables like '%connect_timeout%';

The system should 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.

Source Instance Parameter Check

Last updated : 2022-10-17 18:15:04

Check details

- In Redis migration scenarios, if the target database is TencentDB for Redis, the number of databases in the source instance must be not greater than or equal to that in the target instance.
- Check whether the status of the source instance is normal.

Troubleshooting

- Modify the number of databases of the source and target instances and restart the verification task.
- Confirm the status of the source instance.
Source Instance Type (Master or Replica)

Last updated : 2023-07-06 15:34:37

Check Details

The source database must be a replica node; otherwise, an alarm will be triggered. The alarm will not stop the task and can be ignored, but you should assess the impact of ignoring it. During migration, DTS will perform a BGSAVE operation on the source database, which will use the database

memory and resources. If the source database is a master node, business writes will be greatly affected. If the source database is a TencentDB for Redis database, a replica node will be used for migration by default. If the source database is a local self-built Redis database, the master node may be used for migration. If an alarm is triggered during the check, we recommend that you change the source database to a replica node.

Troubleshooting

Reconfigure the migration task by changing the parameters of the source database to the information of a replica node.

Parameter Configuration Conflict Check

Last updated : 2021-11-15 15:55:33

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.
 - Ic_monetary: sets the instance currency mode. If this parameter has different values, currency numbers may be incorrect after migration.
 - Ic_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';
```

Target Database Content Conflict Check

Last updated : 2022-09-21 18:04:04

Check Details

MySQL/MariaDB/Percona/TDSQL-C for MySQL/TDSQL for MySQL check requirements

- The target instance cannot have objects with the same name as those in the source database. If a conflict causes an error, you can fix it in one 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.
- During entire instance migration, 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 have conflicting advanced objects. If a conflict causes an error, you need to delete the conflicting objects.

PostgreSQL check requirements

- The target instance cannot have objects with the same name as those in the source database.
- During entire instance migration, the target instance must be empty. If a conflict causes an error, you need to delete the instance content.

MongoDB check requirements

The target instance can have tables with the same name as those in the source database, but they must be empty tables. If a conflict causes an error, you can fix it in one of the following methods:

- Option 1: Delete tables with the same name in the target database or clear their data.
- Option 2: Remove objects with the same name from the migration objects.

SQL Server check requirements

The target instance cannot have databases with the same name as those in the source database; otherwise, an error will be reported.

If an error is reported, you need to delete or rename databases with the same name from the target database.

Redis check requirements

The target instance must be empty; otherwise, an error will be reported.

If an error is reported, delete the content in the target instance and restart the verification task.

Troubleshooting

Using 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 corresponding 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.

ligration Type *	Structural migration Full migration Full + Incremental migration							
ata Consistency Check 🚯 🔹	Full check No Check							
	If the source instance is read-only, data consistency check will be skipped after data	a migration. Fo	or details, see Migration FAQs 🛂.					
Aligration Object () +	Entire instance Specify object							
	O Up to 1000 search results of source database objects are displayed by default. To view more objects, please click the "More" button or specify the object name for targeted search.							
	Source Database Object		Selected Object 🕕	Batch Renar				
	Search database name, supporting fuzzy match	Q,	Globally search for original object names, with fuzzy	match supported				
	1 database in total, with 1 displayed More		▶ 📃 🗄 db-dst	C				
	▼ 🔄 🗄 db-dst							
	► <mark></mark> Tables							
	▶ Views	+						
		4						

3. Run the verification task again.

Modifying objects with the same name in target database

Log in to the target database, 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 corresponding 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 the 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 : 2022-09-21 18:04:04

Check Details

MySQL/TDSQL-C for MySQL/TDSQL for MySQL/PostgreSQL check requirements

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 size of the tables in the target database may be larger than that in the source instance.

MongoDB check requirements

The storage space of the target database must be at least 1.3 times the size of the tables to be migrated in the source database.

Redis check requirements

The storage space of the target database must be at least 1.5 times that of the source database.

Troubleshooting

- Delete some data from the target database to free up sufficient space.
- Upgrade the storage specification of the target database as instructed in Adjusting Database Instance Specification to use an instance with a larger capacity for migration.

Target Database Load Check

Last updated : 2021-11-15 16:02:04

Check Details

- Check requirements: DTS migration will increase the load in the target database. If there is a business running in the target database during migration, a verification warning will be triggered. It will not block the task but will affect the business. You need to assess and determine whether to ignore the warning.
- Impact on business: MongoDB DTS uses logical sync for data migration, which will cause certain pressure on the CPU load of the target database. If there is a business running in the target database, you need to assess and initiate the migration task with caution.

Fix

Stop any business running in the target database and run the verification task again.

Local Disk Space Check

Last updated : 2021-11-15 16:02:04

Check Details

Check whether the disk space on the server of the source database is sufficient. During migration of a self-built source database, at least 50 GB disk space should be reserved on the server of the source database. We recommend you reserve a space at least 1.5 times the size of the migration data to avoid errors during migration.

Fix

Delete some data from the server of the source database to free up sufficient space.

Binlog Parameter Check

Last updated : 2022-09-27 14:27:59

MySQL/TDSQL for MySQL/TDSQL-C 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 be set to only ON or OFF. We recommend 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 instance, the <code>log_slave_updates</code> 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.

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. As the database needs to be restarted after the log_bin parameter is modified, we recommend you also modify the binlog_format and binlog_row_image parameters based on the verification requirements.

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
binlog_row_image = FULL
```

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 should display a result similar to the following:

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 master. When replicating data, the replica will run the same SQL statements as those in the master. This format can reduce the binlog size. However, the replica may not be able to properly replicate certain functions.
- ROW : The binlog will log the modifications of each data row, and the replica will modify the same data. This format guarantees the correct source-replica 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 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 take effect only after all connections to the database are reset. If the source database is a replica, you also need to restart the source/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 to make the configuration take effect. Then, run the following command to check whether the parameter modification takes effect:

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

The system should 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)
```

5. 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 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 column data information of the pre-image and post-image.
- 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, fix it as follows:

Note:

Changes to this parameter can take effect only after all connections to the database are reset. If the source database is a replica, you also need to restart the source/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 to make the configuration take effect. Then, run the following command to check whether the parameter modification takes effect:

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

The system should 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 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. DTS only allows you to set gtid_mode to ON or OFF. We recommend 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, during incremental data sync, if HA switch occurs in the source instance, 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 recovered.

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 master database and all transactions in the replica database must be anonymous.
- OFF_PERMISSIVE : All new transactions in the master database must be anonymous. Transactions in the replica database can be anonymous or GTID transactions but cannot be only GTID transactions.
- ON_PERMISSIVE : All new transactions in the master database must be GTID transactions, and transactions in the replica database can be anonymous or GTID transactions.
- ON : All new transactions in the master database and all transactions in the replica 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 both the source 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 with no need to restart the database, but you must reset all business connections.

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

3. Set gtid_mode = ON_PERMISSIVE on both the source 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 should 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 both the source 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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code>, 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 set to 0. The system default value of this parameter is 1, but the configuration isn't necessarily correct even if the queried parameter value is 1. You still need to manually set it.

1. Log in to the source database.

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

```
set global server_id = 2; // We recommend you set it to an integer above 1. The
value here is only an example.
```

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

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

The system should 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 replication will be abnormal; therefore, we recommend you not set them. 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 <code>my.cnf</code> configuration file is <code>/etc/my.cnf</code>, 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 should display a result similar to the following:

5. Run the verification task again.

Modifying log_slave_updates parameter

In the source-replica reuse structure, if the <u>log-bin</u> parameter is enabled in the replica database, data operations directly performed in the replica database can be logged in the binlog, but data replications from the source database to the replica database cannot be logged. Therefore, if the replica database is to be used as the source database of another replica database, the <u>log_slave_updates</u> 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 should 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.

Oplog Check

Last updated : 2021-11-15 15:55:34

Check Details

- Check requirements: oplogs can be obtained from the source database during full + incremental migration.
- Check description: incremental migration requires oplogs for replay. If the oplog.rs or oplog.\$main table does not exist in the source local database, oplogs cannot be obtained.

Fix

Start the source database as a replica set or in primary/secondary mode to ensure that oplogs can be generated for operations and recorded in the source local database.

Foreign Key Dependency Check

Last updated : 2022-09-02 15:04:25

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

- Foreign key dependency can only be set to only NO ACTION or RESTRICT .
- During partial table migration, tables with foreign key dependency must be migrated.

TDSQL for MySQL (TDStore Edition) Check Details

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

Troubleshooting

- MySQL/MariaDB/Percona/TDSQL-C/TDSQL for MySQL: Modify the foreign key parameter to a value type supported by DTS.
- TDSQL for MySQL (TDStore Edition): Delete the foreign key parameter content.

Modifying the foreign key rule

When you set a foreign key in MySQL, there are four values that can be selected:

- 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.

If an error occurs, fix it as follows:

Windows

- 1. Log in to the DMC platform 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 Structur	re					
Database: mysql Basic Info Column Info Index Foreign Key Partition										
Add Dates										
	Foreign key name	Include column	Referenced database	Referenced table	Referenced column	When deleting	When changing			
01		请选择	✓ mysql	v help_topic	▼ 清选择	* RESTRICT	r RESTRICT			

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

Linux

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

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

3. Add the foreign key settings again.

```
alter table `table name 1` add constraint `foreign key name 2` foreign key `tab
le name 1`(`column name 1`) references `table name 2`(`column name 1`)
on delete cascade on update cascade;
```

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 : 2021-11-15 15:52:25

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 : 2022-09-21 17:24:22

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:



4. Run the verification task again.

Incremental Migration Precondition Check

Last updated : 2021-11-15 15:55:33

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 **SPGDATA**.
- 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:

5. Run the verification task again.

Extension/Plugin Compatibility Check

Last updated : 2021-11-15 15:55:33

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.

Source Database Balancer Check

Last updated : 2021-11-15 16:02:04

Check Details

- Check requirements: if the source database is a sharded instance, you need to disable the balancer in it before you can start migration.
- Check description: an incremental migration task will get the oplog. If the balancer is enabled, moveChunk in the source database may cause data inconsistency in the target database.

Fix

- 1. Log in to the source database.
- 2. Run the following command to disable the source database balancer:

```
sh.stopBalancer()
sh.getBalancerState()
```

3. Run the verification task again.

Source Database Node Role Check

Last updated : 2021-11-15 16:02:04

Check Details

- Check requirements: in a MongoDB migration task, if the source database is a sharded database, you need to enter the mongos, config server, and mongod node information.
- Check description: the information of the mongos, config server, and mongod nodes cannot be disordered; otherwise, data migration will also be disordered; for example, the mongos node information should not be entered in the box for the mongod node. Note that you only need to enter the information of one mongod node for each shard.

Fix

- Enter the correct node information in the DTS task configuration items.
- Enter only one mongod for each shard.

ShardKey Check

Last updated : 2021-11-15 16:02:04

Check Details

- Check requirement: if the target database is a sharded instance, you can preset the shardkey in it. If the table shardkeys in the source and target databases are different, an warning will be triggered. It will not block the task but will affect the business. You need to assess and determine whether to ignore the warning.
- Impact on the business: if some shardkeys are different, the migration or sync task will fail.

Fix

If the target database has preset shardkeys, run the following command to shard the source database:

```
sh.shardCollection("<database>.<collection>", { <shard key> : "hashed" } , false,
{numInitialChunks: number of preset chunks})
```

Run the verification task again.

Warning Item Check

Last updated : 2021-11-15 15:52:36

MySQL/TDSQL-C Check Details

You need to configure the following parameter as required; otherwise, the system will report warnings during verification, which 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 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 max_allowed_packet parameter in the target database is smaller than that in the source database, it may cause failures in writing data to the target database and lead to full migration failures.
 - Fix: change the value of the max_allowed_packet parameter in the target database to a value greater than that in the source database.
- We recommend 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 event packets. If its value is too small, program errors mat occur, causing backup failure and frequent sending/receiving of network packets, which compromises the system performance.
 - Fix: run the following command to modify the <code>max_allowed_packet</code> parameter:

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

- We recommend 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 you use a database with a specification above 2-core CPU and 4 GB memory.

- If you only perform full data migration, do not write new data into the source database during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.
- For data export with locks: you need to use Flush Table With Read Lock to temporarily lock tables in the source database, and the MyISAM table will be locked until the full data is exported. Currently, the lock wait timeout period is 60s, and if locks cannot be obtained before it elapses, the task will fail.
- For data export without locks: read locks will be added to tables without a primary key or non-null unique key.

TDSQL for MySQL Check Details

You need to configure the following parameter as required; otherwise, the system will report warnings during verification, which 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 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 max_allowed_packet parameter in the target database is smaller than that in the source database, it may cause failures in writing data to the target database and lead to full migration failures.
 - Fix: change the value of the max_allowed_packet parameter in the target database to a value greater than that in the source database.
- We recommend you set <code>max_allowed_packet</code> 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 event packets. If its value is too small, program errors mat occur, causing backup failure 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 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 you use a database with a specification above 2-core CPU and 4 GB memory.
- If you only perform full data migration, do not write new data into the source database during migration; otherwise, the data in the source and target databases will be inconsistent. In scenarios with data writes, to ensure the data consistency in real time, we recommend you select full + incremental data migration.
- For data export with locks: you need to use Flush Table With Read Lock to temporarily lock tables in the source database, and the MyISAM table will be locked until the full data is exported. Currently, the lock wait timeout period is 60s, and if locks cannot be obtained before it elapses, the task will fail.
- For data export without locks: read locks will be added to tables without a primary key or non-null unique key.
- If the source database instance is a distributed database, you need to create sharded tables in the target database in advance; otherwise, the tables will become non-sharded tables after being migrated.

DLL Ring Sync Check for Single Database/Table Object

Last updated : 2021-11-16 16:14:00

Check Details

In scenarios where multiple sync tasks need to be configured, such as two-way sync, many-to-one sync, and active active sync, DDL configuration should not form a ring; otherwise, DDL statements will loop in the system, causing errors.

For example, among the three sync tasks (1, 3, and 5) marked by blue lines in the following figure, you can select DDL in up to two of them, and if you select three, a ring will be formed.



Fix

Modify the sync task configuration. In **Set sync options and objects** > **Data Sync Option** > **SQL Type**, modify the DDL parameter configuration to avoid rings.
DLL Sync Conflict Check for Single Database/Table Object in the Same Target

Last updated : 2021-11-16 16:14:23

Check Details

In scenarios where multiple sync tasks need to be configured, such as many-to-one sync and active-active sync, the same table object cannot receive DDL sync from multiple IDCs; otherwise, such DDL statements may conflict with each other in the target database, causing errors.

For example, as shown below, databases A and C have tables with the same name to be synced to database B. Then, you can select DDL in only one task between tasks 1 and 4.



Fix

Modify the sync task configuration. In **Set sync options and objects** > **Data Sync Option** > **SQL Type**, modify the DDL parameter configuration to avoid the situation where the same table object receives DDL sync from multiple IDCs.

Level-2 Subpartitioned Table Check

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL Check Details

- If the source database is TDSQL for MySQL, the migration task will report a warning if there are level-2 subpartitioned tables. You can ignore the warning and continue with the task, in which case the level-2 subpartitioned tables will be skipped for migration. If there are such tables in a data sync task, the verification will fail, and you need to remove these tables. For more information, see Subpartitioning.
- As the underlying logic of level-2 subpartitioned tables is implemented based on subtables, the target database will report an exception when level-2 subpartitioned tables are migrated or synced.

Troubleshooting

Remove the level-2 subpartitioned tables from the list of tables to be migrated or synced.

Primary Key Check

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL Check Details

If the target database is TDSQL for MySQL, the source database table must have a primary key. Currently, tables without primary key cannot be migrated.

Troubleshooting

Modify the tables without primary key as prompted and execute the verification task again.

DDL Check for Tables to Be Migrated

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL/MariaDB Check Details

- If the target database is TDSQL for MySQL or TencentDB for MariaDB, the verification will fail if a table to be migrated contains any characters other than digits, underscores, or letters in its name or column name.
- Note: TDSQL for MySQL or TencentDB for MariaDB database/table/column names can only contain digits, underscores, or letters.

Troubleshooting

Remove tables that contain invalid characters as prompted and execute the verification task again.

System Database Conflict Check

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL Check Details

- If the target database is TDSQL for MySQL and the database named sysdb is included in the databases to be migrated, it won't be migrated.
- Note: sysdb is the system database in TDSQL for MySQL. If a database to be migrated is named sysdb, the migration task will report a warning but continue to run normally.

Troubleshooting

Remove sysdb from the list of databases to be migrated.

Table Structure Check for Source and Target Instances

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL Check Details

- If the target database is TDSQL for MySQL, we recommend you create a partitioned table and specify a shardkey in the target database in advance; otherwise, the task will report a warning. You can ignore the warning and continue with the task. After the warning is ignored, DTS will create a table in the target database based on the source database table paradigm. If the source database is a standalone instance, DTS will create a single table.
- If you have already created a partitioned table in the target database or don't need one, DTS will check whether the source and target database table structures are consistent. The verification will fail if the field types, field names, or encoding methods are inconsistent.

Troubleshooting

When the verification fails, DTS will display the list of tables with inconsistent table structures. You can modify the target table structure based on the inconsistencies.

InnoDB Table Check

Last updated : 2022-11-24 10:38:58

TDSQL for MySQL/MariaDB Check Details

If the target database is TDSQL for MySQL or TencentDB for MariaDB, only InnoDB tables can be migrated or synced. If the objects to be migrated or synced include non-InnoDB tables, the verification will fail.

Troubleshooting

Remove non-InnoDB tables as prompted and execute the verification task again.

Migration Object Dependency Check

Last updated : 2022-11-24 10:38:58

TencentDB for MariaDB Check Details

If the target database is TencentDB for MariaDB, the views to be migrated are not migrated together with their referenced tables or views, the verification will fail.

Troubleshooting

Select other objects that are referenced with the views to be migrated and migrate them together as prompted, and execute the verification task again.

Constraint Check

Last updated : 2022-08-24 16:28:42

TDSQL for PostgreSQL check details

The subscribed table must have a primary key or have REPLICA IDENTITY as FULL.

Troubleshooting

If the verification fails, modify the corresponding table.