

Data Transfer Service

Error Handling

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice



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

Service Statement

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

Contents

Error Handling

- Common Errors

- Failed Connectivity Test

- Failed or Alarmed Check Item

- Inability to Select Subnet During CCN Access

- Slow or Stuck Migration

- Data Sync Delay

- High Data Subscription Delay

- Data Consumption Exception

Error Handling

Common Errors

Last updated : 2024-07-08 15:45:26

This document describes some common errors that may occur in a DTS task and how to fix them.

Common MySQL Errors

Common errors reported in the MySQL system during database migration, sync, or subscription and their solutions are as detailed below:

Error Code	Description	Scenario	Error Message	Analysis and Solution
1227	There is a permission issue.	Data migration/sync/subscription	Error 1227: Access denied.	<p>Error analysis</p> <p>The account that executes the task doesn't have permission to connect to the source/target database.</p> <p>Solution</p> <p>Authorize the account that executes the task. For more information, see Data Migration (NewDTS).</p>
1040	There are too many database connections.	Data migration/sync	Error 1040: Too many connections.	<p>Error analysis</p> <p>There are too many database connections.</p> <p>Solution</p> <p>Increase the maximum number of connections to the source database by increasing the value of <code>max_connections</code>, or run the task again during low business workload hours.</p>
1045	The operation was denied.	Data migration/sync/subscription	Error 1045 (28000): Access denied for user '{{xx}}'@'{{xx}}'	<p>Error analysis</p> <p>The account permission or password is modified when the task is running.</p>

			(using password: YES)	<p>The DTS service IP addresses haven't been authorized in the source or target database.</p> <p>Solution</p> <p>Check whether you have modified the account or password. If so, undo the modification operation or change the account or password back. For more information, see Data Migration (NewDTS).</p> <p>Authorize the DTS service IP addresses as instructed in Adding DTS IP Address to Database Allowlist.</p>
1050	The table already exists, and the DDL statement is executed repeatedly.	Data migration/sync	<p>Error 1050: Table <code>{{}}</code> <i>already exists</i>, <i>binlog position:</i> <code><{{}}></code>, <i>gtid:</i><code>{{}}</code>, <i>related tables:</i> <code>{{}}</code></p>	<p>Error analysis</p> <p>When data is migrated or synced from multiple source databases to one target database, all the source databases executed the same DDL operation, causing a duplicate DDL operation in the target database. You can only select DDL operation in one of the multiple source databases. When the task is running, the target database has created the same table, and it repeats the same DDL operation when it receives data synced from the source database.</p> <p>The network is abnormal or the statement is executed for too long, causing duplicate DDL operations during the task retry process.</p> <p>Solution</p> <p>Troubleshoot based on the above analysis. To skip the current transaction, submit a ticket.</p>
1054	The table	Data migration/sync	Error 1054:	Error analysis

	contains an unknown column		Unknown column {{{} <i>related tables:</i> {{{}}	<p>The table structure is not selected as the migration/sync object before the task is started, and the target database doesn't contain this column.</p> <p>When the task is running, this column is deleted from the target database.</p> <p>Solution</p> <p>Check whether the target database contains this column, and if not, add this column to the target database and execute the task again.</p>
1062	An error is reported due to the primary key conflict.	Data sync	Error 1062: Duplicate entry '{{{xx}}}' for key 'PRIMARY', <i>related tables:</i> '{{{xx}}}'.	<p>Error analysis</p> <p>If you select Report for Primary Key Conflict Resolution in data sync, DTS will report an error when encountering a primary key conflict between the source and target databases.</p> <p>When the task is running, data is manually written to the target database, causing the target database to contain data with the same primary key.</p> <p>Before the task is started, the unique key check is disabled for the source database. Therefore, the source database already contains data with duplicate primary keys.</p> <p>When the task is running, the DELETE operation is not synced from the source database to the target database, causing a primary key conflict when data is inserted into the source database.</p> <p>Solution</p> <p>Check whether there are duplicate primary keys in the</p>

				<p>source database, and if so, handle them first.</p> <p>Modify or delete the primary key of the corresponding data table from the target database and execute the task again.</p>
1071	The index field is too long.	Data migration/sync	<p>Error 1071 (42000): Specified key was too long; max key length is 767 bytes.</p>	<p>Error analysis</p> <p>By default, a single-field index in the InnoDB engine can contain up to 767 bytes, that is, an index can contain up to 384 two-byte fields or 256 three-byte fields. GBK, UTF-8, and utf8mb4_unicode_ci are two-, three-, and four-byte character sets respectively. On MySQL 5.6 and later versions, all MyISAM tables are automatically converted to InnoDB tables. Therefore, with the MyISAM storage engine, a self-built database may contain combined index column of more than 767 bytes in length; however, the same table creation statement that can normally run in the self-built database won't work on MySQL 5.6 or later versions.</p> <p>Solution</p> <p>Modify the length of index columns in the erroneous rows in the file.</p> <p>Example:</p> <pre>create table test(test varchar(255) primary key) charset=utf8; -- Successful create table test(test varchar(256) primary key) charset=utf8;</pre>
1146	The table	Data migration/	Error 1146:	Error analysis

	doesn't exist.	Data sync	Table '{{xx}}' doesn't exist on query.	<p>When the task is running, the table is deleted from the target database.</p> <p>DDL statements for table structure change are executed in the source database in the data export stage.</p> <p>Before the task is started, the table structure is not selected as the migration/sync object.</p> <p>Solution</p> <p>Execute the <code>show create table xxx</code> statement in the target database to check whether the table exists, and if not, create it in the target database manually.</p>
1213	A deadlock is caused by double write to the source and target databases.	Data migration/sync	<p>Error 1213: Deadlock found when trying to get lock; try restarting transaction, related tables: '{{xx}}'.</p>	<p>Error analysis</p> <p>Your write operation conflicts with that performed by DTS in the target database, which causes a deadlock.</p> <p>Solution</p> <p>Stop the deadlock process and create the task again.</p> <p>We recommend that you control the lock logic for UPDATE operations in the instance, add an index to tables, and use row lock as much as possible to reduce lock overheads.</p>
1236	There is a binlog issue in the source database.	Data migration/sync/subscription	<p>Error 1236 (HY000): Cannot replicate because the master purged required binary logs. Replicate the missing transactions from elsewhere, or provision a new slave from backup.....</p>	<p>Error analysis</p> <p>The binlog retention period configured for the source database is short, so the binlog may have already been cleared when DTS pulls data, or the offset of the pulled binlog is incorrect.</p> <p>Solution</p> <p>Check whether the binlog retention period (<code>expire_logs_days</code>) set in the source database meets</p>

				the business requirements. We recommend that you set it to three days or above and create the task again.
1414	DDL statements for table structure change are executed in the source database in the data export stage.	Data migration	Error 1414: Table definition has changed, please retry transaction.	<p>Error analysis</p> <p>You cannot execute DDL statements for table structure change in the source database in the data export stage; otherwise, an error may be reported.</p> <p>Solution</p> <p>Create the migration task again</p>

Common DTS Errors

Common errors reported in the DTS system during database migration, sync, or subscription and their solutions are as detailed below:

Error Description	Scenario	Error Message	Analysis and Solution
There is a database connection exception.	Data migration/sync/subscription	<code>invalid connection{}</code> . driver: bad connection,{{*}} dial tcp {{*}}: connect: connection refused.	<p>Error analysis</p> <ol style="list-style-type: none"> 1. The source/target database is isolated or offline. 2. The source/target database fails to be restarted for a long time 3. The source/target database fails to implement the source-replica switch for a long time. 4. The source/target database is overloaded 5. The connections in the source/target database are killed manually or by programs. 6. There is a network connection failure caused by the source/target database's network security policy that denies the access request. <p>Solution</p> <p>Troubleshoot based on the above</p>

			<p>analysis and fix the problem one by one.</p> <p>For errors that occur on TencentDB instances, you can go to the product console and the Tencent Cloud Observability Platform to troubleshoot and fix problems. After that, execute the task again in the DTS console.</p> <p>If you cannot troubleshoot or fix the problem, submit a ticket for assistance</p>
There is a database connection exception.	Data migration/sync/subscription	dial tcp {{*}}: connect: connection refused.	<p>Error analysis</p> <ol style="list-style-type: none"> 1. The source/target database is isolated or offline. 2. The source/target database fails to be restarted for a long time 3. The source/target database fails to implement the source-replica switch for a long time. 4. The source/target database is overloaded 5. There is a network connection failure caused by the source/target database's network security policy that denies the access request. <p>Solution</p> <p>Troubleshoot based on the above analysis and fix the problem one by one.</p> <p>For errors that occur on TencentDB instances, you can go to the product console and the Tencent Cloud Observability Platform to troubleshoot and fix problems. After that, execute the task again in the DTS console.</p> <p>If you cannot troubleshoot or fix the problem, submit a ticket for assistance</p>
The table fails to be locked due to a slow	Data migration/sync	Find Resumable Error, src db has long query sql, fix it and try it later.	<p>Error analysis</p> <p>If the source database has long-running (over 5s) SQL statements, DTS needs to wait for these slow</p>

<p>SQL statement in the source database.</p>		<p>Find Resumable Error: Task failed due to table lock failure caused by time-consuming SQL query statements in source instance.</p>	<p>statements to complete before locking the table and exporting the data to avoid disrupting the source database business. The locking duration is set to 60s by default. If the locking operation exceeds this timeout, it will fail, resulting in an error in the task.</p> <p>Solution Handle the slow SQL statement in the source database or create the task again after the slow SQL statement is executed.</p>
<p>The binlog parameter is incorrectly formatted.</p>	<p>Data migration/sync/subscription</p>	<p>Statement binlog format unsupported: {{xx}}. binlog must ROW format, but MIXED now. binlog row before/after image not full, missing column {{xx}}, binlog position:{{xx}}, gtid: {{*}}.</p>	<p>Error analysis To ensure the accuracy and integrity of data, DTS checks the source database binlog parameters in the task check stage, and reports an error for any ineligible check item to not start the task. If you modify a source database binlog parameter after the check is passed and the task is successfully started, the task will still report an error. Therefore, make sure the source database binlog meets the following requirements: <code>binlog_format</code> must be set to <code>ROW</code> . <code>binlog_row_image</code> must be set to <code>FULL</code> .</p> <p>Solution Correct the erroneous parameter as prompted or as instructed in Binlog Parameter Check, and create the task again.</p> <p>Note: You need to restart the thread for the parameter modification to take effect. After the database is restarted, the parameter configurations will be initialized, so you need to check whether they are correct after the restart.</p>

<p>The built-in Kafka is abnormal.</p>	<p>Data subscription</p>	<p>kafka: error while consuming {{*}}. kafka: Failed to produce message to topic.</p>	<p>Error analysis When the built-in Kafka component of a DTS data subscription task produces or consumes data abnormally, the backend service will automatically retry and recover. If the exception occurs, refresh the page and observe the task status.</p> <p>Solution If the task status doesn't change over 10 minutes after you fresh the page, submit a ticket for assistance.</p>
<p>The Kafka data expires because the task has been stopped for over seven days.</p>	<p>Data subscription</p>	<p>kafka server: The requested offset is outside the range of offsets maintained by the server for the given topic/partition.</p>	<p>Error analysis The Kafka data cached in the DTS task will expire if the task has been stopped or abnormal for over seven days, so the Kafka data cannot be read successfully.</p> <p>Solution Terminate the task and create a new one. You can create a monthly subscribed task by resetting it.</p>

Failed Connectivity Test

Last updated : 2024-07-08 15:45:26

Issue

The source or target database connectivity test fails when you create a data migration, sync, or subscription task.

Possible Causes

If the Telnet test fails, the causes may be:

[The server where the source database resides has a security group or firewall configured.](#)

[The source IP addresses are blocked in the source database.](#)

[The source database port is not opened.](#)

[There is a network conflict, such as IP range conflict or incorrect parameter configuration.](#)

[After an access type is selected and the connectivity verification is passed, the access type is changed.](#)

If the Telnet test is passed, but the database connection fails, the causes may be:

[There is an account authorization problem.](#)

[The account or password is incorrect.](#)

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

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:

If the source database is a self-built database, check whether the server where the source database resides is configured with firewall policies, and if so, disable the firewall.

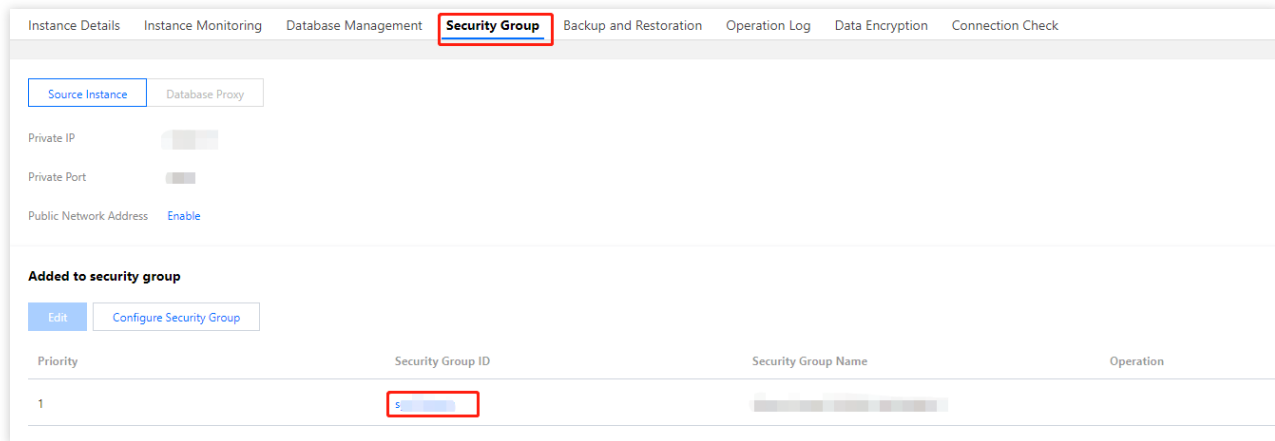
Windows: Open Control Panel and find the Windows Defender Firewall and check whether firewall policies are configured.

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

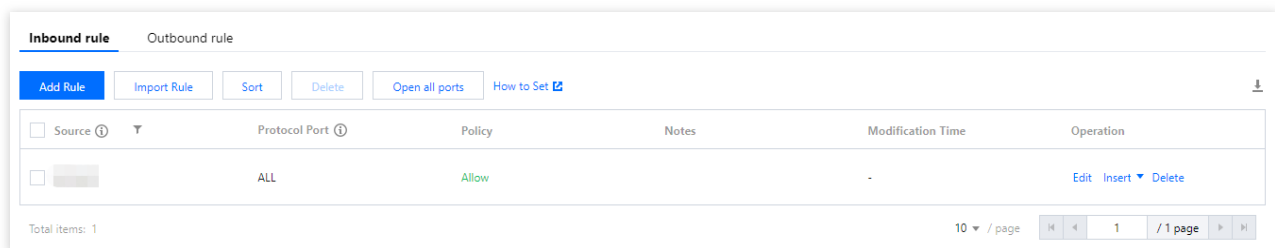
If the source database is a TencentDB database, check whether DTS IP range is blocked in the security group of the database, and if so, modify as follows:

When the connectivity test fails, the IP range that needs to be allowed for DTS will be displayed in the console as shown below:

1. Log in to the source database (with MySQL as an example) and click an instance ID in the instance list to enter the instance management page.
2. On the instance management page, select the **Security Group** tab and check whether there are policies blocking the SNAT IP range of DTS.



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



If the source database is a third-party cloud database, check the security group settings.

Source IP Addresses Blocked in Source Database

Check method

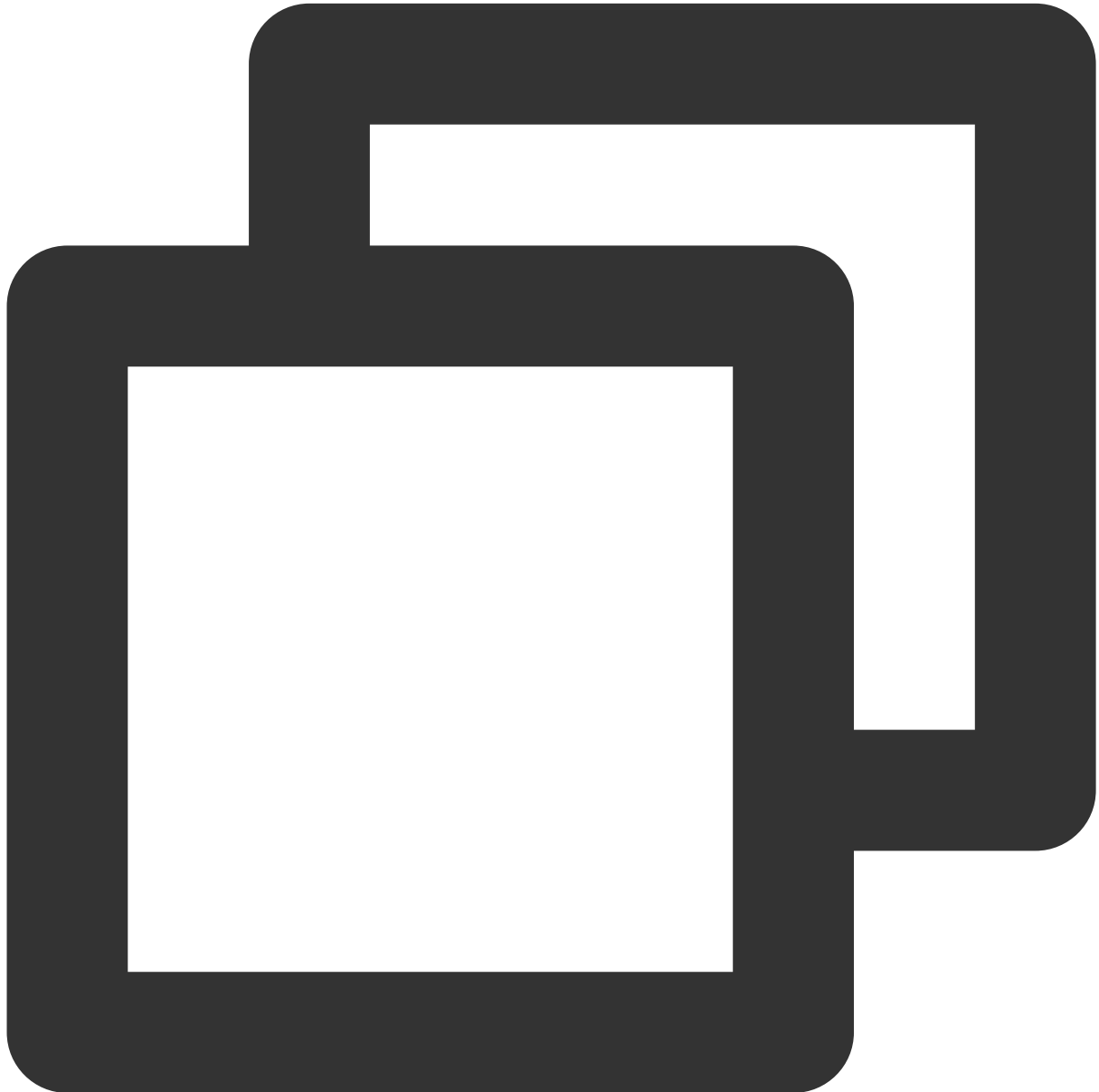
MySQL

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

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

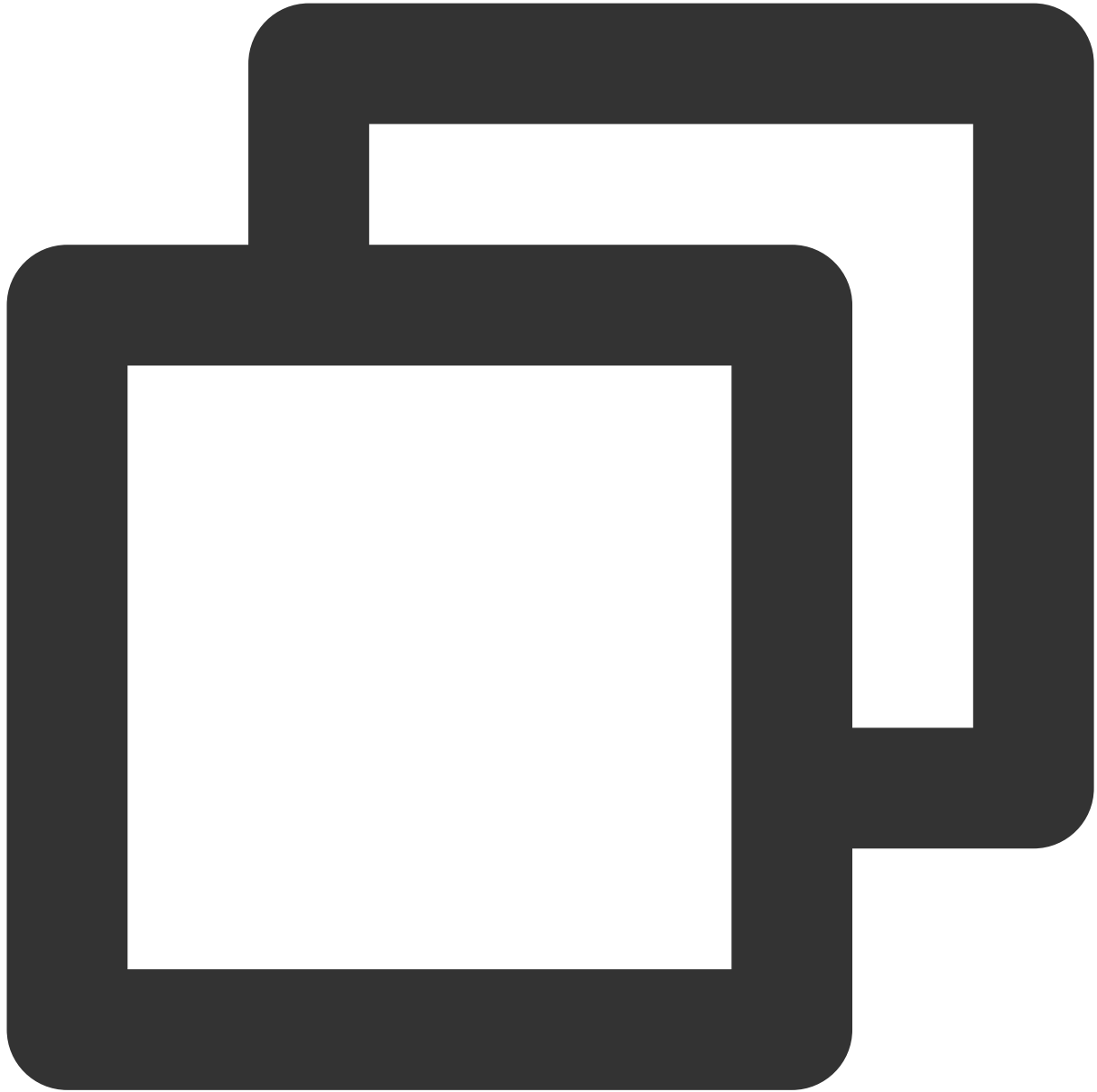
If the source database is MySQL, you can use the MySQL client to connect to it, run the following SQL statement, and check whether the list of authorized IP addresses contains the SNAT IP addresses of DTS in the output result.

When granting database permissions to users, the authorized IPs must include the SNAT IPs; otherwise, they may be blocked; for example:



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

You can verify as follows:



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

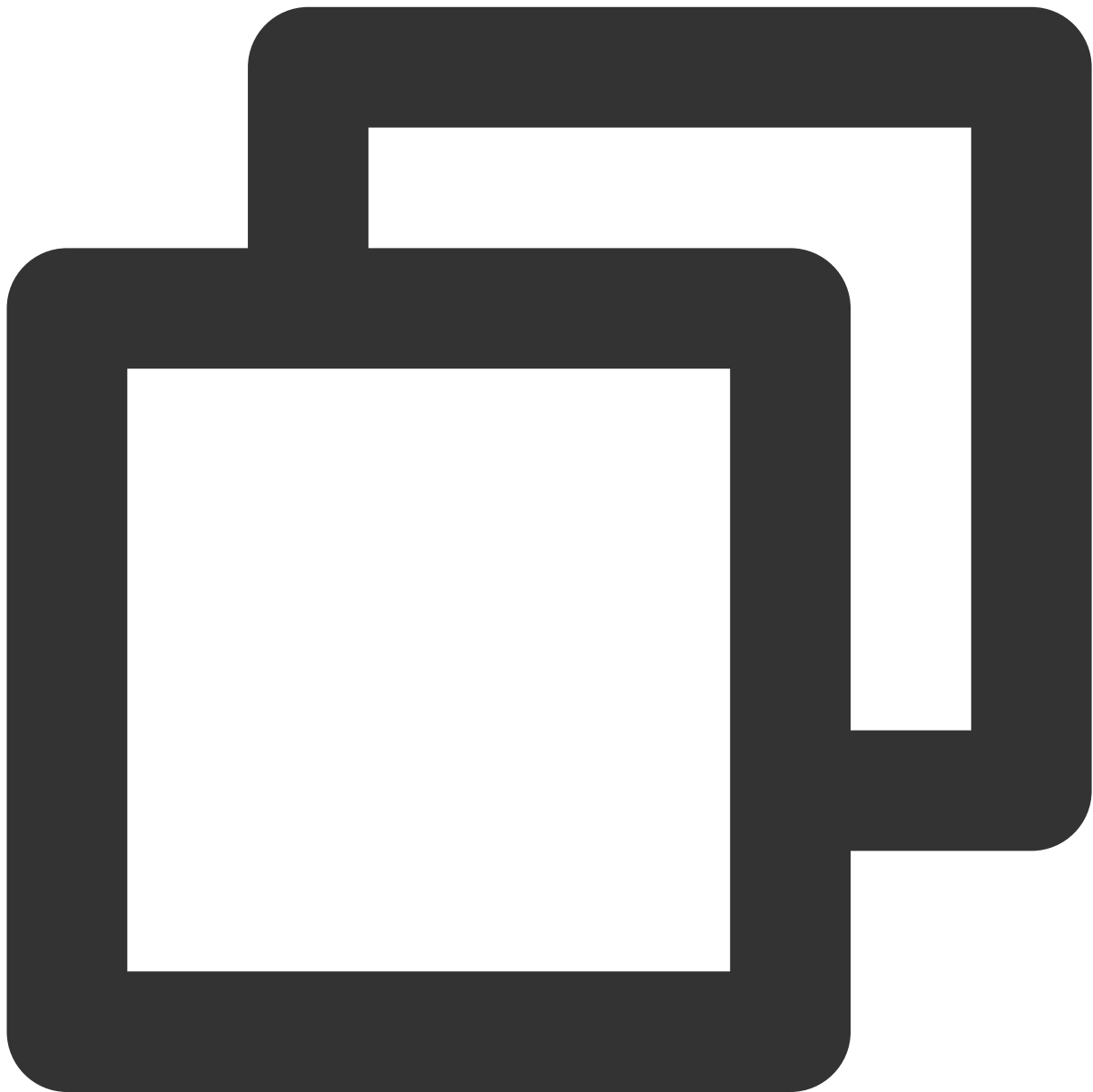
SQL Server

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

PostgreSQL

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

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



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

host	all	all	0.0.0.0/0	md5
host	all	all	172.x.x.0/20	md5

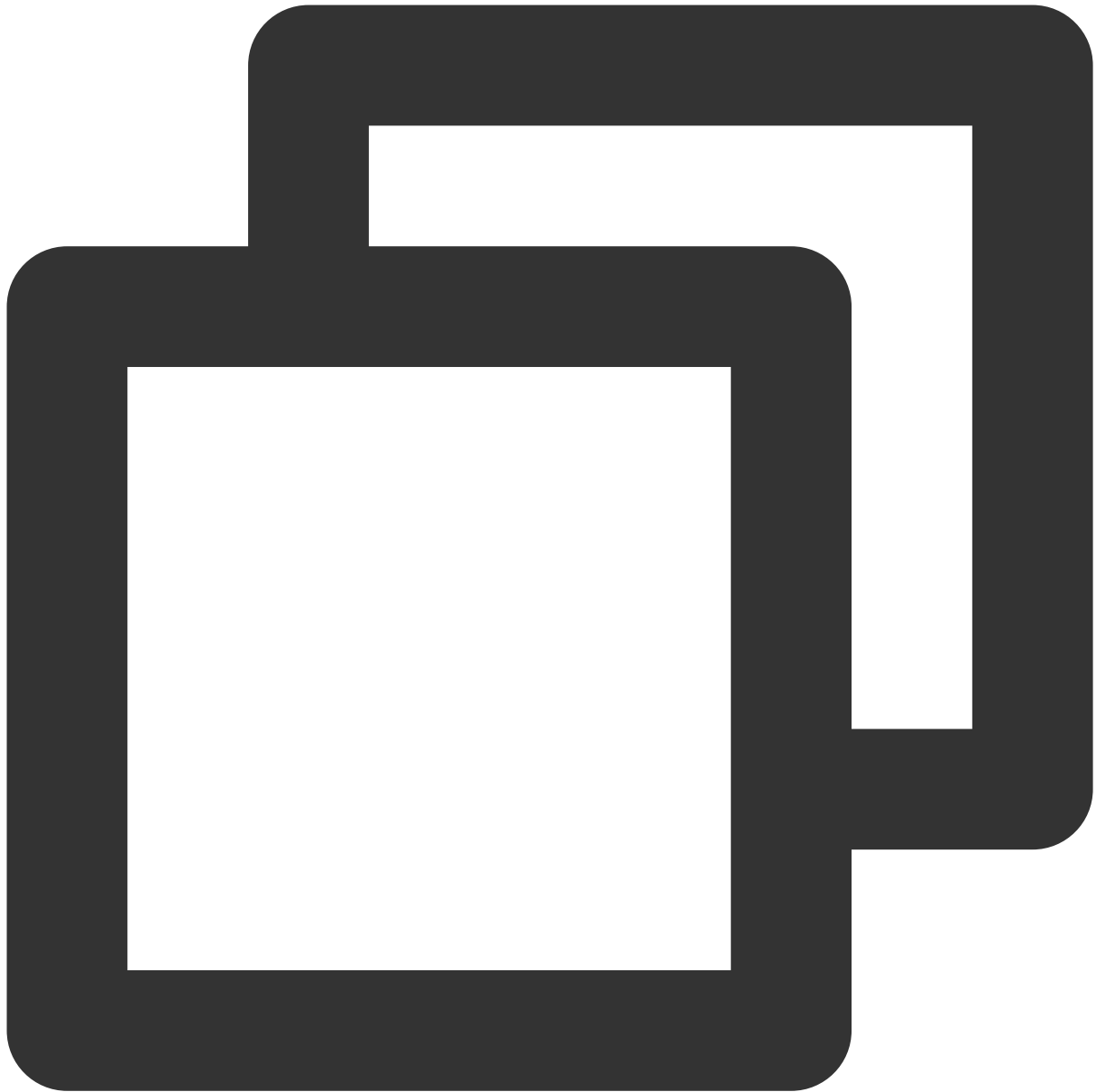
MongoDB/Redis

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

Fix

MySQL

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



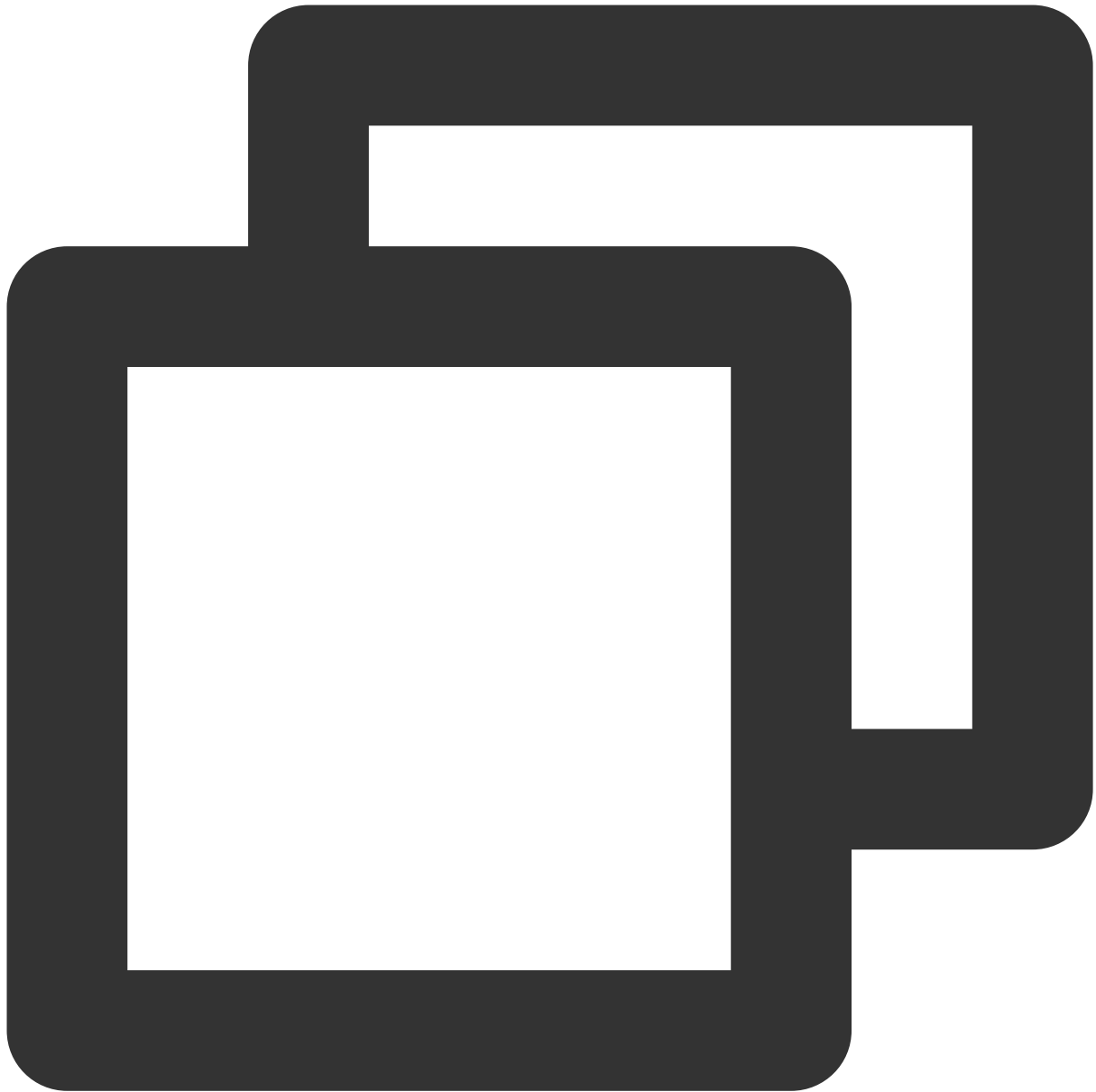
```
mysql> grant all privileges on . to '[\\$UserName]@'%'; // `[\\$Username]` is the
mysql> flush privileges;
```

2. If the source database is a self-built database, you also need to check whether the `bind-address` configuration is abnormal, and if so, modify it as instructed below.

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

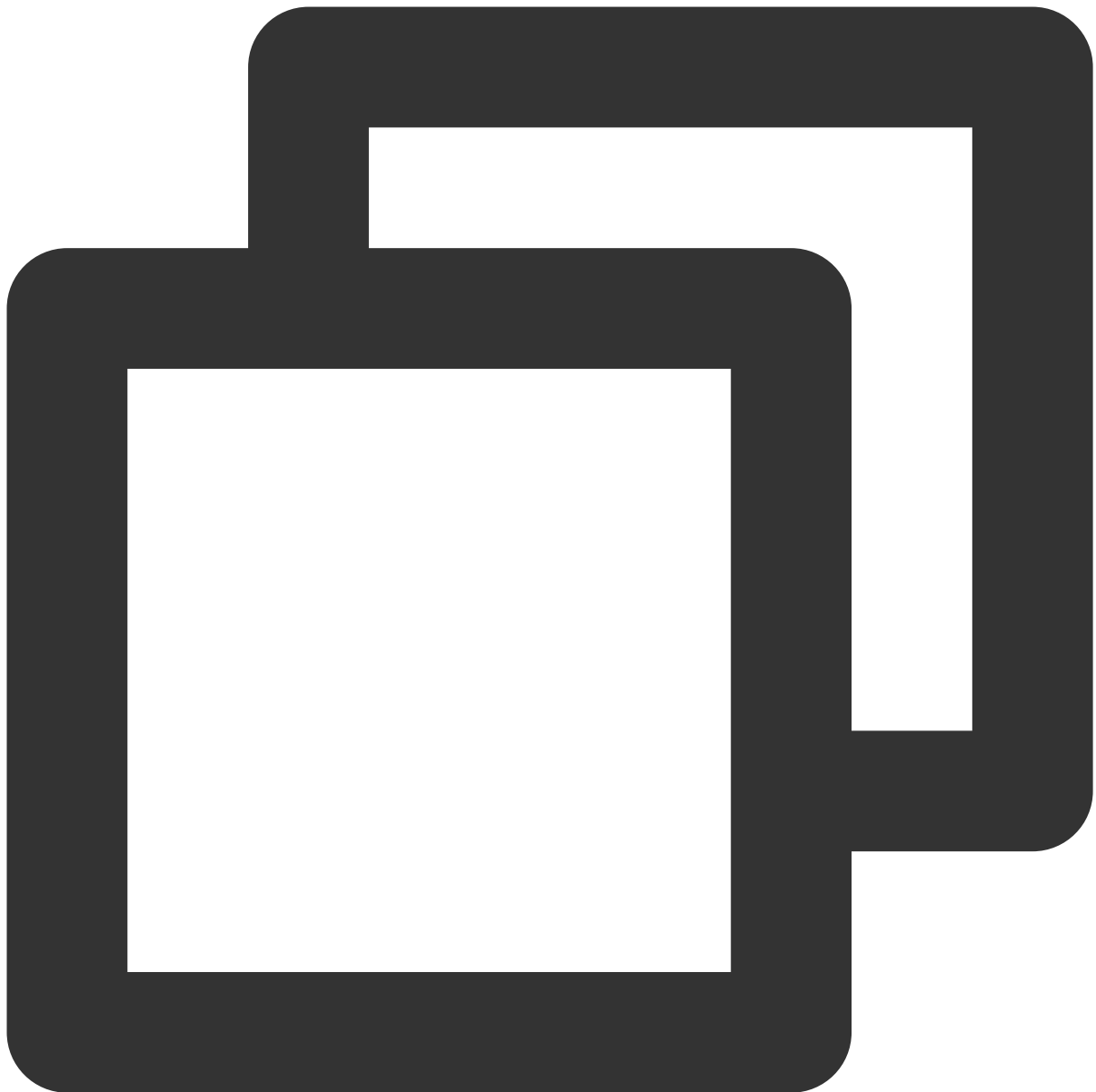
Note:

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



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

2.2. Restart the database.



```
service mysqld restart
```

2.3. Check whether the configuration takes effect.



```
netstat -tln
```

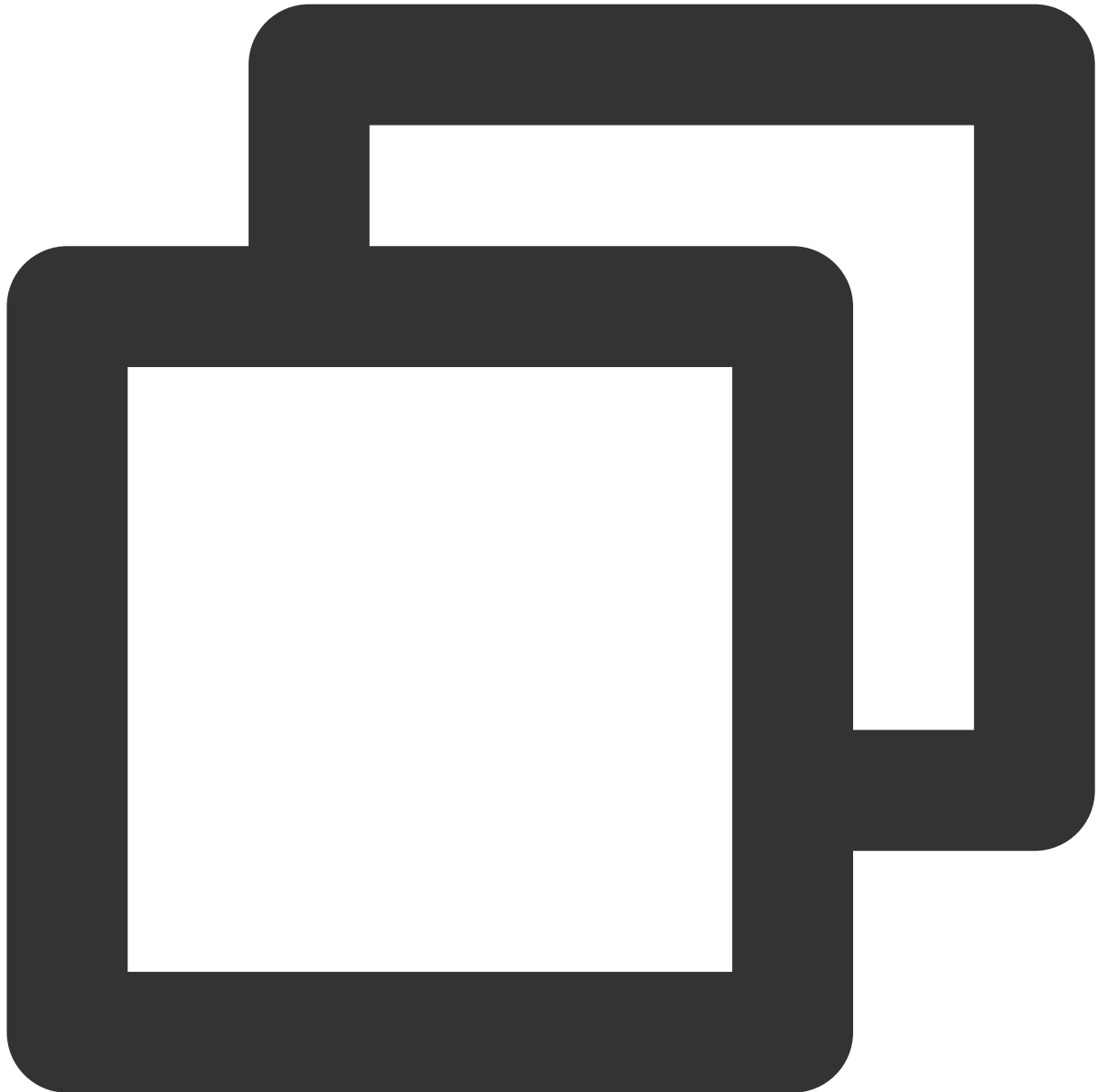
3. Run the verification task again.

SQL Server

Disable the firewall or trigger.

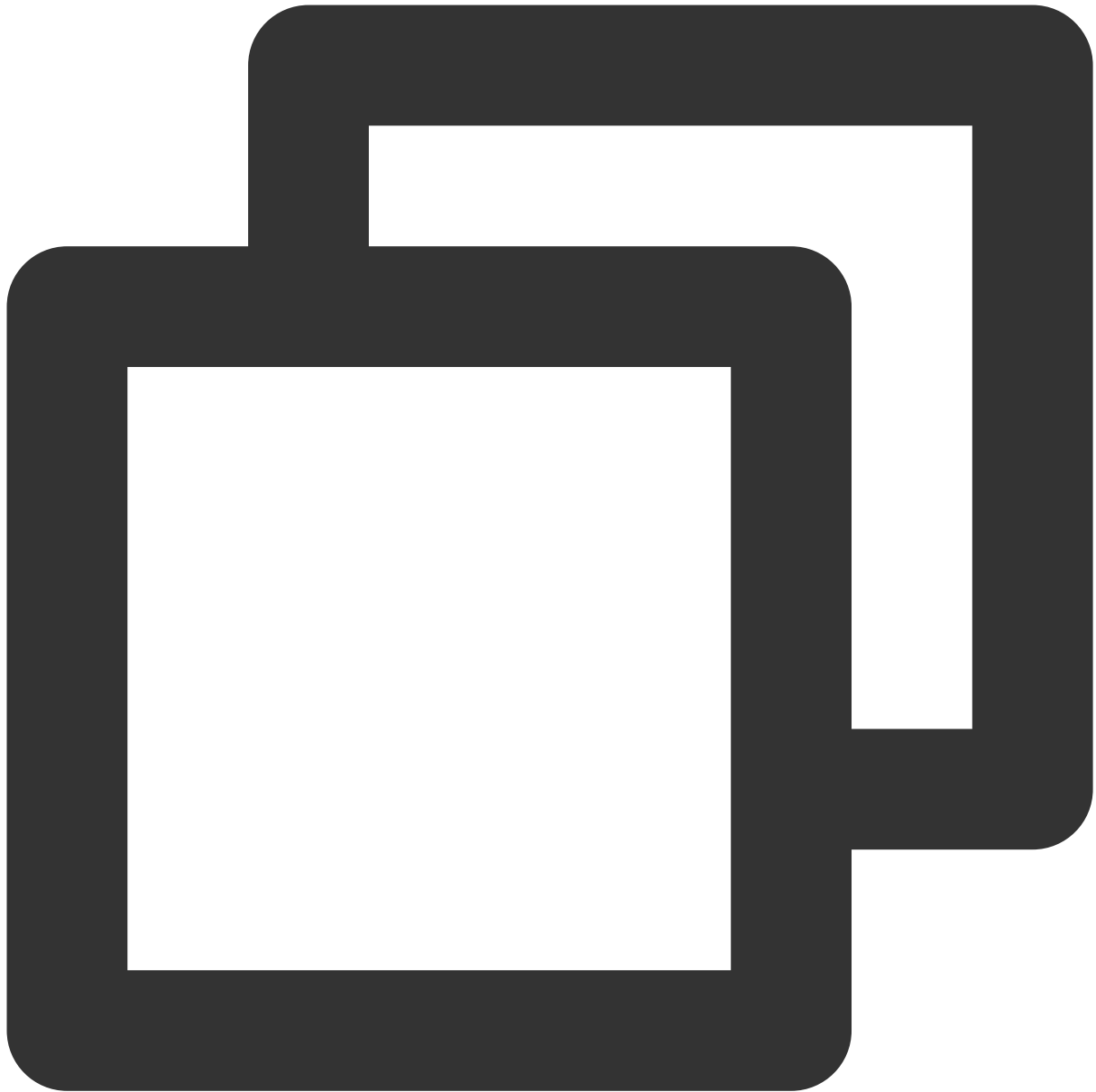
PostgreSQL

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



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

2. After the modification is completed, you can restart the database instance 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](#).

Redis

1. Disable the `bind` configuration in `redis.conf` or change it to `0.0.0.0`.

2. Restart the database to make the configuration take effect and execute the verification task again.

Closed Network Port

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:

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

MySQL: 3306

SQL Server: 1433

PostgreSQL: 5432

MongoDB: 27017

Redis: 6379

Network Conflict

If you select the [VPN/Direct Connect](#) or [CCN](#) access method, you can refer to the documentation for troubleshooting.

Migration Account Authorization

Authorize the migration account again as instructed in the corresponding scenario in [Migration from MySQL to TencentDB for MySQL](#) and [Sync from MySQL/MariaDB/Percona to TencentDB for MySQL](#).

Incorrect Database Account or Password

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

Access Type Change

For the same source and target databases, if an access type such as **Public Network** is selected and the connectivity verification is passed, you cannot switch to another access type such as **Direct Connect**; otherwise, an error will be reported during connectivity verification.

Failed or Alarmed Check Item

Last updated : 2024-07-08 15:45:26

Issue

The result of a check item is **Failed** or **Alarm** during task verification.

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

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

Possible Causes

The check item doesn't meet the requirements.

Solutions

Fix the problem as instructed in [Check Item Overview](#).

Inability to Select Subnet During CCN Access

Last updated : 2024-07-08 15:45:26

Issue

The subnet cannot be selected during CCN access.

Possible Causes

The account of the CCN-associated VPC is different from that running the migration/sync task.

Solutions

The account of the CCN-associated VPC must be the same as that running the migration/sync task.

For example, to migrate an instance from account A to account B, you should use account B to create a task, so the CCN-associated VPC must be under account B.

For more information on CCN configuration, see [CCN Access: Configuring VPC-IDC Interconnection Through CCN](#).

Slow or Stuck Migration

Last updated : 2024-07-08 15:45:26

Issue

The migration/sync task takes too long or gets stuck.

Possible Causes

The migrated data volume is high.

The source database has a slow SQL statement.

The source database content is non-compliant.

The bandwidth is restricted or there are network jitters.

No data is written to the source database during incremental migration or sync.

Note:

If you select **Full + Incremental migration** as the migration type, after the full migration task is completed, the incremental migration task will continue, and you need to manually stop it by clicking **Done** in the **Operation** column; otherwise, the task will keep running, which is not the case of a stuck task.

Solutions

High migrated data volume

The data volume is high, slowing down the migration/sync task.

Slow SQL statement in source database

Check whether the source database has a slow SQL statement; if so, process the statement; if not, check other causes.

Non-compliant source database content

The content in the source database is non-compliant. For example, if the source database has tables without a primary key, large queries involving such tables will slow down the task. We recommend you add a primary key to such tables or not migrate them.

Network problem

If you use CCN for access, check the bandwidth configured in CCN. CCN only provides bandwidth below 10 Kbps between all regions free of charge, which is insufficient for DTS to transfer data. In this case, you need to configure a higher bandwidth.

If you use a self-built database, check whether the network bandwidth is restricted.

No data written to source database during incremental migration or sync

In incremental migration or sync, if no data is written to the source database for a long time or there is an empty binlog, you can write data to the source database to resume the task.

Data Sync Delay

Last updated : 2024-07-08 15:45:26

Issue

The content synced between the source and target databases has a delay.

Possible Causes

The target database has a high load.

The target database has a low specification.

No data is written to the source database for a long time.

The bandwidth is restricted or there are network jitters.

Solutions

High load in target database

View the RPS of the source and target databases in the [monitoring data](#). If the RPS is low, the target database may have a high load.

If the target database has a high load, check whether the task is normal after the business traffic drops or upgrade the target database specification.

Low specification of target database

Upgrade the target database specification.

No data written to source database during incremental migration or sync

In incremental migration or sync, if no data is written to the source database for a long time or there is an empty binlog, you can write data to the source database to resume the task.

Network problem

If you use CCN for access, check the bandwidth configured in CCN. CCN only provides bandwidth below 10 Kbps between all regions free of charge, which is insufficient for DTS to transfer data. In this case, you need to configure a [higher bandwidth](#).

If you use a self-built database, check whether the network bandwidth is restricted.

High Data Subscription Delay

Last updated : 2024-07-08 15:45:26

Issue

The producer delay is too high in the data subscription service. The monitoring data shows that there is a large gap between the numbers of GTIDS in the data subscription service and the source database, and the data subscription service can only parse a small number of transactions per second.

Possible Causes

1. The source database is overloaded.
2. The source database writes data so quickly that the DTS data subscription service cannot parse it all.
3. There are large or complicated transactions written to the source database.

Troubleshooting

1. The source database is overloaded.

Check the source database monitoring metrics. If the write load of the source database is too high, a high subscription delay is to be expected. If the load is normal, further troubleshoot as follows.

2. The source database writes data so quickly that the DTS data subscription service cannot parse it all.

Check the generation speed of source database binlogs. If it exceeds 50 MB/sec, it is very likely that the data parsing capability of the DTS data subscription service has reached the upper limit. In this case, a high subscription delay is to be expected.

If the generation speed of source database binlogs is less than 50 MB/sec, further troubleshoot as follows.

3. There are large or complicated transactions written to the source database.

Check whether there are large transactions being executed in the source database or whether the current table contains large fields such as those of the "JSON" or "BLOB" type.

Data Consumption Exception

Last updated : 2024-07-08 15:45:26

Issue

In data subscription scenarios, if you use your own consumer to consume data, you may encounter the following exceptions:

1. Data cannot be consumed.
2. The consumed data is either lost or duplicated.
3. The consumer delay keeps increasing.

Troubleshooting

1. Data cannot be consumed

If your own consumer fails to consume data, use the demo provided by DTS for consumption test first.

If the demo can normally consume data, you need to check your own consumer.

If the demo also cannot consume data, further troubleshoot as follows:

Check the network environment of the consumer. The consumer must be in the Tencent Cloud private network and in the same region as where the DTS data subscription task is.

Check whether the demo starting parameters are correct, especially the consumer group password.

Check whether the demo version is correct. The required demo version varies by source database type and data format.

Check the number of unconsumed messages on the consumer group management page in the console to see if the subscription task has written data to Kafka.

2. The consumed data is either lost or duplicated

When a data subscription task is restarted, [data duplication](#) may occur in the producer, causing the duplication of the consumed data. This scenario rarely occurs. Data duplication or data loss in other scenarios are not supposed to occur.

Generally, data duplication or data loss is caused by exceptions in your own consumer. To troubleshoot, reproduce the problem first. You can choose one of the following two methods to reproduce the problem:

In the console, change the Kafka consumption offset back to the previous offset and consume data again.

Create a consumer group and use it to consume data again. Consumption in different consumer groups does not affect each other.

If the problem can be reproduced, [submit a ticket](#) to handle it; otherwise, check whether your own consumer is abnormal.

3. The consumer delay keeps increasing

1. The commit logic of the consumer has been modified.

If the consumer only consumes data but doesn't commit the consumption offset, the offset in Kafka won't be updated. The default commit logic of the DTS demo is that, every time a checkpoint message is consumed, the consumption offset will be committed. The subscription service writes a checkpoint message about every 10 seconds. If you modify the commit rule, the consumer delay may keep increasing. To solve this problem, check the commit rule of your consumer first.

2. The consumption efficiency is too low.

The consumption efficiency may be affected by the network condition, the processing efficiency of the consumer, concurrent consumption, or consumption in multiple partitions. You can create a consumer group and compare the consumption efficiency of the DTS demo with that of your own consumer to find the cause of the low efficiency. You can also check the network condition to improve the data processing speed, or increase the number of consumers to implement concurrent consumption for topics in multiple partitions.