数据传输服务

前置校验不通过处理方法

产品文档

腾讯云

文档目录

- 目标实例是否为空
- 目标实例状态是否正常

SQL Server 校验项

- 迁移参数检查
- 源库或目标库连接性检查
- 数据库版本检查
- 用户权限检查

连接 DB 检查
源库存在性检查
目的库存在性检查
周边检查
迁移网络检查
版本检查
源实例权限检查
帐号冲突检查
部分实例参数检查
源实例参数检查
源实例是否为从机
参数配置冲突检查
目标实例内容冲突检查
目标实例空间检查
目的端负载校验
本地磁盘空间检查
Binlog 参数检查
Oplog 校验
外键依赖检查
视图检查
高级对象检查
增量迁移预置条件检查
插件兼容性检查
源端 Balancer 校验
源端节点角色校验
片建校验
警告项检查
二级分区表检查
待迁移的表是否有主键检查
待迁移表的 DDL 检查
源实例迁移库和目标实例系统库冲突检查

目标实例和源实例表结构检查

表是否是 InnoDB 表检查

视图之间以及视图和表之间的互相依赖检查

约束检查

前置校验不通过处理方法检查项汇总

最近更新时间：2021-11-16 18:14:15

各数据库迁移、同步、订阅过程中的校验项检查如下，如果发生校验项报错，请参考对应指导进行修复。

MySQL/TDSQL-C

- 连接 DB 检查
- 周边检查
- 版本检查
- 部分实例参数检查
- 目标实例内容冲突检查
- 目标实例空间检查
- Binlog 参数检查
- 外键依赖检查
- 视图检查
- 警告项检查
- RDS 检查

TDSQL MySQL

- 连接 DB 检查
- 周边检查
- 版本检查
- 部分实例参数检查
- 目标实例内容冲突检查
- 目标实例空间检查
- Binlog 参数检查
- 外键依赖检查
- 警告项检查

PostgreSQL

- 源实例连通性检查
- 源实例版本检查
- 目标实例连通性检查
- 账号冲突检查
- 参数配置冲突检查
- 结构冲突检查
- 插件兼容性检查
- 增量迁移预置条件检查

MongoDB

- 连接 MongoDB 实例校验
- 库表冲突校验
- 源端节点角色校验
- Oplog 校验
- 实例版本校验
- 实例容量校验
- 目的端负载校验
- 片建校验
- 源端 Balancer 校验

SQL Server

- 迁移参数检查
- 迁移网络检查
- 源库连接性检查
- 目的库连接性检查
- 硬盘空间检查
- 数据库版本检查
- 源库存在性检查
- 目的库存在性检查

MySQL/MariaDB/Percona/TDSQL-CMySQL/TDSQLMySQL校验项
版本检查

最近更新时间：2024-02-19 16:06:02

检查详情

检查要求：目标数据库版本必须大于或等于源数据库版本，且所有迁移和同步的版本符合版本要求。
检查说明：此处版本以大版本号区分，如5.6.x支持迁移或同步到5.6.x、5.7.x及以后版本，最后一位属于小版本号，小版本号不限制，如5.6.5可以迁移或同步到5.6.4，但是可能会有兼容性问题。

修复方法

请按 数据迁移支持的数据库 和 数据同步支持的数据库 中的版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例权限检查

最近更新时间：2024-02-19 16:06:19

检查详情

检查用户是否具备对数据库的操作权限，具体可参考以下对应权限要求：

数据迁移权限要求

MySQL/TDSQL-C/MariaDB/Percona 之间的数据迁移

MySQL/MariaDB/Percona/TDSQL MySQL 与 TDSQL MySQL 之间的数据迁移

数据同步权限要求

MySQL/TDSQL-C/MariaDB/Percona 之间的数据同步

MySQL/MariaDB/Percona/TDSQL MySQL 与 TDSQL MySQL 之间的数据同步

数据订阅权限要求

MySQL/MariaDB/TDSQL MySQL/TDSQL-C MySQL 数据订阅

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

部分实例参数检查

最近更新时间：2023-08-25 16:57:28

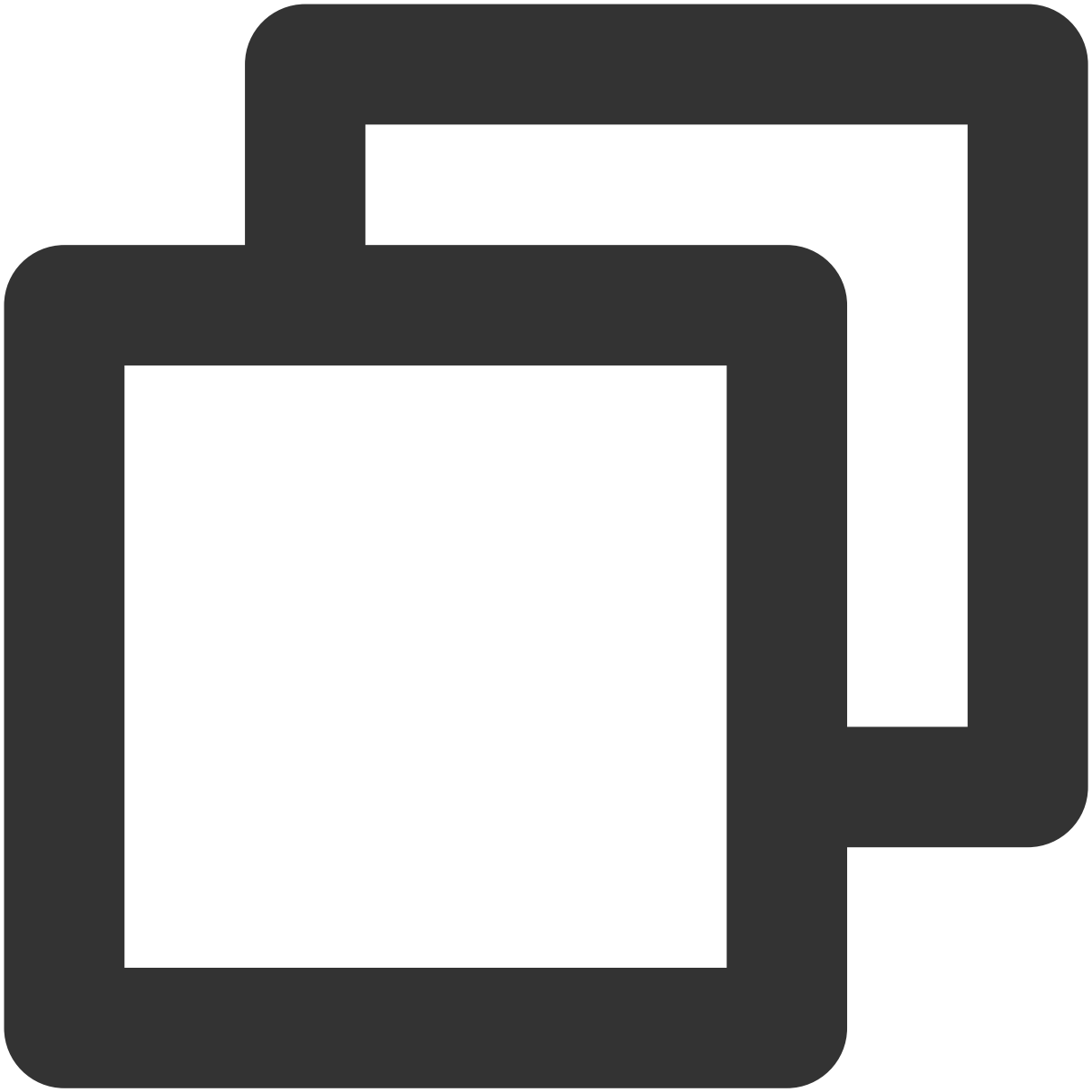
检查详情

源库表的 `row_format` 不能为 `FIXED` 。
源库和目标库 `lower_case_table_names` 变量必须一致。
目标库 `max_allowed_packet` 参数设置至少为4M。
源库变量 `connect_timeout` 必须大于10。
在 MySQL/TDSQL MySQL/TDSQL-C 迁移到 MySQL 的场景中，如果源实例存在耗时较长的 SQL 在运行，则会触发警告，提示“源实例有耗时较长的 SQL 在运行，可能导致锁表，请稍后重试或对源实例中的 SQL 进行处理”。

修复方法

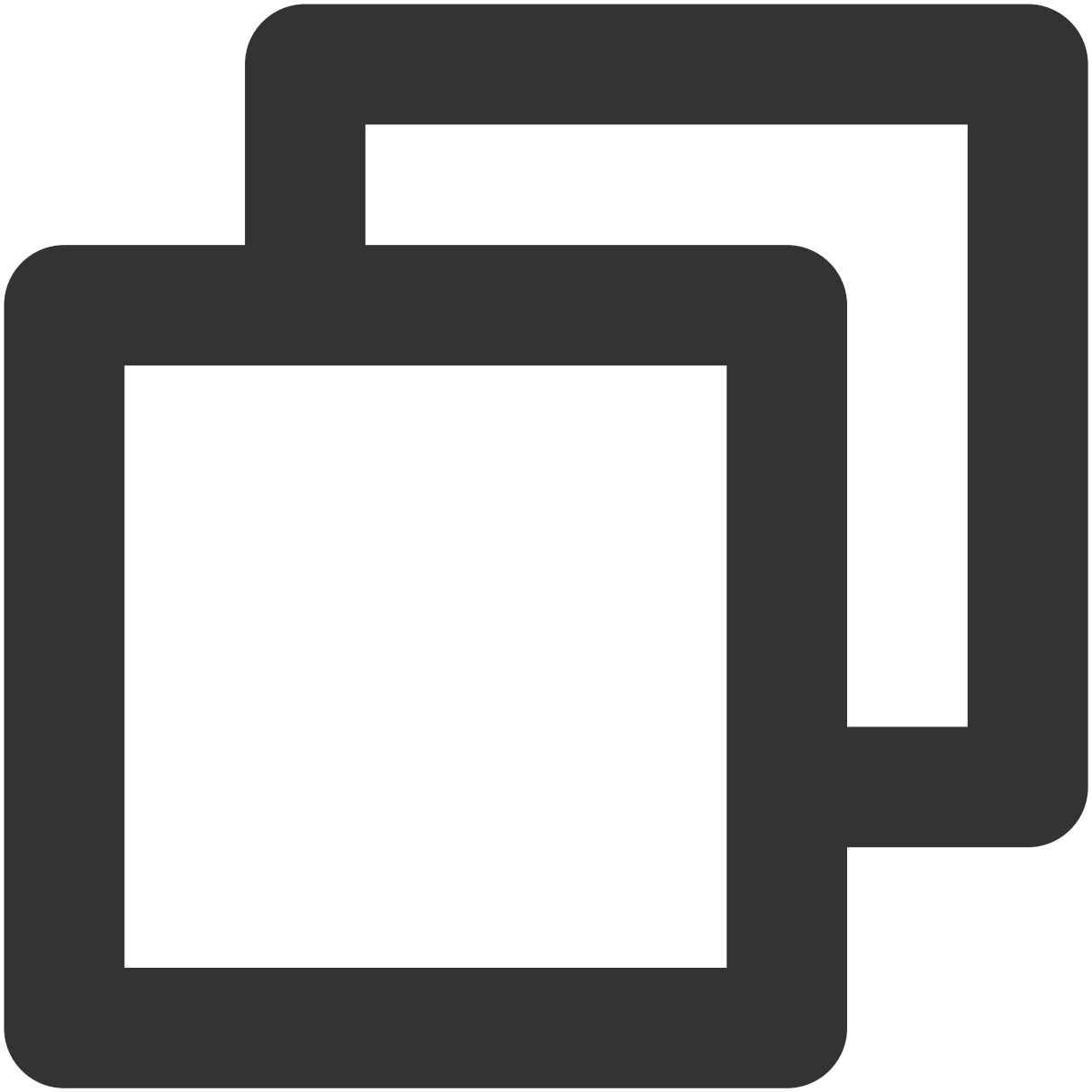
修改源库 row_format 参数

数据库中表的 `row_format` 的取值为 `FIXED` 时，表格中每行的存储长度超过限制值时会溢出，发生报错。因此需要修改为其他模式，如 `DYNAMIC` ，使每行的存储长度会随内容的长度而变化。
如发生此类报错，请参考如下指导进行修复：
1. 登录源数据库。
2. 修改 `row_format` 参数为 `DYNAMIC` 。



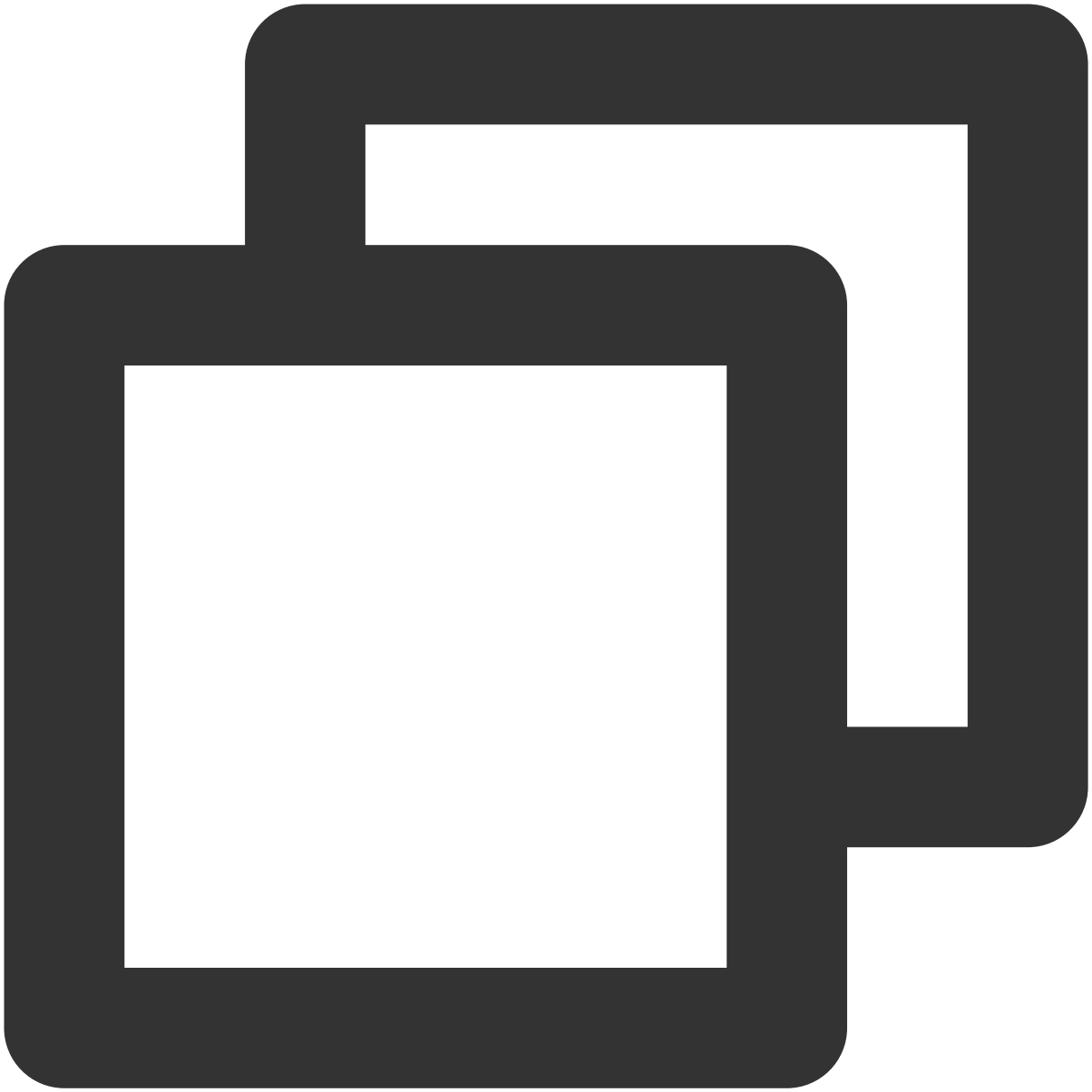
```
alter table table_name row_format = DYNAMIC;
```

3. 查看配置是否生效。



```
show table status like 'table_name'\\G;
```

系统显示结果类似如下：



```
mysql> show table status like 'table_name'\\G;
*************************** 1. row ***************************
        Name: table_name
      Engine: InnoDB
     Version: 10
  Row_format: Dynamic
        Rows: 5
        ......
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库和目标库 lower_case_table_names 变量保持一致

`lower_case_table_names` 是 MySQL 设置大小写是否敏感的一个参数，不同的取值情况如下：

Windows 或 macOS 环境对大小写是不敏感的，但是 Linux 环境却是敏感的，为了保证不同系统的兼容性，需要将大小写敏感规则设置统一。

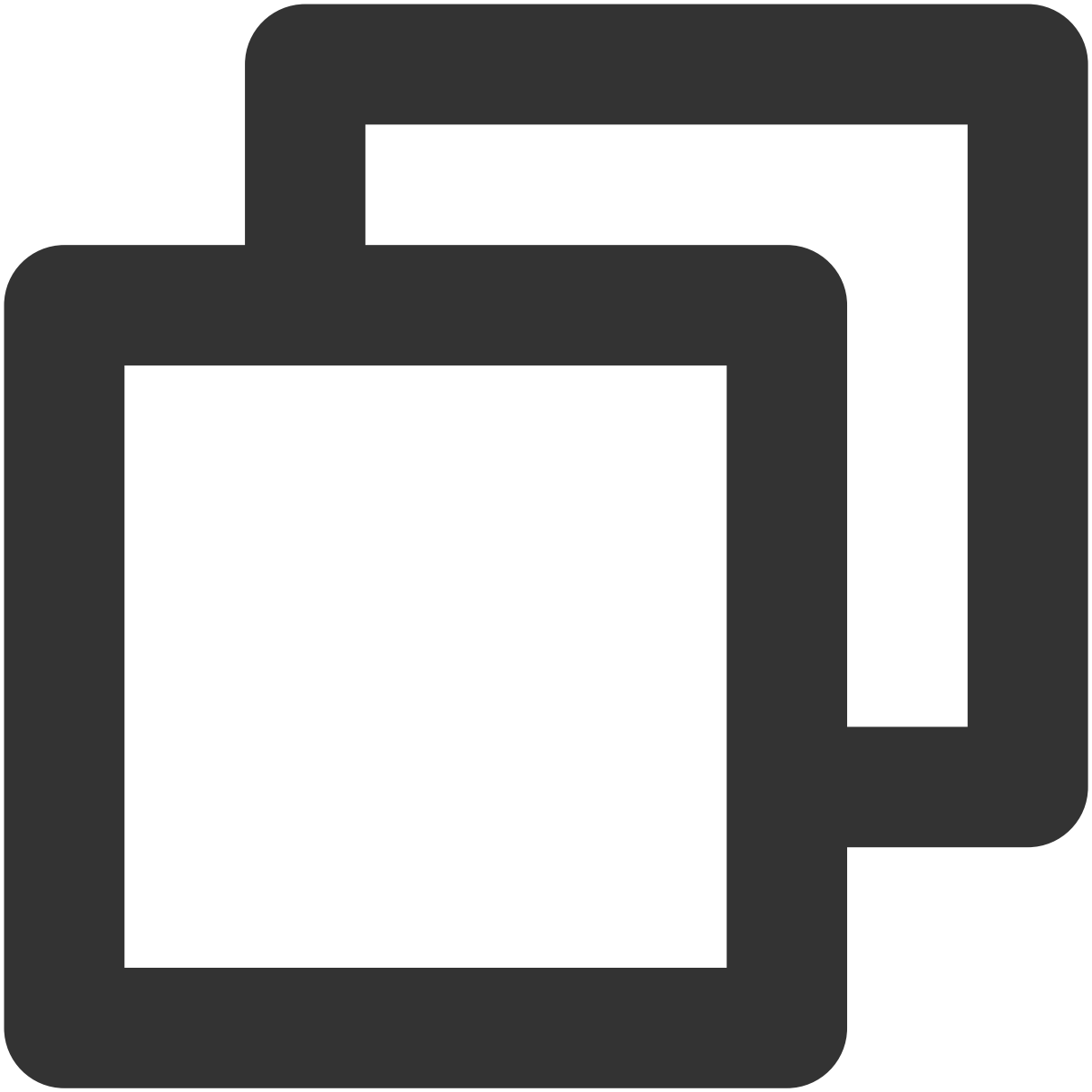
0：表名存储为给定的大小写，比较的时候区分大小写。

1：表名存储在磁盘是小写的，比较的时候不区分大小写。

2：表名存储为给定的大小写，比较的时候是小写的。

如发生此类报错，请参考如下指导将源库和目标库的参数改为统一。

1. 登录源数据库。
2. 查看源库和目标库中的 `lower_case_table_names` 取值。

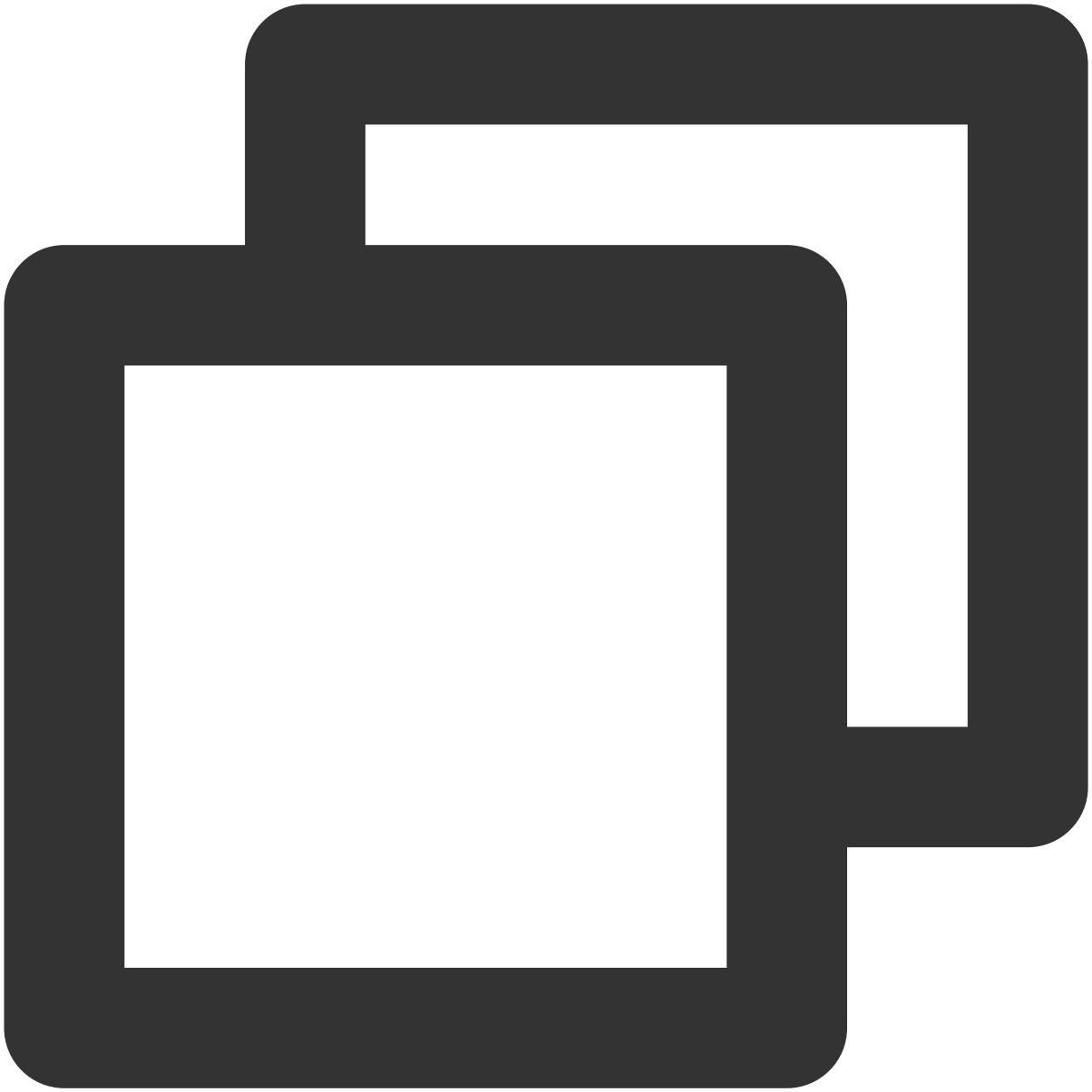


```
show variables like '%lower_case_table_names%';
```

3. 参考如下内容修改源数据库的配置文件 `my.cnf` 。

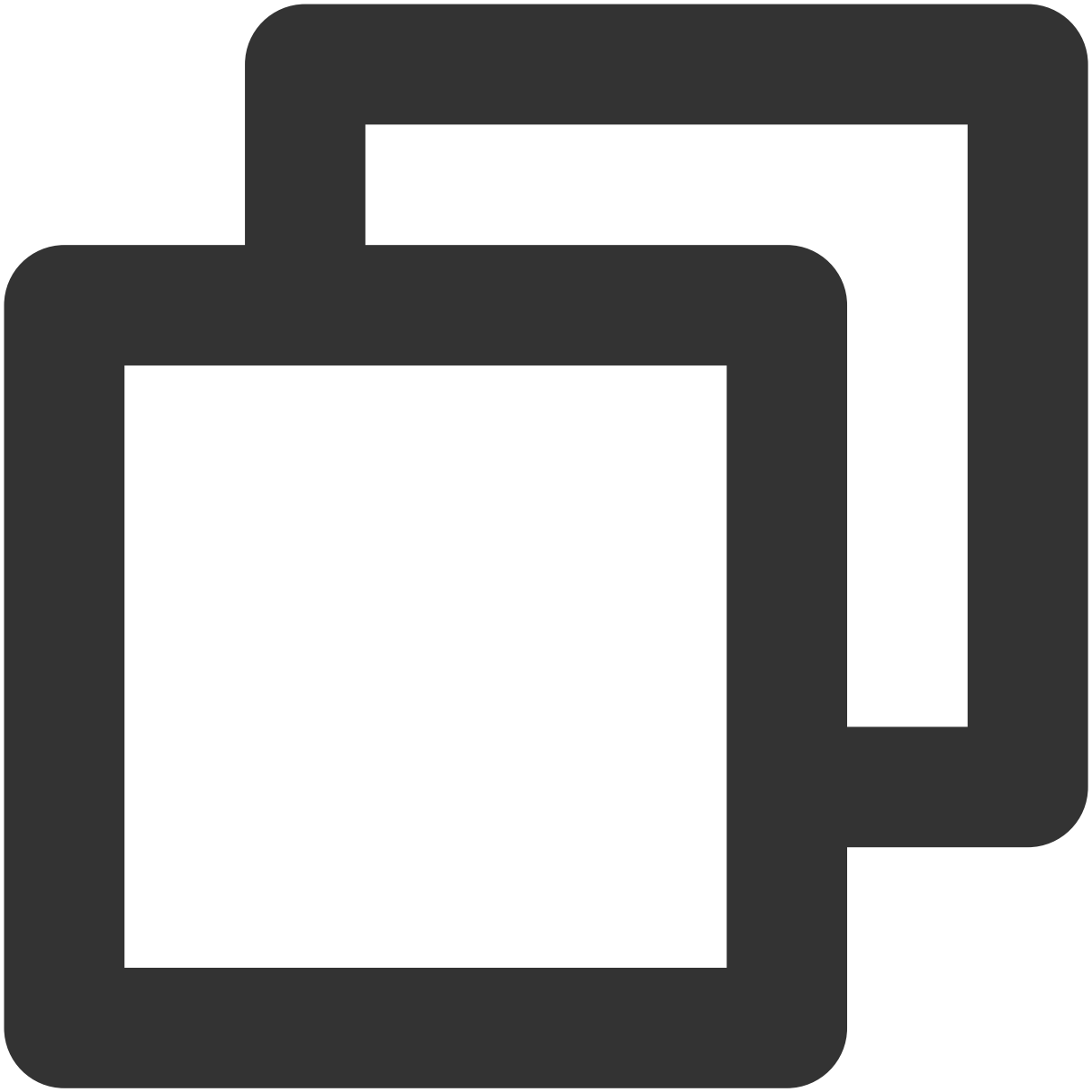
说明

`my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。



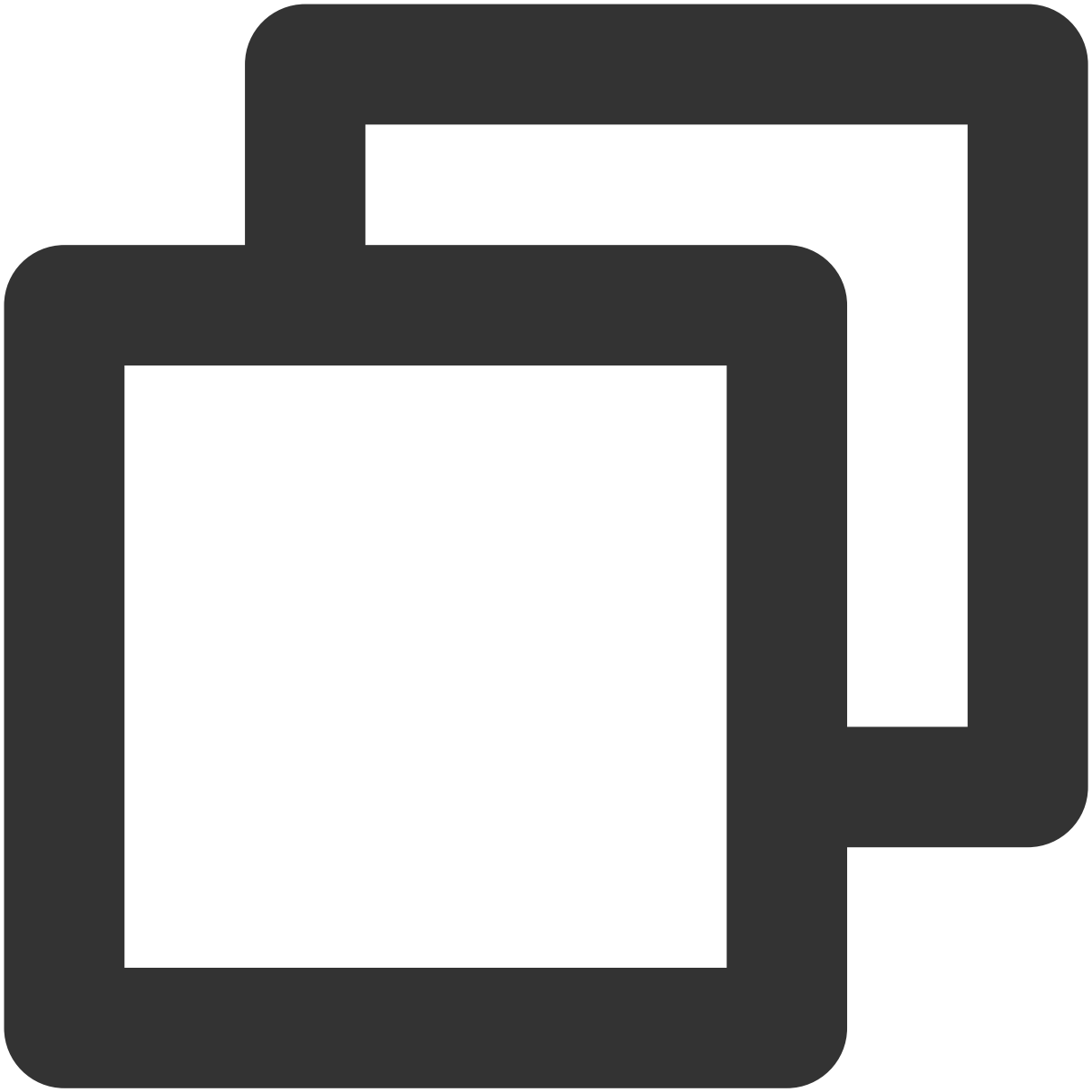
```
lower_case_table_names = 1
```

4. 参考如下命令重启数据库。



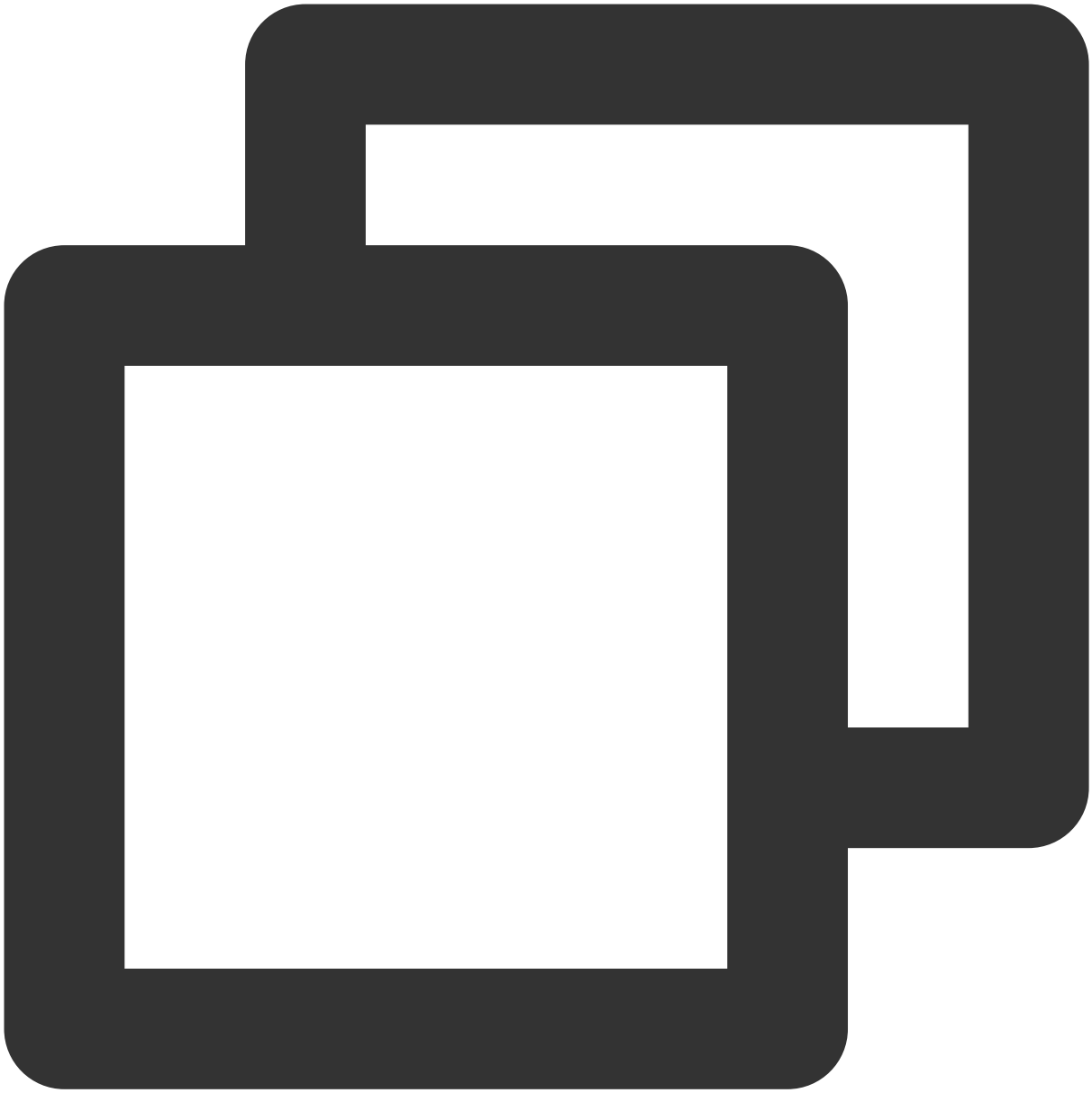
```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

5. 查看配置是否生效。



```
show variables like '%lower_case_table_names%';
```

系统显示结果类似如下：



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

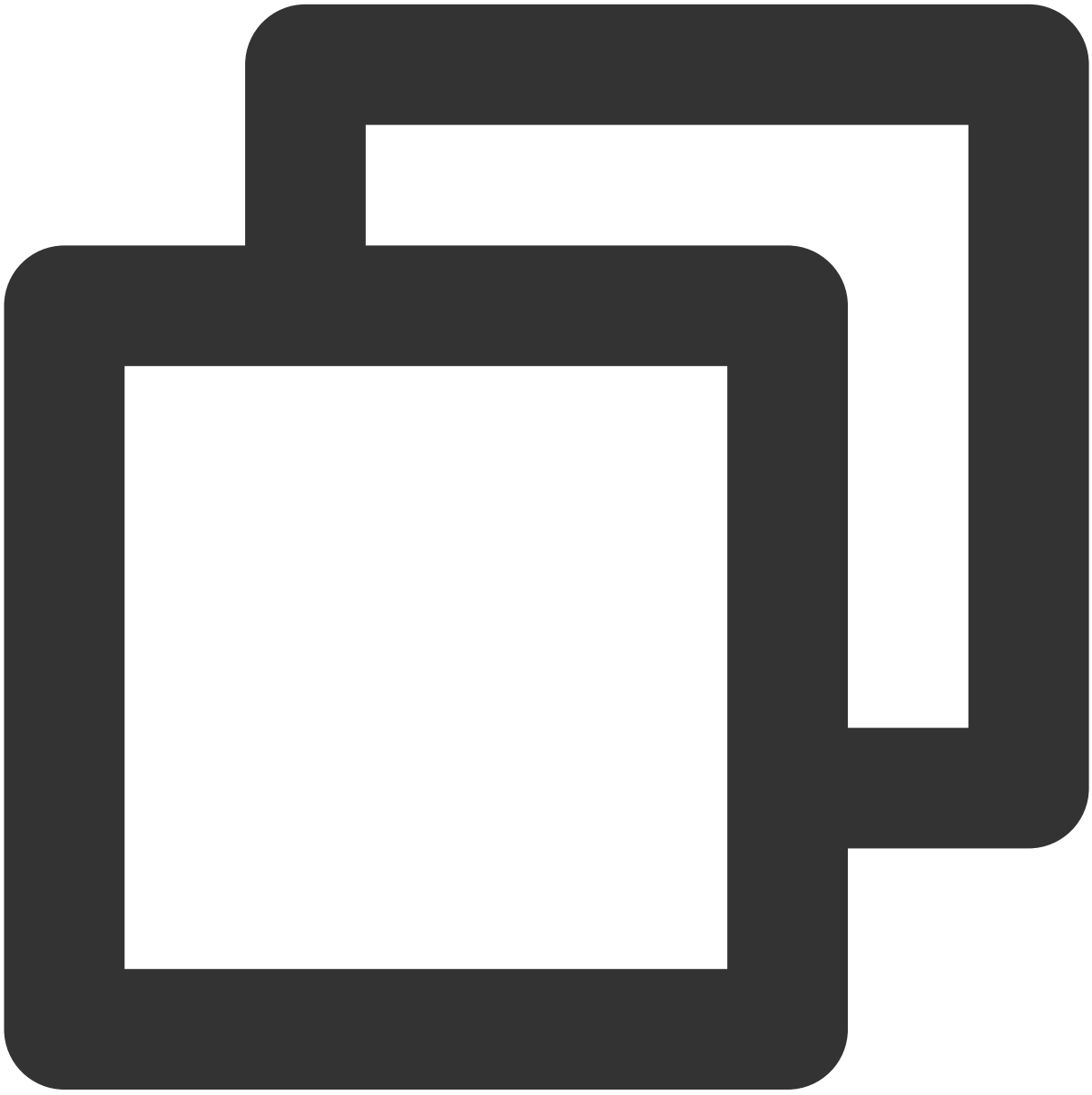
6. 重新执行校验任务。

修改目标库 max_allowed_packet 参数

`max_allowed_packet` 为最大允许的传输包。设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络包，影响系统性能。

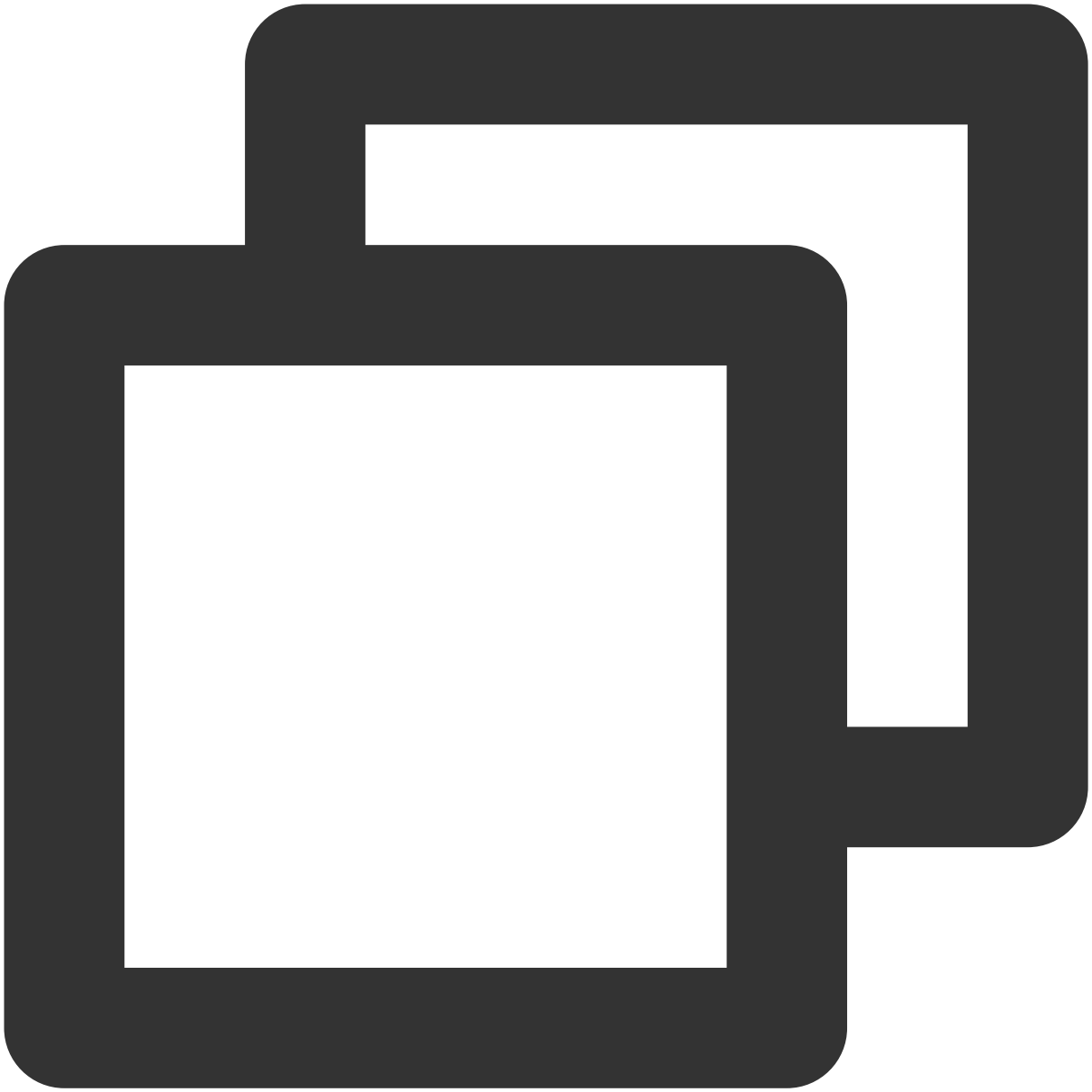
如发生此类报错，请参考如下指导进行修复：

1. 登录目标数据库。
2. 修改 `max_allowed_packet` 参数。



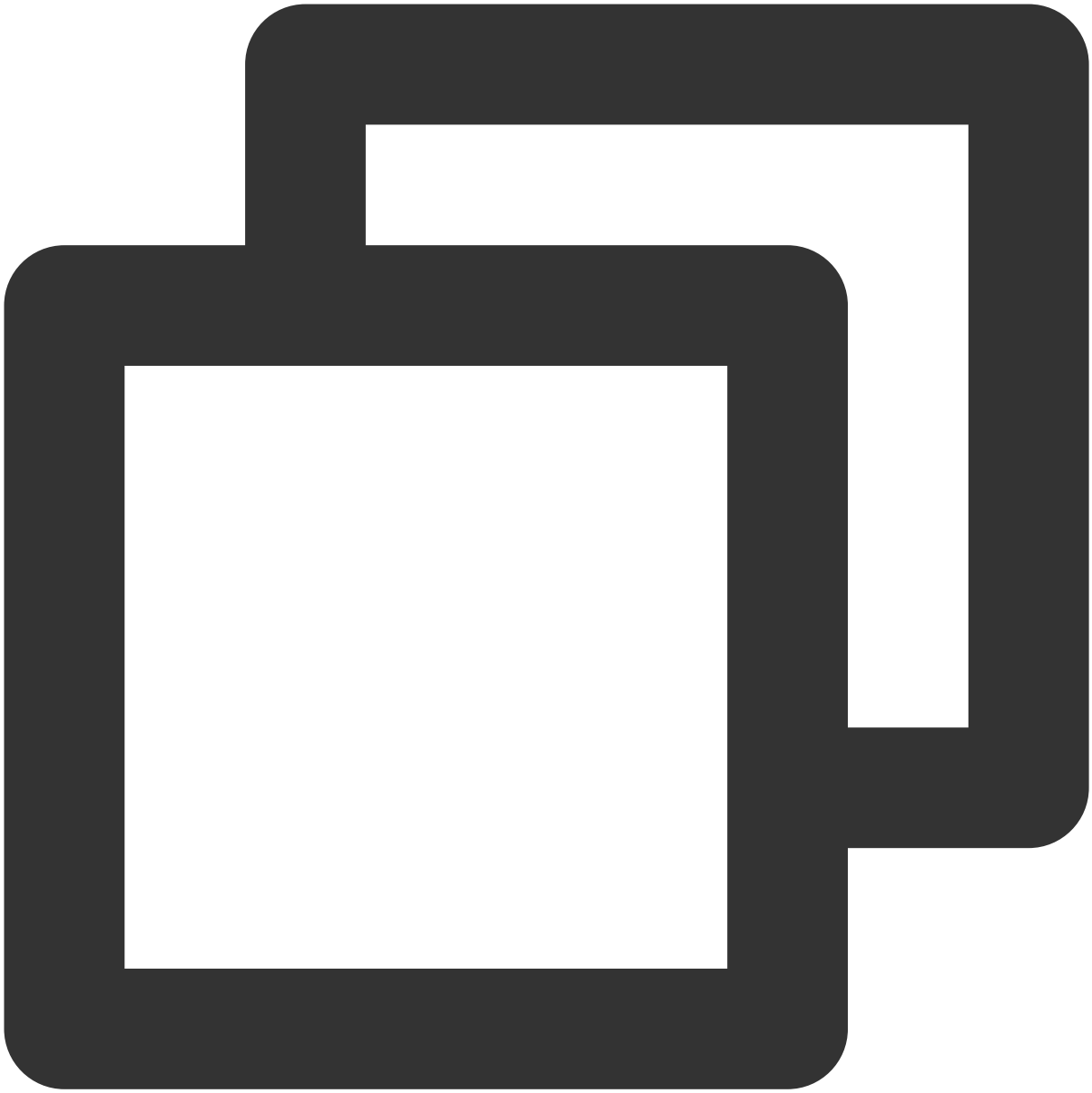
```
set global max_allowed_packet = 4*1024*1024;
```

3. 查看配置是否生效。



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

系统显示结果类似如下：



```
mysql> show global variables like '%max_allowed_packet%';
+--------------------------+---------+
| Variable_name            | Value   |
+--------------------------+---------+
| max_allowed_packet       | 4194304 |
+--------------------------+---------+
1 row in set (0.00 sec)
```

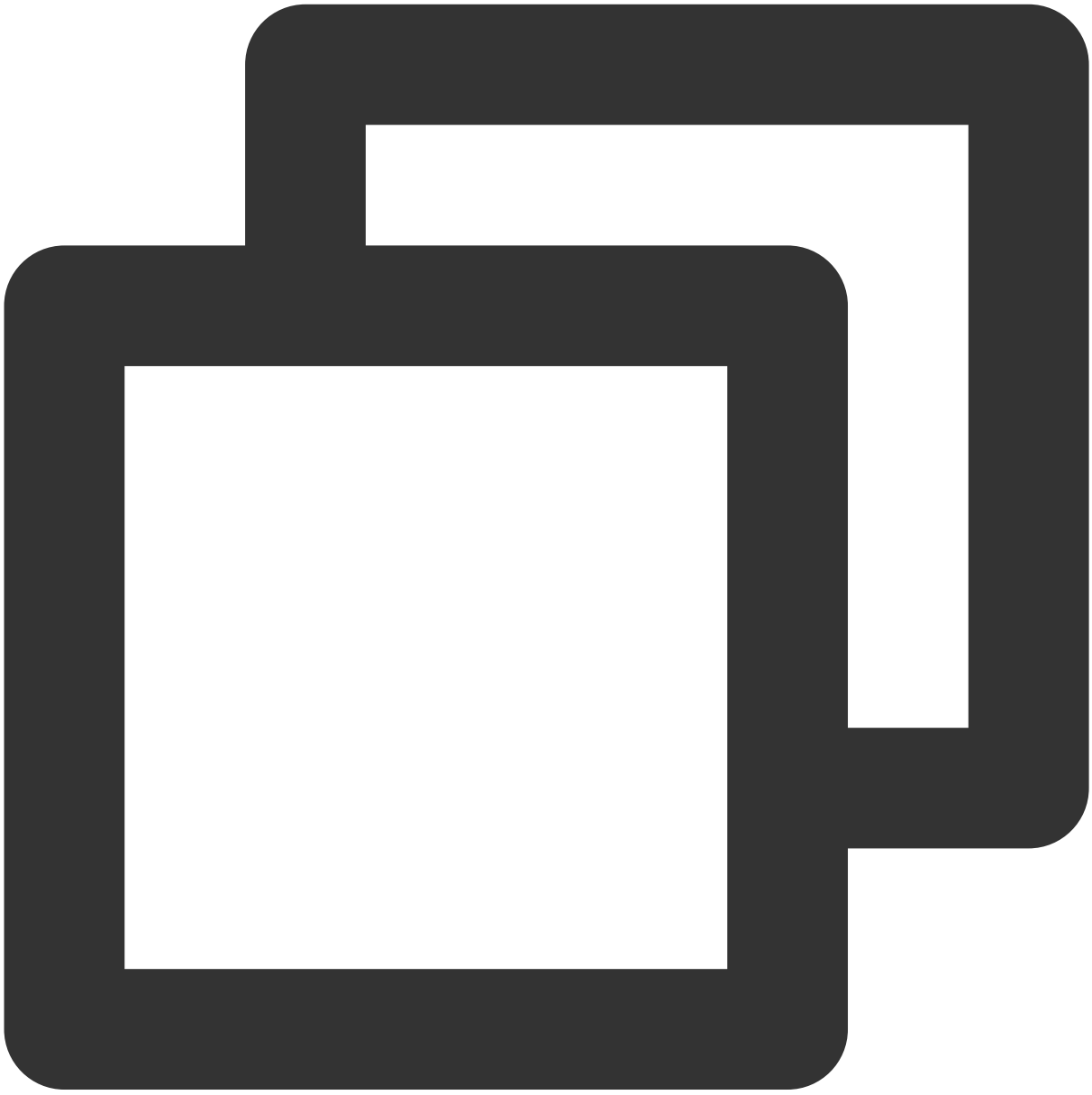
4. 重新执行校验任务。

修改源库变量 connect_timeout

`connect_timeout` 为数据库的连接时间，超过 `connect_timeout` 设置值的连接请求将会被拒绝。如果设置过小，会导致数据库连接频繁断开，影响处理效率，因此建议该参数取值大于10。

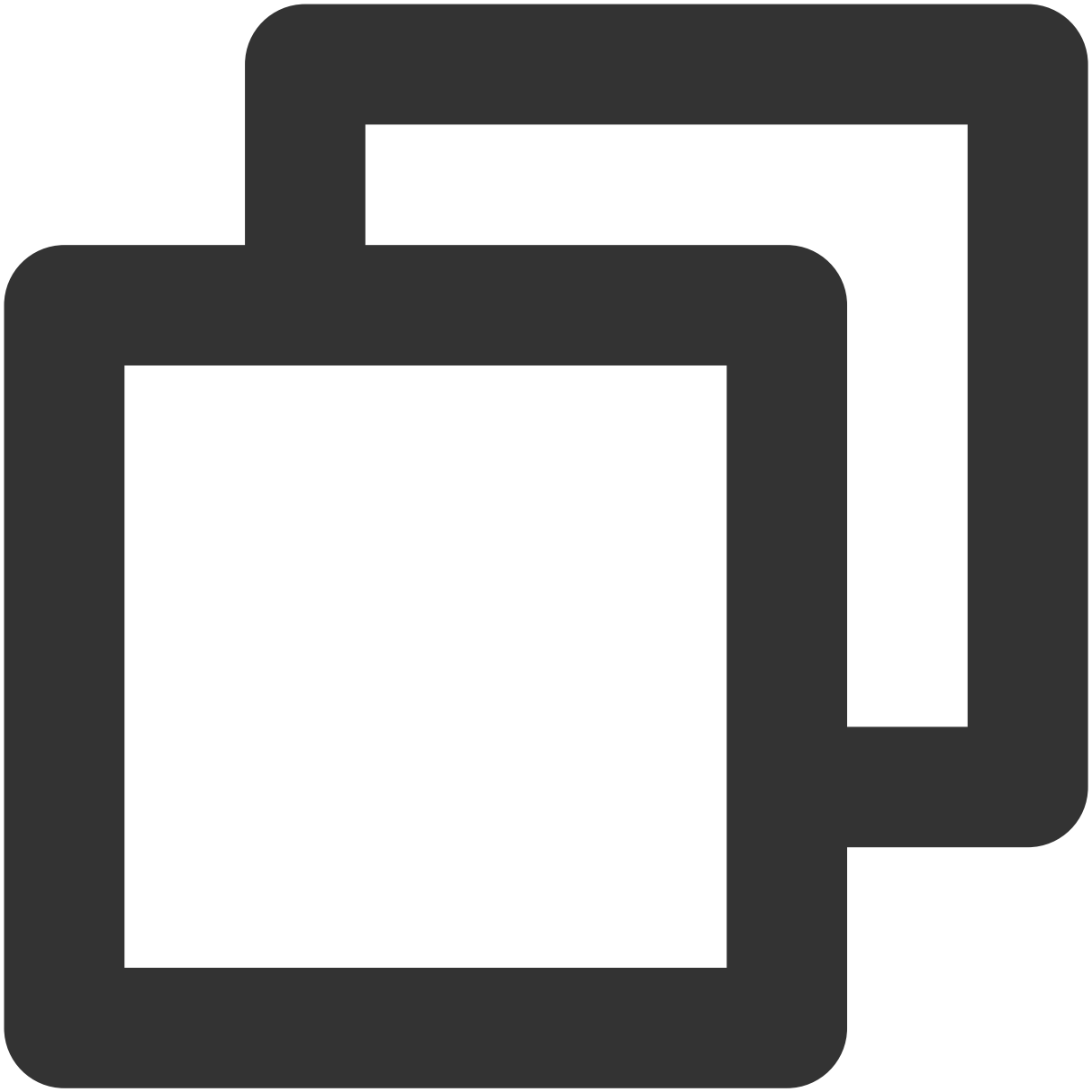
如发生此类报错，请参考如下指导进行修复：

1. 登录源数据库。
2. 修改 `connect_timeout` 参数。



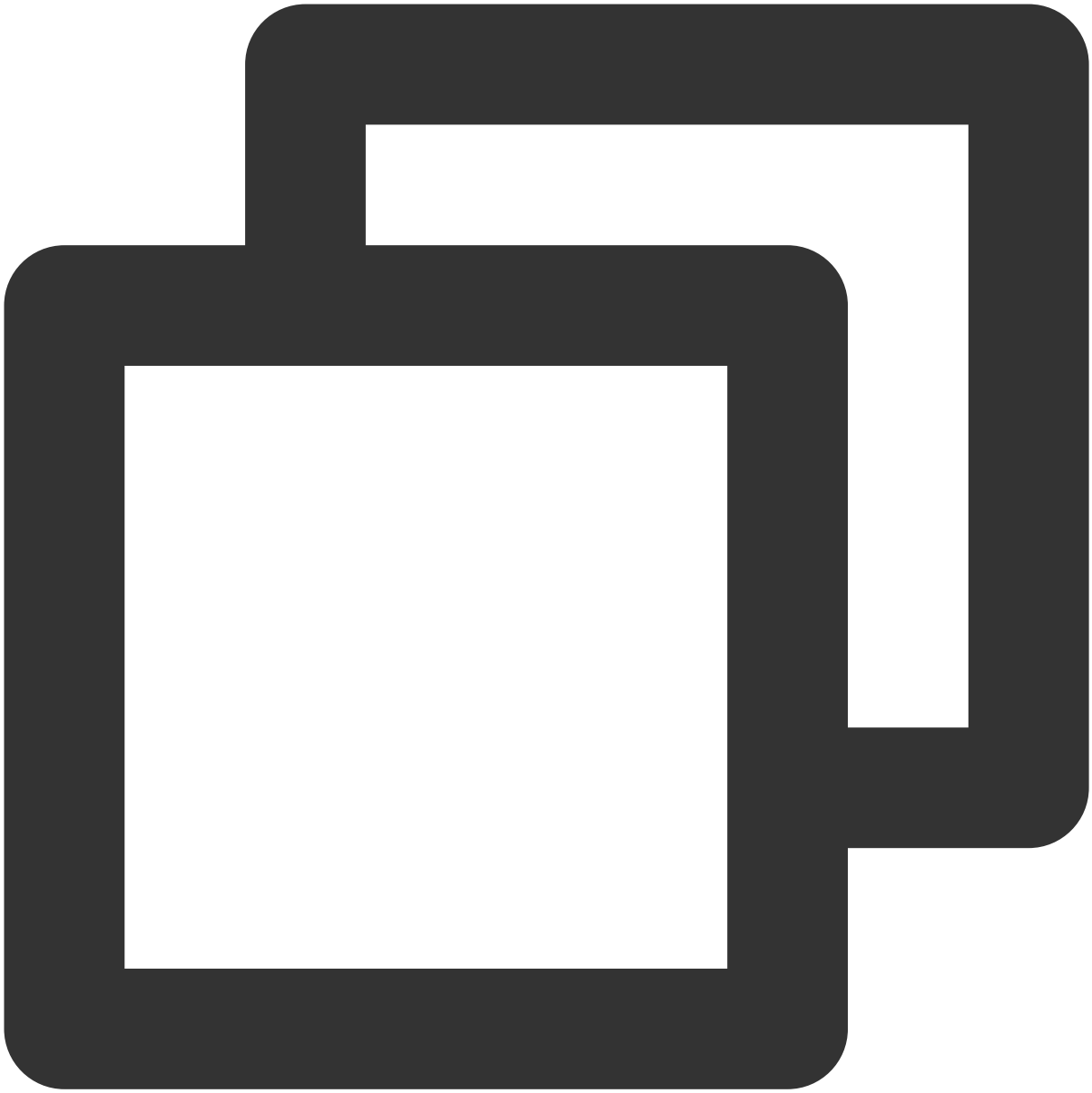
```
set global connect_timeout = 10;
```

3. 查看配置是否生效。



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

系统显示结果类似如下：



```
mysql> show global variables like '%connect_timeout%';
+-----------------------+---------+
| Variable_name         | Value   |
+-----------------------+---------+
| connect_timeout       | 10      |
+-----------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

目标实例内容冲突检查

最近更新时间：2024-02-19 16:08:36

检查详情

目标实例不能有和源库同名的对象。如果存在冲突报错，可任选以下一个方法进行修复。
方法一：使用库表映射。
方法二：修改或删除目标数据库中的同名对象。
方法三：从迁移对象中移除同名对象。
整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。
选择高级对象时， 目标库不能有冲突的高级对象。如果存在冲突报错，需要删除冲突的对象。

修复方法

使用库表映射（仅适用于 MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL）

使用 DTS 库表映射功能，将同名的待迁移对象映射为目标数据库中的其他名称。
1. 登录 DTS 控制台，选择对应的迁移任务，在**操作**列，选择**更多 > 修改**。
2. 在选择迁移对象右侧“已选对象”中，将鼠标悬浮在需要修改的对象上即可显示编辑按钮，然后重命名对象。
3. 重新执行校验任务。

修改目标数据库中的同名对象

登录目标数据库，重命名或删除目标数据库中和迁移对象同名的对象。

从迁移对象中移除同名对象

修改迁移任务配置，从迁移对象中移除同名对象，该对象不进行数据迁移。
1. 登录 DTS 控制台，选择对应的迁移任务，在**操作**列，选择**更多 > 修改**。
2. 在迁移对象中，移除同名的对象。
3. 重新执行校验任务。

删除目标库中的内容

登录目标数据库，删除目标数据库中的同名对象或者整库内容，然后重新执行校验任务。

目标实例空间检查

最近更新时间：2024-02-19 16:09:25

检查详情

目标库存储空间需要在源库待迁移库表空间的1.2倍以上。
全量数据迁移会并发执行 INSERT 操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间很可能会比源实例的表存储空间大。

修复方法

删除目标库中的部分数据，以便腾出足够的空间。
升级目标库存储规格，使用更大容量的实例进行迁移。

Binlog 参数检查

最近更新时间：2024-02-19 16:09:44

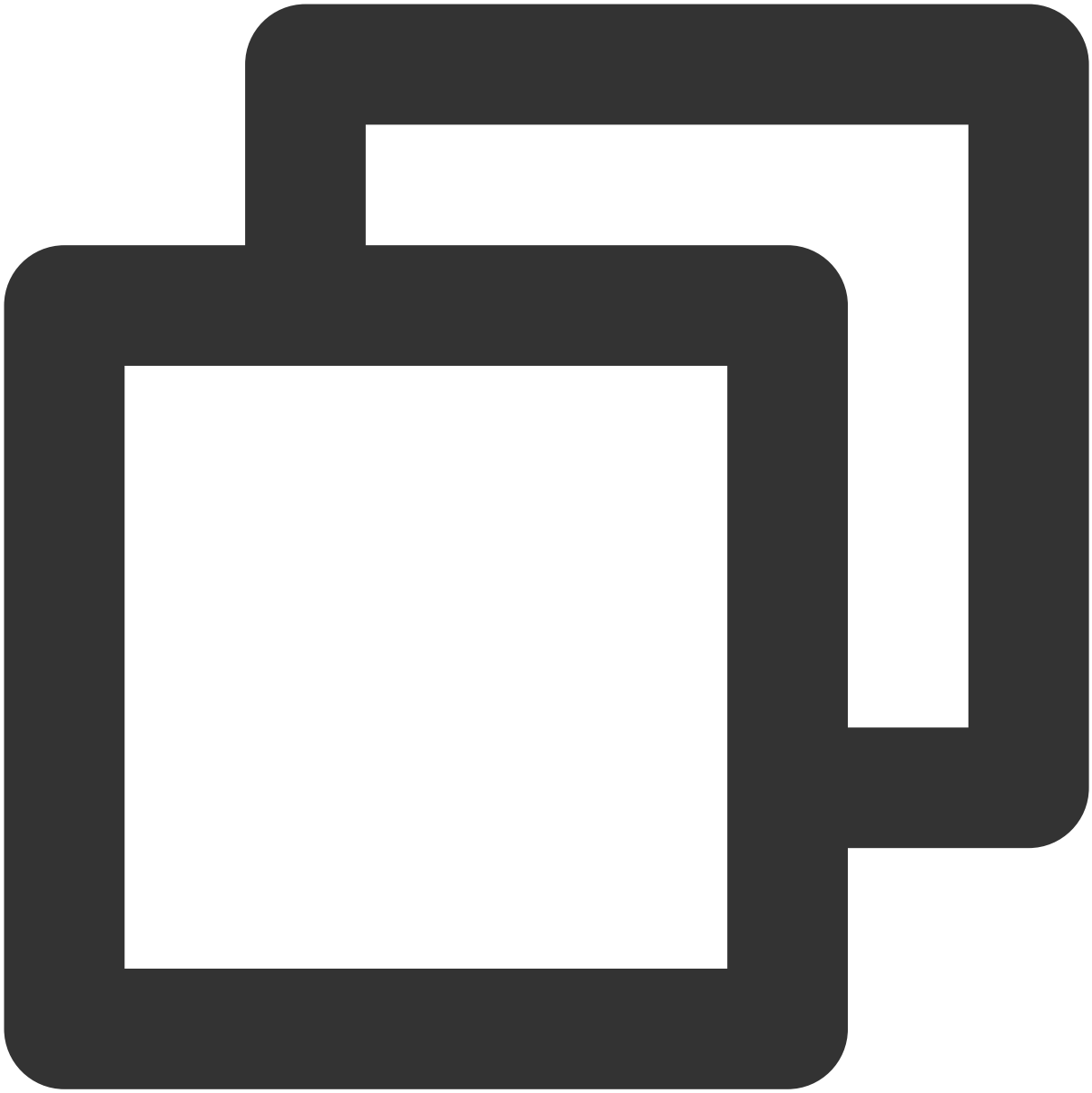
检查详情

源数据库 binlog 相关参数需要按照如下要求配置，如果校验不通过，请参考本文后续指导进行修复。
`log_bin` 变量必须设置为 `ON` 。
`binlog_format` 变量必须设置为 `ROW` 。
`binlog_row_image` 必须设置为 `FULL` 。
如果源数据库为 MySQL 5.6 及以上版本， `gtid_mode` 只支持设置为 `ON` 和 `OFF` ，建议将 `gtid_mode` 设置为 `ON` ，设置为 `OFF` 会报警告，设置为 `ON_PERMISSIVE` 和 `OFF_PERMISSIVE` 会报错。
`server_id` 参数需要手动设置，且值不能设置为0。
不允许设置 `do_db` ， `ignore_db` 。
对于源实例为从库的情况， `log_slave_updates` 变量必须设置为 `ON` 。
建议源库 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。

修复方法

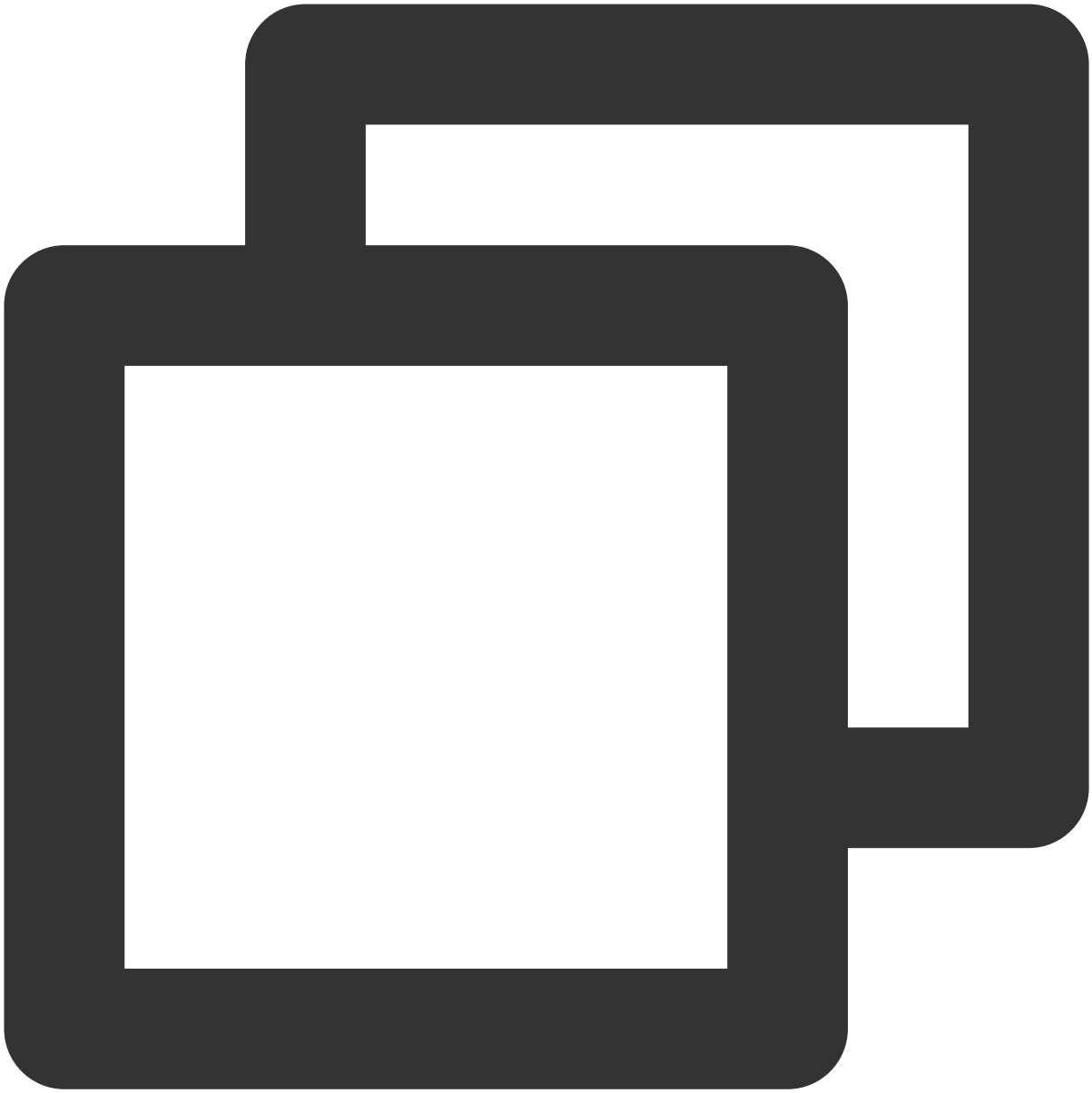
开启 binlog

`log_bin` 是 binlog 的开关控制参数，需要将 binlog 打开，以便记录所有的数据库表结构和表数据变更日志。
如发生类似报错，请参考如下指导进行修复：
1. 登录源数据库。
2. 参考如下内容修改源数据库的配置文件 `my.cnf` 。
因 log_bin 参数修改后需要重启数据库后才能生效，如果 binlog_format、server_id、binlog_row_image、expire_logs_days 这几个参数也在校验阶段提示报错，请一并修改完成后再重启数据库，让所有参数都生效。
说明
`my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。



```
log_bin = MYSQL_BIN
binlog_format = ROW
server_id = 2      //建议设为大于1的整数，此处仅为示例
binlog_row_image = FULL
expire_logs_days = 3      //修改 binlog 的保留时间，建议大于等于3天
```

3. 参考如下命令重启源数据库。

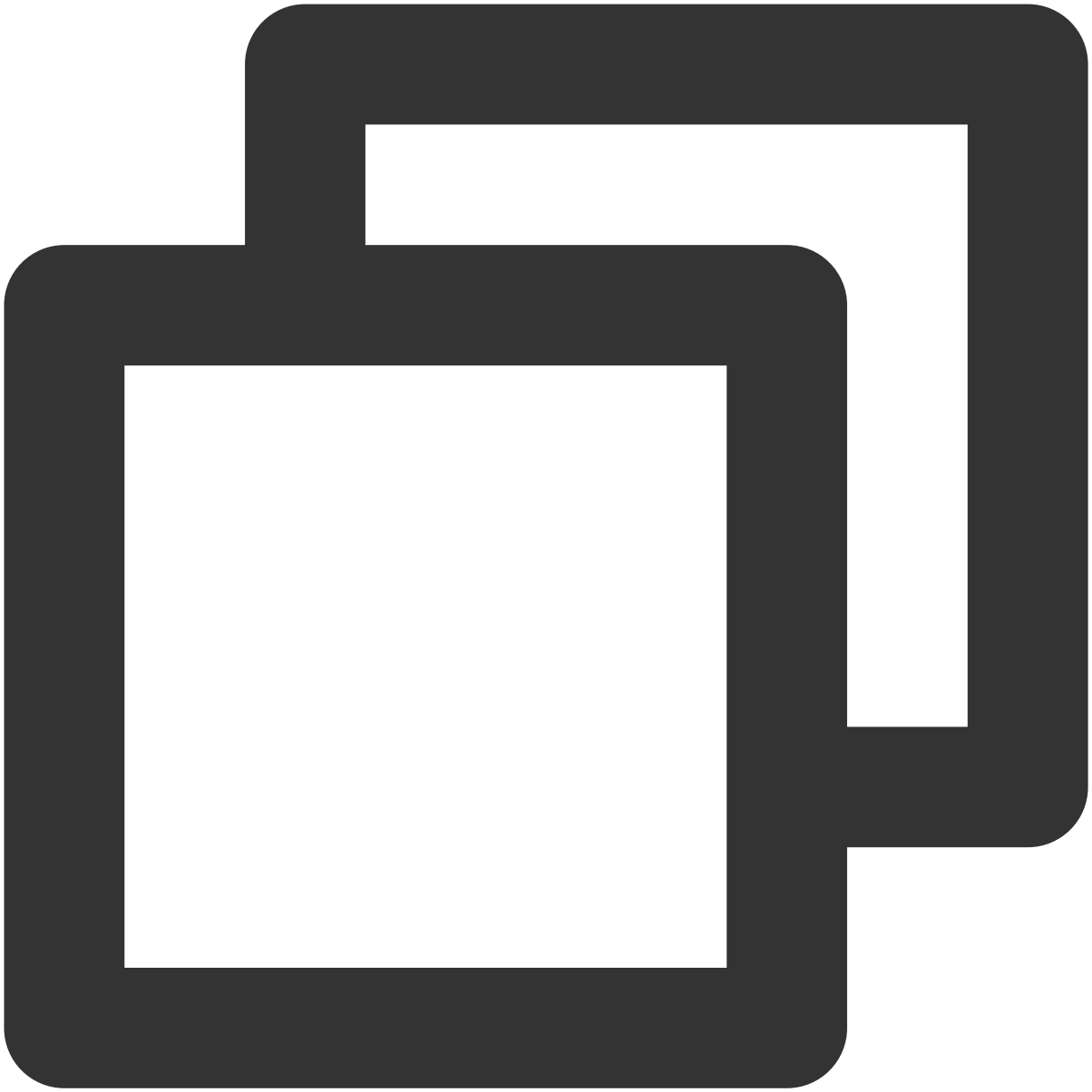


```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

说明

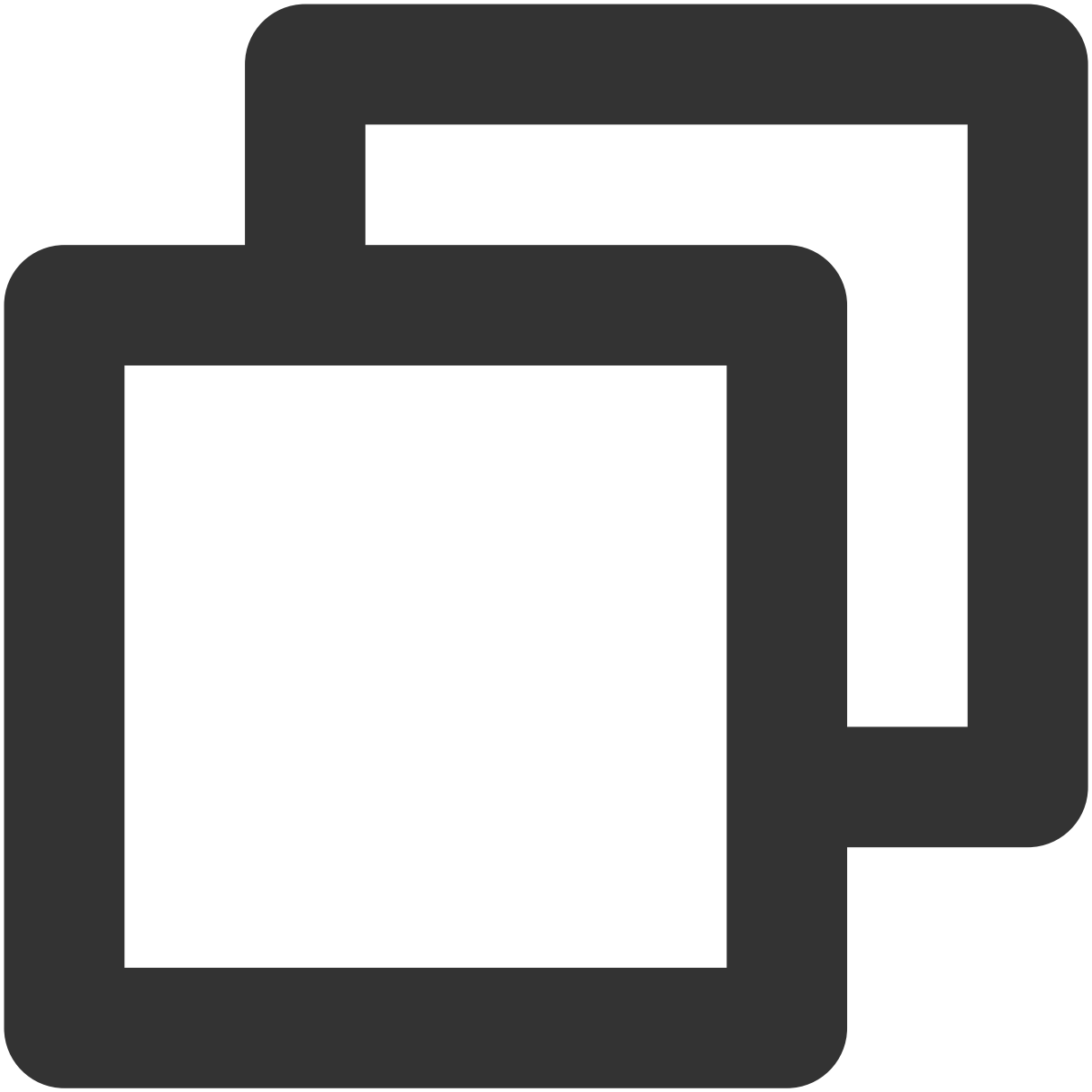
[$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认 binlog 功能是否已启用。



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

系统显示结果类似如下：



```
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. 重新执行校验任务。

修改 binlog_format 参数

`binlog_format` 为 binlog 的记录模式，有以下三种：

`STATEMENT` ：每一条会修改数据的 SQL 都会记录到 master 的 binlog 中，slave 在复制的时候，会执行和原来 master 端相同的 SQL。该模式可以减少 binlog 日志量，但是对某些特定的函数进行复制时，slave 端不能正确复制。

`ROW` ：binlog 日志中会记录成每一行数据修改的形式，然后在 slave 端再对相同的数据进行修改。该模式会保证 master 和 slave 的正确复制，但是 binlog 日志量会增加。

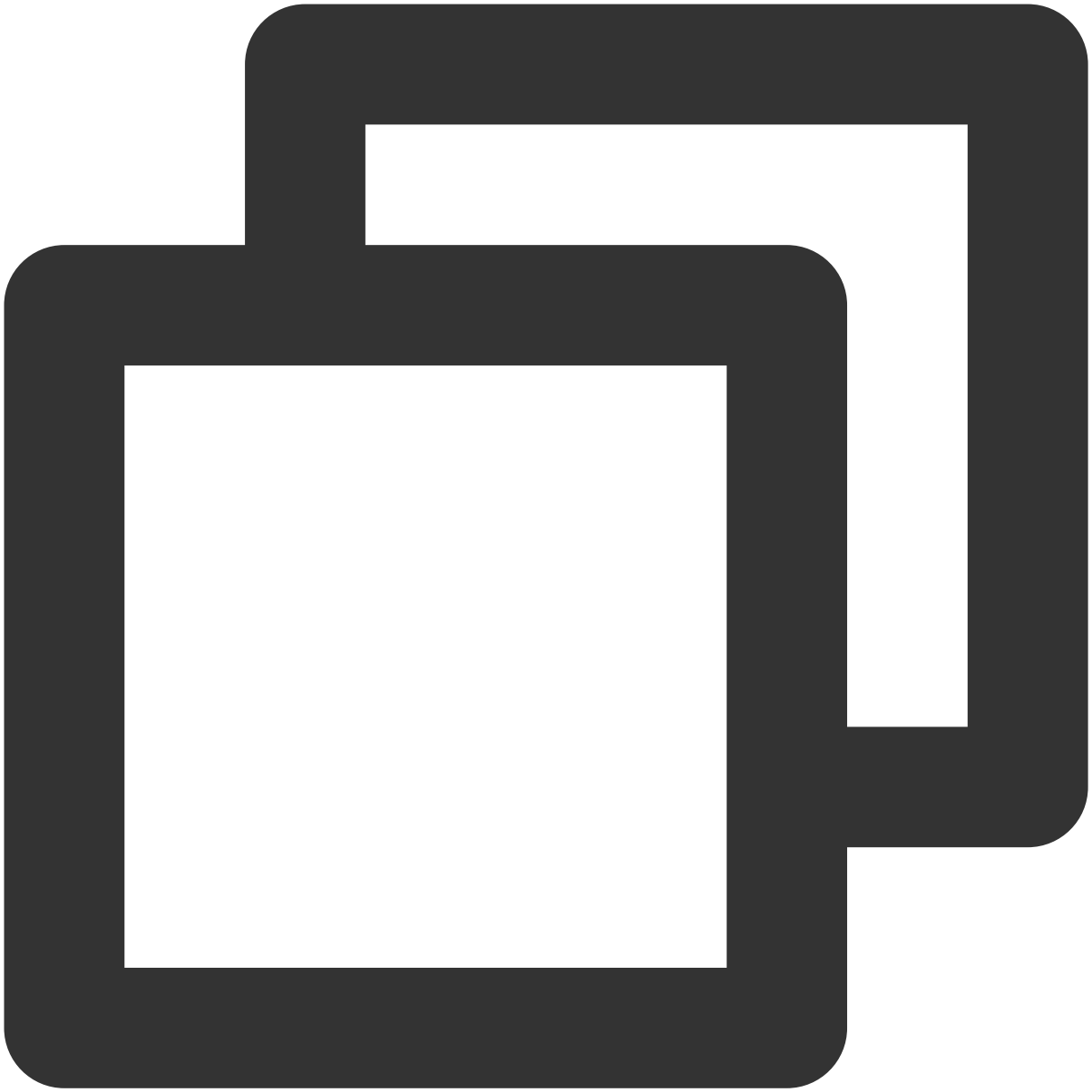
`MIXED` ：前两种模式的结合，MySQL 会根据执行的每一条具体的 SQL 语句来区分对待记录的日志形式，在 `STATEMENT` 和 `ROW` 之间选择一种。

综上，为了保证 master 和 slave 的正确复制， `binlog_format` 参数需要设置为 `ROW` 。如发生类似报错，请参考如下指导进行修复。

说明

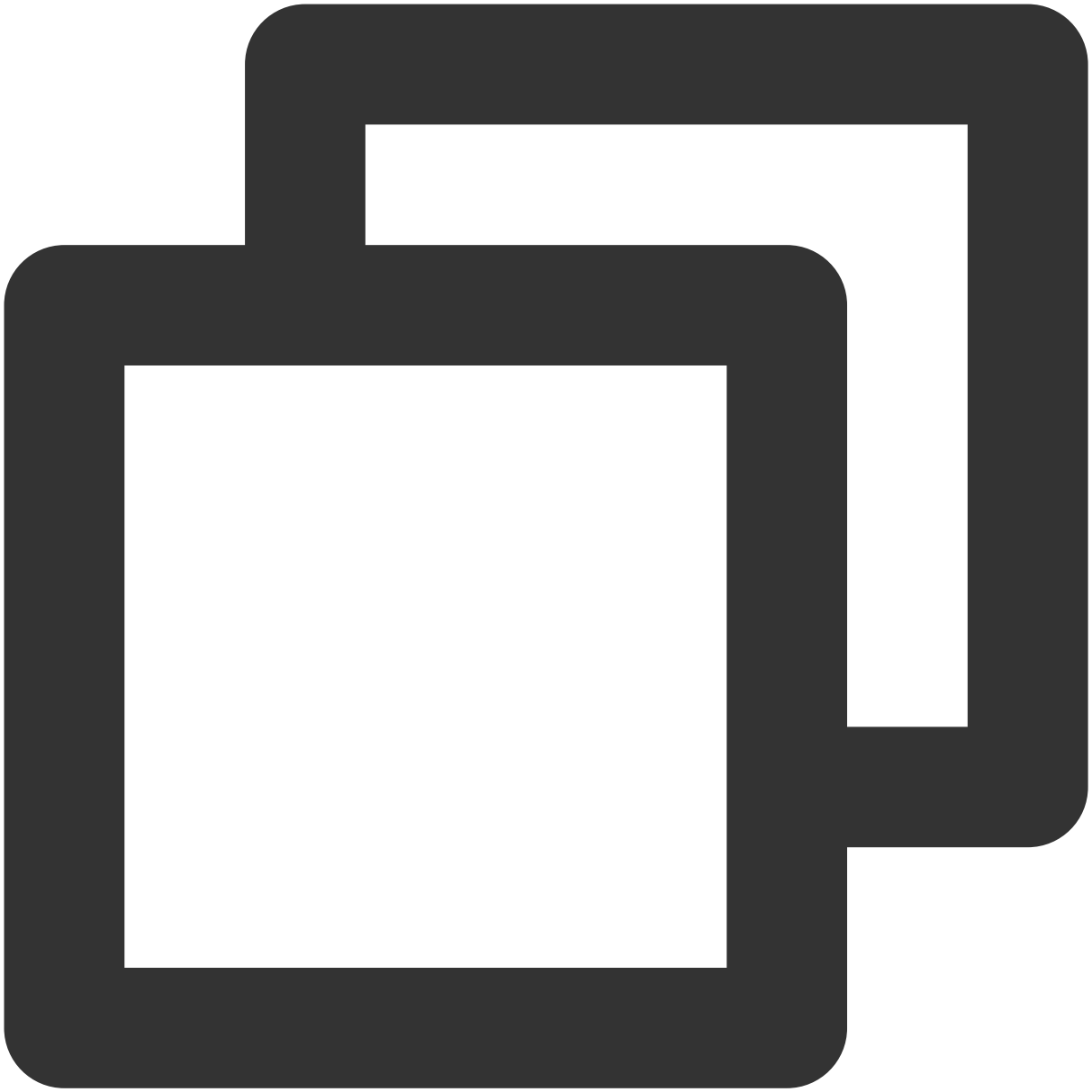
该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。
2. 参考如下命令修改 `binlog_format` 。



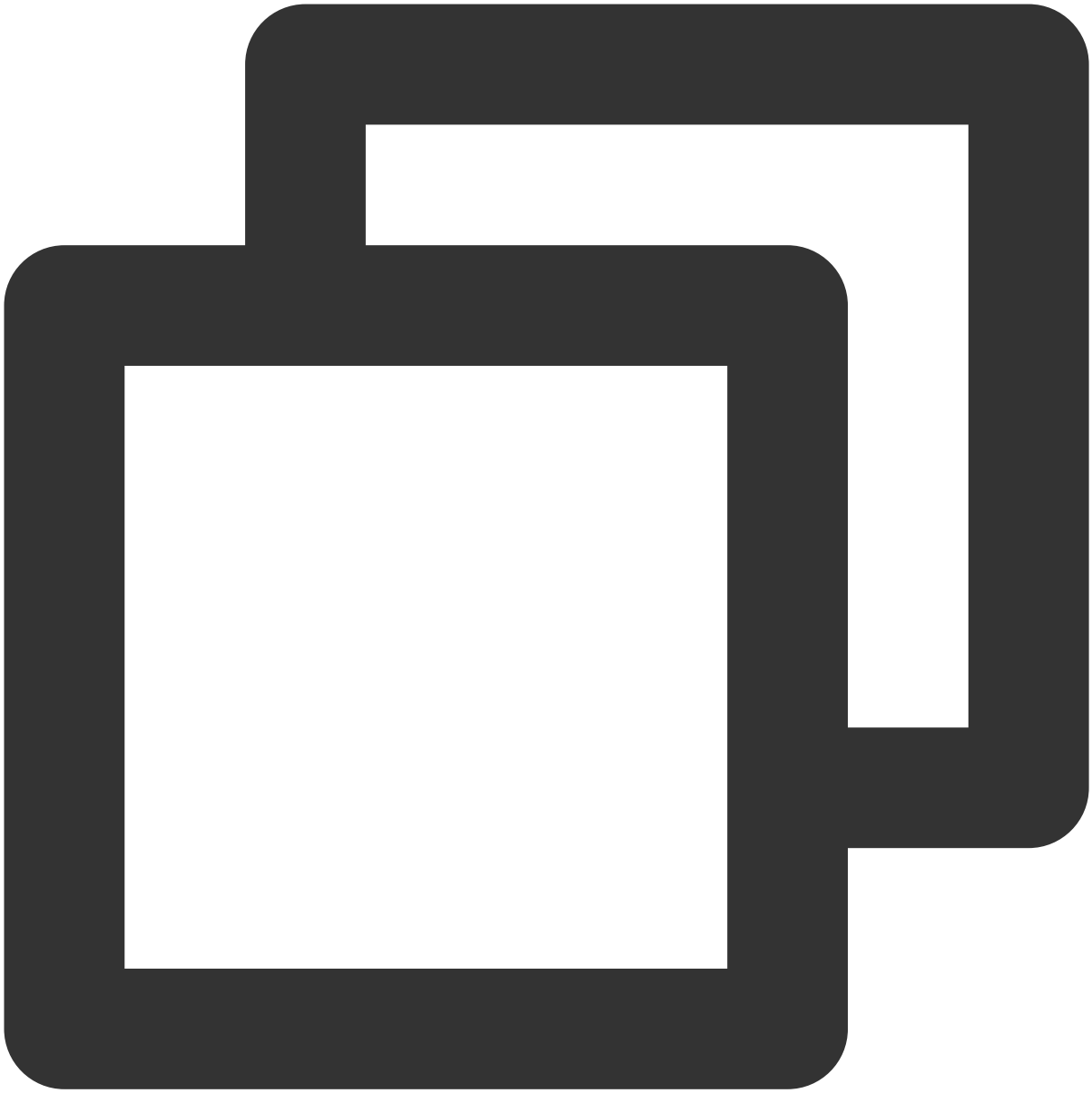
```
set global binlog_format = ROW;
```

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。



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

系统显示结果类似如下：



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

4. 重新执行校验任务。

修改 binlog_row_image 参数

`binlog_row_image` 参数决定了 binlog 是如何记录前镜像（记录修改前的内容）和后镜像（记录修改后的内容）的，这会直接影响到数据闪回、主从复制等功能。

`binlog_row_image` 参数只在 `binlog_format` 配置为 `ROW` 模式下生效。具体取值影响如下：

`FULL` ：在 `ROW` 模式下，binlog 会记录前镜像和后镜像的所有列的数据信息。

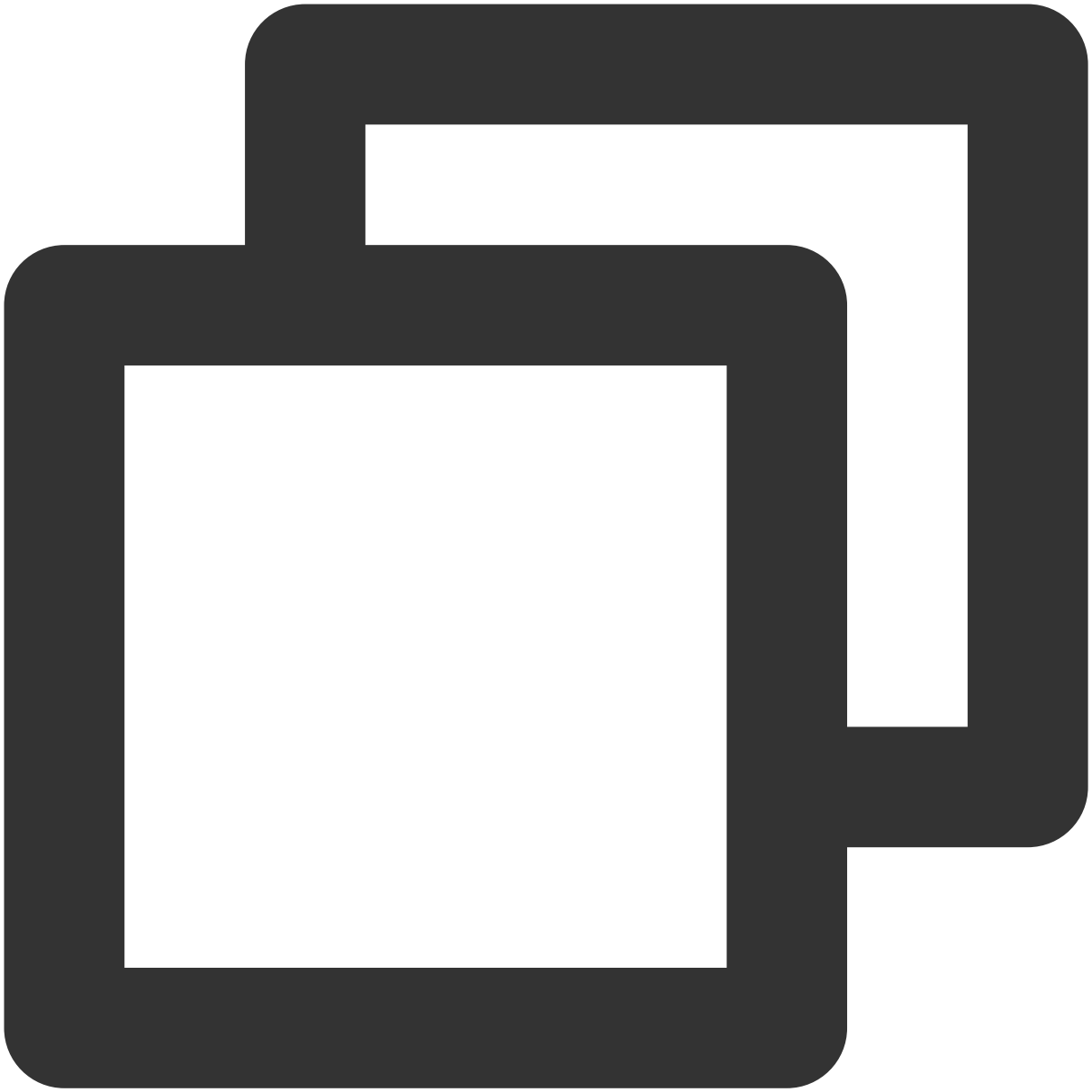
`MINIMAL` ：在 `ROW` 模式下，当表没有主键或唯一键时，前镜像记录所有列，后镜像记录被修改的列；如果存在主键或唯一键，不管是前镜像还是后镜像，都只记录有影响的列。

综上， `binlog_row_image` 需要配置为 `FULL` ，源数据库的 binlog 记录全镜像。如发生报错，请参考如下步骤进行修复。

说明

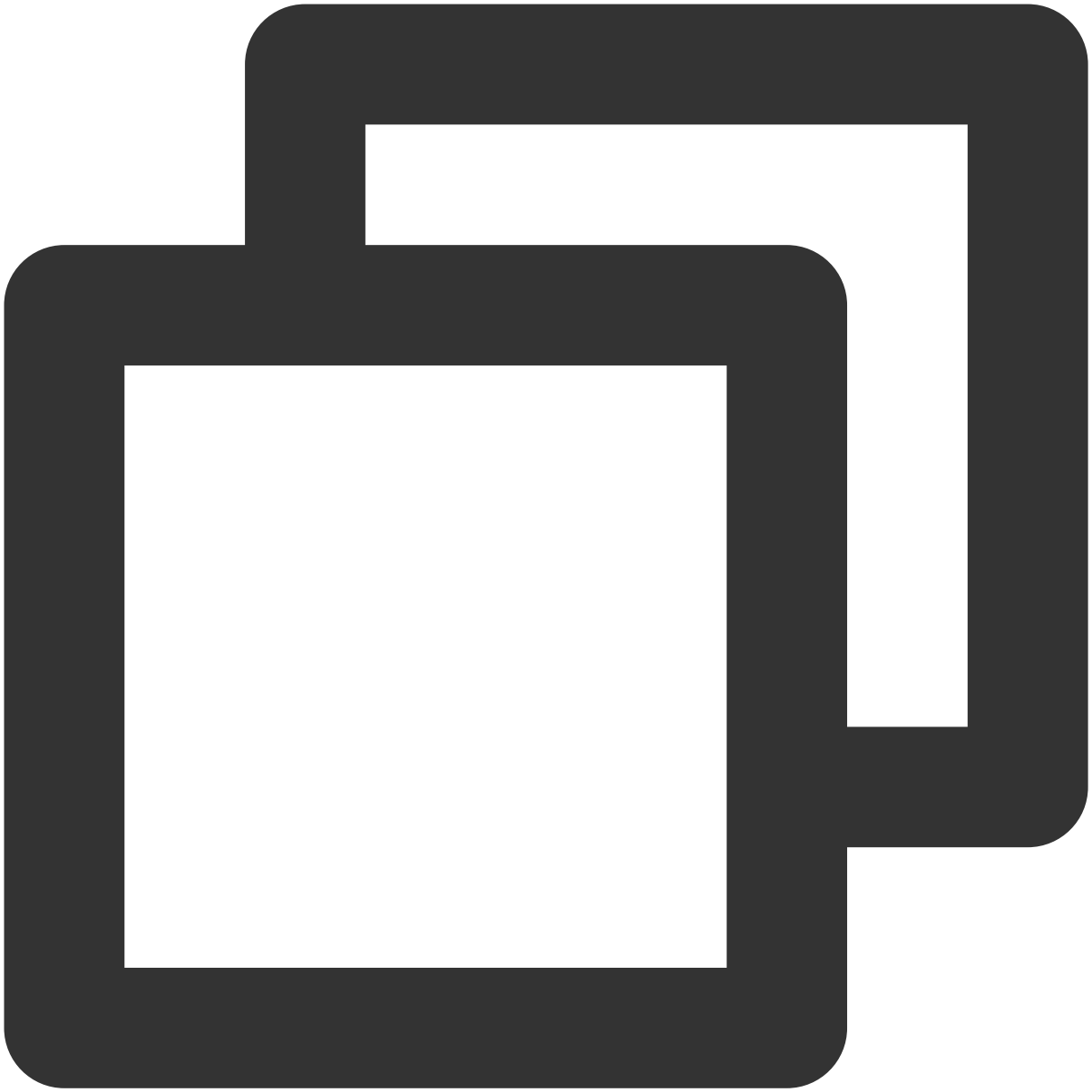
该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。
2. 参考如下内容修改 `binlog_row_image` 。



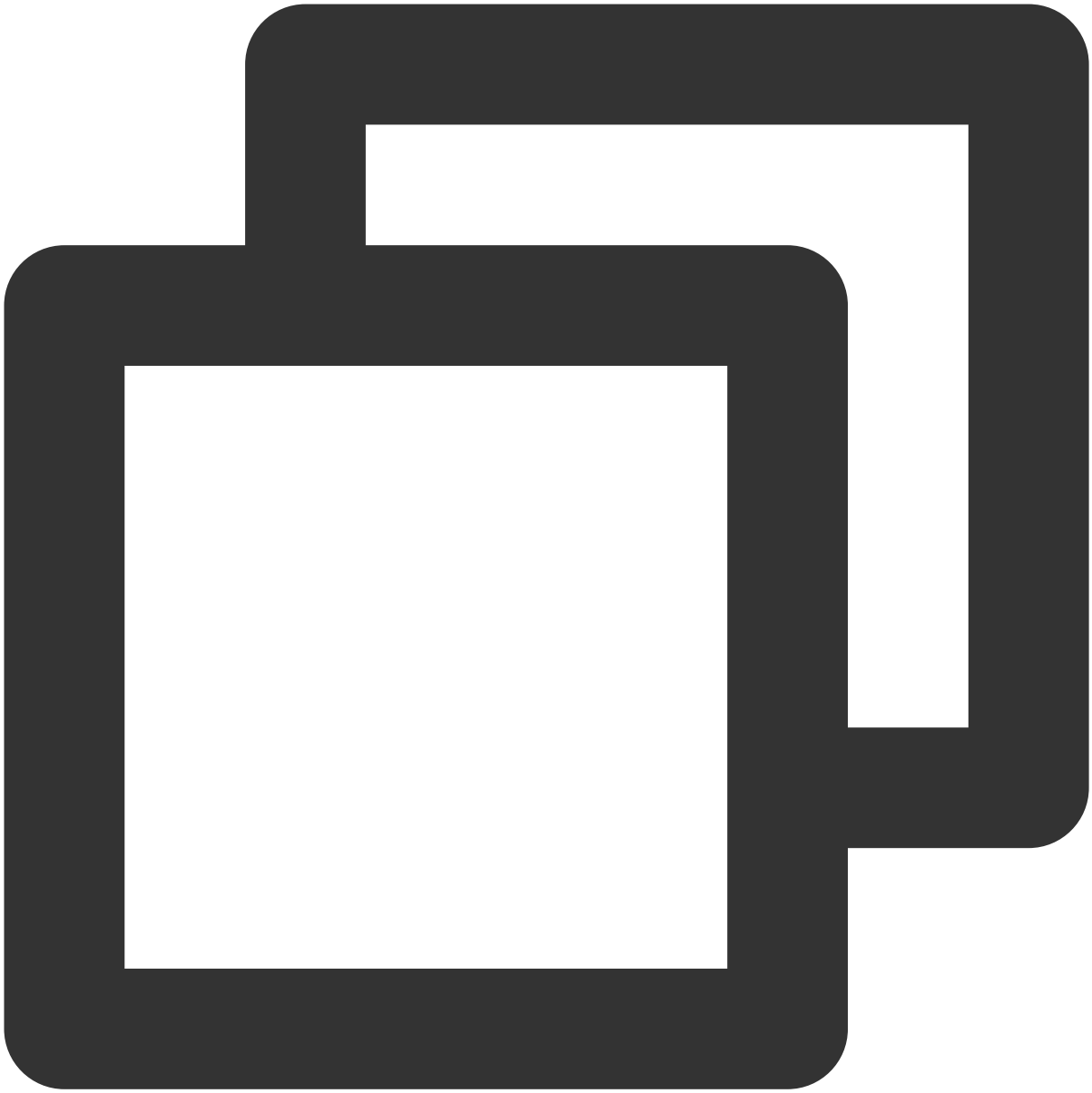
```
set global binlog_row_image = FULL;
```

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。



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

系统显示结果类似如下：



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

4. 重新执行校验任务。

修改 gtid_mode 参数

GTID（Global Transaction Identifier， 全局事务标识），用于在 binlog 中唯一标识一个事务，使用 GTID 可以避免事务重复执行导致数据混乱或者主从不一致。

GTID 是 MySQL 5.6 的新特性，所以 MySQL 5.6 及之后版本存在此问题。DTS 只支持 `gtid_mode` 设置为 `ON` 和 `OFF`，建议将 `gtid_mode` 设置为 `ON`，否则校验时会报警告。

警告不影响迁移或同步任务进行，但是会对业务造成一定的影响：设置 GTID 后，在增量数据同步阶段，如果源实例发生 HA 切换，DTS 服务切换重连，任务几乎无感知；反之，任务会中断后失败，且不可恢复。

`gtid_mode` 的取值如下，在修改 `gtid_mode` 的值时，只能从以下四个值逐级修改，例如，需要从 `OFF` 修改为 `ON`，则 `gtid_mode` 修改顺序为 `OFF` <-> `OFF_PERMISSIVE` <-> `ON_PERMISSIVE` <-> `ON`。

`OFF`：主库所有新启的事务以及从库的事务都要求是匿名事务。

`OFF_PERMISSIVE`：主库新启的事务是匿名事务，但从库事务允许是匿名的或者是 GTID 事务，但不允许只是 GTID 模式。

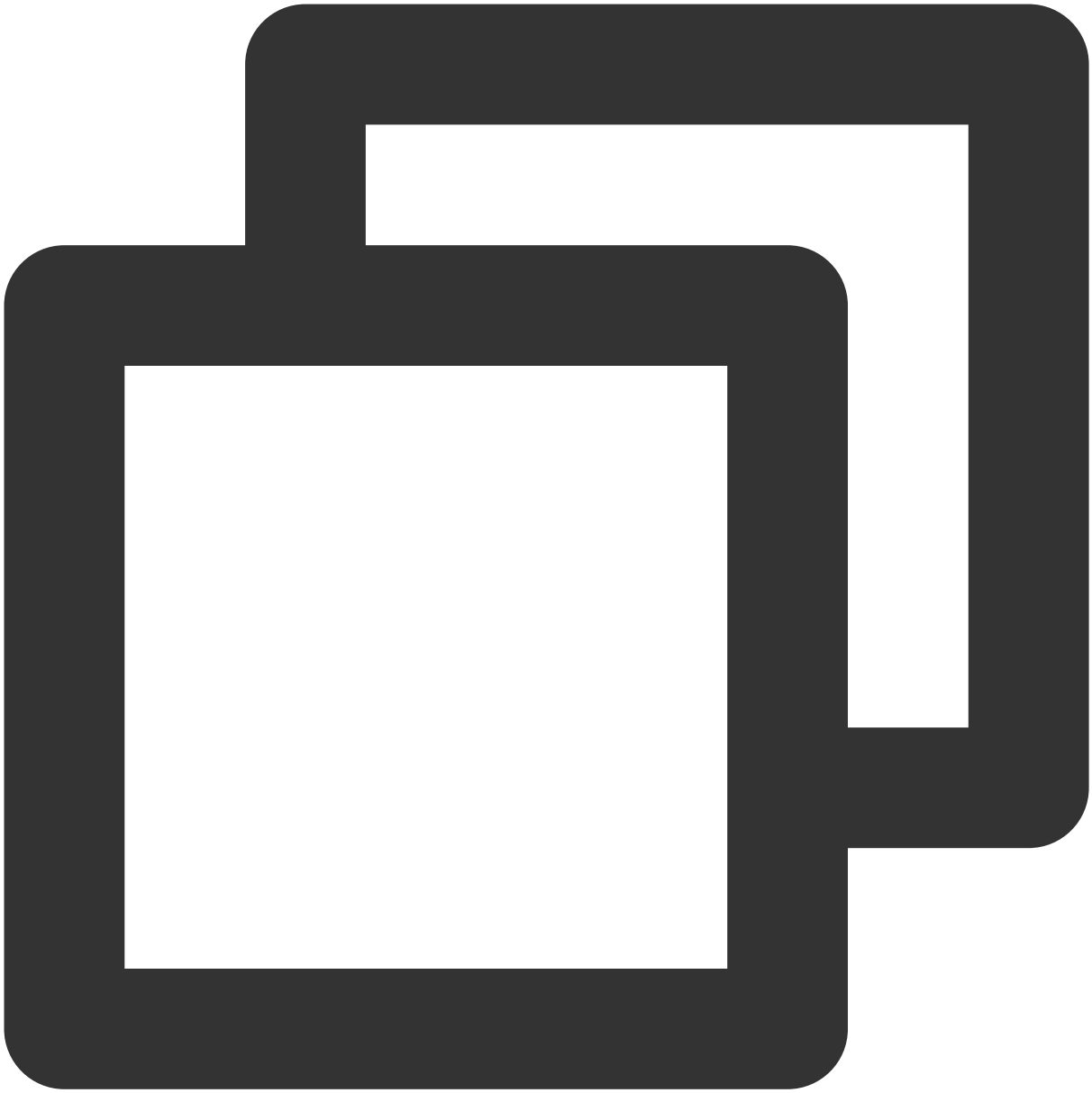
`ON_PERMISSIVE`：主库新启的事务是 GTID 事务，从库事务允许是匿名的或者是 GTID 事务。

`ON`：主库新启的事务是 GTID 事务，从库的事务也要求是 GTID 事务。

如果发生类似警告，请按照如下指导进行修复：

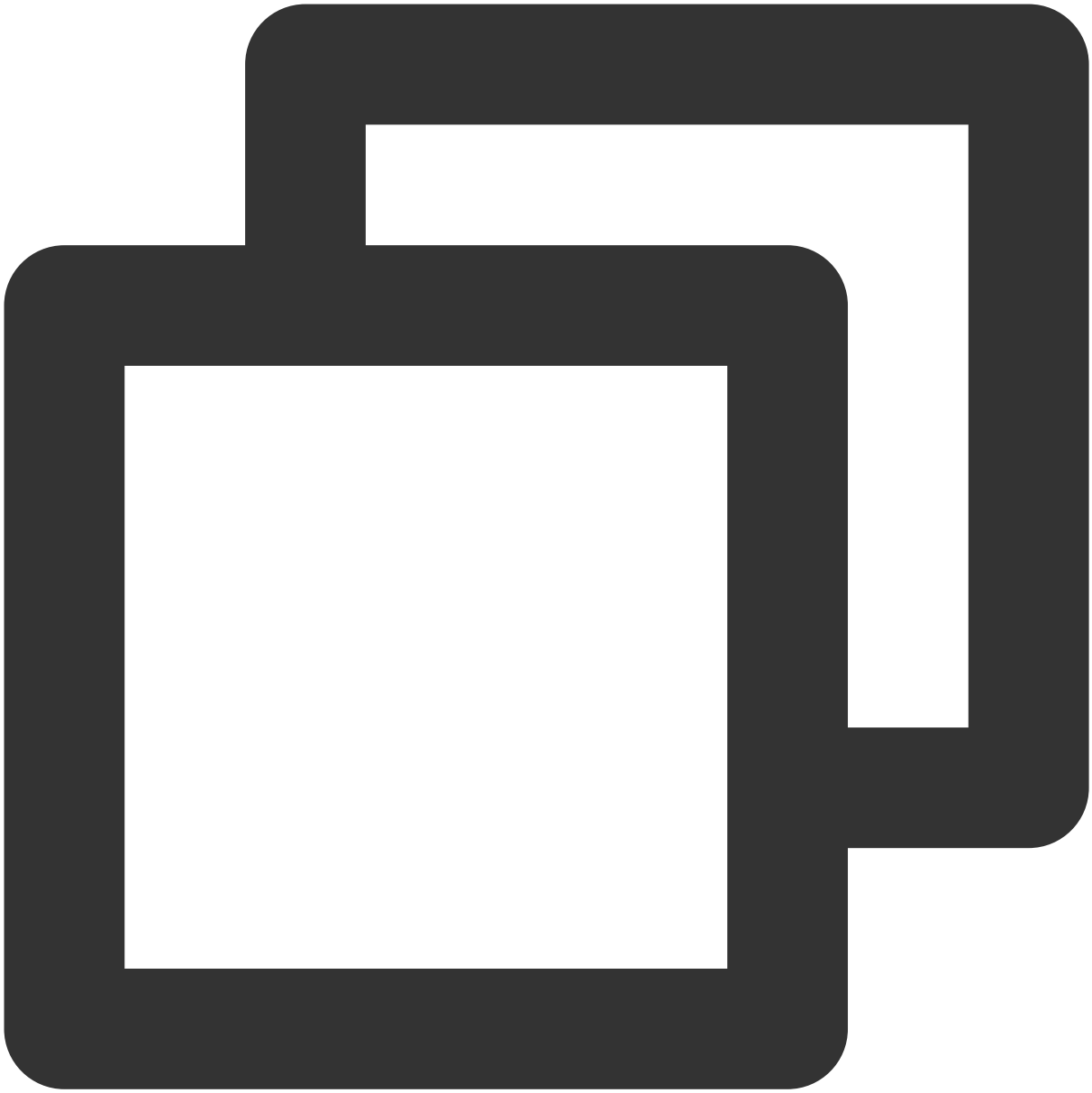
1. 登录源数据库。
2. 在主从复制结构两边都设置 `gtid_mode = OFF_PERMISSIVE`。

MySQL 5.7.6 之前的版本需要在 `my.cnf` 配置文件中修改并重启数据库才能生效，5.7.6 及之后的版本通过全局命名修改，不需要重启数据库，但是需要重置所有业务连接。



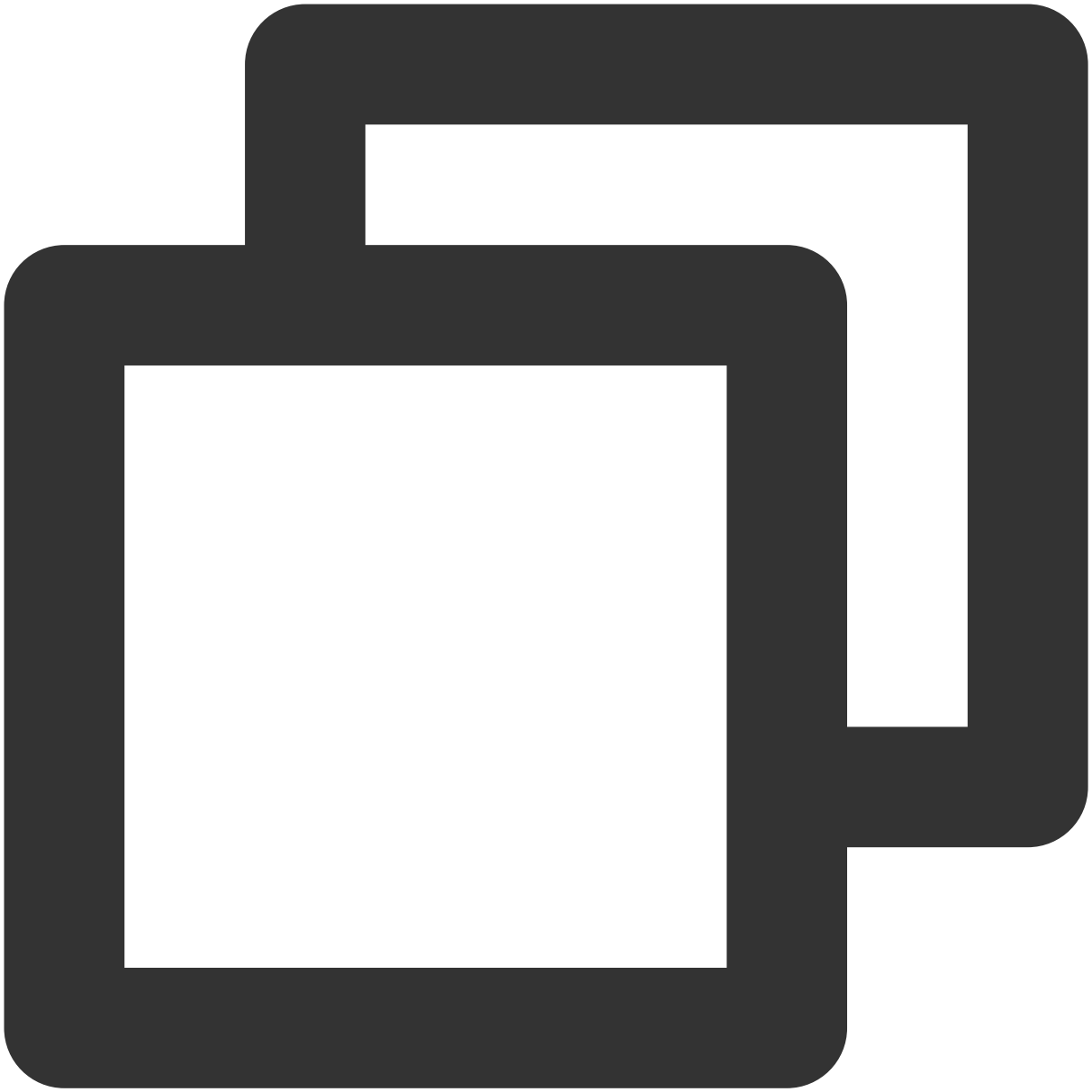
```
set global gtid_mode = OFF_PERMISSIVE;
```

3. 在主从复制结构两边都设置 `gtid_mode = ON_PERMISSIVE` 。



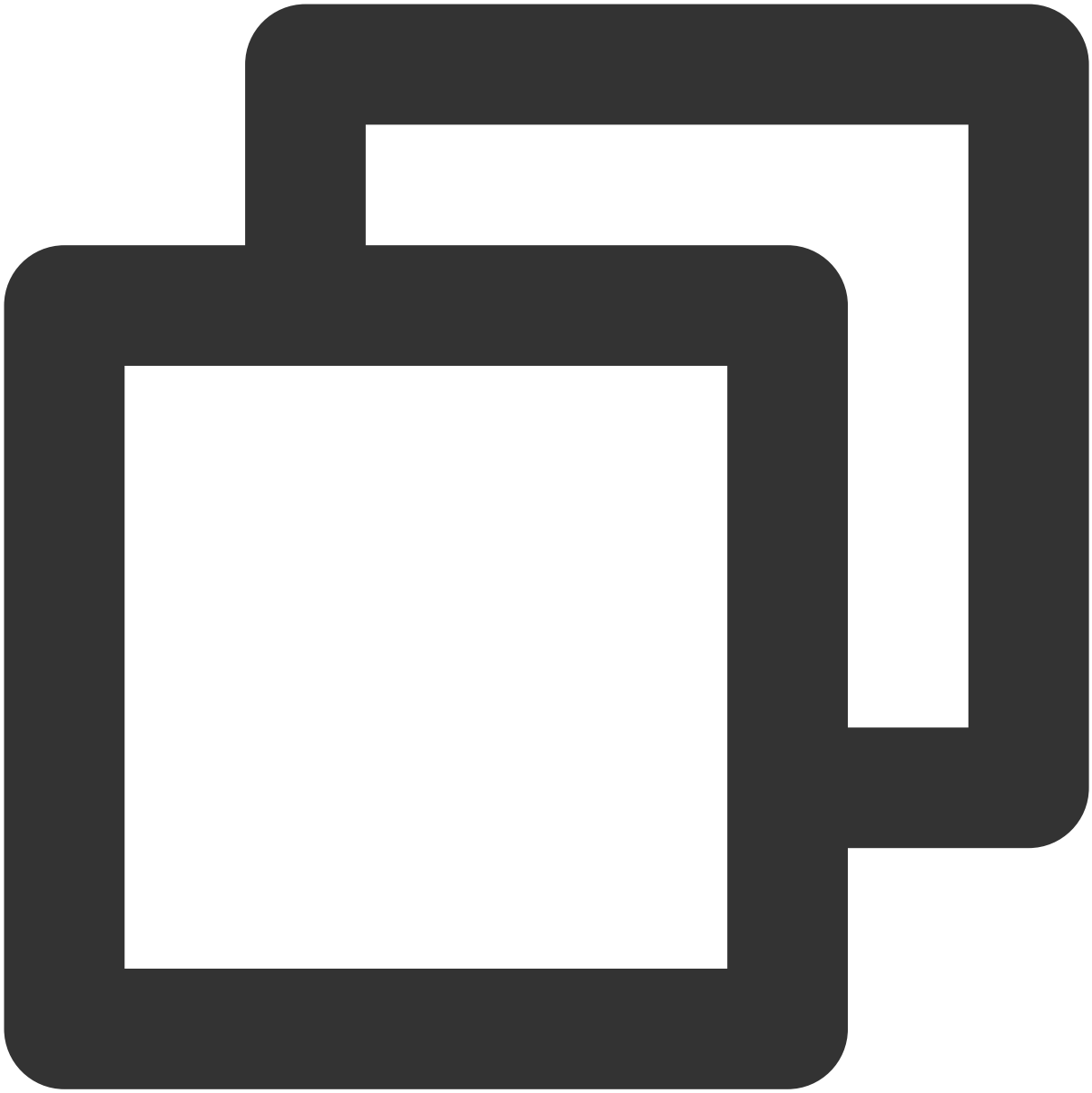
```
set global gtid_mode = ON_PERMISSIVE;
```

4. 在各个实例节点上执行如下命令，检查匿名事务是否消耗完毕，参数值为 `0` 则代表消耗完毕。



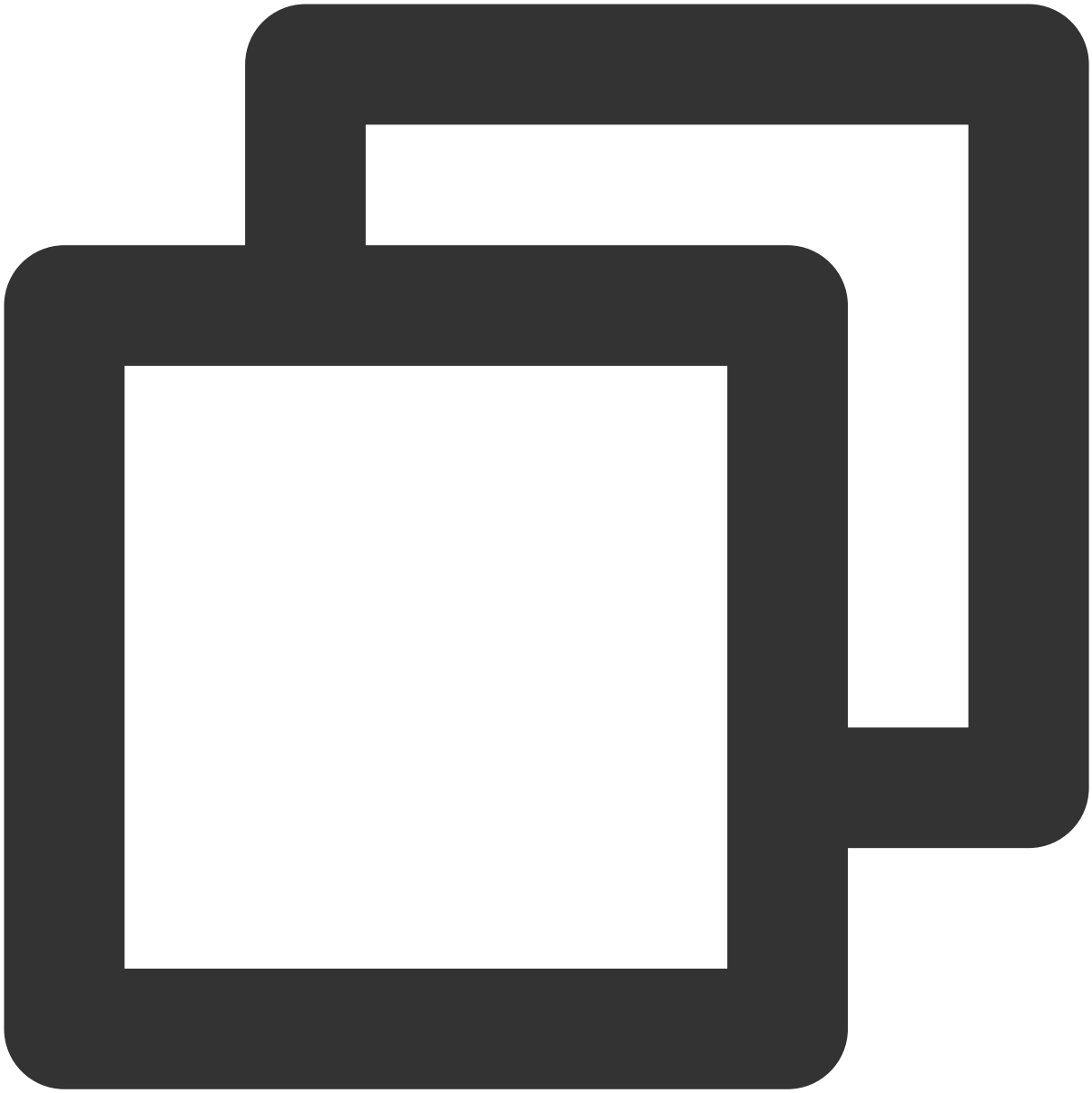
```
show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
```

系统显示结果类似如下：



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

5. 在主从复制结构两边都设置 `gtid_mode = ON` 。

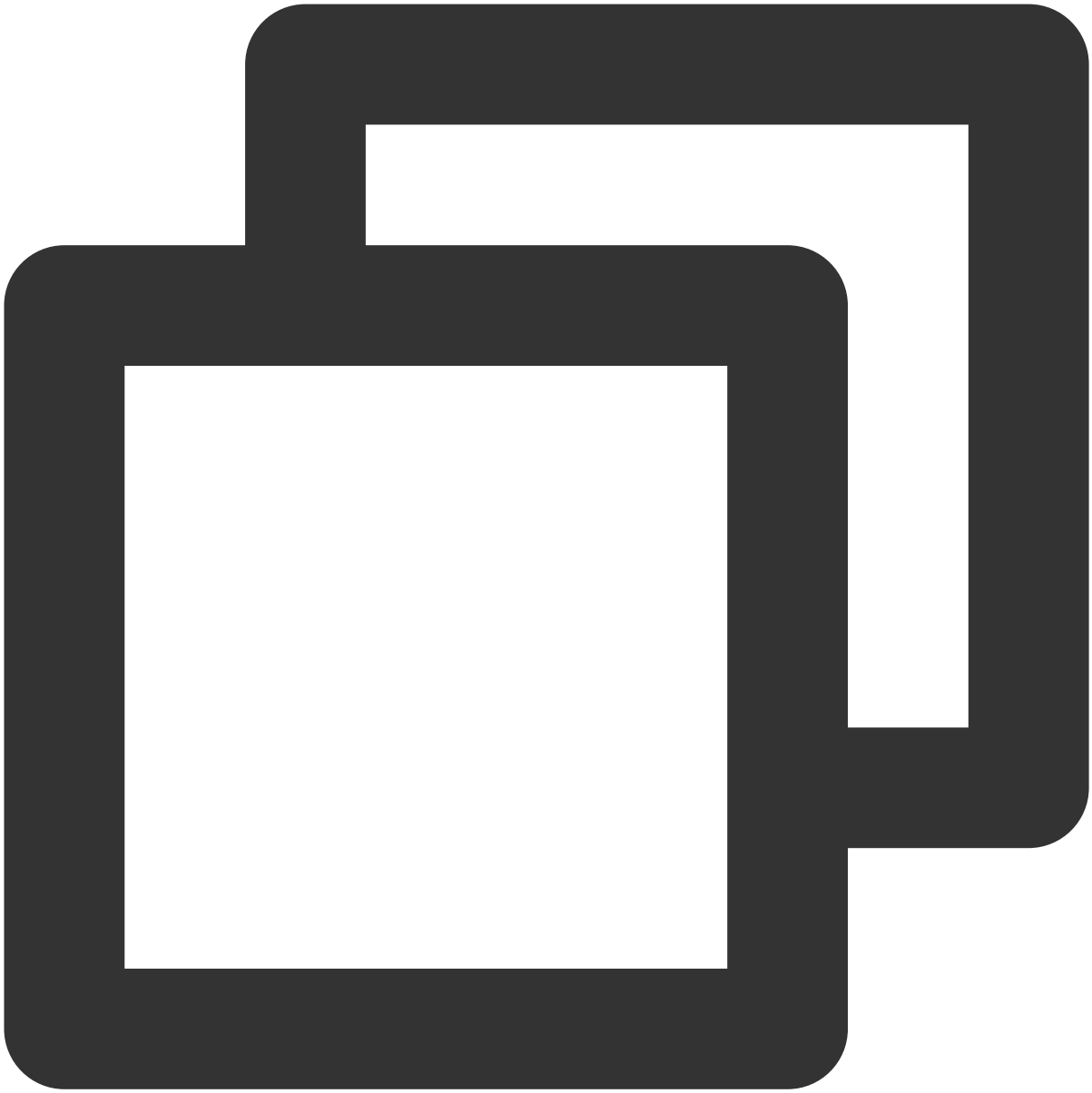


```
set global gtid_mode = ON;
```

6. 在 `my.cnf` 文件中添加如下内容，后续重启数据库后使初始值生效。

说明

`my.cnf` 配置文件的默认路径为 `/etc/my.cnf`，现场以实际情况为准。



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

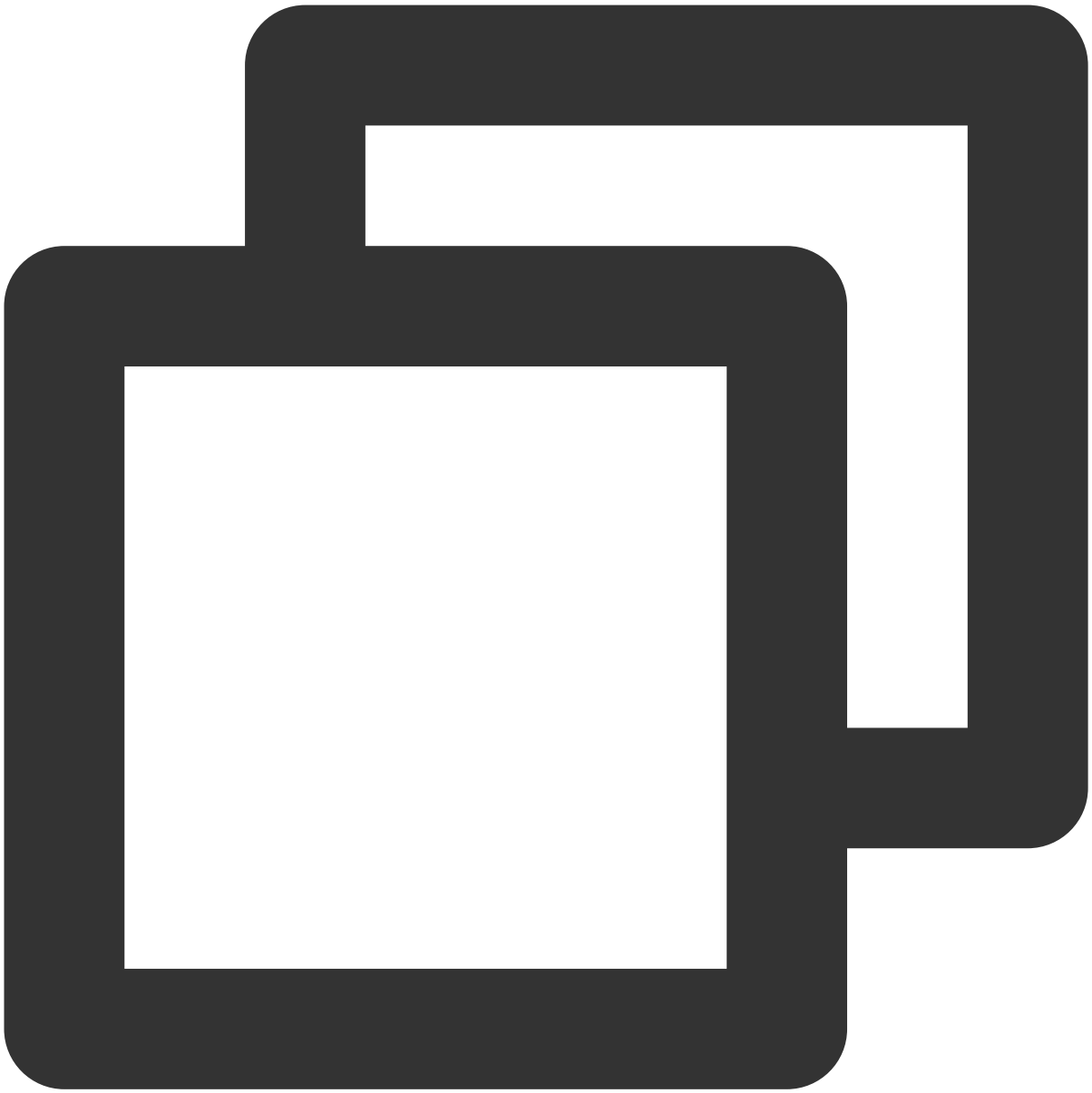
7. 重新执行校验任务。

修改 server_id 参数

`server_id` 参数需要手动设置，且值不能设置为0。该参数系统预设值为 `1` ，如果查询该参数显示为 `1` 不一定正确，需要手动进行配置。

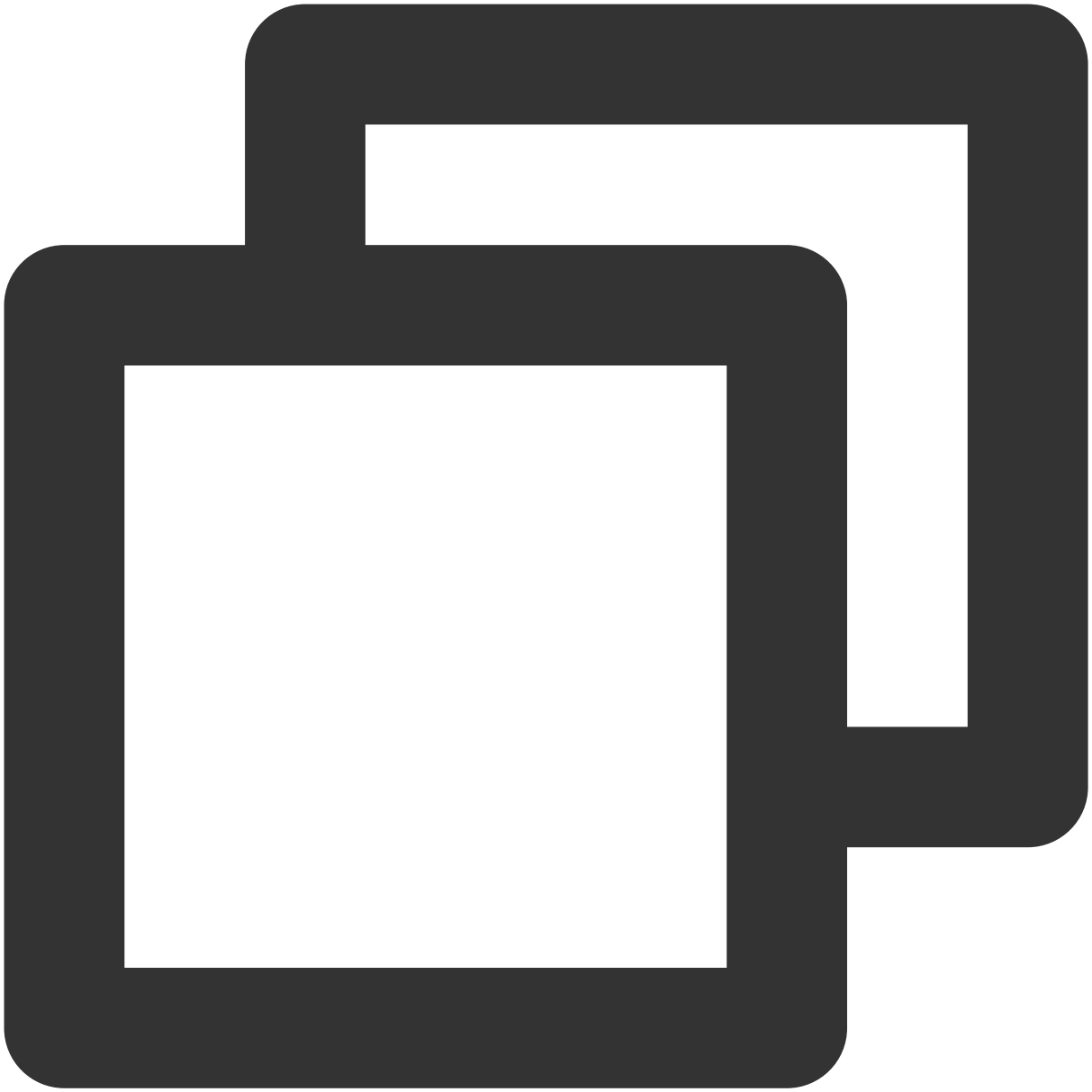
1. 登录源数据库。

2. 参考如下内容修改 `server_id` 。



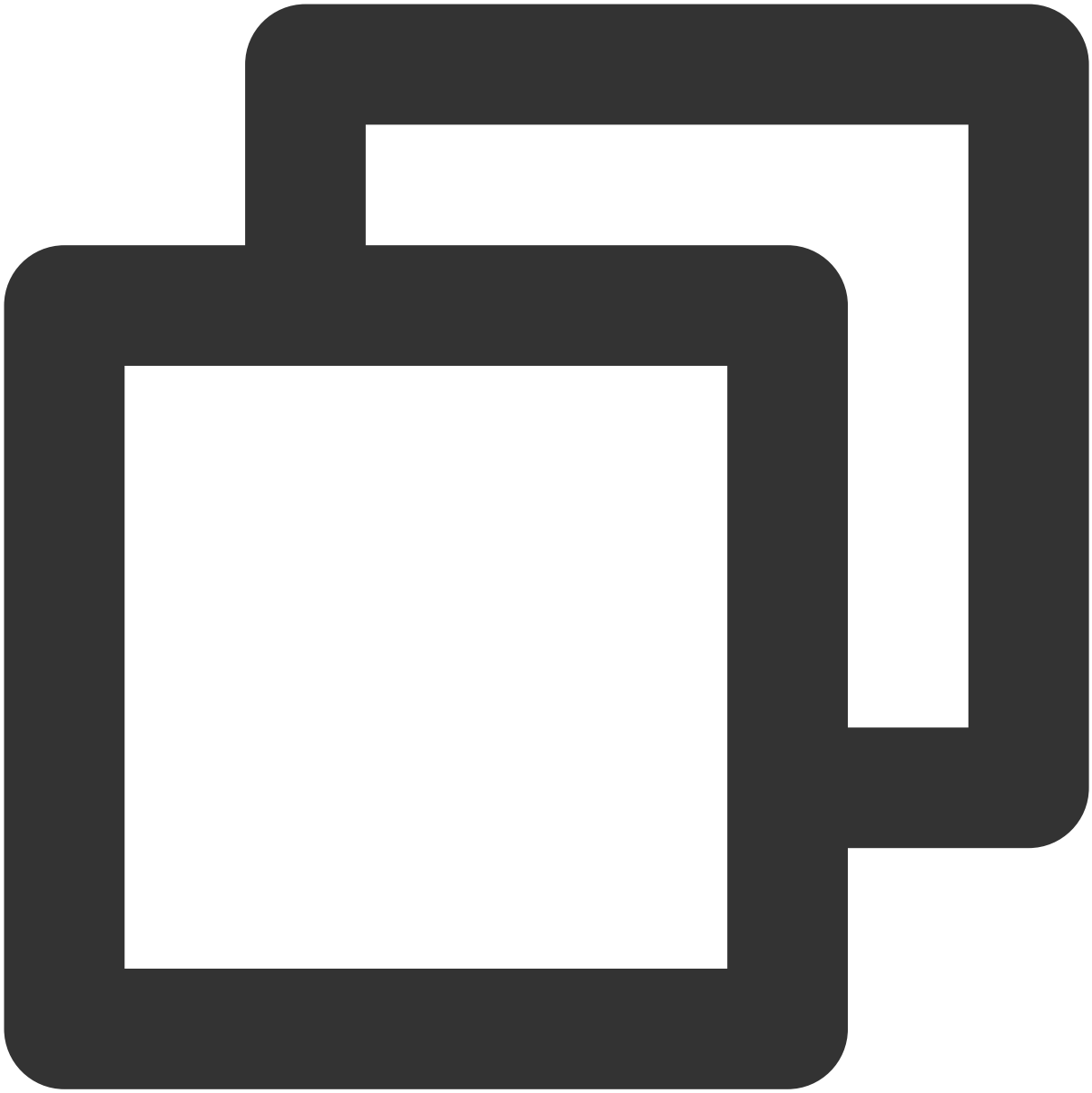
```
set global server_id = 2;   //建议设为大于1的整数，此处仅为示例
```

3. 通过如下命令查看参数修改是否生效。



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

系统显示结果类似如下：



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

4. 重新执行校验任务。

删除 do_db，ignore_db 设置

binlog 会记录数据库所有执行的 DDL 和 DML 语句，而 do_db，ignore_db 则是设置 binlog 记录的过滤条件。

`binlog_do_db` ：只记录指定数据库的二进制日志，默认全部记录。

`binlog_ignore_db` ：不记录指定的数据库的二进制日志。

设置 do_db，ignore_db 后，会导致一些跨库操作 binlog 记录不全，主从复制出现异常，因此不建议设置。如发生类似报错，请参考如下指导进行修复：

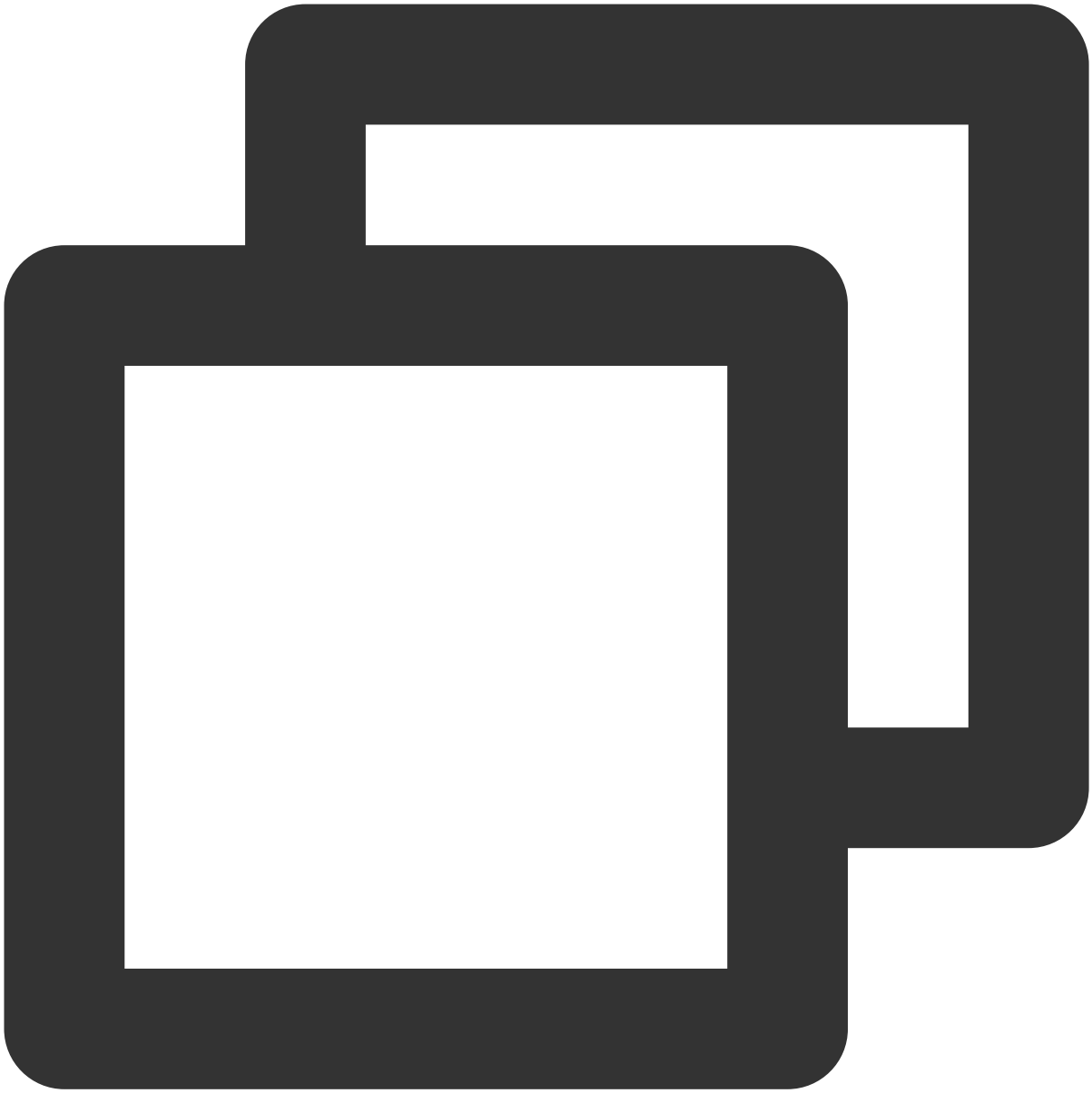
1. 登录源数据库。

2. 修改源数据库的配置文件 `my.cnf` ，删除 do_db，ignore_db相关设置。

说明

`my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。

3. 参考如下命令重启源数据库。

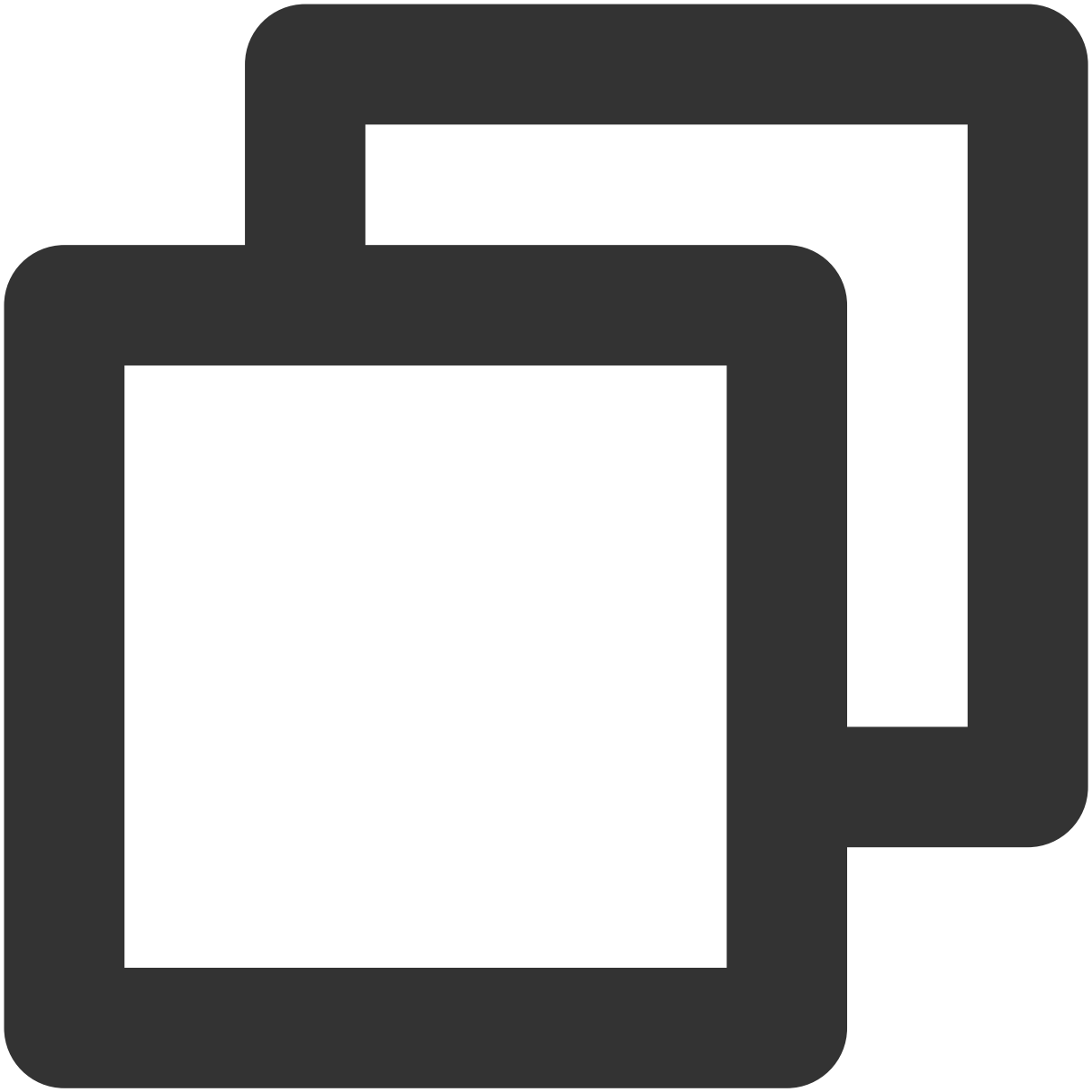


```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

说明

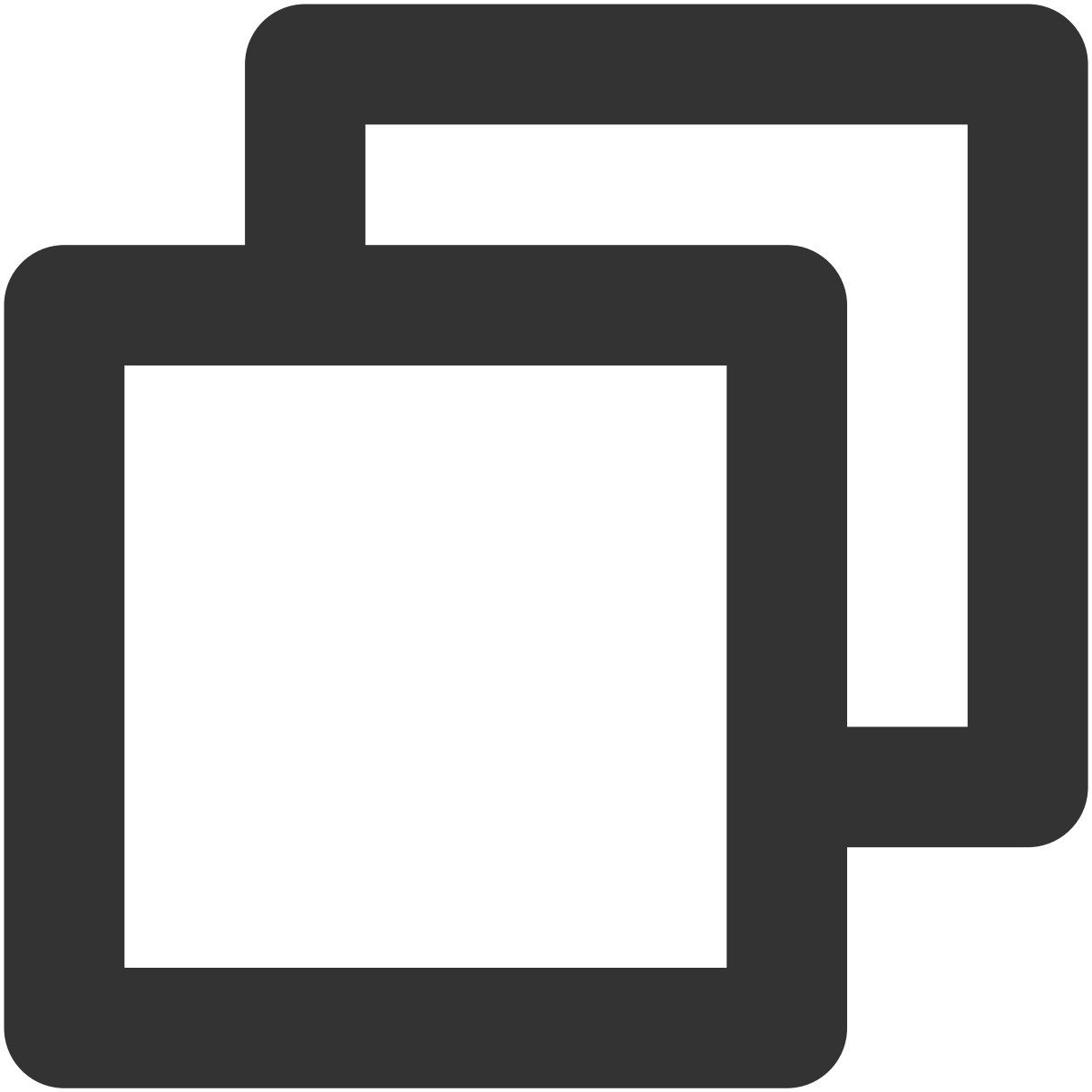
[$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认参数修改是否生效。



```
show master status;
```

系统显示结果类似如下：



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

5. 重新执行校验任务。

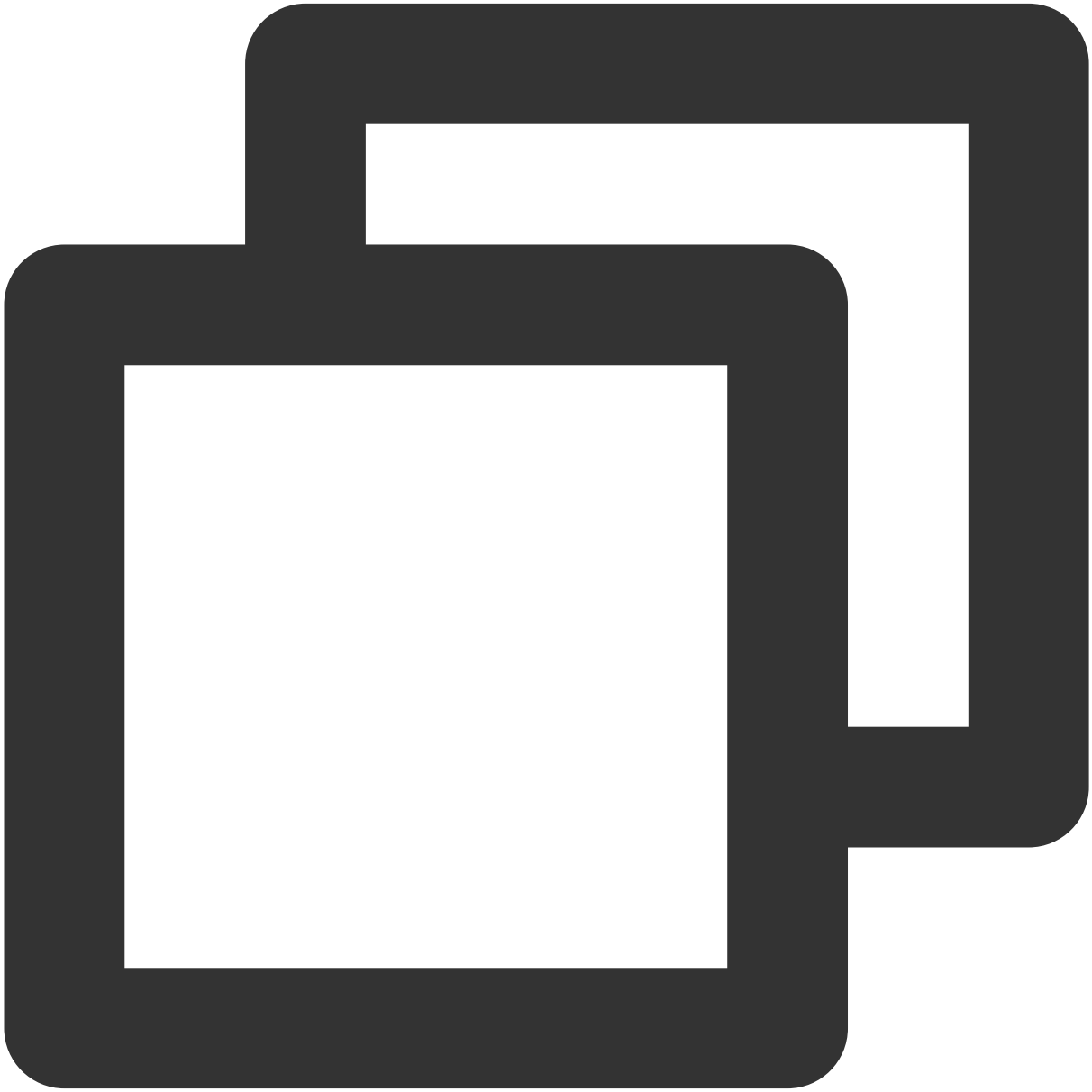
修改 log_slave_updates 参数

在主从复用结构中，从库开启 `log-bin` 参数，直接在从库操作数据时，可以记录在 binlog 中，但是从库从主库上复制数据时，不能记录在 binlog 中，所以从库作为其他从库的主库时，需要打开 `log_slave_updates` 参数。

1. 登录源数据库。
2. 在源数据库的配置文件 `my.cnf` 中增加如下内容。

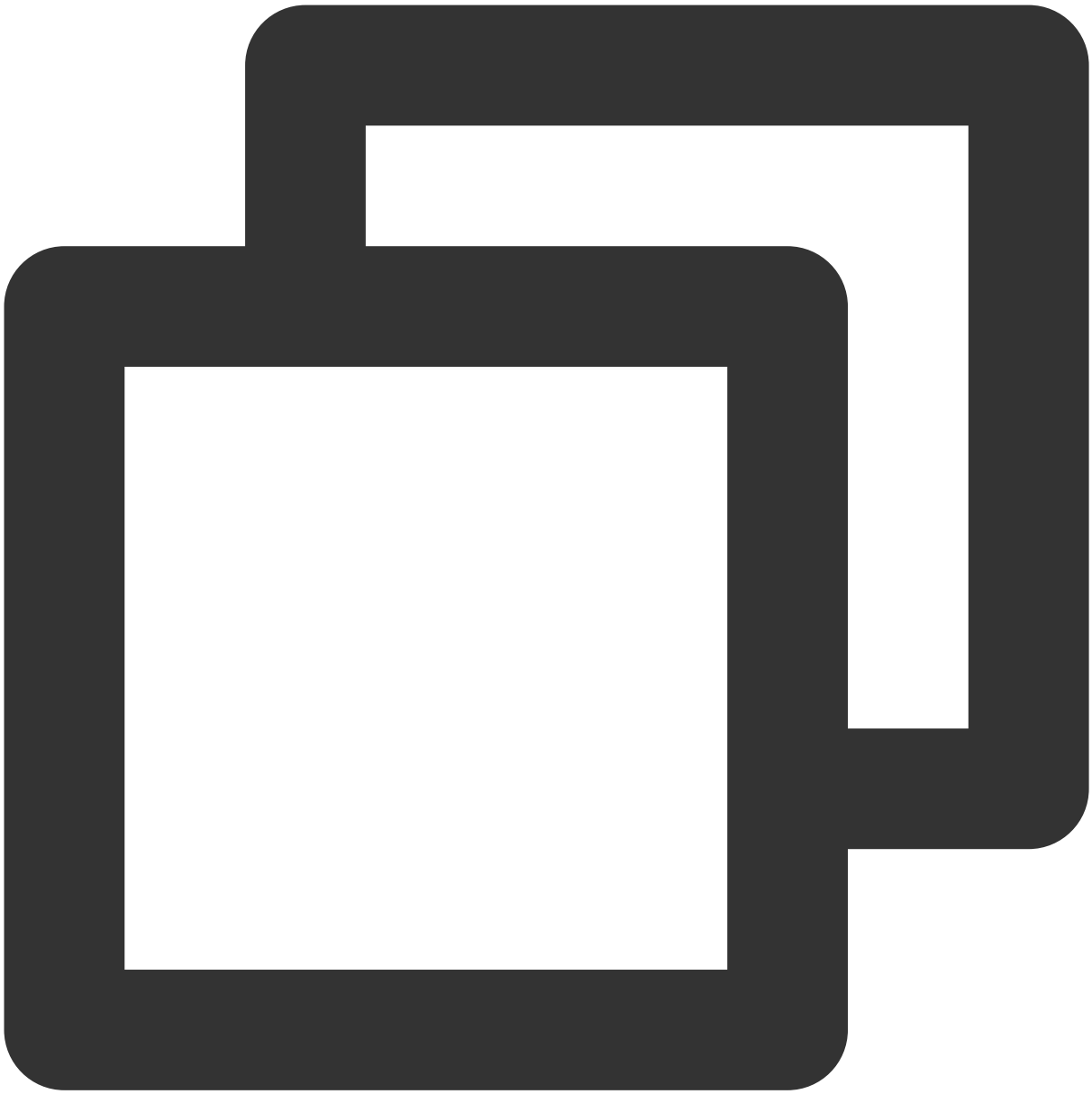
说明

`my.cnf` 配置文件的默认路径为 `/etc/my.cnf`，现场以实际情况为准。



```
log_slave_updates = ON
```

3. 参考如下命令重启源数据库。

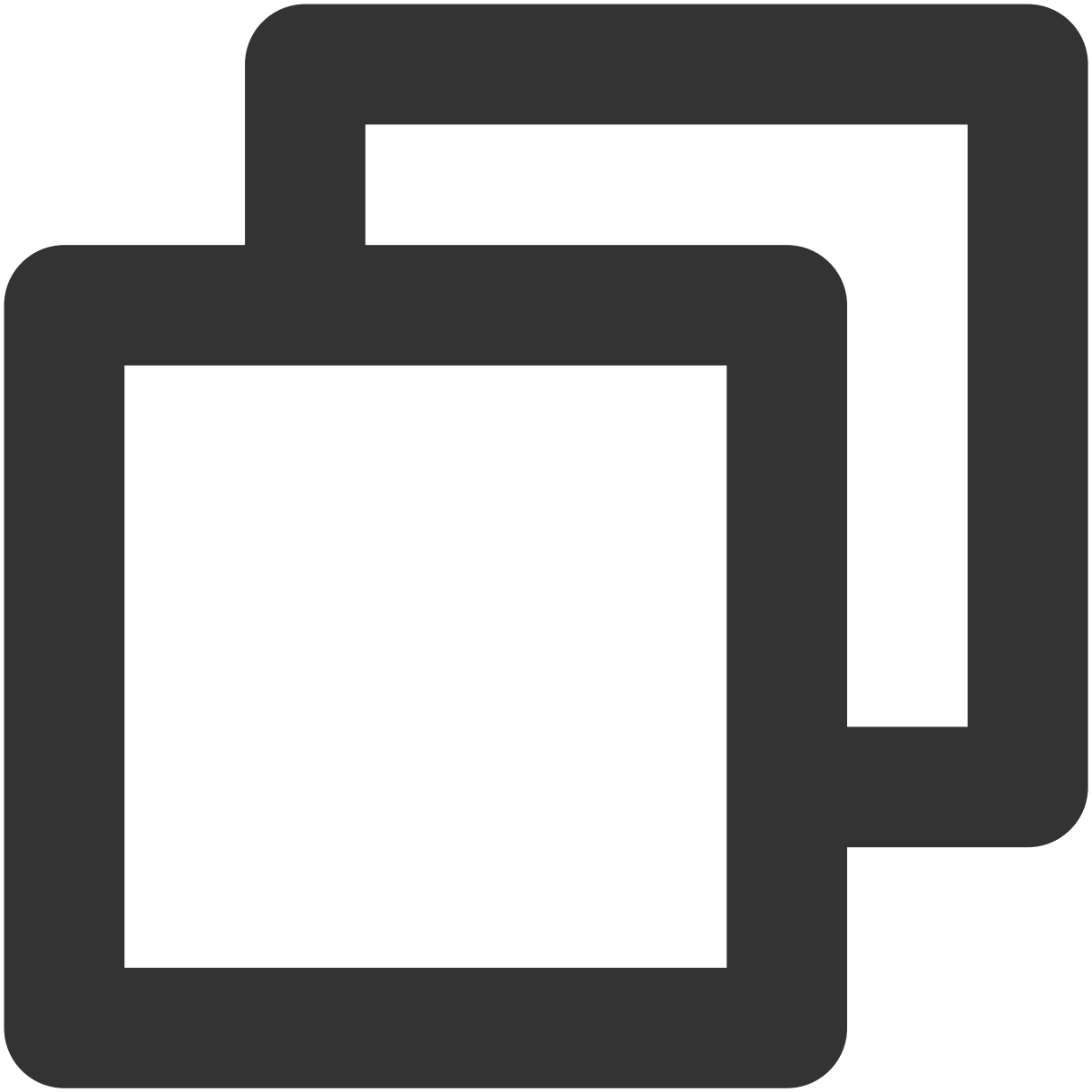


```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

说明

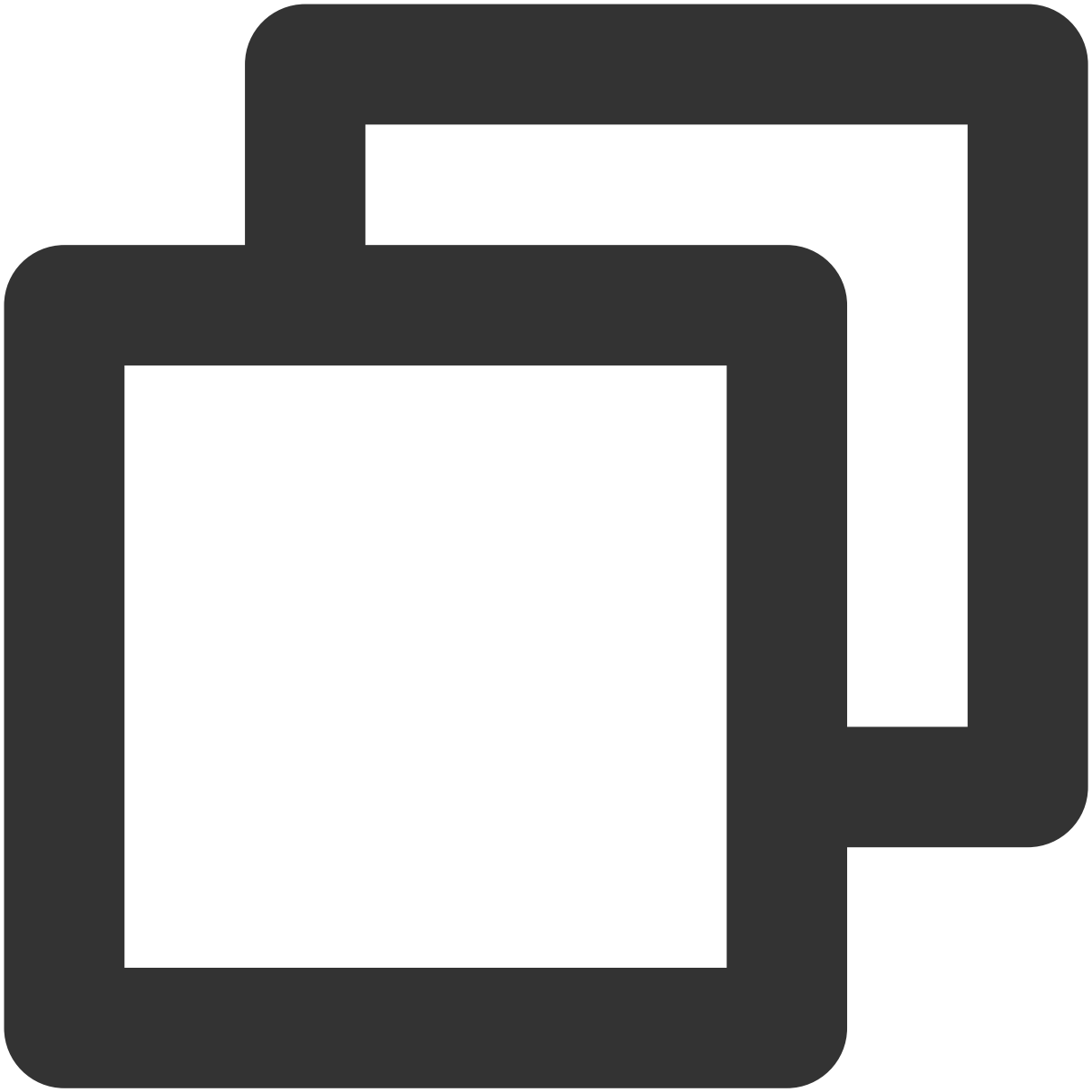
[$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 查看配置是否生效。



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

系统显示结果类似如下：



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

5. 重新执行校验任务。

MongoDB 校验项
连接 MongoDB 实例校验

最近更新时间：2024-02-19 16:12:03

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

源数据库所在网络或服务器设置了安全组或防火墙。
源数据库对来源 IP 地址进行了限制。
网络端口未放通。
数据库账号或密码不正确。

修复方法

请按照问题原因中的对应原因进行处理。

库表冲突校验

最近更新时间：2024-02-19 16:12:16

检查要求

MongoDB 迁移场景中，目标实例可以存在与源库同名，但是不能存在数据（只能为空表）。

修复方法

如果存在冲突报错，删除目标库中的对应库表，或者删除目标库同名库表内的数据。

源端或目标端账户权限校验

最近更新时间：2023-08-25 16:28:56

检查详情

检查用户是否具备对数据库的操作权限，具体可参考如下对应文档。
数据迁移权限要求：MongoDB 数据迁移

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

实例版本校验

最近更新时间：2024-02-19 16:13:10

源数据库和目标数据的版本符合 MongoDB 支持的版本。

实例容量校验

最近更新时间：2024-02-19 16:13:29

检查要求

MongoDB 迁移场景，目标库存储空间需要在源库待迁移库表空间的1.3倍以上。

修复方法

删除目标库中的部分数据，以便腾出足够的空间。
升级目标库存储规格，使用更大容量的实例进行迁移。

PostgreSQL 校验项
源实例或目标实例连通性检查

最近更新时间：2024-02-19 16:30:16

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

源数据库所在网络或服务器设置了安全组或防火墙。
源数据库对来源 IP 地址进行了限制。
网络端口未放通。
数据库账号或密码不正确。

修复方法

请按照问题原因中的对应链接进行处理。

源实例版本检查

最近更新时间：2024-02-19 16:30:41

检查详情

迁移场景

PostgreSQL10.x以前的版本（如9.x）不支持“全量+增量迁移”，若源库配置了增量迁移任务，则会导致校验失败。

当版本不一致时，会存在部分特殊的兼容性问题，任务会发出警告。请自行阅读各版本的兼容性报告，来确认业务是否有使用到不兼容的特性。

同步场景

PostgreSQL 10之前的版本不支持同步，数据同步时目标实例版本必须大于等于源实例版本。

当版本不一致时，会存在部分特殊的兼容性问题，任务会发出警告。请自行阅读各版本的兼容性报告，来确认业务是否有使用到不兼容的特性。

修复方法

请按 数据迁移支持的数据库 和 数据同步支持的数据库 中的版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

账号冲突检查

最近更新时间：2023-08-25 16:41:02

检查要求

PostgreSQL 迁移场景中，目标实例不能有和源库同名的对象。

PostgreSQL 整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。

修复方法

如果检测到冲突，请删除冲突内容，然后重新校验。

目的库存在性检查

最近更新时间：2024-02-19 16:34:02

检查详情

检查要迁移的目标库是否存在，如果不存在则会报错。

修复方法

在迁移过程中不要删除目标库，如果删除了需要重新构建迁移任务。

主键依赖检查

最近更新时间：2024-02-19 16:34:26

检查详情

PostgreSQL 同步场景，检查待同步表格是否有主键，若没有主键，任务校验会告警，会自动设置 ALTER TABLE XXX REPLICA IDENTITY。

修复方法

请按照界面提示，设置主键，然后重新执行校验任务。

实例关键参数检查

最近更新时间：2024-02-19 16:34:48

检查详情

PostgreSQL 同步场景会检查复制参数 wal_level 是否等于 logical ， 同时 replication_slots 需满足选择的待同步库数量+replication_slots <= max_wal_senders

修复方法

设置实例参数 wal_level 为 logical。
减少待同步的库数量。

目标实例只读库检查

最近更新时间：2024-02-19 16:35:11

检查详情

PostgreSQL 在配置同步任务时，需要检查目标实例是否为只读实例，如果为只读实例，则无法进行数据同步，任务会报错。

修复方法

配置任务时不能选择目标实例为只读。

多任务冲突检测

最近更新时间：2024-02-19 16:35:28

检查详情

PostgreSQL 同步任务中，在配置任务时不支持环形链路，如果为环形链路则任务校验会报错。如果源端同步对象已经存在多个任务，则会告警，请检查任务配置是否合理。

修复方法

请按照界面提示，取消已经存在回环或者重复配置同步的任务，然后重新执行校验任务。

TDSQL PostgreSQL
连接 DB 检查

最近更新时间：2023-08-25 16:52:46

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

源数据库所在网络或服务器设置了安全组或防火墙。
源数据库对来源 IP 地址进行了限制。
网络端口未放通。
数据库帐号或密码不正确。

修复方法

请按照问题原因中的对应链接进行处理。

版本检查

最近更新时间：2023-08-25 16:53:31

检查详情

被订阅的表必须拥有主键或 REPLICA IDENTITY 为 FULL。

修复方法

当校验不通过时，请修改对应的表格。

周边检查

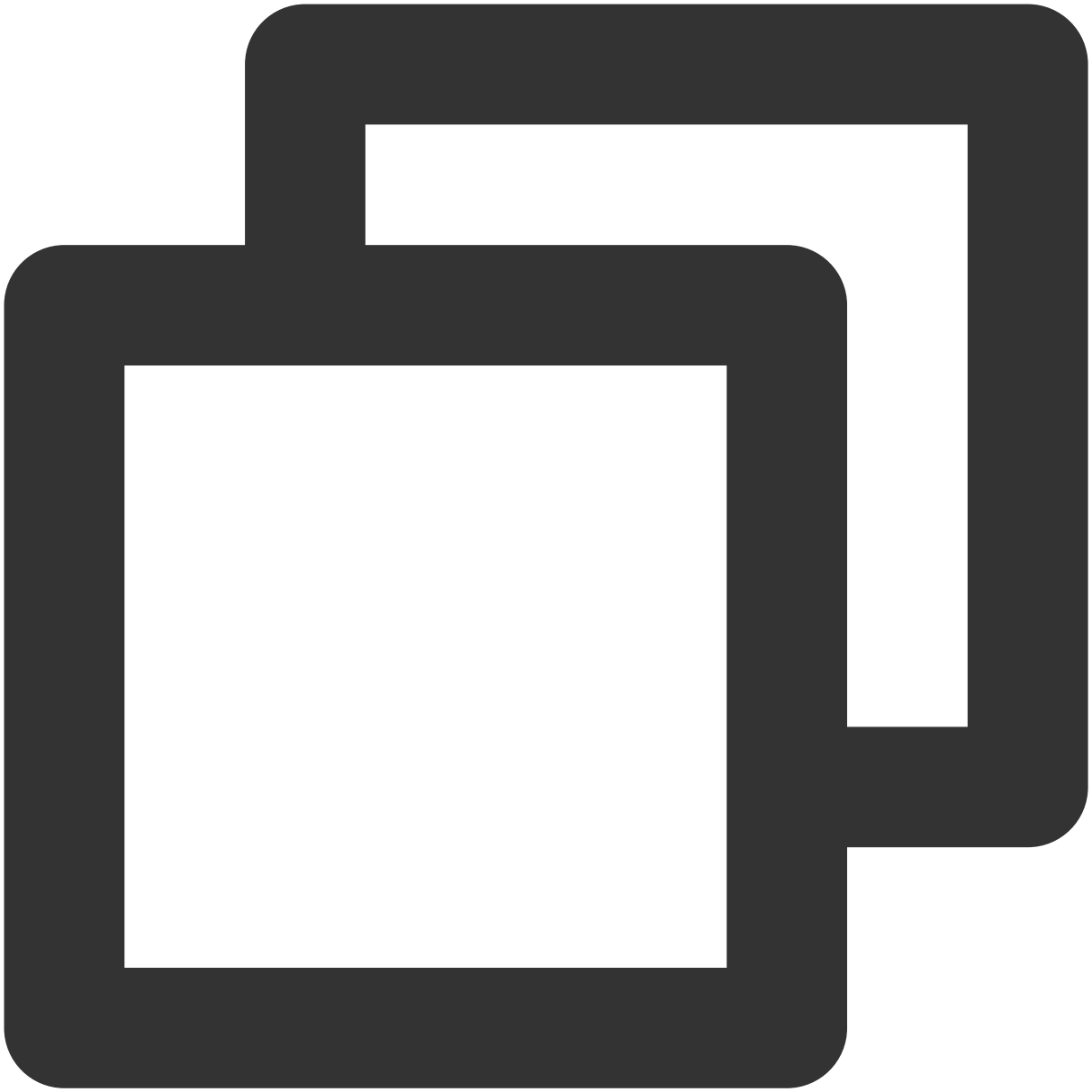
最近更新时间：2023-08-25 16:46:25

检查详情

DN 节点的 wal_level 必须是 logical。

修复方法

1. 登录源数据库。
2. 找到 postgresql.conf 文件，打开此文件，修改 `wal_level` 。



```
wal_level = logical
```

3. 修改完成后，重启数据库实例。

源实例权限检查

最近更新时间：2024-02-19 16:42:15

检查详情

用户必须拥有 REPLICATION 权限，对应 pg_roles.rolreplication。
用户必须拥有被订阅表的 select 权限，如果是整库订阅，那么用户要拥有该 schema 下所有表的 select 权限。

修复方法

用户可能不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

约束检查

最近更新时间：2024-02-21 15:53:00

检查详情

被订阅的表必须拥有主键或 REPLICA IDENTITY 为 FULL。

修复方法

当校验不通过时，请修改对应的表格。

Redis/Tendis 校验项
网络是否可达

最近更新时间：2023-07-06 15:42:50

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

源数据库所在网络或服务器设置了安全组或防火墙。
源数据库对来源 IP 地址进行了限制。
网络端口未放通。
数据库帐号或密码不正确。

修复方法

请按照问题原因中的对应情况进行处理。

源实例目标实例版本是否兼容

最近更新时间：2023-07-06 15:41:50

检查详情

Redis > Redis 迁移，Redis 源实例版本需要大于等于2.2.6，2.2.6以下版本不支持 DTS 迁移。Redis > KeeWiDB 迁移，Redis 源库支持Redis 4.0版本，如果迁移4.0以外的其他版本，需要 提交工单 进行申请。
建议目标数据库版本大于或等于源数据库版本，否则会报警告，低版本向高版本迁移，有兼容性问题。
目标数据库的 Proxy 版本为最新版本。

修复方法

请按 数据迁移支持的数据库 和 数据同步支持的数据库 中的版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例参数检查

最近更新时间：2023-08-25 16:54:36

检查详情

Redis 迁移场景，目标数据库为腾讯云 Redis 时，源实例的数据库个数需要小于等于目标实例的数据库个数。
源实例的状态是否正常。

修复方法

修改源和目标实例的数据库个数，然后重新启动校验任务。
确认源实例的状态。

目标实例容量是否满足要求

最近更新时间：2023-07-06 15:40:27

检查要求

Redis > Redis 迁移，目标库的空间必须大于等于源库待迁移数据所占空间的1.5倍。Redis > KeeWiDB 迁移，目标库的空间必须大于等于源库待迁移数据所占空间的2倍。

修复方法

删除目标库中的部分数据，以便腾出足够的空间。
升级目标库存储规格，使用更大容量的实例进行迁移。

目标实例是否为空

最近更新时间：2023-07-06 15:39:20

检查要求

Redis 迁移场景中，目标实例内容必须为空，否则会报错。

修复方法

如果报错请删除目标实例中的内容，然后重新启动校验任务。

目标实例状态是否正常

最近更新时间：2023-07-06 15:38:32

检查详情

检查要迁移的目标库是否已经不存在了，如果不存在则会报错。

修复方法

在迁移过程中不要删除目标库，如果删除了需要重新构建迁移任务。

SQL Server 校验项
迁移参数检查

最近更新时间：2023-08-25 16:43:09

检查要求

目标实例不能有和源库同名的库，否则会报错。

修复方法

如果报错请在目标库中删除或者重命名同名的库。

源库或目标库连接性检查

最近更新时间：2024-03-05 10:09:07

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

源数据库所在网络或服务器设置了安全组或防火墙。
源数据库对来源 IP 地址进行了限制。
网络端口未放通。
数据库账号或密码不正确。

修复方法

请按照问题原因中的对应情况进行处理。

数据库版本检查

最近更新时间：2023-07-06 15:33:32

仅支持基础版迁移到高可用版本（包括双机高可用和集群版），且目标实例的版本号需要大于源实例的版本号。不支持跨版本迁移。

用户权限检查

最近更新时间：2023-07-06 15:31:37

检查详情

检查用户是否具备对数据库的操作权限，具体操作方法可参见 SQL Server 数据迁移。

修复方法

用户可能不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

连接 DB 检查

最近更新时间：2021-11-15 15:02:07

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报“连接源实例失败”。

问题原因

- 源数据库所在网络或服务器设置了安全组或防火墙。
- 源数据库对来源 IP 地址进行了限制。
- 网络端口未放通。
- 数据库帐号或密码不正确。

源数据库所在网络或服务器设置了安全组或者防火墙

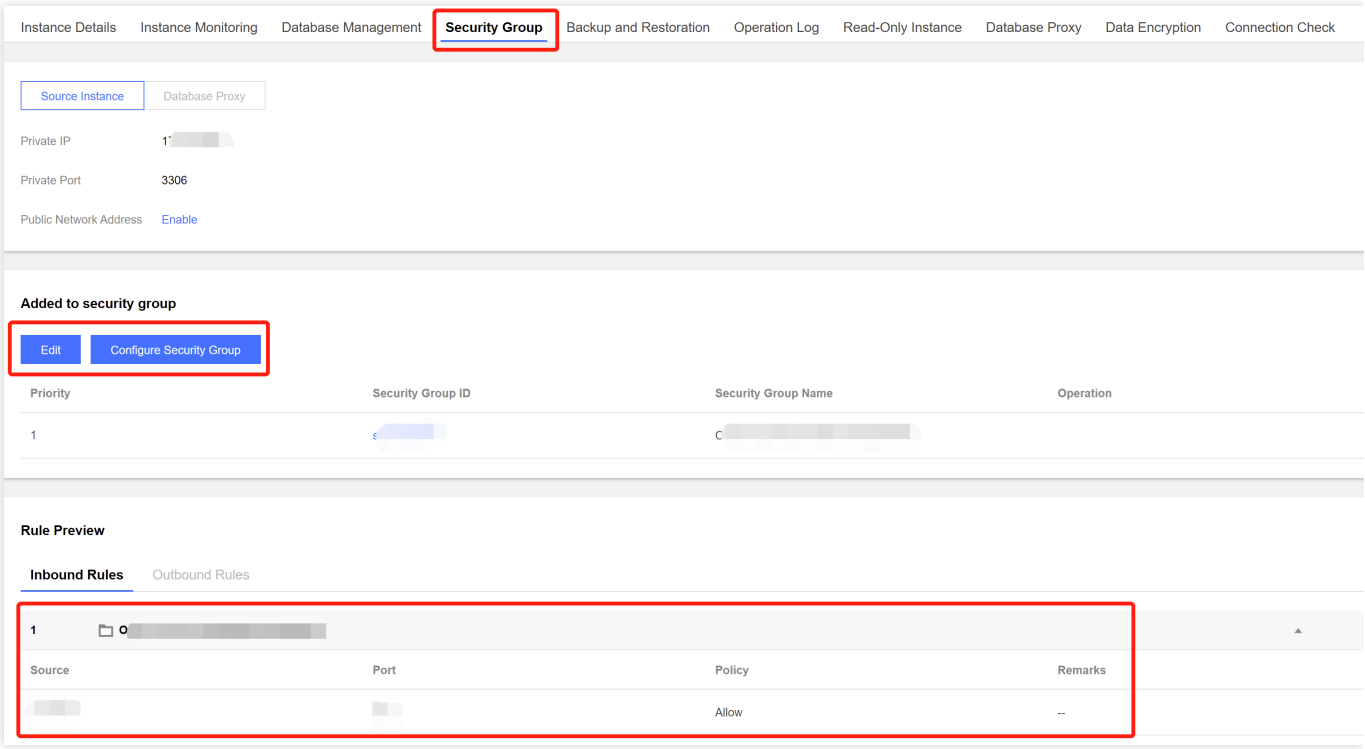
检测方法

安全组功能与防火墙功能类似，安全组是针对云上数据库的网络安全设置。

请根据现场情况，进行以下检查步骤：

- 查看源数据库所在的服务器是否配置了防火墙策略。
  - Windows 系统：打开控制面板找到 Windows 防火墙，查看是否配置了防火墙策略。
  - Linux 系统：请执行 `iptables -L` 命令，检查服务器是否配置了防火墙策略。
- 查看数据库配置的安全组是否限制了 DTS 的 IP 地址段。
  i. 登录 对应的数据库，在实例列表，单击实例 ID，进入实例管理页面。

ii. 在实例管理页面，选择**安全组**页，查看是否存在禁止 DTS 的 SNAT IP 地址段的策略。



修复方法

请根据现场情况，选择以下对应的步骤：

- 服务器开启了防火墙。

  i. 关闭服务器的防火墙，然后登录重新执行校验任务。

  > 说明：
  > 该方法 Windows 和 Linux 系统都适用。

  ii. 将 DTS 的 IP 地址段“策略”配置为**允许**。

- 安全组的配置限制了DTS的SNAT IP地址段。

  i. 在安全组页面，单击对应的安全组 ID。

2. 修改 DTS 的 IP 地址段的策略，配置为**允许**。

源数据库对来源 IP 地址进行了限制

检测方法

MySQL 检测方法

- 在源数据库部署的服务器上，使用数据迁移任务中填入的数据库帐号和数据库密码连接源数据库。如果连接正常，说明源数据库可能限制了来源 IP 地址。
- 如果是自建数据库，需要在数据库上确认 bind-address 的配置，如果不是0.0.0.0，则 IP 受限。
- 如果源数据库为 MySQL，您可以使用 MySQL 客户端连接源数据库，执行以下 SQL 语句进行检查，检查输出结果中的授权 IP 地址列表中是否包含 DTS 的 SNAT IP 地址。
  在给用户进行数据库的授权时，授权的 IP 中需要包含 SNAT IP，否则会发生 IP 受限问题。示例如下：

```
root@10.0.0.0/8 //授权用户通过指定10.0.0.0/8访问，其他 IP 会受限（错误配置）
root@% //授权用户访问所有的 IP，其中需要包含 SNAT IP（正确配置）
```

  您可以通过如下方法验证。

```
select host,user,authentication_string,password_expired,account_locked from
mysql.user WHERE user='[\$Username]'; //[\$Username]为数据迁移任务中填写的数据库帐号
```

SQL Server 检测方法

检查源数据库中是否有 Endpoint 或 Trigger 限制了访问来源 IP 地址。

PostgreSQL 检测方法

- 如果源端数据库为其他云数据库时，请首先检查源端云数据库实例的安全访问策略是否有所限制。请根据不同云厂商的限制方法来进行检查。
- 如果源端数据库为自建的 PostgreSQL 数据库，请进入 $PGDATA 目录下的 data 目录，找到 `pg_hba.conf` 文件。查看此文件中是否存在 deny 策略，或者仅允许部分网络端的 IP 地址访问。

```
# 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 检测方法

如果是自建数据库，需要在数据库上确认 bind-address 的配置，如果不是0.0.0.0，则 IP 受限。

修复方法

MySQL 修复方法

1. 源数据库为 MySQL，请在源数据库中执行以下 SQL 语句，重新给数据迁移使用的用户授权。

```
mysql> grant all privileges on . to '[\$UserName]'@'%'; //[\$Username] 为数据迁移任务中填写的数据库帐号
mysql> flush privileges;
```

2. 对于自建数据库，如果 bind-address 配置异常，请参考如下指导修改。

   2.1. 在 `/etc/my.cnf` 文件中增加如下内容。

> 说明：
> `my.cnf` 配置文件的默认路径为 `/etc/my.cnf`，现场以实际情况为准。

```
bind-address=0.0.0.0 #全部地址或者指定的 IP 地址
```

   2.2. 重启数据库。

```
service mysqld restart
```

   2.3. 验证配置是否生效。

```
netstat -tln
```

3. 重新执行校验任务。

SQL Server 修复方法

关闭防火墙或禁用 trigger。

PostgreSQL 修复方法

1. 请在 `pg_hba.conf` 文件中加入允许 DTS 网络段的访问策略。或者在迁移过程中临时放开所有网段的访问策略。如在此文件中添加一行：

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

2. 修改完成后，可重启数据库实例，让配置生效：

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

3. 重新执行校验任务。

MongoDB 修复方法

参考 MySQL 中的方法配置 bind-address。

网络端口未放通

检测方法

常见数据库默认端口如下，需要确认这些端口已放通。如果用户修改了默认端口，请按实际情况修改放通的端口。

- MySQL：3306
- SQL Server：1433
- PostgreSQL：5432
- MongoDB：27017
- Redis：6379

修复方法

放通相应的数据库端口。

如果源数据库为 SQL Server，还需要同时放通文件共享服务端口445。

数据库帐号或密码不正确

检测方法

登录源数据库，验证帐号和密码是否正确。

修复方法

在 DTS 控制台 修改数据迁移任务，输入正确的数据库帐号和密码后重新执行校验任务。

源库存在性检查

最近更新时间：2021-11-15 16:08:05

检查详情

检查要迁移的源库是否已经不存在了，如果不存在则会报错。

修复方法

在迁移过程中不要删除源库，如果删除了需要重新构建迁移任务。

目的库存在性检查

最近更新时间：2021-11-15 16:08:05

检查详情

检查要迁移的目标库是否已经不存在了，如果不存在则会报错。

修复方法

在迁移过程中不要删除目标库，如果删除了需要重新构建迁移任务。

周边检查

最近更新时间：2022-08-24 16:28:42

MySQL/TDSQL MySQL/TDSQL-C 检查详情

- 检查要求：源数据库环境变量参数 `innodb_stats_on_metadata` 需要设置为 `OFF` 。
- 检查说明：
  - `innodb_stats_on_metadata` 参数开启时，每当查询 `information_schema` 元数据库里的表，Innodb 就会更新 `information_schema.statistics` 表，导致访问时间变长。关闭后可加快对于 schema 库表的访问。
  - MySQL 5.6.6 之前版本 `innodb_stats_on_metadata` 参数预设值为 `ON` ，需要修改为 `OFF` 。MySQL 5.6.6 及其以后的版本预设值为 `OFF` ，不存在问题。

TDSQL PostgreSQL 版检查详情

DN 节点的 wal_level 必须是 logical。

修复方法

1. 登录源数据库。

2. 修改 `innodb_stats_on_metadata` 为 `OFF` 。

```
set global innodb_stats_on_metadata = OFF;
```

3. 查看配置是否生效。

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

系统显示结果类似如下：

```
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. 重新执行校验任务。

迁移网络检查

最近更新时间：2021-11-15 16:08:05

检查详情

检查内部迁移网络是否打通。

修复方法

如果迁移网络检查报错，请 提交工单 处理。

版本检查

最近更新时间：2022-09-21 18:04:04

MySQL/TDSQL MySQL 检查详情

- 检查要求：目标数据库版本必须大于或等于源数据库版本，且所有迁移和同步的版本符合版本要求。
- 检查说明：此处版本以大版本号区分，如5.6.x支持迁移或同步到5.6.x、5.7.x及以后版本，最后一位属于小版本号，小版本号不限制，如5.6.5可以迁移或同步到5.6.4，但是可能会有兼容性问题。

PostgreSQL 检查详情

- PostgreSQL 10.x 以前的版本（如9.x）不支持“全量+增量迁移”，若源库配置了增量迁移任务，则会导致校验失败。
- 当版本不一致时，会存在部分特殊的兼容性问题，迁移时会发出警告。请自行阅读各版本的兼容性报告，来确认业务是否有使用到不兼容的特性。

TDSQL PostgreSQL 版检查详情

登入数据库 `select tbase_version();` 必须返回 Tbase_V2.xx。

MongoDB 检查详情

源数据库和目标数据的版本符合 MongoDB 支持的版本。

SQL Server 检查详情

仅支持基础版迁移到高可用版本（包括双机高可用和集群版），且目标实例的版本号需要大于源实例的版本号。

Redis 检查详情

- Redis 源实例版本需要大于等于2.2.6，2.2.6以下版本不支持 DTS 迁移。
- 建议目标数据库版本大于或等于源数据库版本，否则会报警告，低版本向高版本迁移，有兼容性问题。

- 目标数据库 Redis 的 Proxy 版本为最新版本。

修复方法

请按 数据迁移支持的数据库 和 数据同步支持的数据库 中的版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例权限检查

最近更新时间：2022-09-16 10:06:51

检查详情

检查用户是否具备对数据库的操作权限，具体参考如下对应链接。

- 数据迁移权限要求
  - MySQL/TDSQL-C/MariaDB/Percona 之间的数据迁移
  - PostgreSQL 数据迁移
  - MongoDB 数据迁移
  - SQL Server 数据迁移
- 数据同步权限要求
  - MySQL/MariaDB/Percona/TDSQL MySQL 与 TDSQL MySQL 之间的数据同步

TDSQL PostgreSQL 版检查内容

用户必须拥有 REPLICATION 权限，对应 pg_roles.rolreplication。

用户必须拥有被订阅表的 select 权限，如果是整库订阅，那么用户要拥有该 schema 下所有表的 select 权限。

修复方法

用户可能不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

帐号冲突检查

最近更新时间：2021-11-15 15:55:34

检查详情

检查目标库用户是否与源数据库的用户重复。

修复方法

在整实例迁移的场景下，如果目标实例中存在和源实例一模一样的帐号，需要将目标实例中的同名帐号删除。

- 如果目标库中的帐号为初始化帐号，请使用初始化帐号登录至数据库中，执行以下语句。

```
create user 新用户 with password '密码';
grant pg_tencentdb_superuser to 新用户名;
alter user 新用户 with CREATEDB;
alter user 新用户 with CREATEROLE;
```

- 如果目标库中的帐号为新增用户，则使用新创建的用户登录至数据库中，删除冲突用户。

```
drop user 冲突用户;
# 如果冲突用户存在资源依赖，则请先修改依赖对象的owner，如表的owner修改语句为：
alter table 表名 owner to 新用户;
```

当冲突用户删除完成后，请重新执行校验任务。

部分实例参数检查

最近更新时间：2021-11-15 15:15:58

MySQL/TDSQL MySQL/TDSQL-C/MariaDB/Percona 检查详情

- 源库表的 `row_format` 不能为 `FIXED` 。
- 源库和目标库 `lower_case_table_names` 变量必须一致。
- 目标库 `max_allowed_packet` 参数设置至少为4M。
- 源库变量 `connect_timeout` 必须大于10。
- 在 MySQL/TDSQL MySQL/TDSQL-C 迁移到 MySQL 的场景中，如果源实例存在耗时较长的 SQL 在运行，会报警告，提示“源实例有耗时较长的 SQL 在运行，可能导致锁表，请稍后重试或对源实例中的 SQL 进行处理”。

修复方法

修改源库 row_format 参数

数据库中表的 `row_format` 的取值为 `FIXED` 时，表格中每行的存储长度超过限制值时会溢出，发生报错。因此需要修改为其他模式，如 `DYNAMIC` ，使每行的存储长度会随内容的长度而变化。

如发生此类报错，请参考如下指导进行修复。

1. 登录源数据库。
2. 修改 `row_format` 参数为 `DYNAMIC` 。

```
alter table table_name row_format = DYNAMIC
```

3. 查看配置是否生效。

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

系统显示结果类似如下：

```
mysql> show table status like '%row_format%';
+---------------+----------+
| Variable_name | Value |
+---------------+----------+
| row_format | DYNAMIC |
+---------------+----------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库和目标库 lower_case_table_names 变量保持一致

`lower_case_table_names` 是 MySQL 设置大小写是否敏感的一个参数，不同的取值情况如下：Windows 或 macOS 环境对大小写是不敏感的，但是 Linux 环境却是敏感的，为了保证不同系统的兼容性，需要将大小写敏感规则设置统一。

- 0：表名存储为给定的大小写，比较的时候区分大小写。
- 1：表名存储在磁盘是小写的，比较的时候不区分大小写。
- 2：表名存储为给定的大小写，比较的时候是小写的。

如发生此类报错，请参考如下指导将源库和目标库的参数改为统一。

1. 登录源数据库。
2. 查看源库和目标库中的 `lower_case_table_names` 取值。

```
show variables like '%lower_case_table_names%';
```

3. 修改 `lower_case_table_names` 参数。

```
alter global lower_case_table_names = 1
```

4. 参考如下命令重启数据库。

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

5. 查看配置是否生效。

```
show variables like '%lower_case_table_names%';
```

系统显示结果类似如下：

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

6. 重新执行校验任务。

修改目标库 max_allowed_packet 参数

`max_allowed_packet` 为最大允许的传输包。设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络报，影响系统性能。

如发生此类报错，请参考如下指导进行修复。

1. 登录目标数据库。
2. 修改 `max_allowed_packet` 参数。

```
set global max_allowed_packet = 4M
```

3. 查看配置是否生效。

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

系统显示结果类似如下：

```
mysql> show global variables like '%max_allowed_packet%';
+------------------------+---------+
| Variable_name | Value |
+------------------------+---------+
| max_allowed_packet | 4194304 |
+------------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库变量 connect_timeout

`connect_timeout` 为数据库的连接时间，超过 `connect_timeout` 设置值的连接请求将会被拒绝。如果设置过小，会导致数据库连接频繁断开，影响处理效率，因此建议该参数取值大于10。

如发生此类报错，请参考如下指导进行修复。

1. 登录源数据库。
2. 修改 `connect_timeout` 参数。

```
set global connect_timeout = 10
```

3. 查看参数是否修改成功。

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

系统显示结果类似如下：

```
mysql> show global variables like '%connect_timeout%';
+-----------------------+-------+
| Variable_name | Value |
+-----------------------+-------+
| connect_timeout | 10 |
+-----------------------+-------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

源实例参数检查

最近更新时间：2022-09-21 18:04:04

检查详情

- Redis 迁移场景，目标数据库为腾讯云 Redis 时，源实例的数据库个数需要小于等于目标实例的数据库个数。
- 源实例的状态是否正常。

修复方法

- 修改源和目标实例的数据库个数，然后重新启动校验任务。
- 确认源实例的状态。

源实例是否为从机

最近更新时间：2023-07-06 15:34:16

检查详情

源库必须为 Slave 节点，否则校验项会报警告，警告项不会阻止任务的继续进行，用户可以忽略后继续任务，但是需要评估忽略后的影响。
迁移过程中 DTS 会对源库进行 bgsave 操作，消耗源库的内存和资源，源库为 Master 节点时对业务的写入影响较大。
如果源库为腾讯云 Redis 数据库，默认数据迁移过程中使用 Slave 节点来迁移，用户无需关注。
如果源库为本地自建的 Redis 数据库，可能会出现使用 Master 迁移的场景，如果校验报警告，建议修改源库为从库。

修复方法

重新配置迁移任务，源库的参数配置为 Slave 节点信息。

参数配置冲突检查

最近更新时间：2021-11-15 15:55:33

检查详情

- 检查要求：对源和目标数据库的如下参数值进行检查，如果源和目标库的参数值不一致，则会提示校验警告。警告不会阻塞迁移任务，但是会对业务有一定的影响，请评估后自行决定是否修改。
  TimeZone，lc_monetary，lc_numeric，array_nulls，server_encoding，DateStyle，extra_float_digits，gin_fuzzy_search_limit，xmlbinary，constraint_exclusion。

- 业务影响：如果参数未设置一致，可能会导致源库和目标库的数据不一致，具体影响如下。
  - TimeZone：设置实例的时区，此参数值如果不一致，可能会导致迁移后数据错误。
  - lc_monetary：设置实例货币模式，此参数值如果不一致，可能会导致迁移后货币数字错误。
  - lc_numeric：设置实例数字模式，此参数值不一致，可能会导致迁移后数据错误。
  - array_nulls：设置数组是否允许为空，此参数值不一致，可能会导致迁移数据不一致，某一些数据无法迁移成功。
  - server_encoding：设置实例的字符集，此参数值不一致，可能会导致数据保存乱码。
  - DateStyle：设置日期显示格式，此参数值不一致，可能会导致数据无法迁移成功。
  - extra_float_digits ：设置浮点值的输出精度，此参数值不一致，会影响数据精度问题，在高精度的数据库使用场景，会导致迁移后的数据不一致问题。
  - gin_fuzzy_search_limit：设置 GIN 索引返回的集合尺寸的软上限，此参数值不一致，会导致迁移后数据显示结果不一致的问题。
  - xmlbinary：设置 xml 函数转换的结果问题，此参数值不一致，可能会导致在目标库中执行相应函数时与源库的行为不一致的问题。
  - constraint_exclusion：设置约束是否生效，此参数值不一致，可能会导致迁移后数据不一致的问题。

修复方法

1. 使用 superuser 账号登录源数据库。
2. 执行下列示例语句修改对应的参数：
   - 用户可先选择源库的参数修改，如源数据库对应参数不能修改，则需要修改目标库对应参数，目标库的修改请通过 提交工单 处理。
   - server_encoding 参数无法在源库修改，如果该参数异常，请检查该参数在目标库是否已经创建，如果已经创建，且和源库不一致，则需要申请新的实例，如果未创建，则参考如下方法修改（当前云数据库实例仅支持 UTF8 与 LATIN 两种字符集）。

```
alter system set timezone='参数值';
alter system set lc_monetary='参数值';
alter system set lc_numeric='参数值';
```

目标实例内容冲突检查

最近更新时间：2022-09-21 18:04:04

检查详情

MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL 检查要求

- 目标实例不能有和源库同名的对象。如果存在冲突报错，可任选如下一个方式进行修复。
  - 方法一：使用库表映射。
  - 方法二：修改或删除目标数据库中的同名对象。
  - 方法三：从迁移对象中移除同名对象。
- 整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。
- 选择高级对象时， 目标库不能有冲突的高级对象。如果存在冲突报错，需要删除冲突的对象。

PostgreSQL 检查要求

- 目标实例不能有和源库同名的对象。
- 整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。

MongoDB 检查要求

目标实例可以存在与源库同名，但是不能存在数据（只能为空表）。如果存在冲突报错，可任选如下一个方式进行修复。

- 方法一：删除目标库中的对应库表，或者删除目标库同名库表内的数据。
- 方法二：从迁移对象中移除同名对象。

SQL Server 检查要求

目标实例不能有和源库同名的库，否则会报错。

如果报错请在目标库中删除或者重命名同名的库。

Redis 检查要求

目标实例内容必须为空，否则会报错。

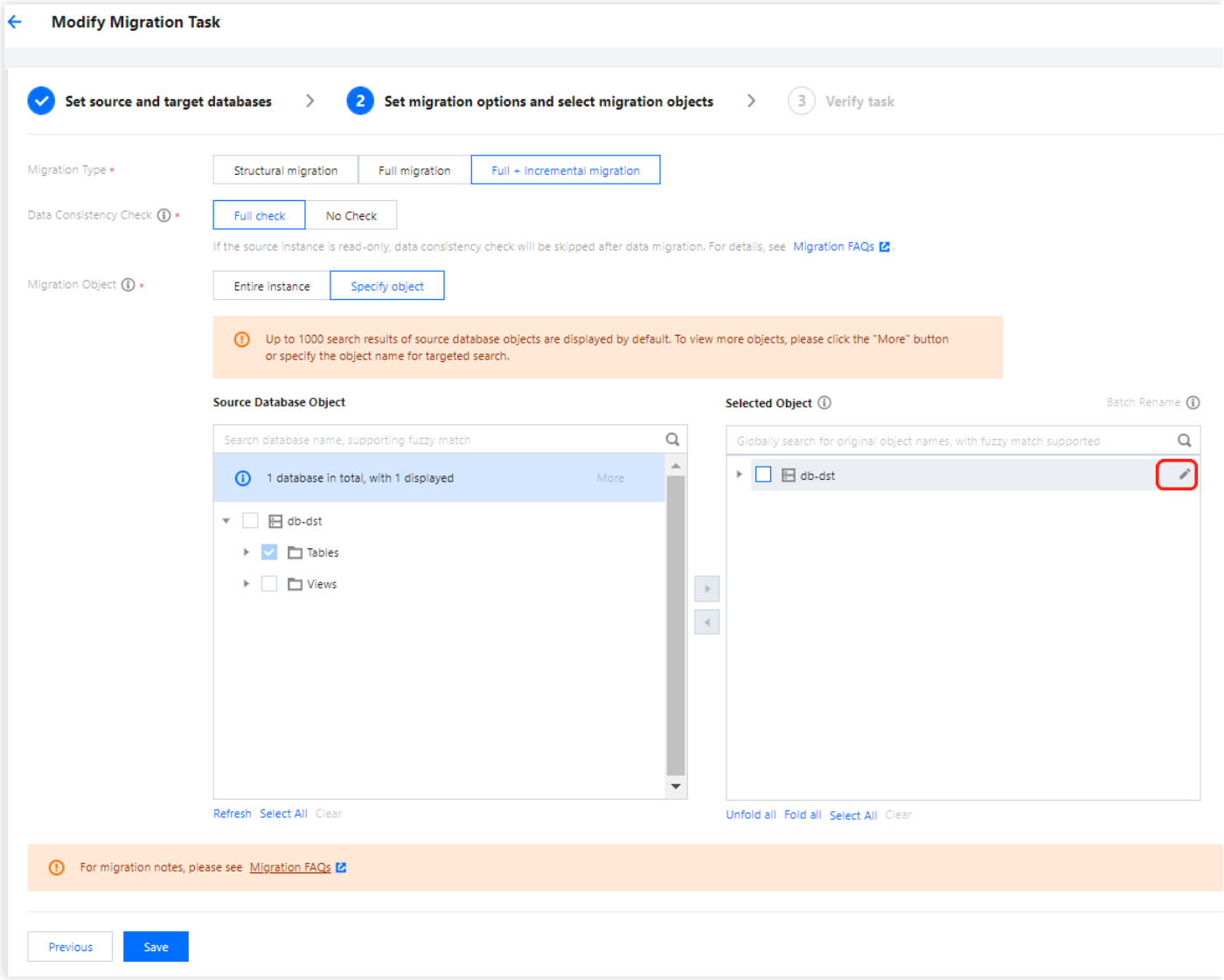
如果报错请删除目标实例中的内容，然后重新启动校验任务。

修复方法

使用库表映射（仅适用于 MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL）

使用 DTS 库表映射功能，将同名的待迁移对象映射为目标数据库中的其他名称。

1. 登录 DTS 控制台，选择对应的迁移任务，在**操作**列，选择**更多 > 修改**。
2. 在选择迁移对象右侧“已选对象”中，将鼠标悬浮在需要修改的对象上即可显示编辑按钮，然后重命名对象。



3. 重新执行校验任务。

修改目标数据库中的同名对象

登录目标数据库，重命名或删除目标数据库中和迁移对象同名的对象。

从迁移对象中移除同名对象

修改迁移任务配置，从迁移对象中移除同名对象，该对象不进行数据迁移。

1. 登录 DTS 控制台，选择对应的迁移任务，在**操作**列，选择**更多 > 修改**。
2. 在迁移对象中，移除同名的对象。
3. 重新执行校验任务。

删除目标库中的内容

登录目标数据库，删除目标数据库中的同名对象或者整库内容，然后重新执行校验任务。

目标实例空间检查

最近更新时间：2022-09-21 18:04:05

检查详情

MySQL/TDSQL-C MySQL/TDSQL MySQL/PostgreSQL 检查要求

目标库存储空间需要在源库待迁移库表空间的1.2倍以上。
全量数据迁移会并发执行 INSERT 操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间很可能会比源实例的表存储空间大。

MongoDB 检查要求

目标库存储空间需要在源库待迁移库表空间的1.3倍以上。

Redis 检查要求

目标库存储空间大于等于源库存储空间的1.5倍以上。

修复方法

- 删除目标库中的部分数据，以便腾出足够的空间。
- 升级目标库存储规格，使用更大容量的实例进行迁移。

目的端负载校验

最近更新时间：2021-11-15 16:02:04

检查详情

- 检查要求：DTS 迁移会导致目的端负载变高，如果在迁移过程中，目的端有业务使用，则会发出校验警告。警告不会阻塞任务的继续，但会对业务有一定影响，请用户评估后自行决定是否忽略警告。
- 业务影响：MongoDB DTS 采用逻辑同步的方式进行数据迁移，会对目的端 CPU 负载造成一定压力，如果目的端有业务使用，请谨慎评估发起。

修改方法

停止目标端的业务使用，重新执行校验任务。

本地磁盘空间检查

最近更新时间：2021-11-15 16:02:04

检查详情

检查源数据库所在服务器硬盘空间是否足够，源实例自建迁移场景中，源数据库所在服务器硬盘空间至少需要预留50GB，建议预留空间是您迁移数据所占空间的1.5倍，避免迁移过程中报错。

修复方法

删除源数据库所在服务器中的部分数据，以便腾出足够的空间。

Binlog 参数检查

最近更新时间：2022-10-11 16:19:39

MySQL/TDSQL MySQL/TDSQL-C 检查详情

源数据库 binlog 相关参数需要按照如下要求配置，如果校验不通过，请参考本文后续指导进行修复。

- `log_bin` 变量必须设置为 `ON` 。
- `binlog_format` 变量必须设置为 `ROW` 。
- `binlog_row_image` 必须设置为 `FULL` 。
- 如果源数据库为 MySQL 5.6 及以上版本， `gtid_mode` 只支持设置为 `ON` 和 `OFF` ，建议将 `gtid_mode` 设置为 `ON` ，设置为 `OFF` 会报警告，设置为 `ON_PERMISSIVE` 和 `OFF_PERMISSIVE` 会报错。
- `server_id` 参数需要手动设置，且值不能设置为0。
- 不允许设置 `do_db` ， `ignore_db` 。
- 对于源实例为从库的情况， `log_slave_updates` 变量必须设置为 `ON` 。
- 建议源库 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。

修复方法

开启 binlog

`log_bin` 是 binlog 的开关控制参数，需要将 binlog 打开，以便记录所有的数据库表结构和表数据变更日志。

如发生类似报错，请参考如下指导进行修复。

1. 登录源数据库。
2. 参考如下内容修改源数据库的配置文件 `my.cnf` 。因 log_bin 参数修改后需要重启数据库，所以建议同步修改 binlog_format 和 binlog_row_image 参数为校验要求配置。

> 说明：
> `my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。

```
log_bin = MYSQL_BIN
binlog_format = ROW
binlog_row_image = FULL
```

3. 参考如下命令重启源数据库。

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

说明：
[$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。
4. 确认 binlog 功能是否已启用。

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

系统显示结果类似如下：

```
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. 重新执行校验任务。

修改 binlog_format 参数

`binlog_format` 为 binlog 的记录模式，有以下三种：

- `STATEMENT` ：每一条会修改数据的 SQL 都会记录到 master 的 binlog 中，slave 在复制的时候，会执行和原来 master 端相同的 SQL。该模式可以减少 binlog 日志量，但是对某些特定的函数进行复制时，slave 端不能正确复制。

- `ROW`：binlog 日志中会记录成每一行数据修改的形式，然后在 slave 端再对相同的数据进行修改。该模式会保证 master 和 slave 的正确复制，但是 binlog 日志量会增加。
- `MIXED`：前两种模式的结合，MySQL 会根据执行的每一条具体的 SQL 语句来区分对待记录的日志形式，在 `STATEMENT` 和 `ROW` 之间选择一种。

综上，为了保证 master 和 slave 的正确复制，`binlog_format` 参数需要设置为 `ROW`。如发生类似报错，请参考如下指导进行修复。

> 说明：
> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。

2. 参考如下命令修改 `binlog_format`。

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

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

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

系统显示结果类似如下：

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

5. 重新执行校验任务。

修改 binlog_row_image 参数

`binlog_row_image` 参数决定了 binlog 是如何记录前镜像（记录修改前的内容）和后镜像（记录修改后的内容）的，这会直接影响到数据闪回、主从复制等功能。

`binlog_row_image` 参数只在 `binlog_format` 配置为 `ROW` 模式下生效。具体取值影响如下：

- `FULL` ：在 `ROW` 模式下，binlog 会记录前镜像和后镜像的所有列的数据信息。
- `MINIMAL` ：在 `ROW` 模式下，当表没有主键或唯一键时，前镜像记录所有列，后镜像记录被修改的列；如果存在主键或唯一键，不管是前镜像还是后镜像，都只记录有影响的列。

综上，`binlog_row_image` 需要配置为 `FULL` ，源数据库的 binlog 记录全镜像。如发生报错，请参考如下步骤修复。

> 说明：
> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步SQL线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。
2. 参考如下内容修改 `binlog_row_image` 。

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

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

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

系统显示结果类似如下：

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

4. 重新执行校验任务。

修改 gtid_mode 参数

GTID（Global Transaction Identifier， 全局事务标识），用于在 binlog 中唯一标识一个事务，使用 GTID 可以避免事务重复执行导致数据混乱或者主从不一致。
GTID 是 MySQL 5.6 的新特性，所以 MySQL 5.6 及之后版本存在此问题。DTS 只支持 `gtid_mode` 设置为 `ON` 和 `OFF`，建议将 `gtid_mode` 设置为 `ON`，否则校验时会报警告。

警告不影响迁移或同步任务进行，但是会对业务造成一定的影响：设置 GTID后，在增量数据同步阶段，如果源实例发生 HA 切换，DTS 服务切换重连，任务几乎无感知；反之，任务会中断后失败，且不可恢复。

`gtid_mode` 的取值如下，在修改 `gtid_mode` 的值时，只能从以下四个值逐级修改，例如，需要从 `OFF` 修改为 `ON`，则 `gtid_mode` 修改顺序为 `OFF` <-> `OFF_PERMISSIVE` <-> `ON_PERMISSIVE` <-> `ON`。

- `OFF`：主库所有新启的事务以及从库的事务都要求是匿名事务。
- `OFF_PERMISSIVE`：主库新启的事务是匿名事务，但从库事务允许是匿名的或者是 GTID 事务，但不允许只是 GTID 模式。
- `ON_PERMISSIVE`：主库新启的事务是 GTID 事务，从库事务允许是匿名的或者是 GTID 事务。
- `ON`：主库新启的事务是 GTID 事务，从库的事务也要求是 GTID 事务。

如果发生类似警告，请按照如下指导进行修复。

1. 登录源数据库。

2. 在主从复制结构两边都设置 `gtid_mode = OFF_PERMISSIVE`。
   MySQL 5.7.6 之前的版本需要在 `my.cnf` 配置文件中修改并重启数据库才能生效，5.7.6 及之后的版本通过全局命名修改，不需要重启数据库，但是需要重置所有业务连接。

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

3. 在主从复制结构两边都设置 `gtid_mode = ON_PERMISSIVE`。

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

4. 在各个实例节点上执行如下命令，检查匿名事务是否消耗完毕，参数值为 `0` 则代表消耗完毕。

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

系统显示结果类似如下：

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

5. 在主从复制结构两边都设置 `gtid_mode = ON` 。

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

6. 在 `my.cnf` 文件中添加如下内容，后续重启数据库后使初始值生效。

> 说明：
> `my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。

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

7. 重新执行校验任务。

修改 server_id 参数

`server_id` 参数需要手动设置，且值不能设置为0。该参数系统预设值为 `1` ，如果查询该参数显示为 `1` 不一定正确，需要手动进行配置。

1. 登录源数据库。

2. 参考如下内容修改 `server_id` 。

```
set global server_id = 2; //建议设为大于1的整数，此处仅为示例
```

3. 通过如下命令查看参数修改是否生效。

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

系统显示结果类似如下：

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

4. 重新执行校验任务。

删除 do_db，ignore_db 设置

binlog 会记录数据库所有执行的 DDL 和 DML 语句，而 do_db，ignore_db 则是设置 binlog 记录的过滤条件。

- `binlog_do_db` ：只记录指定数据库的二进制日志，默认全部记录。
- `binlog_ignore_db` ：不记录指定的数据库的二进制日志。

设置 do_db，ignore_db 后，会导致一些跨库操作 binlog 记录不全，主从复制出现异常，因此不建议设置。如发生类似报错，请参考如下指导进行修复。

1. 登录源数据库。
2. 修改源数据库的配置文件 `my.cnf` ，删除 do_db，ignore_db相关设置。

> 说明：
> `my.cnf` 配置文件的默认路径为 `/etc/my.cnf` ，现场以实际情况为准。

3. 参考如下命令重启源数据库。

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

> 说明：
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。
> 4. 确认参数修改是否生效。

```
show master status;
```

系统显示结果类似如下：

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

5. 重新执行校验任务。

修改 log_slave_updates 参数

在主从复用结构中，从库开启 `log-bin` 参数，直接在从库操作数据时，可以记录在 binlog 中，但是从库从主库上复制数据时，不能记录在 binlog 中，所以从库作为其他从库的主库时，需要打开 `log_slave_updates` 参数。

1. 登录源数据库。

2. 在源数据库的配置文件 `my.cnf` 中增加如下内容。

> 说明：
> `my.cnf` 配置文件的默认路径为 `/etc/my.cnf`，现场以实际情况为准。

```
log_slave_updates = ON
```

3. 参考如下命令重启源数据库。

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

> 说明：

[$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 查看配置是否生效。

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

系统显示结果类似如下：

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

5. 重新执行校验任务。

Oplog 校验

最近更新时间：2021-11-15 15:55:34

检查详情

- 检查要求：进行全量 + 增量迁移时，能够从源端获取到 Oplog。
- 检查说明：增量迁移需要通过 Oplog 进行回放，如果源端 local 库下不存在 oplog.rs 或者 oplog.$main 表格，则无法获取 Oplog。

修复方法

将源端以副本集或者主从方式启动，保证操作能够产生 Oplog，并且记录在源端 local 库下。

外键依赖检查

最近更新时间：2022-09-02 15:04:25

MySQL/MariaDB/Percona/TDSQL-C/TDSQL MySQL 检查详情

- 外键依赖只能设置为 `NO ACTION` 、 `RESTRICT` 。
- 部分库表迁移时，有外键依赖的表必须齐全。

TDSQL MySQL（TDStore）检查详情

不支持外键依赖数据，如果源库有外键数据任务校验会报错。

修复方法

- MySQL/MariaDB/Percona/TDSQL-C/TDSQL MySQL：请修改外键参数为 DTS 支持的取值类型。
- TDSQL MySQL（TDStore）：请删除对应的外键参数内容。

修改外键规则

MySQL 在设置外键的时候，删除时和更新时有四个值可以选择。

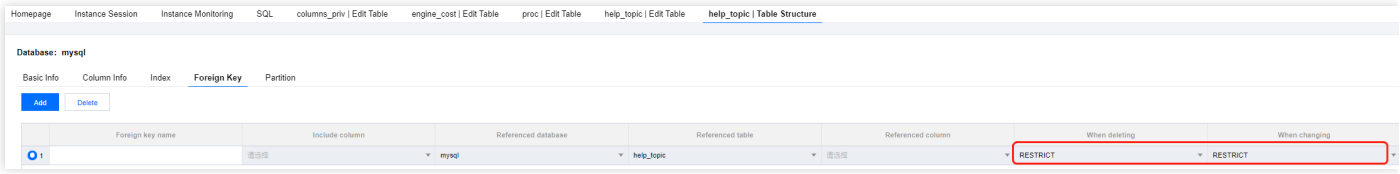
- `CASCADE` ：父表进行删除或者更新记录时，子表会同步删除或更新关联记录。
- `SET NULL` ：父表进行删除或者更新记录时，子表会将关联记录的外键字段所在列设为 `null` （子表外键不能设为 `not null` ）。
- `RESTRICT` ：父表执行删除或更新记录时，如果子表中有关联该父表的记录，则拒绝该父表删除请求。
- `NO ACTION` ：同 RESTRICT，也是首先检查外键。

如果发生报错，请参考如下指导修复。

Windows 操作指导

1. 登录源数据库 DMC 平台。

2. 在左侧目标树上选中要修改的表，在打开的表编辑界面上，单击**外键**页签，修改外键参数，如下图所示。



3. 修改完成后，单击**保存**。
4. 重新执行校验任务。

Linux 操作指导

1. 登录源数据库。

2. 删除原来的外键设置。

```
alter table `表名称1` drop foreign key `外键名称1`;
```

3. 重新添加外键设置。

```
alter table `表名称1` add constraint `外键名称2` foreign key `表名称1`(`列名1`) ref
erences `表名称2`(`列名1`)
on delete cascade on update cascade;
```

4. 重新执行校验任务。

完善迁移对象

修改迁移任务配置，在迁移对象中勾选具有关联关系的对象。

1. 在 DTS 控制台，选择对应的迁移任务，在**操作**列选择**更多 > 修改**。
2. 在**迁移对象**中勾选具有关联关系的对象。
3. 重新执行校验任务。

视图检查

最近更新时间：2021-11-15 15:52:31

MySQL/TDSQL-C 检查详情

- 检查要求：在导出视图结构时，DTS 会检查源库中 `DEFINER` 对应的 user1（[DEFINER = user1]）和迁移目标的 user2 是否一致。
  - 如果一致则迁移后不做改动。
  - 如果不一致，则迁移后修改 user1 在目标库中的 `SQL SECURITY` 属性，由 `DEFINER` 转换为 `INVOKER`（[INVOKER = user1]），同时设置目标库中 `DEFINER` 为迁移目标的 user2（[DEFINER = 迁移目标user2]）。

- 检查说明：`SQL SECURITY` 参数用来表示用户访问指定视图时，系统按照谁的权限来执行。
  - `DEFINER`：表示只有定义者才能执行。
  - `INVOKER`：表示拥有权限的调用者可以执行。
    默认情况下，系统指定为 `DEFINER`。

TDSQL MySQL 检查详情

只允许和迁移目标 user@host 相同的 definer，即在导出视图结构时，DTS 检查源库中 definer 对应的 user1（[DEFINER = user1]）和迁移目标的 user2（user@host）是否一致，一致则支持迁移视图，否则不支持。

对于和迁移目标 user@host 不同的 definer，如果需要迁移，则要修改源数据库视图的 definer（修改为迁移目标用户），或者在迁移/同步任务中不勾选，等任务结束后手动同步视图。

高级对象检查

最近更新时间：2022-09-21 17:24:22

MySQL/MariaDB/Percona 检查详情

选择迁移/同步高级对象时，DTS 会对如下内容进行校验。报错项必须要处理才能继续任务，警告项用户评估业务风险后可忽略，继续任务。

- 报错项：目标实例参数 log_bin_trust_function_creators 必须为 ON。
- 警告项：
  - 迁移/同步高级对象与库表重命名功能冲突，选择高级对象后需要取消库表重命名。
  - 选择高级对象的函数，存储过程时，DTS 会检查源库中 `DEFINER` 对应的 user1（[DEFINER = user1]）和执行任务账号 user2 是否一致。
    - 如果一致则迁移/同步后不做改动。
    - 如果不一致，则迁移/同步后修改 user1 在目标库中的 `SQL SECURITY` 属性，由 `DEFINER` 转换为 `INVOKER`（[INVOKER = user1]），同时设置目标库中 `DEFINER` 为执行任务账号的 user2（[DEFINER = 执行任务账号user2]）。
  - 高级对象的迁移/同步时间：
    - 存储过程和函数，在“源库导出”阶段进行迁移/同步。
    - 触发器和事件，没有增量任务，在任务结束时进行迁移/同步；有增量任务，在用户单击**完成**操作后开始迁移/同步，所以单击**完成**后任务的过渡时间会长一些。

修复方法

修改 log_bin_trust_function_creators 参数。

log_bin_trust_function_creators 用于控制是否信任用户将存储函数写入 binlog 日志中。设置为 OFF，仅 SUPER 权限的用户可将创建的存储函数操作写入 binlog 日志，设置为 ON，非 SUPER 权限的用户也可将创建的存储函数操作写入 binlog 日志中。

发生报错时，请参考如下步骤进行修改。

1. 登录源数据库。

2. 参考如下内容修改 log_bin_trust_function_creators 参数。

```
set global log_bin_trust_function_creators = ON;
```

3. 通过如下命令查看参数修改是否生效。

```
show variables like '%log_bin_trust_function_creators%';
```

系统显示结果类似如下：

```
mysql> show variables like '%log_bin_trust_function_creators%';
+---------------------------------+-------+
| Variable_name | Value |
+---------------------------------+-------+
| log_bin_trust_function_creators | ON |
```

4. 重新执行校验任务。

增量迁移预置条件检查

最近更新时间：2022-06-21 11:13:25

检查详情

当迁移类型选择增量迁移时，需要对如下条件进行检查，否则校验失败。

- 源和目标库的主版本号需要为 PostgreSQL 10.x 之前。
- 源实例的 `wal_level` 必须为 `logical` 。
- 目标库 `max_replication_slots` 和 `max_wal_senders` 参数需要大于待迁移的数据库总数。
- 目标实例的 `max_worker_processes` 必须大于 `max_logical_replication_workers` 的值。
- 待迁移表中不能存在 unlogged table，否则无法迁移。

修复方法

如果版本不符合要求，请升级版本。修改参数 `wal_level` ， `max_replication_slots` ， `max_worker_processes` 和 `max_wal_senders` 的方法如下。

1. 登录源数据库。

> 说明：
>
> - 如源数据库为自建数据库，需要登录至数据库的运行服务器上，进入数据库数据主目录中，一般为 $PGDATA。
> - 如源数据库为其他云数据库，请使用相关云平台的参数修改方法。
> - 如需要修改目标实例的参数，请通过 提交工单处理。

2. 找到 postgresql.conf 文件，打开此文件，修改 `wal_level` 。

```
wal_level = logical
```

3. 修改完成后，重启数据库实例。

4. 登录至数据库实例中，使用以下命令查看参数值是否设置正确：

```
postgres=> select name,setting from pg_settings where name='wal_level';
name | setting
-----------+---------
wal_level | logical
(1 row)
postgres=> select name,setting from pg_settings where name='max_replication_slo
ts';
name | setting
----------------------+---------
max_replication_slots | 10
(1 row)
postgres=> select name,setting from pg_settings where name='max_wal_senders';
name | setting
-----------------+---------
max_wal_senders | 10
(1 row)
```

5. 重新执行校验任务。

插件兼容性检查

最近更新时间：2021-11-15 15:55:33

检查详情

- 检查要求：检查源库已存在的插件，目标库是否也同样存在。
  在迁移过程前，无需在目标库中创建插件，迁移过程中 DTS 会自动为目标实例创建与源端匹配的插件。如果目标端无法创建对应的插件，或者目标库插件版本与源库不一致时，则会提示校验警告。警告不会阻塞迁移任务，但是会对业务造成一样的影响。

- 业务影响：PostgreSQL 的插件较多，大多数插件兼容性问题并不影响数据迁移，但是存储引擎类的插件兼容性问题（例如 timescaledb，pipelinedb，postgis），可能会导致迁移任务失败。

修复方法

- 非引擎类的插件兼容性问题（如 pg_hint_plan，pg_prewarm，tsearch2，hll，rum，zhparser），一般可以忽略，用户可自行评估业务情况处理。
- 引擎类的插件兼容性问题（如 timescaledb，pipelinedb，postgis），建议通过 提交工单 处理。

源端 Balancer 校验

最近更新时间：2021-11-15 16:02:04

检查详情

- 检查要求：源端为分片实例情况下，源端必须关闭 Balancer 才能发起迁移。
- 检查说明：增量迁移会获取 Oplog，开启 Balancer 的情况下，源端 moveChunk 可能会导致最终目的端数据不一致。

修复方法

1. 登录源数据库。
2. 使用如下命令关闭源端 Balancer。

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

3. 重新执行校验任务。

源端节点角色校验

最近更新时间：2021-11-15 16:02:04

检查详情

- 检查要求：MongoDB 迁移任务，源端为分片时，需要填写对应 mongos，config server，mongod 节点信息。
- 检查说明：mongos，config server 和 mongod 节点信息填写不能混乱，否则会导致数据迁移错乱，例如将 mongos 节点信息填入 mongod 填写框内。注意，每个分片只需要填写一个 mongod 节点。

修复方法

- DTS 任务填写框内填写正确节点信息。
- 每个分片只填写一个 mongod。

片建校验

最近更新时间：2021-11-15 16:02:04

检查详情

- 检查要求：当目的端为分片实例时，可以在目的端预设片建，如果目的端与源端表格片建不一致，则会发出警告提示。警告不会阻塞任务的继续，但会对业务有一定影响，请用户评估后自行决定是否忽略警告。
- 业务影响：部分片建不一致场景会导致迁移或者同步任务失败。

修复方法

如果目的端预设片建，参考如下命令在源端进行分片操作。

```
sh.shardCollection("<database>.<collection>", { <shard key> : "hashed" } , false, {numInitialChunks: 预置的chunk个数})
```

重新执行校验任务。

警告项检查

最近更新时间：2021-11-15 15:52:42

MySQL/TDSQL-C 检查详情

如下参数需要按照要求配置，否则校验时系统会发出警告，警告不影响迁移任务的进行，但是会对业务造成一定的影响，请用户评估后自行决定是否修改。

- 建议目标库 `max_allowed_packet` 的取值大于源库。
  - 业务影响：目标库的 `max_allowed_packet` 参数设置小于源库，会导致目标库数据无法写入，从而造成全量迁移失败。
  - 处理建议：修改目标库的 `max_allowed_packet` 参数，大于源库取值。

- 建议目标库的 `max_allowed_packet` 设置大于1GB。
  - 业务影响： `max_allowed_packet` 设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络报，影响系统性能。
  - 处理建议：参考如下命令修改 `max_allowed_packet` 参数。

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

- 建议源库和目标库的字符集保持一致。
  - 业务影响：源库和目标库的字符集不一致可能会导致乱码。
  - 处理建议：参考如下命令将源库和目标库字符集修改为一致。

```
set character_set_server = 'utf8';
```

- 建议使用2CPU，4G Mem以上规格的实例。

- 如果仅执行全量数据迁移，请勿在迁移过程中向源实例中写入新的数据，否则会导致源和目标数据不一致。针对有数据写入的场景，为实时保持数据一致性，建议选择全量+增量数据迁移。

- 有锁导出时：源实例需要使用 Flush Table With Read Lock 短暂加锁，其中的 MyISAM 表会锁定到全量数据导出完成。当前等待加锁超时时间设置为60秒, 该时间内无法获取锁将导致任务失败。

- 无锁导出时：对没有主键或非空唯一键的表会加读锁。

TDSQL MySQL 检查详情

如下参数需要按照要求配置，否则校验时系统会发出警告，警告不影响迁移任务的进行，但是会对业务造成一定的影响，请用户评估后自行决定是否修改。

- 建议目标库 `max_allowed_packet` 的取值大于源库。
  - 业务影响：目标库的 `max_allowed_packet` 参数设置小于源库，会导致目标库数据无法写入，从而造成全量迁移失败。
  - 处理建议：修改目标库的 `max_allowed_packet` 参数，大于源库取值。

- 建议目标库的 `max_allowed_packet` 设置大于1GB。
  - 业务影响： `max_allowed_packet` 设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络报，影响系统性能。
  - 处理建议：参考如下命令修改 `max_allowed_packet` 参数。

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

- 建议源库和目标库的字符集保持一致。
  - 业务影响：源库和目标库的字符集不一致可能会导致乱码。
  - 处理建议：参考如下命令将源库和目标库字符集修改为一致。

```
set character_set_server = 'utf8';
```

- 建议使用2CPU，4G Mem以上规格的实例。

- 如果仅执行全量数据迁移，请勿在迁移过程中向源实例中写入新的数据，否则会导致源和目标数据不一致。针对有数据写入的场景，为实时保持数据一致性，建议选择全量+增量数据迁移。

- 有锁导出时：源实例需要使用 Flush Table With Read Lock 短暂加锁，其中的 MyISAM 表会锁定到全量数据导出完成。当前等待加锁超时时间设置为60秒, 该时间内无法获取锁将导致任务失败。

- 无锁导出时：对没有主键或非空唯一键的表会加读锁。

- 源数据库实例为分布式数据库时，需要提前在目标库建立分表，否则这些表被迁移后都将是单表。

二级分区表检查

最近更新时间：2022-11-24 10:38:58

TDSQL MySQL 检查详情

- 检查要求：源库为 TDSQL MySQL 时，如果在数据迁移任务中遇到 二级分区 表，则系统会给出警告，警告不会影响迁移任务的继续，用户可以选择忽略警告，忽略后二级分区表会被跳过迁移；如果在数据同步任务中遇到二级分区表，则校验不通过，需要移除二级分区表。
- 业务影响：二级分区表由于底层是通过子表实现的，迁移或同步二级分区表会出现目标库数据异常。

修复方法

请根据提示从待迁移或同步列表中移除二级分区表。

待迁移的表是否有主键检查

最近更新时间：2024-02-22 11:09:27

TDSQL MySQL 检查详情

当目标库是 TDSQL MySQL 时，源数据库中的表必须有主键，暂不支持无主键表迁移。

修复方法

请按照界面提示修改无主键的表，然后重新执行校验任务。

待迁移表的 DDL 检查

最近更新时间：2024-02-22 11:12:35

TDSQL MySQL/MariaDB 检查详情

检查要求：当目标库是腾讯云数据库 TDSQL MySQL 或腾讯云数据库 MariaDB 时，如果待迁移的表列名或是表名包含除数字、下划线和英文字母外的其他字符，则校验不通过。
检查说明：TDSQL MySQL 和云数据库 MariaDB 的库、表、列名只支持数字、下划线和英文字母。

修复方法

按照界面提示去掉含有特殊字符的表，然后重新执行校验任务。

源实例迁移库和目标实例系统库冲突检查

最近更新时间：2022-11-24 10:38:58

TDSQL MySQL 检查详情

- 检查要求：当目标库是 TDSQL MySQL 时，如果待迁移的库中有名为 `sysdb` 的库，该库将不会被迁移。
- 检查说明： `sysdb` 是 TDSQL MySQL 的系统库，如果待迁移的库和系统库重名，会给出警告提醒，警告不会影响迁移任务的运行。

修复方法

请从待迁移库列表中移除 `sysdb` 。

目标实例和源实例表结构检查

最近更新时间：2022-11-24 10:38:58

TDSQL MySQL 检查详情

目标库是 TDSQL MySQL 时，建议提前在目标库建立好分表，并指定好 shardkey，否则会报警告，警告不影响任务的进行，用户可以忽略警告继续任务，忽略后 DTS 则会按照源库的表样式来在目标库创建表，如果源库为单机实例，则目标库会创建为单表。
如果用户已经在目标库创建好了分表或者不需要创建分表，DTS 检查源库和目标库表的表结构是否一致，当字段类型、字段名、编码不一致时校验会不通过。

修复方法

当校验不通过时，DTS 界面会提示结构不一致的表列表，对比这些表在源库和目标库结构的差异，修改目标库表结构。

表是否是 InnoDB 表检查

最近更新时间：2024-02-22 14:45:18

TDSQL MySQL/MariaDB 检查详情

当目标库是 TDSQL MySQL 或腾讯云数据库 MariaDB 时，只支持迁移或同步 InnoDB 表，如果待迁移或同步对象中含有非 InnoDB 的表，则校验不通过。

修复方法

按照界面提示在迁移或同步对象中移除非 InnoDB 的表，然后重新执行校验任务。

视图之间以及视图和表之间的互相依赖检查

最近更新时间：2022-11-24 10:38:58

MariaDB 检查详情

当目标库是腾讯云数据库 MariaDB 时，如果选择迁移的视图，没有和与其相依赖的表或视图一并迁移，则校验不通过。

修复方法

请按照界面提示，选择与迁移视图相依赖的其他对象一并迁移，然后重新执行校验任务。

约束检查

最近更新时间：2022-08-24 16:28:42

TDSQL PostgreSQL 版检查详情

被订阅的表必须拥有主键或 REPLICA IDENTITY 为 FULL。

修复方法

当校验不通过时，请修改对应的表格。