

基于 docker 的 mysql 主从同步

作者: [yexuejc](#)

原文链接: <https://ld246.com/article/1596531153493>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



目标：一台已经在运行mysql，增加从库

环境： docker mysql5.7.26 不同主机

步骤:

1. 在需要安装从库的主机上安装docker(自行百度)
2. 在从库主机上安装mysql (基于docker)

```
docker run -d \  
--name mysql-slave \  
-e MYSQL_ROOT_PASSWORD=123456 \  
-v /mnt/up/docker/mysql-slave/conf:/etc/mysql \  
-v /mnt/up/docker/mysql-slave/data:/var/lib/mysql \  
-v /mnt/up/docker/mysql-slave/logs:/var/log/mysql \  
-p 3306:3306 \  
-d mysql:5.7.26
```

解释:

- `docker run -d` 后台运行容器
- `--name mysql-slave` 给容器起名为 `mysql-slave`
- `-e MYSQL_ROOT_PASSWORD=123456` 设置mysql的root的密码为 `123456`
- `-v /mnt/up/docker/mysql-slave/conf:/etc/mysql` 挂载宿主机 `/mnt/up/docker/mysql-slave/conf`目录到容器 `/etc/mysql`,mysql的配置文件放在该目录下，映射到宿主机好配置，**注3 my.cnf(下方)**
- `-v /mnt/up/docker/mysql-slave/data:/var/lib/mysql` mysql的数据库文件，映射出来可以久化数据库文件（自行看需求是否需要映射）

- `-v /mnt/up/docker/mysql-slave/logs:/var/log/mysql \` mysql的日志文件, (自行看需求是否需映射)
- `-p 3306:3306 \` 暴露到宿主机上的端口
- `-d mysql:5.7.26` 指定mysql镜像版本

3. 查看容器是否启动正常

`docker logs mysql-slave`

出现如下内容代表配置的my.cnf未生效, 解决参考[docker-mysql使用中的坑](#)

[Warning] World-writable config file '/etc/mysql/my.cnf' is ignored.

4. 主库机上创建用于主从同步的账号, 建议创建一个专用账号

```
CREATE USER 'repl'@'%' IDENTIFIED BY 'passwd' REQUIRE SSL;
GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%';
FLUSH PRIVILEGES;
```

5. 主库机修改my.cnf配置,注意 `server_id`尽量唯一, 建议主从my.cnf对mysql的配置一致, 避免发奇葩问题

[mysqld]

#0,区分大小写; 1,不区分

#lower_case_table_names=1

时间少8个小时

default-time-zone = '+8:00'

#skip-grant-tables

sql_mode=ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,
O_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

#解决: mysql5.7 timestamp默认值 '0000-00-00 00:00:00' 报错

#GTID 主从复制#

gtid_mode=on

enforce_gtid_consistency=on

server_id=1

#binlog

log_bin=mysqlbin

log_slave_updates=1

#强烈建议, 其他格式可能造成数据不一致

binlog_format=row

#relay log

skip_slave_start=1

5. 重启主库生效

6. 从库机连接主库

```
docker run -it --rm mysql:5.7.26 mysql -h172.17.0.1 -P3306 -uroot -p123456 \
-e "CHANGE MASTER TO MASTER_HOST='192.168.1.120', MASTER_PORT=3306, MASTER_US
```

```
R='repl', MASTER_PASSWORD='passwd', MASTER_AUTO_POSITION=1, MASTER_SSL=1;" \
-e "START SLAVE;"
```

解释:

- 这段运行是创建一个临时的mysql容器进去登录到172.17.0.1:3306, 为什么这样配置? 因为在没有定docker网卡时, 使用默认网卡, ip段为172.17.0.x 宿主机的ip为172.17.0.1。上面我们把 `mysql-sla`容器的3306端口映射到宿主机的3306, 所以这里登录上去的mysql即为容器 `mysql-slave`的mysql当然可以选择自己喜欢的方式去登录, 毕竟我们的目的只是去执行sql, 不是纠结怎么登录
- 登录近从库后执行 `CHANGE MASTER TO MASTER_HOST='192.168.1.120', MASTER_PORT=306, MASTER_USER='repl', MASTER_PASSWORD='passwd', MASTER_AUTO_POSITION=1, MASTER_SSL=1;和 START SLAVE;`

前一句表示把本mysql (即从库) 的主库指向192.168.1.120并且登录账户是repl, 后一句表示是从库同步

7. 查看主从库同步状态

```
docker run -it --rm mysql:5.7.26 mysql -h172.17.0.1 -P3306 -uroot -p123456 -e "show slave status\G"
```

如果如下2个字段不是Yes, 请先查看错误消息如下:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

`Last_Error`会有错误消息, 请结合 `docker logs mysql-slave`的日志信息分析

```
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 212.64.35.217
Master_User: repl
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysqlbin.000002
Read_Master_Log_Pos: 17984
Relay_Log_File: 525d536d1c17-relay-bin.000002
Relay_Log_Pos: 411
Relay_Master_Log_File: mysqlbin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: No
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 1032
Last_Error: Could not execute Update_rows event on table openshop.QRTZ_SCHEDULER_STATE; Can't find record in 'QRTZ_SCHEDULER_STATE', Error_code: 1032; handler error A_ERR_END_OF_FILE; the event's master log mysqlbin.000002, end_log_pos 1755
Skip_Counter: 0
Exec_Master_Log_Pos: 1378
Relay_Log_Space: 17231
```

```
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: Yes
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 1032
Last_SQL_Error: Could not execute Update_rows event on table openshop.QRTZ_SC
EDULER_STATE; Can't find record in 'QRTZ_SCHEDULER_STATE', Error_code: 1032; handler err
r HA_ERR_END_OF_FILE; the event's master log mysqlbin.000002, end_log_pos 1755
Replicate_Ignore_Server_Ids:
Master_Server_Id: 8041418
Master_UUID: a1221a75-d8fe-11e9-acf9-0242ac110005
Master_Info_File: /var/lib/mysql/master.info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State:
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp: 200804 07:38:08
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set: a1221a75-d8fe-11e9-acf9-0242ac110005:4-44
Executed_Gtid_Set: 0235edd8-d61f-11ea-9d3c-0242ac11000e:1-761,a1221a75-d8fe-1
e9-acf9-0242ac110005:1-3
Auto_Position: 1
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0.00 sec)
```

注1：已运行的mysql进行主从同步，那么开始的从库是没有数据库，所以会无法同步，请先把主库上的数据拉一份到从库，再在从库机重新开启主从同步

```
docker run -it --rm mysql:5.7.26 mysql -h172.17.0.1 -P3306 -uroot -p123456 -e "stop slave;re
et slave;" && docker restart mysql-slave
```

解释:停止同步，重置主从链接并重启容器 `mysql-slave`，这个时候容器就恢复到第5步了，继续开始行第6步

注2：上面出现的错误

```
Last_Error: Could not execute Update_rows event on table openshop.QRTZ_SCHEDULER_STAT
; Can't find record in 'QRTZ_SCHEDULER_STATE', Error_code: 1032; handler error HA_ERR_END
```

OF_FILE; the event's master log mysqlbin.000002, end_log_pos 1755

这个错误是因为我手动拉了一份主库数据库到从库后再次操作第6步出现的，说的是数据库 `openshop` 表 `QRTZ_SCHEDULER_STATE` 的数据无法同步，经过查询：[解决方法http://www.itpub.net/thread-566441-1-1.html](http://www.itpub.net/thread-566441-1-1.html)，还有其它类似的错误基本都是数据约束或者主从配置不一致导致的，所以建议主配置一致

注3：my.cnf配置，具体参数请自行配置

```
[mysqld]
#0,区分大小写; 1,不区分
#lower_case_table_names=1
# 时间少8个小时
default-time-zone = '+8:00'

#skip-grant-tables
#解决: mysql5.7 timestamp默认值 '0000-00-00 00:00:00' 报错
sql_mode=ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,
O_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

#GTID 主从复制#
gtid_mode=on
enforce_gtid_consistency=on
server_id=8413

#binlog
log_bin=mysqlbin
log_slave_updates=1
#强烈建议，其他格式可能造成数据不一致
binlog_format=row

#relay log
skip_slave_start=1

#主从复制跳过1146错误 https://blog.csdn.net/gua\_\_gua/article/details/52869614
slave_skip_errors=1146
```

注4：通过docker容器备份数据库

1. 创建一个临时容器并挂载一个目录作为备份sql的存放

```
docker run -it -v /tmp/mysql:/etc/tmp --rm mysql:5.7.26 /bin/bash
```

2. 容器中执行如下命令备份全部数据库

```
mysqldump -h192.168.1.120 -uroot -p123456 --all-databases > /etc/tmp/backup-all.sql
```

3. 如果上述执行错误，那么就一个库一个库的备份了

```
mysqldump -h192.168.1.120 -uroot -p123456 --databases test1 > /etc/tmp/backup-test1.sql
mysqldump -h192.168.1.120 -uroot -p123456 --databases test2 > /etc/tmp/backup-test2.sql
```

4. 备份完成应该在宿主机的/tmp/mysql目录下能看到刚才备份的sql

5. 还原到从库

```
//登录到从库
mysql -h172.17.0.1 -P3306 -uroot -p123456
//使用source命令还原
source /etc/tmp/backup-all.sql
```

注5：遇到一些问题的参考

1. 主从复制错误处理总结

```
//已解决的错误
Last_Errno: 1146
Last_Error: Error executing row event: 'Table 'panda.t' doesn't exist'
```

2. mysql主从同步错误解决和Slave_IO_Running:NO

已解决的错误 Duplicate entry

错误提示如下

```
Error 'Duplicate entry '1' for key 1' on query. Default database: 'movivi1'. Query: 'INSERT INTO `v1vid0_user_samename` VALUES(null,1,'123','11','4545','123)'
```

```
Error 'You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near " at line 1' on query. Default database: 'club'. Query: 'INSERT INTO club.point_process ( GIVEID, GETID, POINT, CREATETIME, DEMO ) VALUES ( 0, 4971112, 5, '2010-12-19 16:29:28',' 1 ro in set (0.00 sec)
```

```
Mysql > Slave status\G;
```

显示: Slave_SQL_Running 为 NO

解决方法:

```
Mysql > stop slave;
Mysql > set global sql_slave_skip_counter =1;
Mysql > start slave;
```

参考: [Docker 部署 MySQL 5.7 & 8.0 主从集群](#)