> MySQL主主复制结构区别于主从复制结构。在主主复制结构中，两台服务器的任何一台上面的数据库存发生了改变都会同步到另一台服务器上，这样两台服务器互为主从，并且都能向外提供服务。

1 主机环境

1.1 软件版本

```
centos 7.4
docker 18.06.1-ce
mysql 5.7
```

1.2 目录结构

```
/opt/docker-mysql/
├── master
│   └── etc
│       └── my.cnf
├── run.sh
└── slave
    └── etc
        └── my.cnf
```

1.3 mysql启动脚本

```
# cat /opt/docker-mysql/run.sh
#!/bin/bash

echo run mysql master
docker run -d \
--restart=always \
--privileged=true \
--name=mysql-master \
--hostname=mysql-master \
-p 3307:3306 \
-e MYSQL_ROOT_PASSWORD=root \
-v /etc/localtime:/etc/localtime \
-v /opt/docker-mysql/master/etc/my.cnf:/etc/mysql/my.cnf \
-v /opt/docker-mysql/master/data:/var/lib/mysql:rw \
harbor.mxnet.io/library/mysql:5.7

echo run mysql slave
docker run -d \
--restart=always \
--privileged=true \
--name=mysql-slave \
--hostname=mysql-slave \
-p 3308:3306 \
-e MYSQL_ROOT_PASSWORD=root \
-v /etc/localtime:/etc/localtime \
-v /opt/docker-mysql/slave/etc/my.cnf:/etc/mysql/my.cnf \
-v /opt/docker-mysql/slave/data:/var/lib/mysql:rw \
harbor.mxnet.io/library/mysql:5.7
```

1.4 mysql配置文件

1.4.1 master

```
cat /opt/docker-mysql/master/etc/my.cnf
[mysql]

[mysqld]
pid-file    = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
datadir        = /var/lib/mysql
#log-error    = /var/log/mysql/error.log
# By default we only accept connections from localhost
#bind-address    = 127.0.0.1
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

lower_case_table_names = 1 #不区分大小写
```

```
character_set_server = utf8 #字符编码

log-bin=mysql-bin # 开启bin-log 日志，MySQL主从配置，必须开启
log-bin-index=mysql-bin

server_id=1 # 唯一的标识，与slave不同

log-slave-updates = true # 双主互备必须开启，否则只是主从关系
relay-log= relaylog
relay-log-index=relaylog
relay-log-purge=on

binlog-do-db=test #开启同步的数据库

#auto-increment-increment = 2
#auto-increment-offset = 1
```

1.4.2 slave

```
cat /opt/docker-mysql/slave/etc/my.cnf
[mysql]

[mysqld]
pid-file    = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
datadir        = /var/lib/mysql
#log-error    = /var/log/mysql/error.log
# By default we only accept connections from localhost
#bind-address    = 127.0.0.1
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

lower_case_table_names = 1 #不区分大小写
character_set_server = utf8 #字符编码

log-bin=mysql-bin # 开启bin-log 日志，MySQL主从配置，必须开启
log-bin-index=mysql-bin

server_id=2 # 唯一的标识，与master不同

log-slave-updates = true # 双主互备必须开启，否则只是主从关系
relay-log= relaylog
relay-log-index=relaylog
relay-log-purge=on

binlog-do-db=test #开启同步的数据库

#auto-increment-increment = 2
#auto-increment-offset = 2
```

> 注：二都只有server-id不同和auto-increment-offset不同
> auto-increment-offset是用来设定数据库中自动增长的起点的，为这两能服务器都设定了一次自动增长值2，所以它们的起点必须得不同，这样才能避免两台服务器数据同步时出现主键冲突

replicate-do-db 指定同步的数据库，我们只在两台服务器间同步test数据库

另：auto-increment-increment的值应设为整个结构中服务器的总数，本案例用到两台mysql服务器，所以值设为2

2 mysql配置双主同步

2.1 master上查看状态

```
docker exec -it mysql-master bash

ifconfig

172.17.0.2

mysql -uroot -proot

mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000003 |      154 | test         |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
```

```
1 row in set (0.01 sec)

mysql>
```

2.2 slave上配置同步

```
docker exec -it mysql-slave bash

ifconfig

172.17.0.3

mysql -uroot -proot

mysql> change master to master_host='172.17.0.2',master_user='root',master_password='root',master_port=3306, master_log_file='mysql-bin.
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> start slave;
Query OK, 0 rows affected (0.04 sec)

mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 172.17.0.2
                  Master_User: root
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 154
               Relay_Log_File: relaylog.000002
                Relay_Log_Pos: 320
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 154
              Relay_Log_Space: 520
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID: 72c4a4a3-4117-11e9-b71c-0242ac110002
             Master_Info_File: /var/lib/mysql/master.info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
           Master_Retry_Count: 86400
                  Master_Bind:
      Last_IO_Error_Timestamp:
     Last_SQL_Error_Timestamp:
               Master_SSL_Crl:
           Master_SSL_Crlpath:
           Retrieved_Gtid_Set:
            Executed_Gtid_Set:
                Auto_Position: 0
         Replicate_Rewrite_DB:
                 Channel_Name:
           Master_TLS_Version:
```

```
1 row in set (0.00 sec)

ERROR:
No query specified

mysql>


mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000003 |      154 | test         |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)

mysql>
```

2.3 master上配置同步

```
docker exec -it mysql-master bash

mysql -uroot -proot

mysql>
mysql> change master to master_host='172.17.0.3',master_user='root',master_password='root',master_port=3306, master_log_file='mysql-bin.
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> start slave;
Query OK, 0 rows affected (0.00 sec)

mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 172.17.0.3
                  Master_User: root
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 154
               Relay_Log_File: relaylog.000002
                Relay_Log_Pos: 320
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 154
              Relay_Log_Space: 520
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 2
                  Master_UUID: 7303adb5-4117-11e9-be41-0242ac110003
             Master_Info_File: /var/lib/mysql/master.info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
```

```
           Master_Retry_Count: 86400
                  Master_Bind:
      Last_IO_Error_Timestamp:
     Last_SQL_Error_Timestamp:
               Master_SSL_Crl:
           Master_SSL_Crlpath:
           Retrieved_Gtid_Set:
            Executed_Gtid_Set:
                Auto_Position: 0
         Replicate_Rewrite_DB:
                 Channel_Name:
           Master_TLS_Version:
1 row in set (0.00 sec)

ERROR:
No query specified

mysql>
```

3 数据库同步

> 备份数据前先锁表，保证数据一致性

```
mysql> FLUSH TABLES WITH READ LOCK;
# mysqldump -uroot -proot test> /tmp/test.sql;
mysql> UNLOCK TABLES;
```

3.1 master上创建数据库

```
docker exec -it mysql-master bash

mysql -uroot -proot

mysql> CREATE DATABASE `test` CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.01 sec)

mysql> USE test;
Database changed
mysql> SET FOREIGN_KEY_CHECKS=0;
Query OK, 0 rows affected (0.00 sec)

mysql> DROP TABLE IF EXISTS `info`;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> CREATE TABLE `info` (
    ->   `id` int(11) NOT NULL AUTO_INCREMENT,
    ->   `name` varchar(255) NOT NULL,
    ->   `email` varchar(255) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.11 sec)

#当数据表中有自增长主键时，当用SQL插入语句中插入语句带有id列值记录的时候,可以把id的值设置为null或者0，这样子mysql都会自己做处理

mysql> INSERT INTO `info` VALUES ('0', 'xiongjun', 'xiongjun@cpms.com.cn');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO `info` VALUES ('0', 'liuyanwen', 'liuyanwen@cpms.com.cn');
Query OK, 1 row affected (0.02 sec)

#方法2: insert into info(name,email) values('zabbix','zabbix@cpms.com.cn');

mysql>
```

3.2 slave上查询数据库

```
docker exec -it mysql-slave bash

mysql -uroot -proot

mysql> select * from test.info;
+----+-----------+-----------------------+
| id | name      | email                 |
+----+-----------+-----------------------+
```

```
|  1 | xiongjun  | xiongjun@cpms.com.cn  |
|  2 | liuyanwen | liuyanwen@cpms.com.cn |
+----+-----------+-----------------------+
2 rows in set (0.00 sec)

mysql>
```

3.3 slave上插入新数据

```
docker exec -it mysql-slave bash

mysql -uroot -proot

mysql> USE test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO `info` VALUES ('0', 'liuxiaolei', 'liuxiaolei@cpms.com.cn');
Query OK, 1 row affected (0.02 sec)

mysql>
```

3.4 master上查询数据库

```
docker exec -it mysql-master bash

mysql -uroot -proot

mysql> select * from test.info;
+----+------------+------------------------+
| id | name       | email                  |
+----+------------+------------------------+
|  1 | xiongjun   | xiongjun@cpms.com.cn   |
|  2 | liuyanwen  | liuyanwen@cpms.com.cn  |
|  3 | liuxiaolei | liuxiaolei@cpms.com.cn |
+----+------------+------------------------+
3 rows in set (0.00 sec)

mysql>
```

数据库双主配置同步成功!