1. 版本：Mysql:5.6

2. 两台机器（以下服务器地址、用户均为举例，相同例子处保持一致）：

master1:192.168.1.2

master2:192.168.1.100

3. 单独编译安装MySQL，并测试连接成功。

4. 两台机器上都建立用户，并赋予复制权限：

master1：

1. mysql> CREATE USER 'logistics'@'%';
2. mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'logistics'@'%' IDENTIFIED BY 'password';

master2：

1. mysql> CREATE USER 'logistics'@'%;
2. mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'logistics'@'%' IDENTIFIED BY 'password';

host是随便设置的，只要是对方可以连接上即可。权限方面，只需要有REPLICATION SLAVE权限即可。

5. master1上的my.cnf添加：

1. [mysqld]
2. # for repl
3. server-id = 1
4. log-bin = mysql-bin
5. auto-increment-increment= 2 # 应设为整个结构中服务器的总数
6. auto-increment-offset = 1 # 设定数据库中自动增长的起点，避免两台服务器数据同步时出现主键冲突

6. master2上的my.cnf添加：

1. [mysqld]
2. # for repl
3. server-id = 2
4. log-bin = mysql-bin
5. auto-increment-increment= 2
6. auto-increment-offset = 1

7. 重启两台机器上的服务。

8. master1上指定master2为主：

1. mysql> CHANGE MASTER TO
2. -> MASTER\_HOST='192.168.1.100',
3. -> MASTER\_USER='logistics',
4. -> MASTER\_PASSWORD='password';
5. Query OK, 0 rows affected (0.19 sec)
6. mysql> START SLAVE;
7. Query OK, 0 rows affected (0.00 sec)

9. master2上指定master1为主：

1. mysql> CHANGE MASTER TO
2. -> MASTER\_HOST='192.168.1.2',
3. -> MASTER\_USER='logistics',
4. -> MASTER\_PASSWORD='password';
5. Query OK, 0 rows affected (0.18 sec)
6. mysql> START SLAVE;
7. Query OK, 0 rows affected (0.00 sec)

使用命令：show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.1.100

Master\_User: logistics

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000006

Read\_Master\_Log\_Pos: 235

Relay\_Log\_File: bussiness-relay-bin.000007

Relay\_Log\_Pos: 398

Relay\_Master\_Log\_File: mysql-bin.000006

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: 235

Relay\_Log\_Space: 738

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: 667891fd-4233-11e4-b7e0-e0db5504d8b8

Master\_Info\_File: /home/data/mysql/master.info

SQL\_Delay: 0

SQL\_Remaining\_Delay: NULL

Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it

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

10. 测试

实验1：启动两个服务，在任一mysql上进行操作，在另一mysql上均能体现出来。

实验2：启动两个服务，在master1上通过tpcc工具添加20个warehouse，数据占用硬盘约1.9G，在两台机器上计算count(\*)，数据一致。

实验3：启动master1，在master1上通过tpcc工具添加一个warehouse，启动master2，同步一段时间后，计算count(\*)，结果一致。

ps，MySQL的主主复制实际上就是双向的主从复制。