主从复制主要是my.cnf文件配置。主从节点都要配置。

**主节点master的my.cnf文件在[mysqld] 下面增加如下配置。**

## 设置 server\_id，一般设置为 IP

server\_id=231

## 复制过滤：需要备份的数据库，输出 binlog

#binlog-do-db=data-test

## 复制过滤：不需要备份的数据库，不输出（mysql 库一般不同步）

binlog-ignore-db=mysql

## 开启二进制日志功能，可以随便取，最好有含义

log-bin=mysql-bin-firsttest

## 为每个 session 分配的内存，在事务过程中用来存储二进制日志的缓存

binlog\_cache\_size=1M

## 主从复制的格式（mixed,statement,row，默认格式是 statement）

binlog\_format=mixed

## 二进制日志自动删除/过期的天数。默认值为 0，表示不自动删除。

expire\_logs\_days=7

## 跳过主从复制中遇到的所有错误或指定类型的错误，避免 slave 端复制中断。

## 如：1062 错误是指一些主键重复，1032 错误是因为主从数据库数据不一致

slave\_skip\_errors=1062

**从节点slave的my.cnf文件在[mysqld] 下面增加如下配置。**

## 设置 server\_id，一般设置为 IP

server\_id=116

## 复制过滤：需要备份的数据库，输出 binlog

#binlog-do-db=data-test

##复制过滤：不需要备份的数据库，不输出（mysql 库一般不同步）

binlog-ignore-db=mysql

## 开启二进制日志，以备 此mysql服务作为其它 Slave 的 Master 时使用

log-bin=mysql-bin-own

## 为每个 session 分配的内存，在事务过程中用来存储二进制日志的缓存

binlog\_cache\_size = 1M

## 主从复制的格式（mixed,statement,row，默认格式是 statement）

binlog\_format=mixed

## 二进制日志自动删除/过期的天数。默认值为 0，表示不自动删除。

expire\_logs\_days=7

## 跳过主从复制中遇到的所有错误或指定类型的错误，避免 slave 端复制中断。

## 如：1062 错误是指一些主键重复，1032 错误是因为主从数据库数据不一致

slave\_skip\_errors=1062

## relay\_log 配置中继日志

relay\_log=edu-mysql-relay-bin

## log\_slave\_updates 表示 slave 将复制事件写进自己的二进制日志

log\_slave\_updates=1

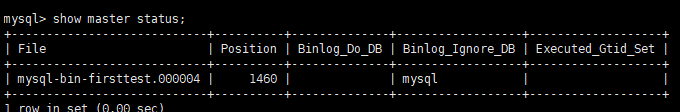
## 防止改变数据(除了特殊的线程)

read\_only=1

一些有用的命令

主节点的my.cnf配置好后 重启服务。然后可以查看master状态。

show master status;



当然可以新增一个数据同步的账号：

grant replication slave, replication client on \*.\* to 'repl'@'%' with grant option;

从节点my.cnf配置好后，重启服务。

**设置主节点参数：**

change master to master\_host='192.168.31.231', master\_user='repl',

master\_password='123456', master\_port=3306, master\_log\_file='mysql-bin-firsttest.000004',

master\_log\_pos=1460, master\_connect\_retry=30;

## 查看主从同步状态

mysql> show slave status\G;

可看到 Slave\_IO\_State 为空， Slave\_IO\_Running 和 Slave\_SQL\_Running 是 No，表明 Slave 还没有开始复制过程。

## 开启主从同步

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

## 停止主从同步

mysql> stop slave;

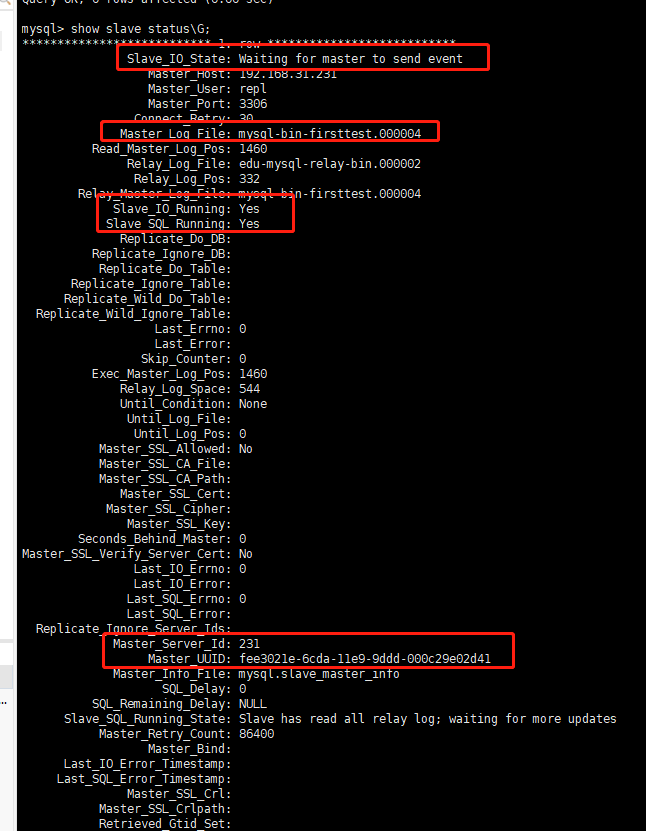
Query OK, 0 rows affected (0.00 sec)

## 再查看主从同步状态

mysql> show slave status\G;

可查看 master 和 slave 上线程的状态。在 master 上，可以看到 slave 的 I/O 线程创建的连接：

mysql> show processlist\G;



Slave是mysql数据同步是个进程，可以启动和停止，重新设置同步，master是只要mysql开启二进制日志就是。一个mysql可以同时是master和slave

主从同步出错，不能再自动同步，需要可在 Slave 上重置主从复制设置（选操作）：

(1) mysql> reset slave;

(2) mysql> change master to master\_host='192.168.1.205',

master\_user='repl',

master\_password='roncoo.123',

master\_port=3306,

master\_log\_file='edu-mysql-bin.00000x',

master\_log\_pos=xx,

master\_connect\_retry=30;

(此时，master\_log\_file 和 master\_log\_pos 要在 Master 中用 show master status 命令查看)

上面所搭建的是单向复制的主从，也是用的比较多的，而双向主从其实就是 Master 和 Slave 都开启日志功能，然后在 Master 执行授权用户（这里授权的是自己作为从服务器，也就是这里的 IP 地址是 Master 的 IP 地址），然后再在 Master 上进行 chang master 操作。

核心：数据库读写分离，就是靠用户权限和read\_only状态来实现的。

主从切换的主要是通过mysql的命令 change master to来实现的

######################################

对于数据库读写状态，主要靠 “read\_only”全局参数来设定；默认情况下，数据库是用于读写操作的，所以read\_only参数也是0或faluse状态，这时候不论是本地用户还是远程访问数据库的用户，都可以进行读写操作；如需设置为只读状态，将该read\_only参数设置为1或TRUE状态，但设置 read\_only=1 状态有两个需要注意的地方：

      1.read\_only=1只读模式，不会影响slave同步复制的功能，所以在MySQL slave库中设定了read\_only=1后，通过 show slave status\G 命令查看salve状态，可以看到salve仍然会读取master上的日志，并且在slave库中应用日志，保证主从数据库同步一致；

      2.read\_only=1只读模式，可以限定普通用户进行数据修改的操作，但不会限定具有super权限的用户的数据修改操作；在MySQL中设置read\_only=1后，普通的应用用户进行insert、update、delete等会产生数据变化的DML操作时，都会报出数据库处于只读模式不能发生数据变化的错误，但具有super权限的用户，例如在本地或远程通过root用户登录到数据库，还是可以进行数据变化的DML操作；

-------锁表---

      为了确保所有用户，包括具有super权限的用户也不能进行读写操作，就需要执行给所有的表加读锁的命令 “flush tables with read lock;”，这样使用具有super权限的用户登录数据库，想要发生数据变化的操作时，也会提示表被锁定不能修改的报错。

        这样通过 设置“read\_only=1”和“flush tables with read lock;”两条命令，就可以确保数据库处于只读模式，不会发生任何数据改变，在MySQL进行数据库迁移时，限定master主库不能有任何数据变化，就可以通过这种方式来设定。

       但同时由于加表锁的命令对数据库表限定非常严格，如果再slave从库上执行这个命令后，slave库可以从master读取binlog日志，但不能够应用日志，slave库不能发生数据改变，当然也不能够实现主从同步了，这时如果使用 “unlock tables;”解除全局的表读锁，slave就会应用从master读取到的binlog日志，继续保证主从库数据库一致同步。

       为了保证主从同步可以一直进行，在slave库上要保证具有super权限的root等用户只能在本地登录，不会发生数据变化，其他远程连接的应用用户只按需分配为select,insert,update,delete等权限，保证没有super权限，则只需要将salve设定“read\_only=1”模式，即可保证主从同步，又可以实现从库只读。

       相对的，设定“read\_only=1”只读模式开启的解锁命令为设定“read\_only=0”;设定全局锁“flush tables with read lock;”，对应的解锁模式命令为：“unlock tables;”.

      当然设定了read\_only=1后，所有的select查询操作都是可以正常进行的。