

MySQL主從複製配置詳解

原創 平凡、世界 SQL數據庫開發 每日

點擊關注“ [SQL數據庫開發](#)”，
設為“置頂或星標”，第一時間文件乾貨

SQL專欄

SQL基礎知識第二版SQL基礎
知識第二版

之後很多小伙伴想知道 MySQL 主從復制的配置步驟，今天它來了。

配置環境

操作系統：兩台CentOS 7.6的Linux系統

數據庫版本：MySQL 5.6.39

主服務器IP：192.168.0.1

從服務器IP：192.168.0.2

安裝數據庫

之前已經給小伙伴們詳細的講解了CentOS安裝MySQL的操作步驟了，沒有的小伙伴可以戳這裡：

《Linux環境下安裝MySQL步驟詳解》

配置前提

- 1、需要保證3306端口開啟或關閉防火牆，在MySQL的安裝裡有介紹。
- 2、兩台服務器之間可以互相ping通

```
--在192.168.0.2上輸入ping命令  
ping 192.168.0.1  
--在192.168.0.1上輸入ping命令  
ping 192.168.0.2
```

- 3、安裝成功一個MySQL後，使用虛擬機統一作為從服務器

配置主（碩士）數據庫

1、修改數據庫配置文件

```
[root@localhost ~]# vi /etc/my.cnf
```

將裡面的內容修改為

```
[mysqld]  
#开启二进制日志  
log-bin=mysql-bin  
#标识唯一id（必须），一般使用ip最后位
```

```
server-id=1
#不同步的数据库，可设置多个
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
binlog-ignore-db=mysql
#指定需要同步的数据库（和slave是相互匹配的），可以设置多个
binlog-do-db=test
```

添加存儲方式和規則（選填）

```
#设置存储模式不设置默认
binlog_format=MIXED
#日志清理时间
expire_logs_days=7
#日志大小
max_binlog_size=100m
#缓存大小
binlog_cache_size=4m
#最大缓存大小
max_binlog_cache_size=521m
```

注：日誌的存儲容量我設置的都比較小，當然你可以根據實際情況修改得大一點。

2、重启数据库服务mysqld

```
service mysqld restart
```

如果你按照上面的正确安装mysql了，这里是是可以正常重启的。如果启动不正常出现如下报错：

```
The server quit without updating PID file.....
```

你需要使用如下命令查看是否还存在mysqld进程

```
ps -ef|grep mysqld
```

如果有，可以使用命令：**kill -9 mysqld的进程号** 结束它，然后重新启动mysqld

我就遇到过上述的情况。当然也有其他原因，这里贴一个其他可能原因的解决办法供参考：

<https://javawind.net/p141>

3、登陆MySQL数据库允许从库获得主库日志

```
[root@localhost ~]# mysql -u root -p
```

注：第一次登陆是不需要输入root的密码的。

进入后做如下配置：

```
#给从库放权限
mysql>GRANT FILE ON *.* TO 'root'@'192.168.0.2' IDENTIFIED BY 'root password'; #创建用户
mysql>GRANT REPLICATION SLAVE ON *.* TO 'root'@'192.168.0.2' IDENTIFIED BY 'root password'; #修改用户权限
mysql>select host ,user ,password from mysql.user; #查看是否修改成功
mysql>FLUSH PRIVILEGES; #刷新权限
```

4、重启MySQL服务，登录MySQL，查看主库信息

```
[root@localhost ~]# service mysqld restart #重启mysql服务
[root@localhost ~]# mysql -u root -p #登陆mysql
mysql> show master status; #查看master状态
```

显示大概如下内容

```
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000006 | 120 | ufind_db | information_schema,performance_schema,mysql | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

注：如果执行这个步骤始终为`Empty set(0.00 sec)`，那说明前面的`my.cnf`没配置对，请回去重新检查配置步骤。

配置从(Slave)数据库

1、修改从库的数据库配置文件

```
[root@localhost ~]# vi /etc/my.cnf
```

将里面的内容修改为

```
#开启二进制日志
log-bin=mysql-bin
server-id=2
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
binlog-ignore-db=mysql
#与主库配置保持一致
replicate-do-db=test
replicate-ignore-db=mysql
log-slave-updates
slave-skip-errors=all
slave-net-timeout=60
```

2、重启MySQL服务，登录MySQL

```
[root@localhost ~]# service mysqld restart

[root@localhost ~]# mysql -u root -p
```

并作如下修改：

```
#关闭Slave
mysql> stop slave; #设置连接主库信息
mysql> change master to master_host='192.168.0.1',master_user='root',master_password='root password',master_log_file=
#开启Slave
mysql> start slave;
```

注：上面的master_log_file是在配置Master的时候的File字段， master_log_pos是在配置Master的Position 字段。一定要一一对应

3、查看从库状态信息

```
mysql> show slave status \G;
```

成功的话会显示如下信息：

```
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 192.168.0.1
        Master_User: root
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000006
        Read_Master_Log_Pos: 120
        Relay_Log_File: localhost-relay-bin.000006
        Relay_Log_Pos: 520
        Relay_Master_Log_File: mysql-bin.000006
        Slave_IO_Running: Yes //显示yes为成功
        Slave_SQL_Running: Yes //显示yes为成功，如果为no，一般为没有启动master
        Replicate_Do_DB: test
        Replicate_Ignore_DB: mysql//上面的都是配置文件中的信息
        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: 357
    Relay_Log_Space: 697
    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: //如果为no，此处会显示错误信息
    Last_SQL_Errno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:
    Master_Server_Id: 2
        Master_UUID: be0a41c0-2b40-11e8-b791-000c29267b6a
    Master_Info_File: /usr/local/mysql/data/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
1 row in set (0.00 sec)

ERROR:
No query specified
```

注：如果`Slave_IO_Running: No`并且出现下面的错误

Last_IO_Error: Fatal error: The slave I/O thread stops because master and slave have equal MySQL server UUIDs; these UUIDs must be different for replication to work.

说明主服务器的UUID和从服务器的UUID重复，因为我是安装成功一台数据库后直接克隆的，所以他们的UUID是一样的，就会报这个错。可以修改一下从库的UUID即可。

我们先在从库的数据库中生成一个UUID

```
mysql>select UUID();
```

将数据库中查询出来的这个UUID复制出来，然后编辑从库的UUID配置文件

如果你也安装的跟我一样，那么这个配置文件的路径就应该在这里：

```
[root@localhost ~]# vi /usr/local/mysql/data/auto.cnf
```

进去后，将一串32位长的UUID，替换成我们刚在数据库中查询生成的UUID即可。

如果*Slave_IO_Running: No* 并出现下面错误

```
Got fatal error 1236 from master when reading data from binary log: 'Could not find first log file name in binary log index file'
```

解决方法：复位

```
mysql>stop slave; //停止  
mysql>reset slave; //复位  
mysql>start slave; //开启
```

至此整个过程就配置好了。

可能有小伙伴会问，这些配置文件我都配好了，信息也和你的一样，我还是不确定是否配置成功。

那么你可以在主服务器上创建一个表，然后在从服务器上查询刚创建的这个表，看是否存在就可以啦。

Tips

1、关于增删改查，主从数据不一致问题：

```
#select 语句，暂时没有发现问题

#insert 语句，暂时没有发现问题

#update 语句，暂时没有发现问题

#delete 语句，主库删除多条数据，发现数据不一致
```

原因：在主库的logbin中的确有执行删除语句，但是在从库的logbin中却没有删除语句

解决：使用 use database 选取当前数据库架构中的需要操作的数据库，然后在执行删除，OK同步成功

2、查询binlog主从日志的方法

```
#查看binlog全部文件
mysql>show binary logs;

#查看binlog是否开启NO为开启
mysql> show variables like 'log_bin%';

#详细信息
mysql> show variables like 'binlog%';

#查看binlog日志
mysql> show binlog events in 'mysql-bin.000019';

#或者使用mysqlbinlog，如果报错使用--no-defaults（使用全路径）
[root@localhost ~]# /usr/local/mysql/bin/mysqlbinlog --no-defaults /usr/local/mysql/data/mysql-bin.000019
```

3、手动清理master日志，最好关闭日志，在/etc/my.cnf

```
#手动刷新日志
mysql> show master status;
#删除全部
mysql> reset slave;或 rest master;
#删除MySQL-bin.004
mysql> PURGE MASTER LOGS TO 'MySQL-bin.004';
```

此外，如果你在修改最大连接数时，可能会存在已经将mysql配置文件的连接数改成1000或更大，但是查询数据库的最大连接数始终都是214，可以尝试如下方法：

<https://www.cnblogs.com/brucetang/p/9733998.html>

参考

- <https://javawind.net/p141>
- <https://www.cnblogs.com/brucetang/p/9733998.html>



最后给大家分享我写的SQL两件套：《SQL基础知识第二版》和《SQL高级知识第二版》的PDF电子版。里面有各个语法的解释、大量的实例讲解和批注等等，非常通俗易懂，方便大家跟着一起来实操。

有需要的可以下载学习，只需要在下面的公众号「**数据前线**」（非本号），后台回复关键字：**SQL**，就行

数据前线



后台回复关键字：**1024**，获取一份精心整理的技术干货

后台回复关键字：**进群**，带你进入高手如云的交流群。

推荐阅读

- 没想到《天龙八部》这段，只有搞IT的才看得懂
- 曾经排名第一的安全软件，为啥会变成流氓软件？
- 北大数学系「扫地僧」韦东奕爆红！
- 中国月入过万的，究竟有多少人？
- 大公司病，太形象了
- 算机的女生都怎么样了？

喜欢此内容的人还喜欢

手把手教你在 Linux 环境下部署 Oracle 数据库!

杰哥的IT之旅



Nginx 常用配置清單

Linux就該這麼學



Nginx 常用配置清單

蟒蛇那些事

