



## MySQL 主从同步配置

### 1. 主从同步的定义

主从同步使得数据可以从一个数据库服务器复制到其他服务器上，在复制数据时，一个服务器充当主服务器（master），其余的服务器充当从服务器（slave）。因为复制是异步进行的，所以从服务器不需要一直连接着主服务器，从服务器甚至可以通过拨号断断续续地连接主服务器。通过配置文件，可以指定复制所有的数据库，某个数据库，甚至是某个数据库上的某个表。

使用主从同步的好处：

- (1) 通过增加从服务器来**提高数据库的性能**，在主服务器上执行写入和更新，在从服务器上向外提供读功能，可以动态地调整从服务器的数量，从而调整整个数据库的性能。
- (2) **提高数据安全**，因为数据已复制到从服务器，从服务器可以终止复制进程，所以，可以在从服务器上备份而不破坏主服务器相应数据
- (3) 在主服务器上生成实时数据，而在从服务器上分析这些数据，从而**提高主服务器的性能**

### 2. 主从同步的机制

Mysql 服务器之间的主从同步是基于二进制日志机制，主服务器使用二进制日志来记录数据库的变动情况，从服务器通过读取和执行该日志文件来保持和主服务器的数据一致。

在使用二进制日志时，主服务器的所有操作都会被记录下来，然后从服务器会接收到该日志的一个副本。从服务器可以指定执行该日志中的哪一类事件（譬如只插入数据或者只更新数据），默认会执行日志中的所有语句。

每一个从服务器会记录关于二进制日志的信息：文件名和已经处理过的语句，这样意味着不同的从服务器可以分别执行同一个二进制日志的不同部分，并且从服务器可以随时连接或者中断和服务器的连接。

主服务器和每一个从服务器都必须配置一个唯一的 ID 号（在 my.cnf 文件的 [mysqld] 模块下有一个 server-id 配置项），另外，每一个从服务器还需要通过 CHANGE MASTER TO 语句来配置它要连接的主服务器的 ip 地址，日志文件名称和



该日志里面的位置（这些信息存储在主服务器的数据库里）

### 3. 配置主从同步的基本步骤

有很多种配置主从同步的方法，可以总结为如下的步骤：

- (1) 在主服务器上，必须开启二进制日志机制和配置一个独立的 ID
- (2) 在每一个从服务器上，配置一个唯一的 ID，创建一个用来专门复制主服务器数据的账号
- (3) 在开始复制进程前，在主服务器上记录二进制文件的位置信息
- (4) 如果在开始复制之前，数据库中已经有数据，就必须先创建一个数据快照（可以使用 mysqldump 导出数据库，或者直接复制数据文件）
- (5) 配置从服务器要连接的主服务器的 IP 地址和登陆授权，二进制日志文件名和位置

### 4. 详细配置主从同步的方法

主和从的身份可以自己指定，我们将 Ubuntu 虚拟机中 MySQL 作为主服务器，将 Windows 中的 MySQL 作为从服务器。

在主从设置前，要保证 Ubuntu 与 Windows 间的网络连通。

#### 4.1 备份主服务器原有数据到从服务器

如果在设置主从同步前，主服务器上已有大量数据，可以使用 mysqldump 进行数据备份并还原到从服务器以实现数据的复制。

1. 在主服务器 Ubuntu 上进行备份，执行命令：

```
mysqldump -uroot -pmysql --all-databases --lock-all-tables > ~/master_db.sql
```

```
python@ubuntu:~$ mysqldump -uroot -pmysql --all-databases --lock-all-tables > ~/.master_db.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
python@ubuntu:~$ ls
dbs      Documents  dump.rdb   master_db.sql  Pictures  Templates  workspace
Desktop  Downloads  examples.desktop  Music      Public    Videos
```

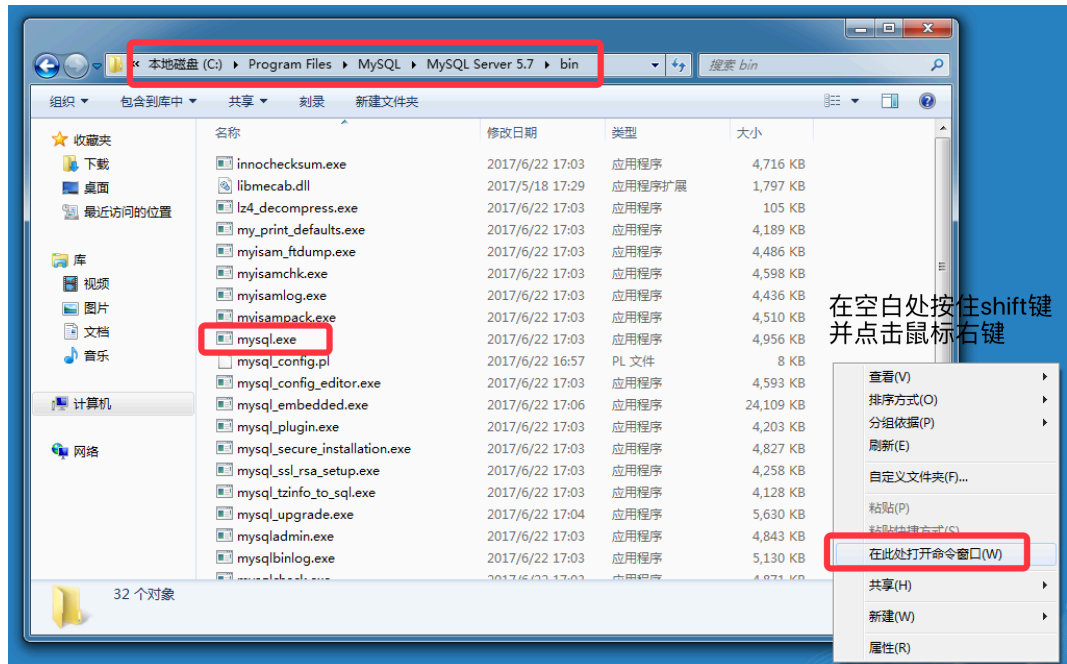
-u : 用户名

-p : 示密码

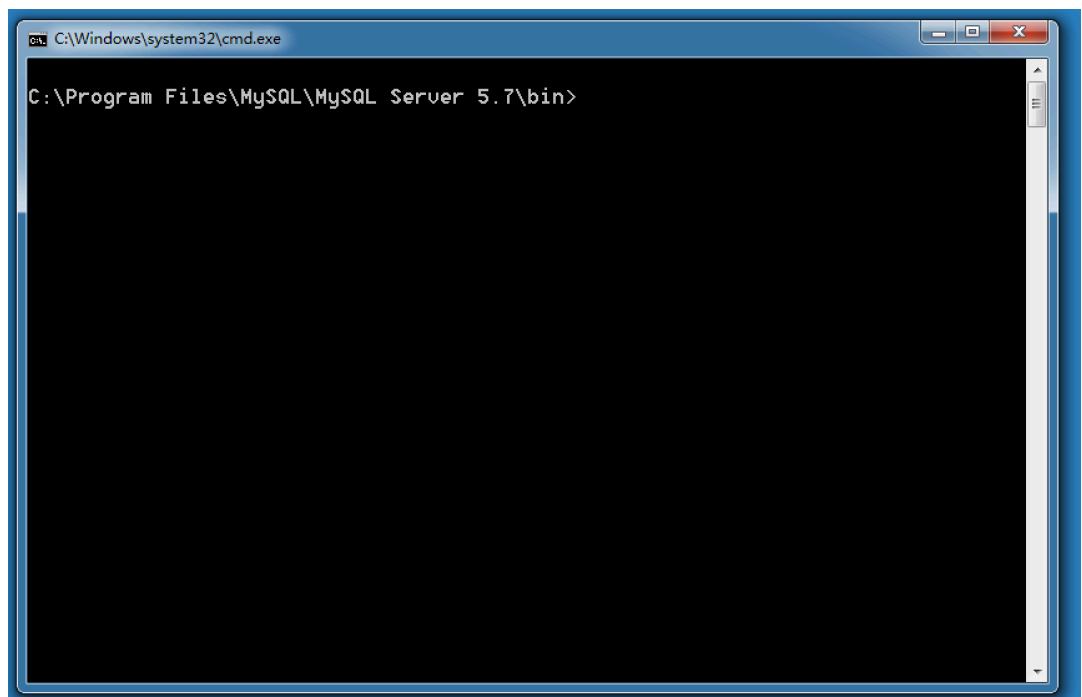
--all-databases : 导出所有数据库  
--lock-all-tables : 执行操作时锁住所有表，防止操作时有数据修改  
~/master\_db.sql : 导出的备份数据 (sql 文件) 位置，可自己指定

## 2. 在从服务器 Windows 上进行数据还原

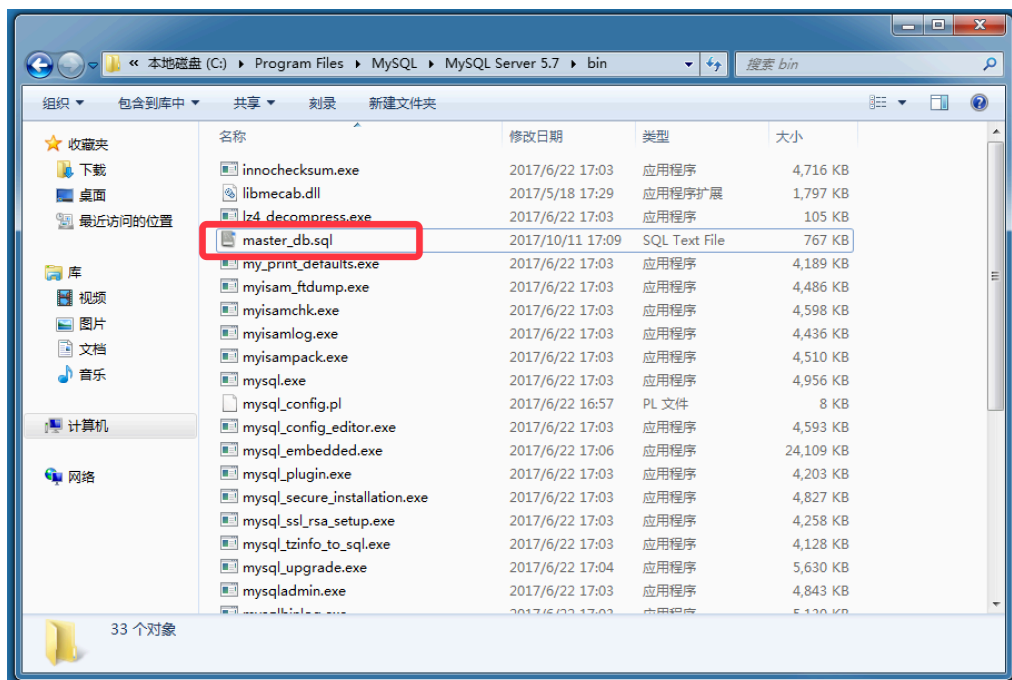
找到 Windows 上 mysql 命令的位置



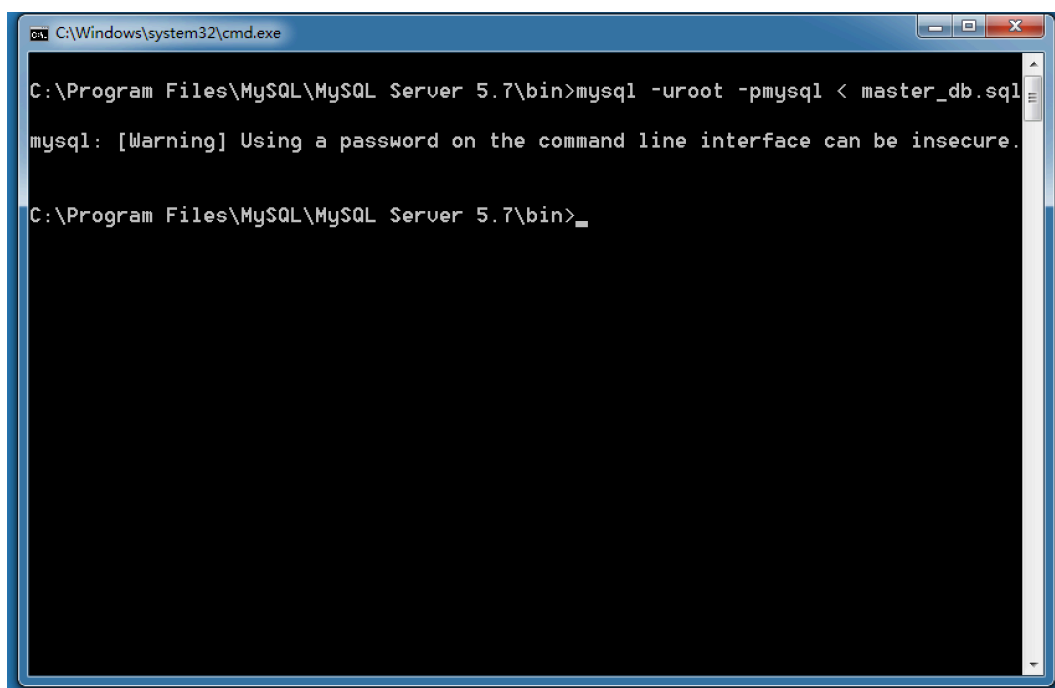
新打开的命令窗口，在这个窗口中可以执行类似在 Ubuntu 终端中执行的 mysql 命令。



将从主服务器 Ubuntu 中导出的文件复制到从服务器 Windows 中，可以将其放在上面 mysql 命令所在的文件夹中，方便还原使用。



在刚打开的命令黑窗口中执行还原操作：  
`mysql -uroot -pmysql < master_db.sql`





## 4.2 配置主服务器 master (Ubuntu 中的 MySQL)

1. 编辑设置 mysqld 的配置文件，设置 log\_bin 和 server-id

```
sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
python@ubuntu:~$ sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
python@ubuntu:~$ sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
71 # Error log - should be very few entries.
72 #
73 log_error = /var/log/mysql/error.log
74 #
75 # Here you can see queries with especially long duration
76 #log_slow_queries = /var/log/mysql/mysql-slow.log
77 #long_query_time = 2
78 #log-queries-not-using-indexes
79 #
80 # The following can be used as easy to replay backup logs or for replication.
81 # note: if you are setting up a replication slave, see README.Debian about
82 #      other settings you may need to change.
83 #server-id      = 1
84 log_bin        = /var/log/mysql/mysql-bin.log
85 expire_logs_days = 10
86 max_binlog_size = 100M
87 #binlog_do_db   = include_database_name
88 #binlog_ignore_db = include_database_name
89 #
90 # * InnoDB
91 #
92 # InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
93 # Read the manual for more InnoDB related options. There are many!
94 #
95 # * Security Features
96 #
97 # Read the manual, too, if you want chroot!
98 # chroot = /var/lib/mysql/
```

2. 重启 mysql 服务

```
sudo service mysql restart
```

```
python@ubuntu:~$ sudo service mysql restart
python@ubuntu:~$
```

3. 登入主服务器 Ubuntu 中的 mysql，创建用于从服务器同步数据使用的帐号

```
mysql -uroot -pmysql
```

```
GRANT REPLICATION SLAVE ON *.* TO 'slave'@'%' identified by 'slave';
```

```
FLUSH PRIVILEGES;
```



```
python@ubuntu: ~  
python@ubuntu:~$ mysql -uroot -pmysql  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 4  
Server version: 5.7.19-0ubuntu0.16.04.1-log (Ubuntu)  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> GRANT REPLICATION SLAVE ON *.* TO 'slave'@'%' identified by 'slave';  
Query OK, 0 rows affected, 1 warning (0.02 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql>
```

#### 4. 获取主服务器的二进制日志信息

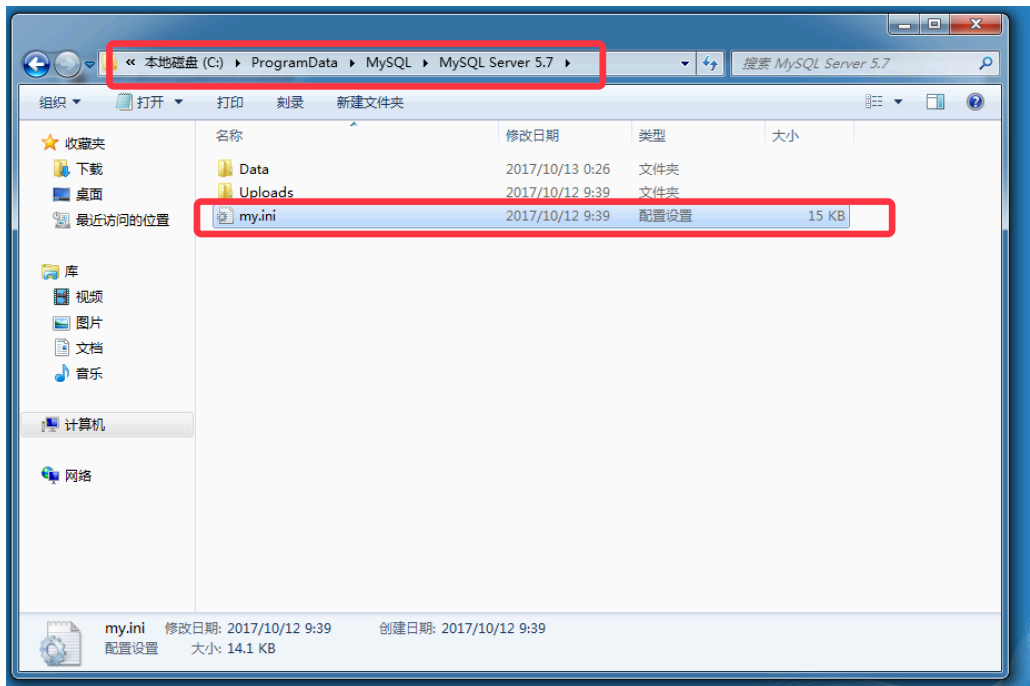
SHOW MASTER STATUS;

```
mysql> SHOW MASTER STATUS;  
+-----+-----+-----+-----+-----+  
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |  
+-----+-----+-----+-----+-----+  
| mysql-bin.000006 |      590 |              |                  |                  |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

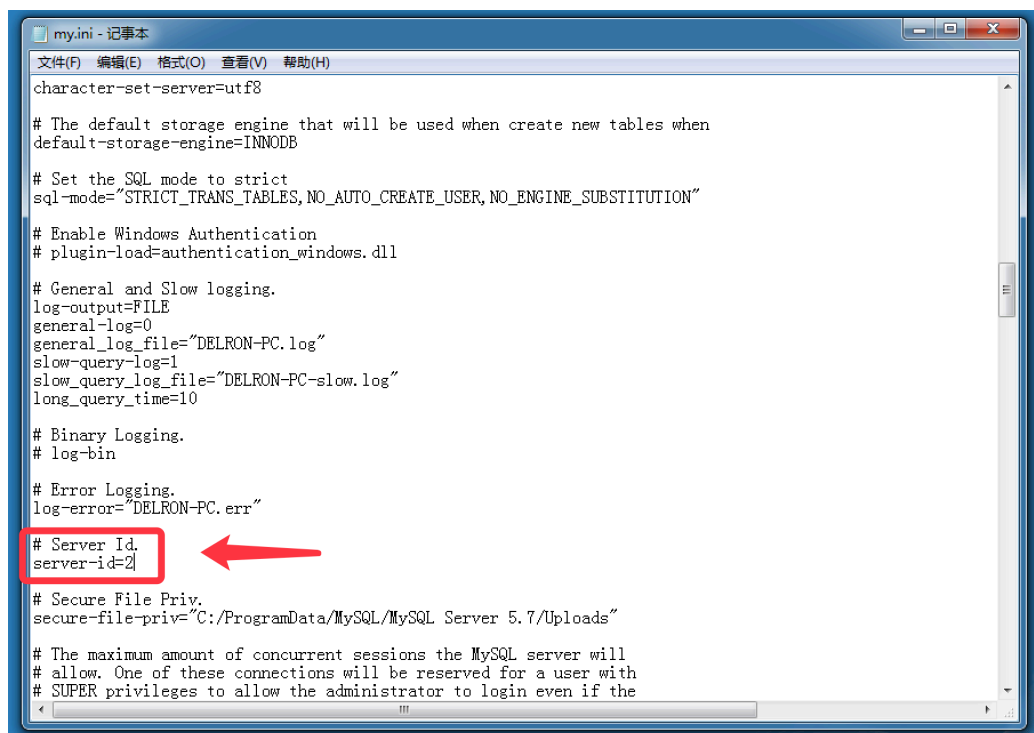
File 为使用的日志文件名字，Position 为使用的文件位置，这两个参数须记下，配置从服务器时会用到。

### 4.3 配置从服务器 slave (Windows 中的 MySQL)

#### 1. 找到 Windows 中 MySQL 的配置文件

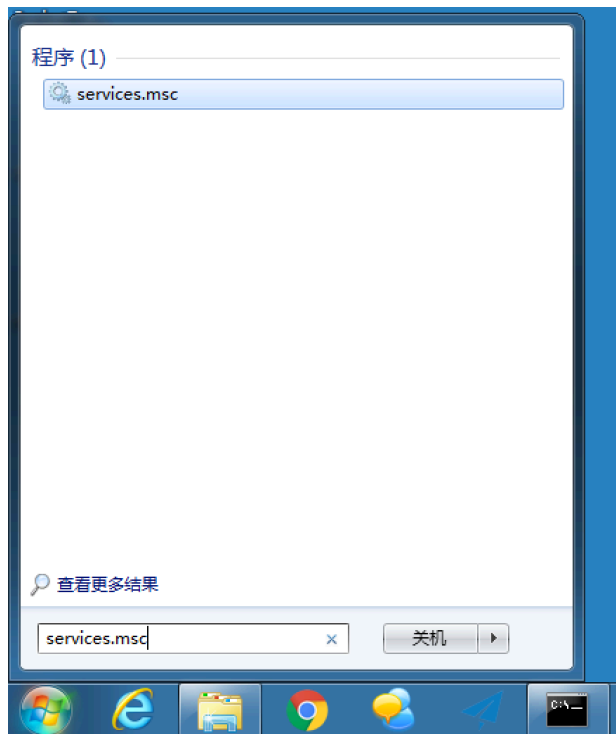


2. 编辑 my.ini 文件，将 server-id 修改为 2，并保存退出。

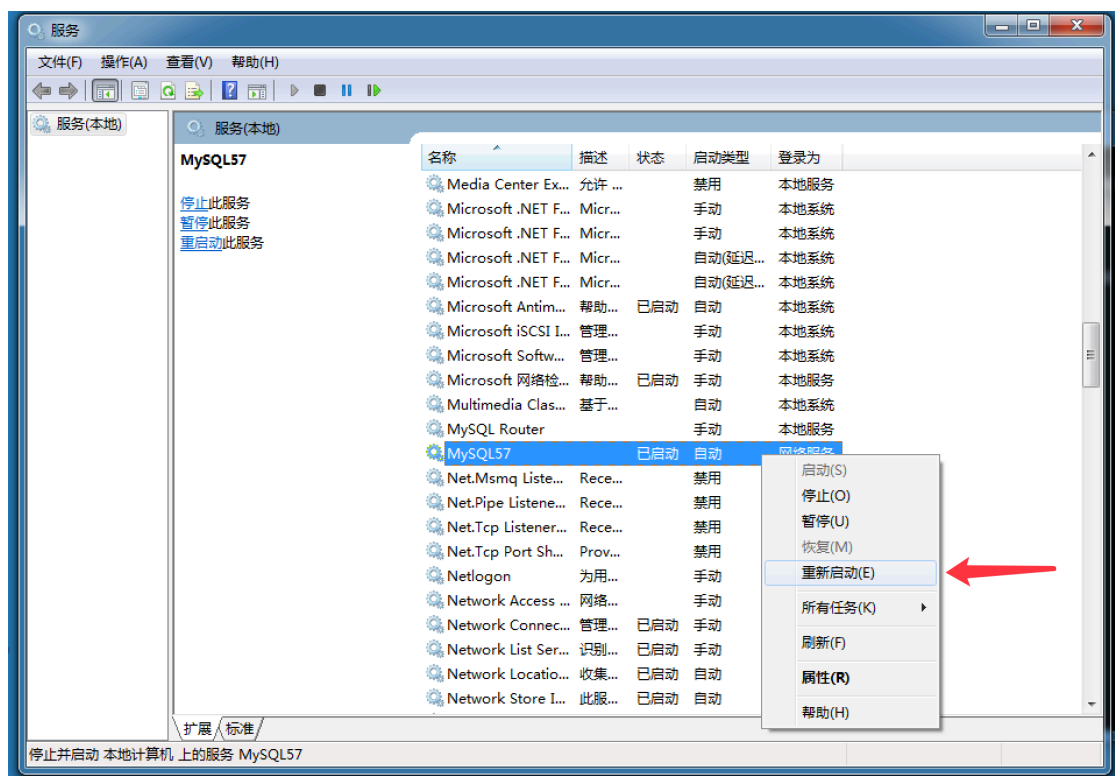


3. 打开 windows 服务管理

可以在开始菜单中输入 services.msc 找到并运行。



4. 在打开的服务管理中找到 MySQL57，并重启该服务







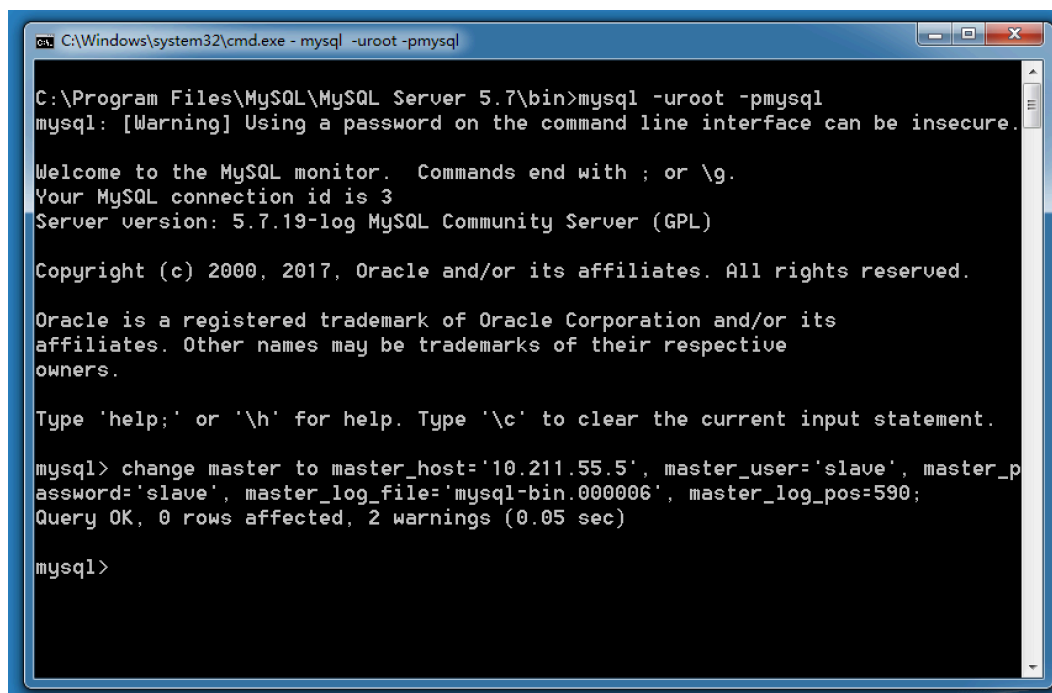
## 5. 进入 windows 的 mysql，设置连接到 master 主服务器

```
Change master to master_host='10.211.55.5', master_user='slave',  
master_password='slave',master_log_file='mysql-bin.000006',  
master_log_pos=590;
```

注: master\_host: 主服务器 Ubuntu 的 ip 地址

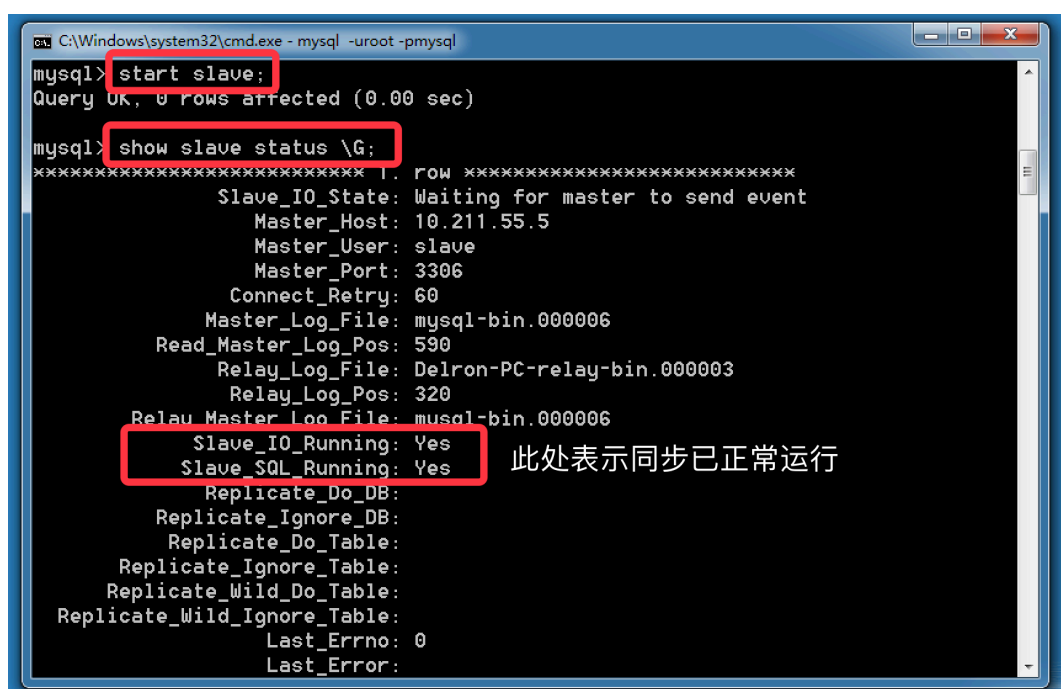
master\_log\_file: 前面查询到的主服务器日志文件名

master\_log\_pos: 前面查询到的主服务器日志文件位置



```
C:\Windows\system32\cmd.exe - mysql -uroot -pmysql  
  
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -uroot -pmysql  
mysql: [Warning] Using a password on the command line interface can be insecure.  
  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 3  
Server version: 5.7.19-log MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> change master to master_host='10.211.55.5', master_user='slave', master_p  
assword='slave', master_log_file='mysql-bin.000006', master_log_pos=590;  
Query OK, 0 rows affected, 2 warnings (0.05 sec)  
  
mysql>
```

## 6. 开启同步，查看同步状态



```
C:\Windows\system32\cmd.exe - mysql -uroot -pmysql  
  
mysql> start slave;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> show slave status \G;  
+-----+  
Slave_IO_State: Waiting for master to send event  
Master_Host: 10.211.55.5  
Master_User: slave  
Master_Port: 3306  
Connect_Retry: 60  
Master_Log_File: mysql-bin.000006  
Read_Master_Log_Pos: 590  
Relay_Log_File: Delron-PC-relay-bin.000003  
Relay_Log_Pos: 320  
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:
```

此处表示同步已正常运行



## 4.4 测试主从同步

在 Ubuntu 的 MySQL 中（主服务器）创建一个数据库

```
mysql> create database dailyfresh default charset=utf8;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dailyfresh |
| ihome |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)
```

在 Windows 的 MySQL 中（从服务器）查看新建的数据库是否存在。

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dailyfresh |
| ihome |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
8 rows in set (0.00 sec)
```