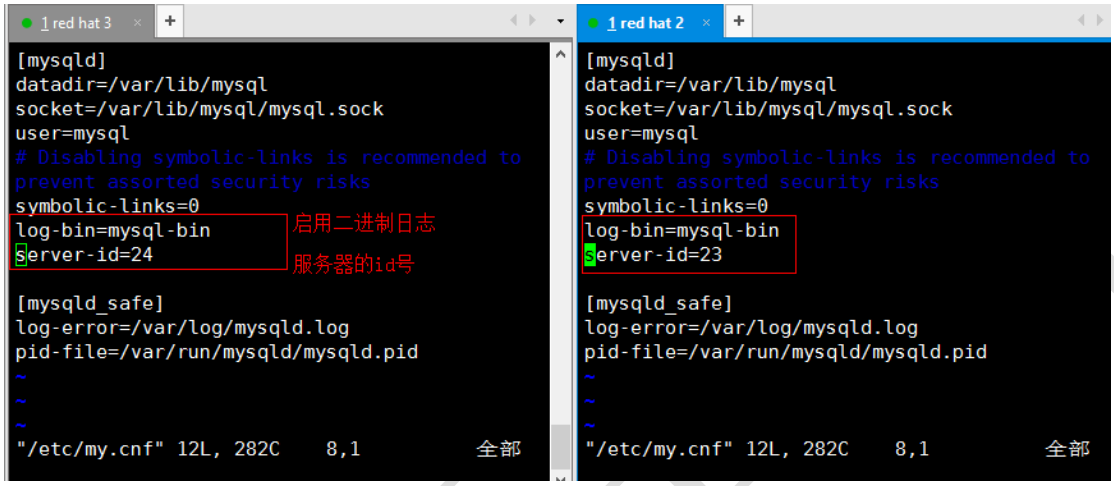
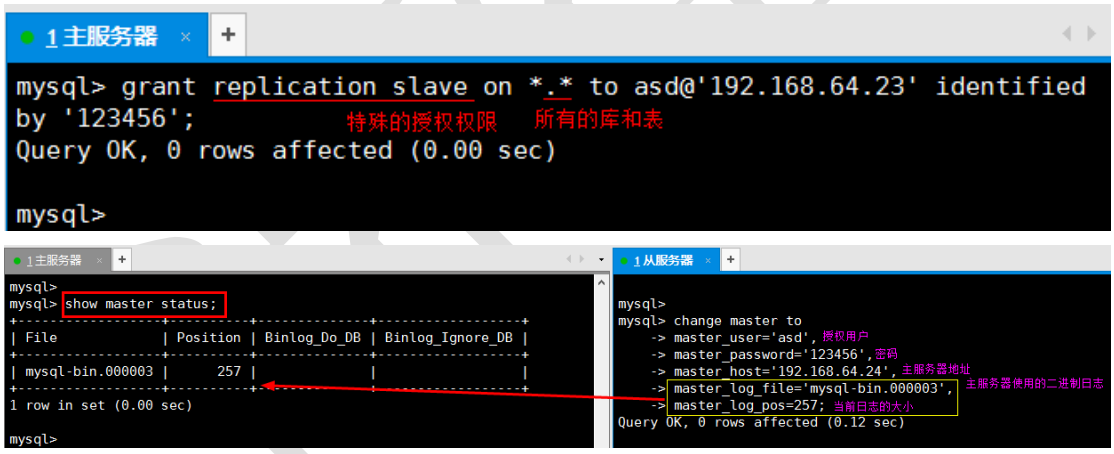
MySQL 集群

1. MySQL 主从备份

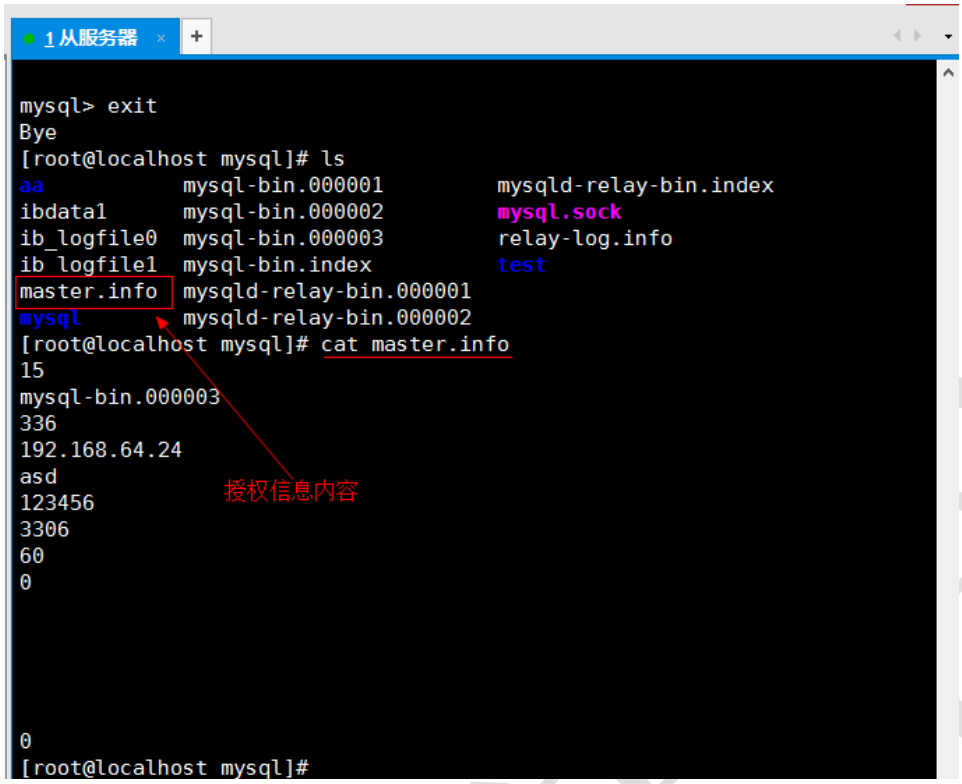
前提条件：安装了 mysql,开启了二进制日志



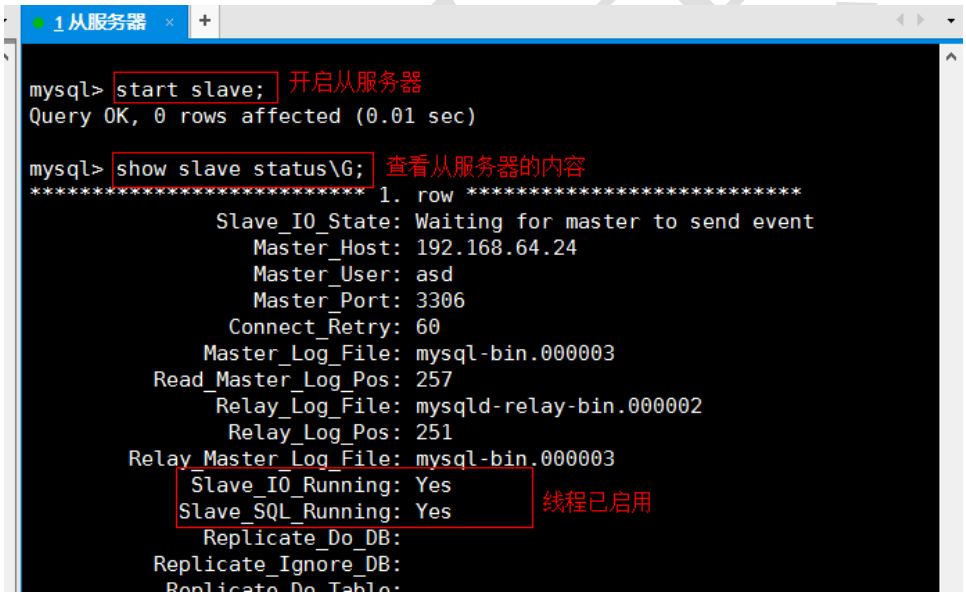
在主服务器上授权，从服务器保存授权的信息



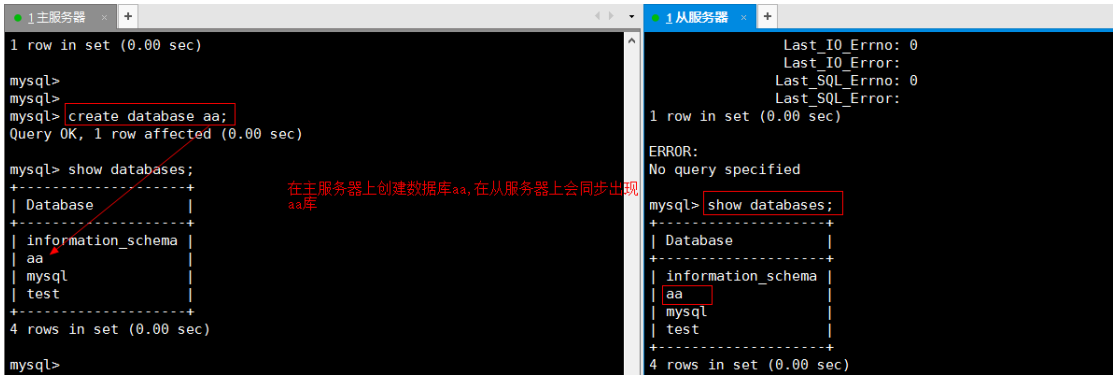
之后在从服务器会产生授权信息文件

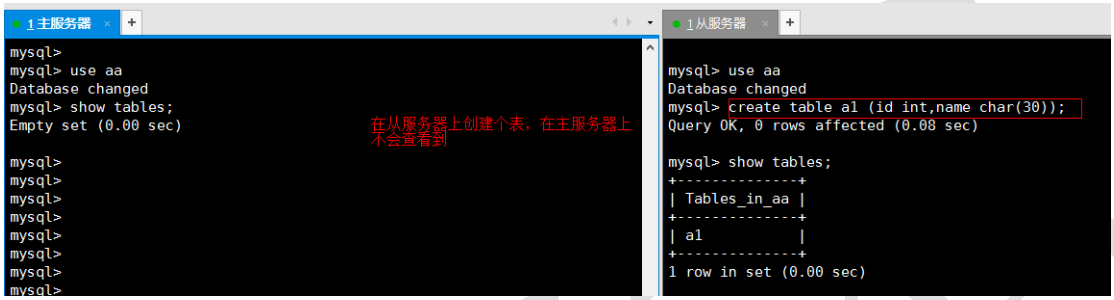


开启从服务器 start slave, 并查看



测试

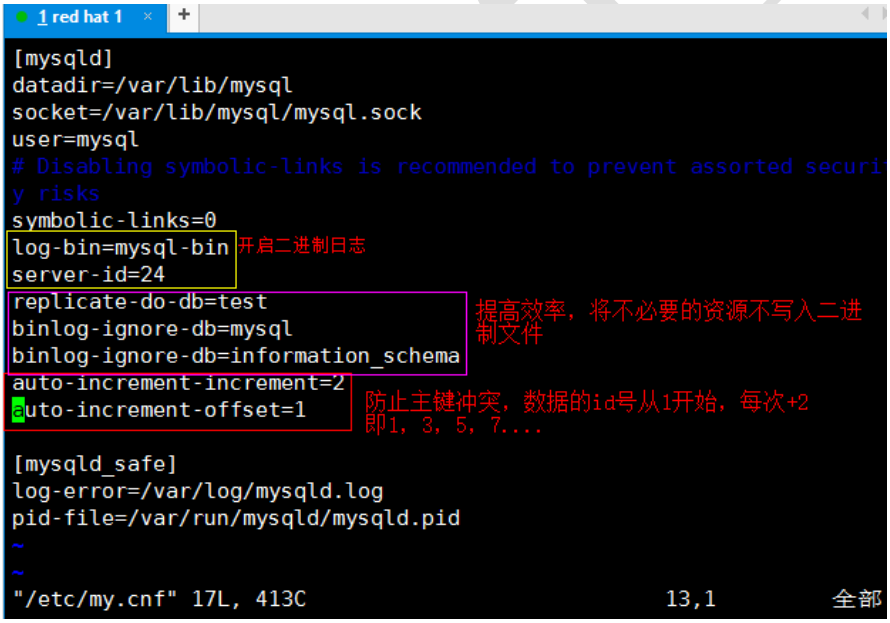




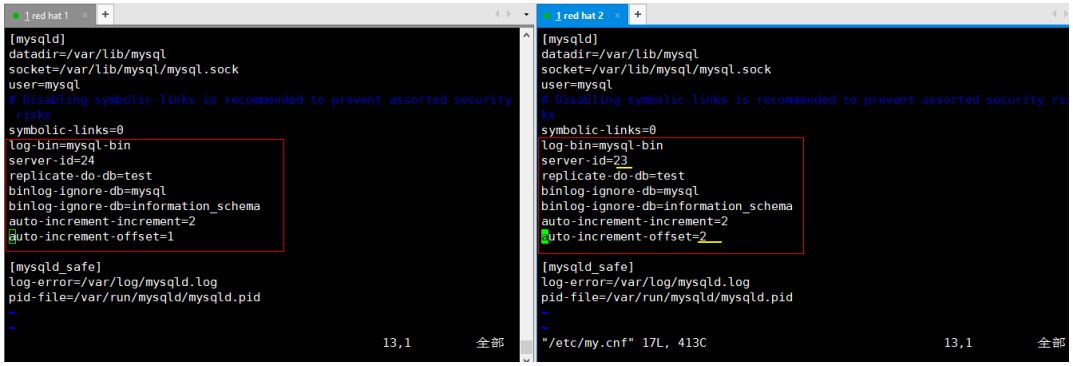
2. MySQL 主主备份

1. 以 1 为主，2 为从配置一遍主从

在主配置文件中配置一下（开启二进制日志和其他内容）



2. 在 2 上做相同的配置

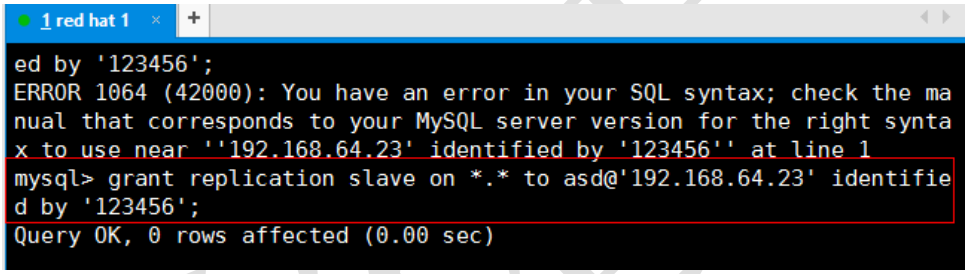


3. 启动服务器

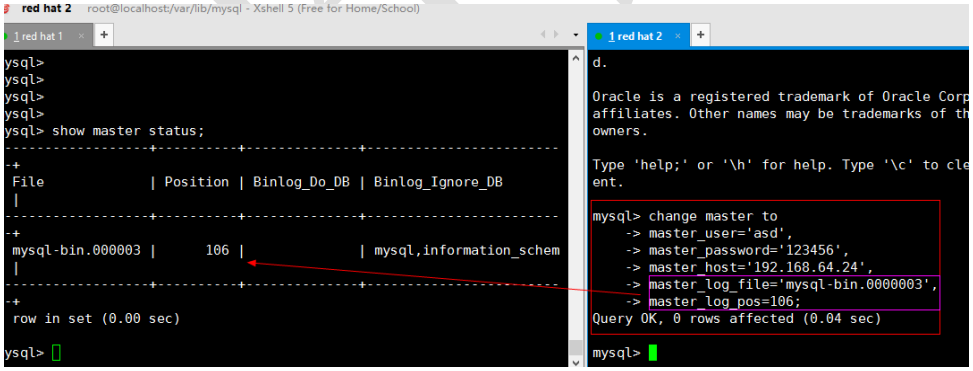
```
[root@localhost mysql]# vim /etc/my.cnf
[root@localhost ~]# service mysqld start
初始化 MySQL 数据库：  Installing MySQL system tables...
OK
Filling help tables...
OK
```

1 为主 2 为从：

在主服务器(1)上授权



在从服务器(2)上保存授权信息



2 为主 1 为从：

在主服务器(2)上授权

```
    -> master_log_pos=106;
Query OK, 0 rows affected (0.04 sec)

mysql> grant replication slave on *.* to asd@'192.168.64.24' identified by
 '123456';
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>
```

在从服务器(1)上保存授权信息

```
mysql>
mysql>
mysql>
mysql> change master to
    -> master_user='asd',
    -> master_password='123456',
    -> master_host='192.168.64.23',
    -> master_log_file='mysql-bin.000001',
    -> master_log_pos=106;
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql>
```

```
 '123456';
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>
mysql> show master status;
+------------------+----------+--------------+------------------------
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB
+------------------+----------+--------------+------------------------
| mysql-bin.000001 |      106 |              | mysql,information_schem
+------------------+----------+--------------+------------------------
1 row in set (0.00 sec)
```

1 和 2 都执行 start slave（互为主从）

```
6', master_host='192.168.64.23', master_log_file='mysql-bin.00000
1', master_log_pos=106;
Query OK, 0 rows affected (0.33 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.64.23
                  Master_User: asd
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 190
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 335
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB: test
```

```
mysql> change master to master_user='asd', master_password='123456', master_hos
t='192.168.64.24', master_log_file='mysql-bin.000003', master_log_pos=106;
Query OK, 0 rows affected (0.04 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: 192.168.64.24
                  Master_User: asd
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 190
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 335
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB: test
```

测试

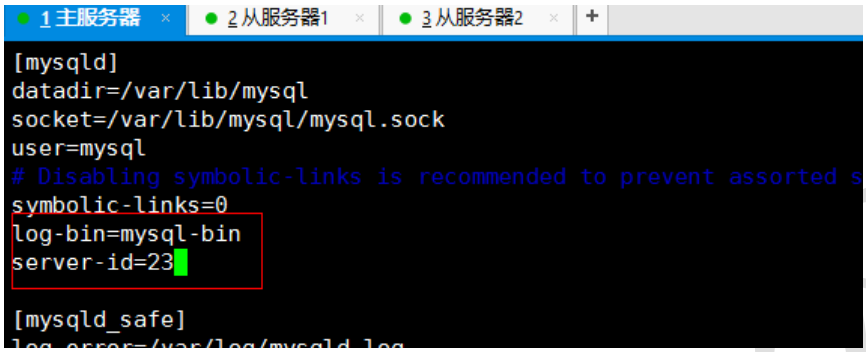
```
mysql> use test;
Database changed
mysql> create table a1(id int);
Query OK, 0 rows affected (0.09 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bb                 |
| mysql              |
| test               |
+--------------------+
4 rows in set (0.00 sec)

mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| a1             |
+----------------+
1 row in set (0.00 sec)

mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| a1             |
| a2             |
+----------------+
2 rows in set (0.00 sec)
```

```
ERROR:
No query specified

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> use test;
Reading table information for completion of table and
You can turn off this feature to get a quicker startup

Database changed
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| a1             |
+----------------+
1 row in set (0.00 sec)

mysql> create table a2;
ERROR 1113 (42000): A table must have at least 1 colum
mysql> create table a2(id int);
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| a1             |
| a2             |
+----------------+
2 rows in set (0.00 sec)

mysql>
```

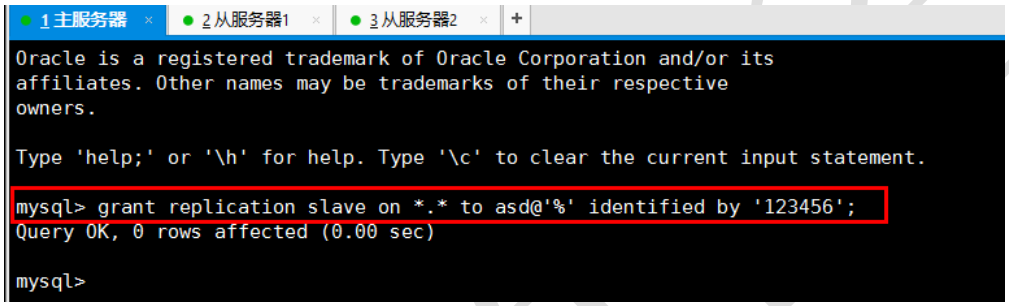
3. MySQL 一主多从

主服务器配置

开启二进制日志，并启动 mysql

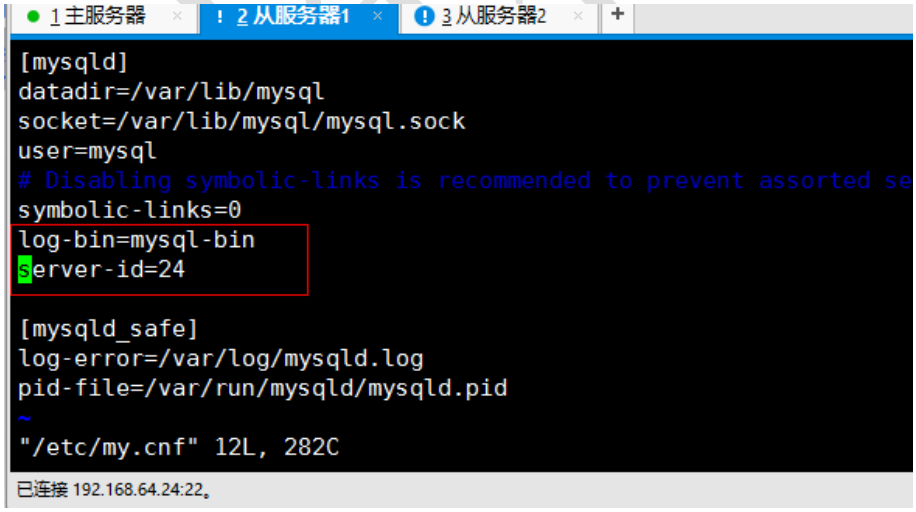


在主服务器上授权

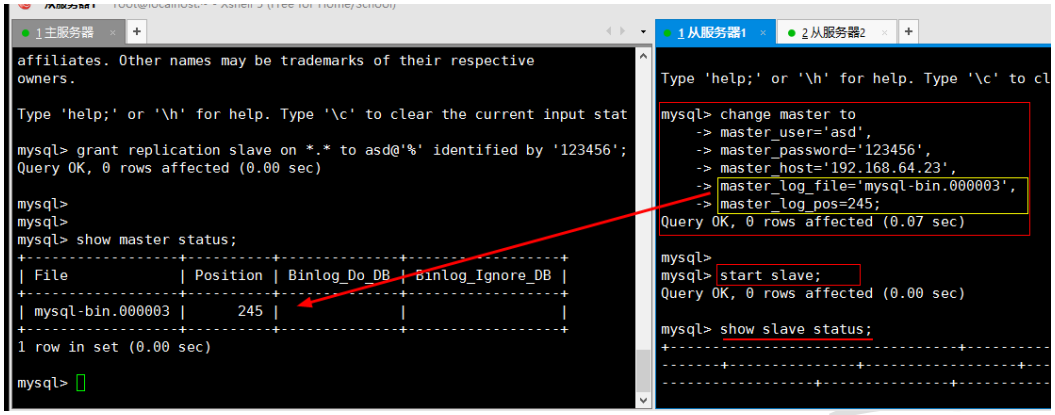


从服务器配置

开启二进制日志，并启动 mysql

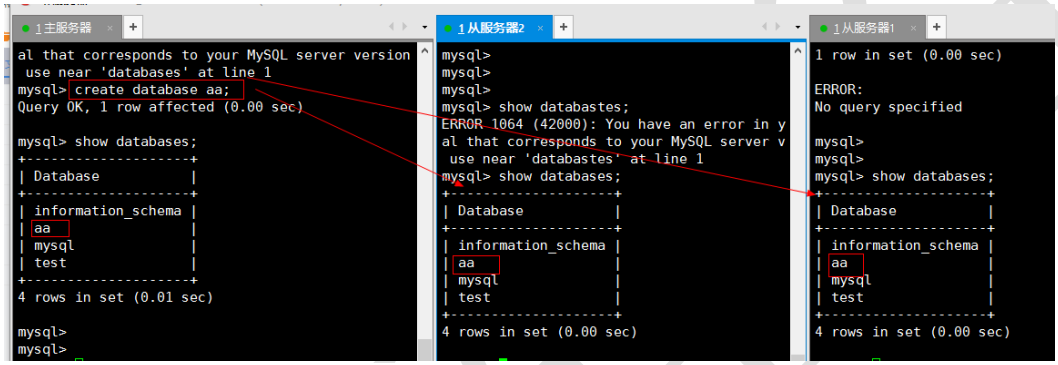


保存授权信息



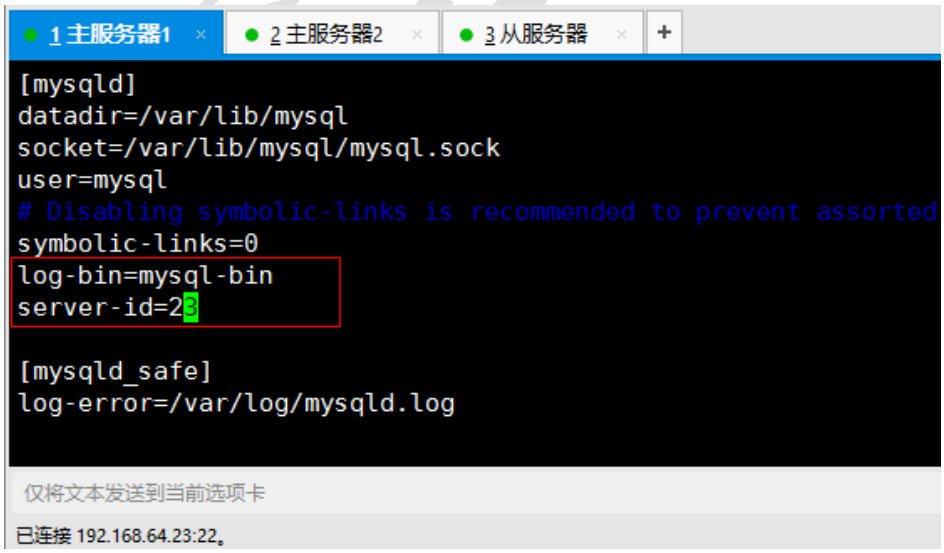
#在另一台从服务器上做相同的配置（注意 id 不能相同）

测试

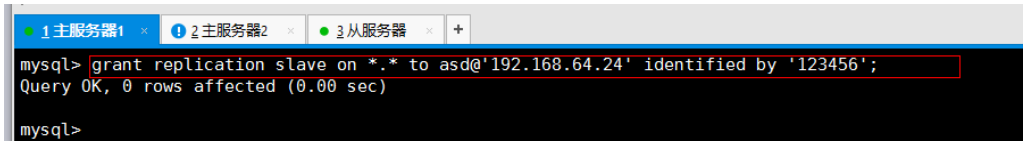


4. MySQL 多主一从

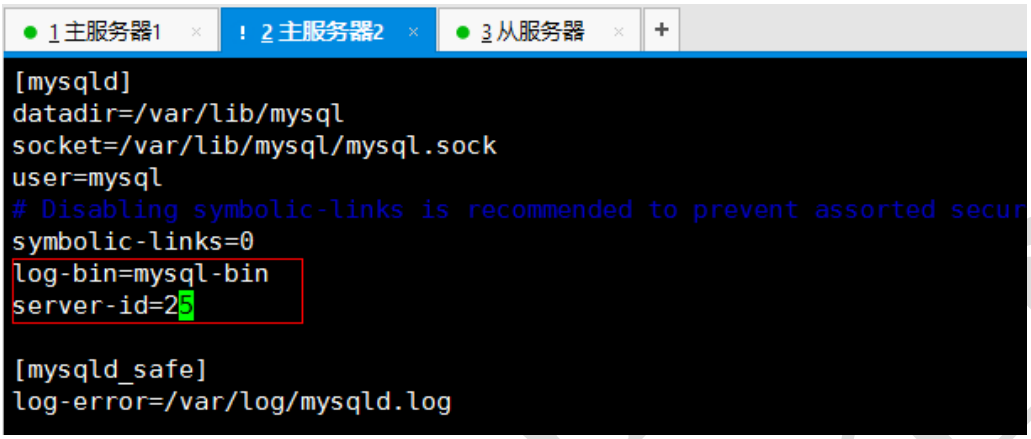
主服务器配置
开启二进制日志，启动服务



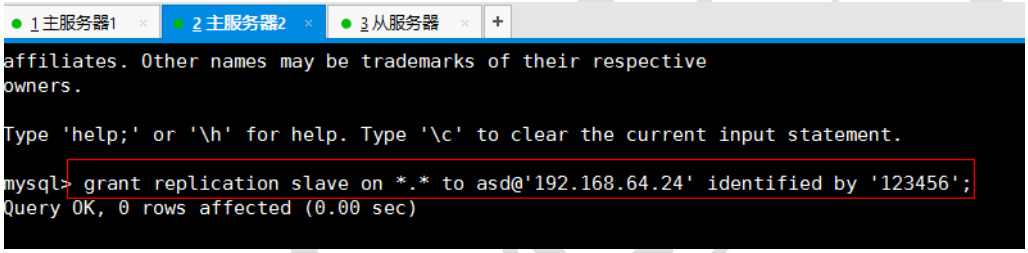
授权



在主服务器 2 上做相同的操作



授权



从服务器操作

对主配置文件操作

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

#M-M-S

[mysqld_multi]
mysqld=/usr/bin/mysqld_safe
mysqladmin=/usr/bin/mysqladmin
log=/tmp/multi.log

[mysqld10]
port=3306
datadir=/var/lib/mysqla/
pid-file=/var/lib/mysqla/mysqld.pid
socket=/var/lib/mysqla/mysql.sock
user=mysql
server-id=30

[mysqld20]
port=3307
datadir=/var/lib/mysqlb/
pid-file=/var/lib/mysqlb/mysqld.pid
socket=/var/lib/mysqlb/mysql.sock
user=mysql
server-id=30
~
~
```

初始化数据库，生成目录 mysqla,mysqlb

```
1 主服务器1   2 主服务器2   3 从服务器   +
[root@localhost ~]# vim /etc/my.cnf
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# mysql_install_db --datadir=/var/lib/mysqla --user=mysql
Installing MySQL system tables...
OK
Filling help tables...
OK
```

初始化数据库，生成mysqla

```
1 主服务器1   2 主服务器2   3 从服务器   +

Please report any problems with the /usr/bin/mysqlbug script!

[root@localhost ~]# mysql_install_db --datadir=/var/lib/mysqlb --user=mysql
Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system
```

生成mysqlb

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# cd /var/lib/
[root@localhost lib]# ls
alternatives  certmonger  fprint      logrotate.status  mysql   nfs       polkit-1  random-see
authconfig    dbus        games       misc              mysqla  ntp       postfix   readahead
cas           dhclient    ipa-client  mlocate           mysqlb  plymouth  prelink   rhsm
[root@localhost lib]# ls -lh mysqla/
总用量 8.0K
drwx------ 2 mysql root 4.0K 2月  25 07:26 mysql
drwx------ 2 mysql root 4.0K 2月  25 07:26 test
[root@localhost lib]# ls -lh mysqlb/
总用量 8.0K
drwx------ 2 mysql root 4.0K 2月  25 07:31 mysql
drwx------ 2 mysql root 4.0K 2月  25 07:31 test
[root@localhost lib]#
```

设置 mysqla,mysqlb 目录及以下文件的属主为 mysql(防止出现权限问题)

```
[root@localhost lib]# chown -R mysql /var/lib/mysqla/
[root@localhost lib]# chown -R mysql /var/lib/mysqlb/
[root@localhost lib]#
[root@localhost lib]#
```

启动从服务器线程

```
[root@localhost ~]# mysqld_multi --defaults-file=/etc/my.cnf start 23
[root@localhost ~]# netstat -anpt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program name
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN      1160/rpcbind
tcp        0      0 0.0.0.0:57680               0.0.0.0:*                   LISTEN      1178/rpc.statd
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN      1382/sshd
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN      1259/cupsd
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN      1458/master
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN      2107/mysqld
tcp        0      0 192.168.64.24:22            192.168.64.1:58561          ESTABLISHED 1717/sshd
tcp        0      0 192.168.64.24:22            192.168.64.1:50013          ESTABLISHED 1646/sshd
tcp        0      0 :::53903                    :::*                        LISTEN      1178/rpc.statd
tcp        0      0 :::111                      :::*                        LISTEN      1160/rpcbind
tcp        0      0 :::22                       :::*                        LISTEN      1382/sshd
tcp        0      0 ::1:631                     :::*                        LISTEN      1259/cupsd
tcp        0      0 ::1:25                      :::*                        LISTEN      1458/master
```

```
[root@localhost ~]# mysqld_multi --defaults-file=/etc/my.cnf start 25
[root@localhost ~]# netstat -anpt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program name
tcp        0      0 0.0.0.0:3307                0.0.0.0:*                   LISTEN      5245/mysqld
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN      1160/rpcbind
tcp        0      0 0.0.0.0:57680               0.0.0.0:*                   LISTEN      1178/rpc.statd
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN      1382/sshd
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN      1259/cupsd
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN      1458/master
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN      2107/mysqld
tcp        0    264 192.168.64.24:22            192.168.64.1:58561          ESTABLISHED 1717/sshd
tcp        0      0 192.168.64.24:22            192.168.64.1:50013          ESTABLISHED 1646/sshd
tcp        0      0 :::53903                    :::*                        LISTEN      1178/rpc.statd
tcp        0      0 :::111                      :::*                        LISTEN      1160/rpcbind
tcp        0      0 :::22                       :::*                        LISTEN      1382/sshd
tcp        0      0 ::1:631                     :::*                        LISTEN      1259/cupsd
tcp        0      0 ::1:25                      :::*                        LISTEN      1458/master
[root@localhost ~]#
```

登录并保存授权信息

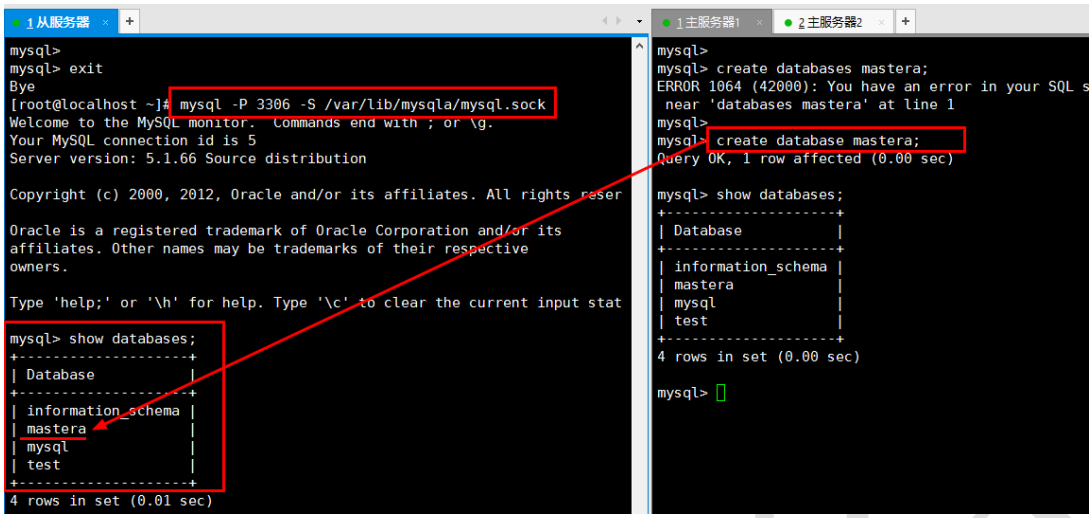
1 主服务器1 | 2 主服务器2 | 3 从服务器

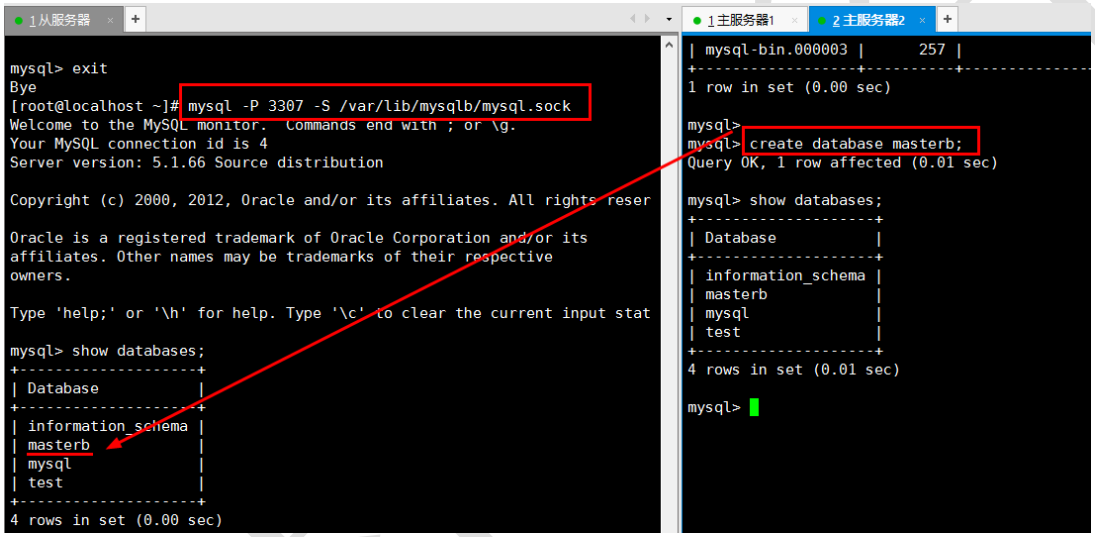
```
[root@localhost ~]# mysql -P 3306 -S /var/lib/mysqla/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1                    登录主服务器1
Server version: 5.1.66 Source distribution

Copyright (c) 2000, 2012, 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_user='asd',
    -> master_password='123456',
    -> master_host='192.168.64.23',
    -> master_log_file='mysql-bin.000003',
    -> master_log_pos=257;
Query OK, 0 rows affected (0.09 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: 192.168.64.23
                  Master_User: asd
                  Master_Port: 3306
                Connect_Retry: 60
```

1 主服务器1 | 2 主服务器2 | 3 从服务器

```
mysql>
mysql>
mysql> exit
Bye
[root@localhost ~]# mysql -P 3307 -S /var/lib/mysqlb/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1                    登录主服务器2，同步2上的数据
Server version: 5.1.66 Source distribution

Copyright (c) 2000, 2012, 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_user='asd',
    -> master_password='123456',
    -> master_host='192.168.64.25',
    -> master_log_file='mysql-bin.000003',
    -> master_log_pos=257;
Query OK, 0 rows affected (0.10 sec)

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

mysql> show slave status;
```

测试





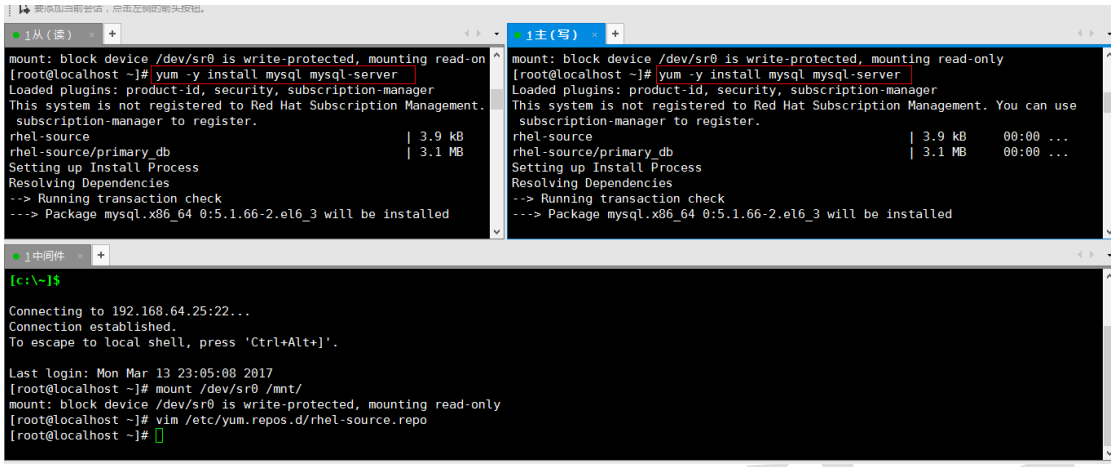
5. MySQL 中间件-Amoeba

中间件：一种提供在不同技术、不同的软件之间共享资源的程序，更大化了利用了数据库的性能，可以无限扩展（注：真实环境中并非如此）

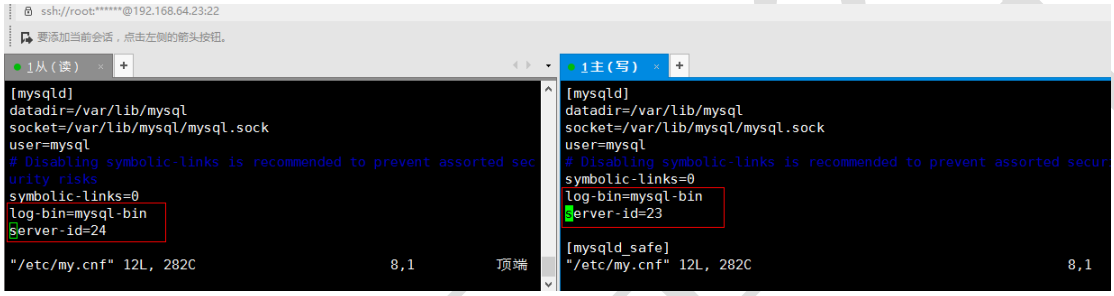
数据库的中间件：

mysql proxy（官方版本）性能低，需要 lua 脚本
atlas 性能低，响应时间长
amoeba 陈思儒研发的

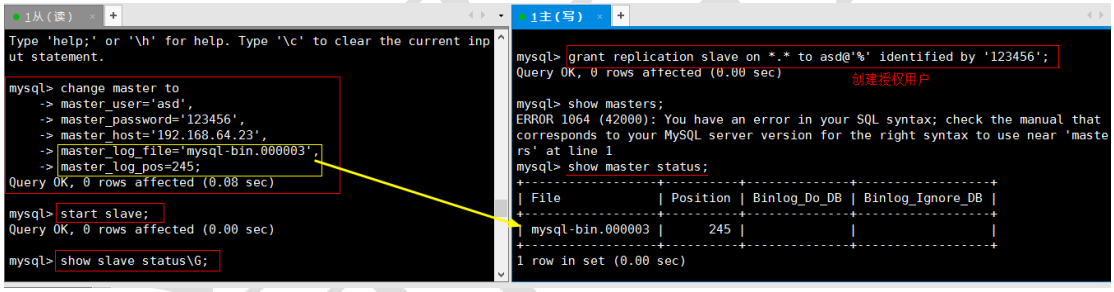
一．先搭建一个主从关系的服务器
在主、从服务器上安装 mysql mysql-server



1. 开启二进制日志



2. 在主服务器上授权，从服务器上保存授权信息，并开启从服务线程。



3. 关闭从服务器线程，为了做读写分离时，测试有明显的实验效果（实际生产环境中不能停掉。。）

```
1从（读）

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

mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State:
                  Master_Host: 192.168.64.23
                  Master_User: asd
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000003
          Read_Master_Log_Pos: 324
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 330
        Relay_Master_Log_File: mysql-bin.000003
             Slave_IO_Running: No
            Slave_SQL_Running: No
              Replicate_Do_DB:
```

二．配置读写分离

1.安装 gcc 环境（amoeba 需要源码安装）

```
[root@localhost ~]# yum -y install gcc*
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscri
rhel-source
rhel-source/primary_db
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package gcc.x86_64 0:4.4.7-3.el6 will be installed
--> Processing Dependency: cpp = 4.4.7-3.el6 for package: gcc-4.4.7-3.el6.x86_64
```

2.拷贝第三方软件，创建单独的目录

```
[root@localhost ~]# mount /dev/sr0 /mnt/
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@localhost ~]# cd /mnt/
[root@localhost mnt]# ls
amoeba-mysql-1.3.1-BETA.zip  jdk-7u40-linux-x64.gz
[root@localhost mnt]# cp * /usr/src/
[root@localhost mnt]# cd !$
cd /usr/src/
[root@localhost src]# ls
amoeba-mysql-1.3.1-BETA.zip  debug  jdk-7u40-linux-x64.gz  kernels
```

```
[root@localhost src]# cd
[root@localhost ~]# mkdir /amoeba
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
```

创建单独的目录，用来存储安装ameoeba相关的东西

3.先安装 jdk(amoeba 是由 java 语言编写的，所以先安装 jdk),配置 java 环境

```
[root@localhost src]# tar -xf jdk-7u40-linux-x64.gz -C /amoeba/
[root@localhost src]#                                          解压到指定的目录下
[root@localhost src]#
[root@localhost src]# cd /amoeba/
[root@localhost amoeba]# ls
jdk1.7.0_40
[root@localhost amoeba]# ln -s jdk1.7.0_40/ jdk
[root@localhost amoeba]# ls                                    创个软链接去除版本号，方便操作
jdk  jdk1.7.0_40
[root@localhost amoeba]#
```

4. 声明用 java 写出来的程序如何调用（/etc/profile）

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.
JAVA_HOME=/amoeba/jdk
export JAVA_HOME

PATH=$JAVA_HOME/bin:$PATH
export PATH

CLASSPATH=.:$JAVA_HOME/bin/tools.jar:$JAVA_HOME/lib/dt.jar:$CLASSPATH
export CLASSPATH
```

```
[root@localhost amoeba]#
[root@localhost amoeba]# source /etc/profile          加载一下，使之生效
[root@localhost amoeba]#
[root@localhost amoeba]#
```

```
[root@localhost amoeba]# java -version                测试一下
java version "1.7.0_40"
Java(TM) SE Runtime Environment (build 1.7.0_40-b43)
Java HotSpot(TM) 64-Bit Server VM (build 24.0-b56, mixed mode)
[root@localhost amoeba]#
```

5. 安装 amoeba

```
[root@localhost amoeba]# cd /usr/src/
[root@localhost src]# ls
amoeba-mysql-1.3.1-BETA.zip  debug  jdk-7u40-linux-x64.gz  kernels
[root@localhost src]#
[root@localhost src]#
[root@localhost src]# unzip amoeba-mysql-1.3.1-BETA.zip -d /usr/local/amoeba
Archive:  amoeba-mysql-1.3.1-BETA.zip                  解压到指定位置下，用 -d
   creating: /usr/local/amoeba/bin/
  inflating: /usr/local/amoeba/bin/amoeba
  inflating: /usr/local/amoeba/bin/amoeba.bat
   creating: /usr/local/amoeba/doc/
   creating: /usr/local/amoeba/lib/
   creating: /usr/local/amoeba/lib/classes/
  inflating: /usr/local/amoeba/LICENSE.txt
```

```
[root@localhost src]# chmod -R +x /usr/local/amoeba/bin/
[root@localhost src]#                                  为了amoeba程序下的命令正常执行，给予权限
[root@localhost src]#                                  并不是非做不可的操作
```

配置 amoeba 这个软件

```
[root@localhost src]# cd /usr/local/amoeba/
[root@localhost amoeba]# ls
bin  build.properties  build.xml  conf  doc  lib  LICENSE.txt  README.html  src
[root@localhost amoeba]# cd conf/
[root@localhost conf]# ls
access_list.conf  amoeba.dtd  amoeba.xml  function.dtd  functionMap.xml  log4j.dtd  log4j.xml  rule.dtd  ruleFunctionMap.xml  rule.xml
[root@localhost conf]#
[root@localhost conf]#
[root@localhost conf]#
```

存放了配置文件

主配置文件

<server>.....</server>区域

```
7
8                <property name="port">8066</property>
9
10
11
12               <property name="ipAddress">192.168.64.25</property>
13
14               <property name="readThreadPoolSize">20</property>
15
16
17               <property name="clientSideThreadPoolSize">30</property>
18
19
20               <property name="serverSideThreadPoolSize">30</property>
21
22
23               <property name="netBufferSize">128</property>
24
25
26               <property name="tcpNoDelay">true</property>
27
28
29               <property name="user">asd</property>
30
31
32
33
34               <property name="password">123456</property>
35
36
37               <property name="queryTimeout">60</property>
38
```

端口

server区域的配置

amoeba安装的 ip地址

使用默认值

登录amoeba的用户名，本来是root(和管理员用户没关系)

登录用户使用的密码

```
  <connectionManagerList>
连接时管理组件的声明ectionManager name="defaultManager" class="com.meidusa.amoeba.net.MultiConnectionManagerWrapper">
                <property name="subManagerClassName">com.meidusa.amoeba.net.AuthingableConnectionManager</property>

                <!--
                  default value is avaliable Processors
                <property name="processors">5</property>
                 -->
        </connectionManager>
```

不需要管

<dbServerList>.....</dbServerList>区域

```
63                -->
64                <dbServer name="server1">                    服务器名字
65
66                        <!-- PoolableObjectFactoryÉµÏÓÀà  -->
67                        <factoryConfig class="com.meidusa.amoeba.mysql.net.MysqlServerConnectionFactory">
68                                <property name="manager">defaultManager</property>
69
70                                <!-- Õæ ÉµmysqlÊý¾Ý Ý¿ â¶ É¿ Ú -->
71                                <property name="port">3306</property>
72
73                                <!-- Õæ ÉµmysqlÊý¾Ý Ý¿ âIP -->
74                                <property name="ipAddress">192.168.64.23</property>          登录的是哪个数据库服务器（这里是主）
75                                <property name="schema">test</property>
76
77                                <!-- ÓÃÓÚµÇÂ¼ mysqlµÄÓÃ»§ ÃûÂ -->
78                                <property name="user">asd</property>                         使用的用户名是什么（这里是在主服务器上授权的那个用户）
79
80                                <!-- ÓÃÓÚµÇÂ¼ mysqlµÄÃÜÂë -->
81
82
83                                <property name="password">123456</property>                  使用的密码
84
85
86                        </factoryConfig>
87
88                        <!-- ObjectPoolÉµÏÓÀà  -->
89                        <poolConfig class="com.meidusa.amoeba.net.poolable.PoolableObjectPool">
90                                <property name="maxActive">200</property>
91                                <property name="maxIdle">200</property>
92                                <property name="minIdle">10</property>
93                                <property name="minEvictableIdleTimeMillis">600000</property>
94                                <property name="timeBetweenEvictionRunsMillis">600000</property>
95                                <property name="testOnBorrow">true</property>
96                                <property name="testWhileIdle">true</property>
97                        </poolConfig>                                                        和优化相关的，可以不做操作
98                </dbServer>
```

```
101               <dbServer name="master" virtual="true">
102                       <poolConfig class="com.meidusa.amoeba.server.MultipleServerPool">
103       池的名字，主的，提供写  <!-- ¸ ÔØ ¾ Ù º â² ÎÊý 1=ROUNDROBIN , 2=WEIGHTBASED , 3=HA-->
104                               <property name="loadbalance">1</property>
105
106                               <!-- ² ÎÔë¸ Âpool¸ º ÔØ ¾ Ù º âµÄpoolNameÁÐ ± í ÒÔ¶ º º Å· Ô¸ î -->
107                               <property name="poolNames">server1</property>
108                       </poolConfig>
109               </dbServer>
```

由于只提供了一个服务器模板，需要自己复制另一个填写关于读的

```
<dbServer name="server2">                    服务器2
        <!-- PoolableObjectFactoryÉµÏÓÀà  -->
        <factoryConfig class="com.meidusa.amoeba.mysql.net.MysqlServerConnectionFactory">
                <property name="manager">defaultManager</property>

                <!-- Õæ ÉµmysqlÊý¾Ý Ý¿ â¶ É¿ Ú -->
                <property name="port">3306</property>                端口不变

                <!-- Õæ ÉµmysqlÊý¾Ý Ý¿ âIP -->
                <property name="ipAddress">192.168.64.24</property>  第二台服务器的ip即从服务的
                <property name="schema">test</property>
                                                                     代表同步的库
                <!-- ÓÃÓÚµÇÂ¼ mysqlµÄÓÃ»§ ÃûÂ -->
                <property name="user">asd</property>                 在从服务器上授权的那个用户

                <!-- ÓÃÓÚµÇÂ¼ mysqlµÄÃÜÂë -->


                <property name="password">123456</property>


        </factoryConfig>

        <!-- ObjectPoolÉµÏÓÀà  -->
        <poolConfig class="com.meidusa.amoeba.net.poolable.PoolableObjectPool">
                <property name="maxActive">200</property>
                <property name="maxIdle">200</property>
                <property name="minIdle">10</property>
                <property name="minEvictableIdleTimeMillis">600000</property>
                <property name="timeBetweenEvictionRunsMillis">600000</property>
                <property name="testOnBorrow">true</property>
                <property name="testWhileIdle">true</property>
        </poolConfig>
</dbServer>
```

```
    <dbServer name="slave" virtual="true">
        <poolConfig class="com.meidusa.amoeba.server.MultipleServerPool">
                <!-- ¸ º ÔØ ¾ ù º â² ÎÊý 1=ROUNDROBIN , 2=WEIGHTBASED , 3=HA-->
                <property name="loadbalance">1</property>

                <!-- ² ÎóÈ¡ Âpool¸ º ÔØ ¾ ù º âµÄpoolNameÁÐ ± í ÒÔ¶ º º Â· Ö¸ î -->
                <property name="poolNames">server1,server2</property>
        </poolConfig>
    </dbServer>
```

读从server1 server2

```
181                 <property name="LRUMapSize">1500</property>
182                 <property name="defaultPool">master</property>
183
184
185                 <property name="writePool">master</property>
186                 <property name="readPool">slave</property>
187
188                 <property name="needParse">true</property>
189         </queryRouter>
190 </amoeba:configuration>
-- 插入 --
```

默认访问池

写入池

读取池

启动 amoeba，修改一下启动脚本：/usr/local/amoeba/bin/amoeba

```
DEFAULT_OPTS="-server -Xms256m -Xmx256m -Xss256k"
# DEFAULT_OPTS="$DEFAULT_OPTS -XX:+HeapDumpOnOutOfMemoryError -XX:+AggressiveOpts -XX:+UseParallelG
e=64m"
DEFAULT_OPTS="$DEFAULT_OPTS -Damoeba.home=\"$AMOEBA_HOME\""
DEFAULT_OPTS="$DEFAULT_OPTS -Dclassworlds.conf=\"$AMOEBA_HOME/bin/amoeba.classworlds\""

CMD="exec \"$JAVA_HOME/bin/java\" $DEFAULT_OPTS $OPTS -classpath \"$CLASSPATH\"  $MAIN_CLASS $@"
eval $CMD
```

注意：将-Xss128k 修改为 -Xss256

#在主和从服务器上进行指定用户授权，授权目的为了让 amoeba 能连接到主从服务器进行查询。

nohup bash -x /usr/local/amoeba/bin/amoeba &
把这个放到后台 退出终端也可以继续运行

```
[root@localhost bin]# nohup bash -x /usr/local/amoeba/bin/amoeba &
[1] 2635
[root@localhost bin]# nohup: 忽略输入并把输出追加到"nohup.out"
```

ps aux | grep amoeba
查看一下运行的程序 查看到的话就说明程序已经运行了起来

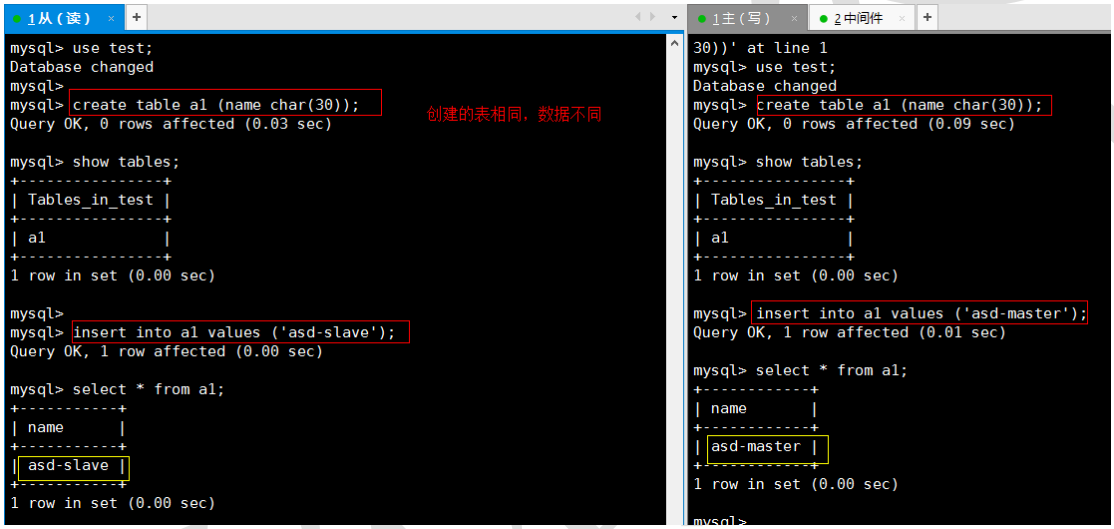
```
[root@localhost bin]# ps aux | grep amoeba
root       2635 17.4  5.5 1072680 56092 pts/0   Sl   00:05   0:03 /amoeba/jdk/bin/java -s
moeba -Dclassworlds.conf=/usr/local/amoeba/bin/amoeba.classworlds -classpath /usr/local/a
cher
root       2663  0.0  0.0 103240   872 pts/0    S+   00:06   0:00 grep amoeba
[root@localhost bin]#
```

测试（安装一个 MySQL 软件包才可以连接）

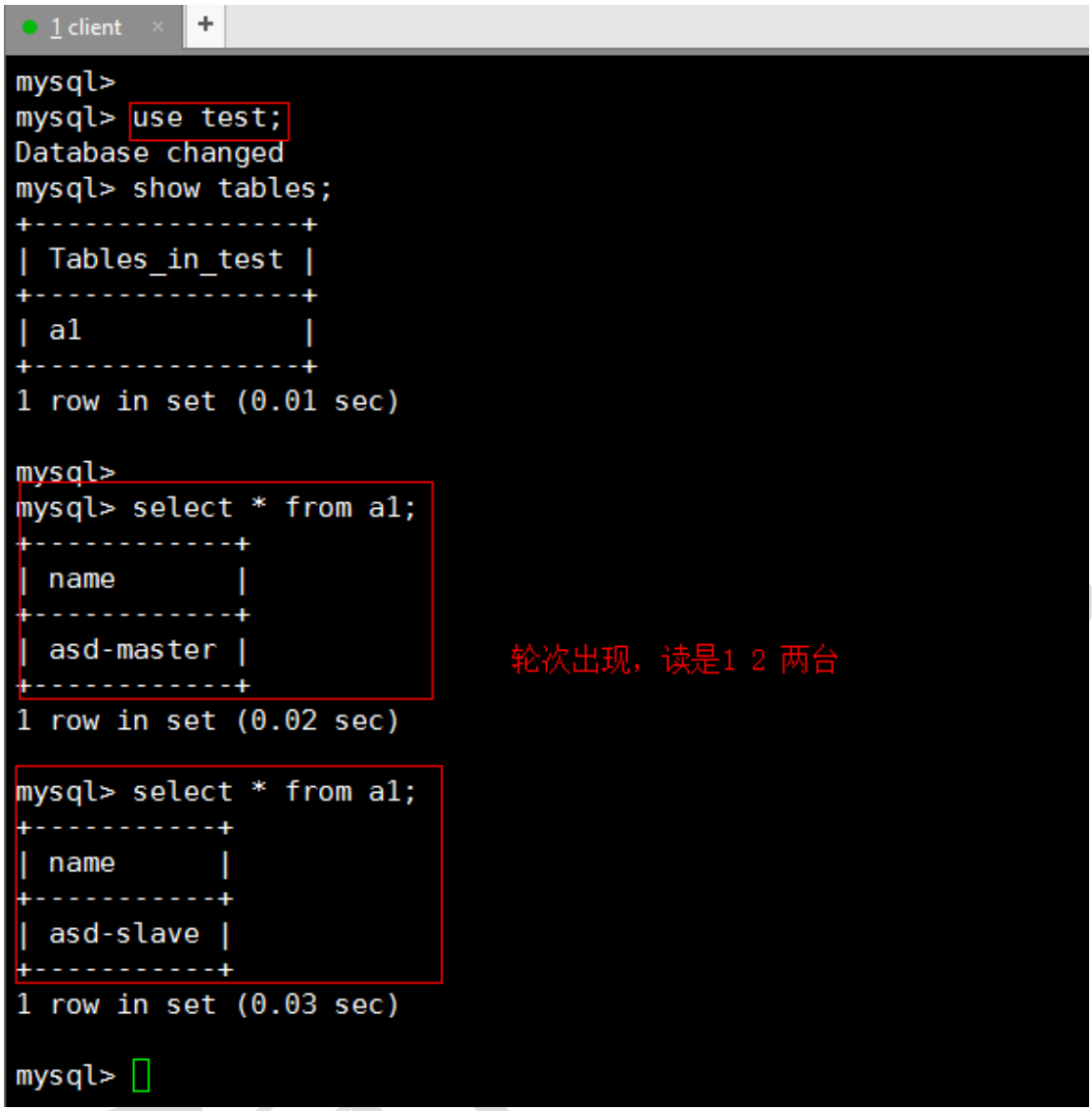
```
[root@localhost conf]# mysql -uasd -p -h 192.168.64.25 -P 8066
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1290243769
Server version: 5.1.45-mysql-amoeba-proxy-1.3.1-BETA Source distribution

Copyright (c) 2000, 2012, 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>
```

指定登录的地址，服务的端口

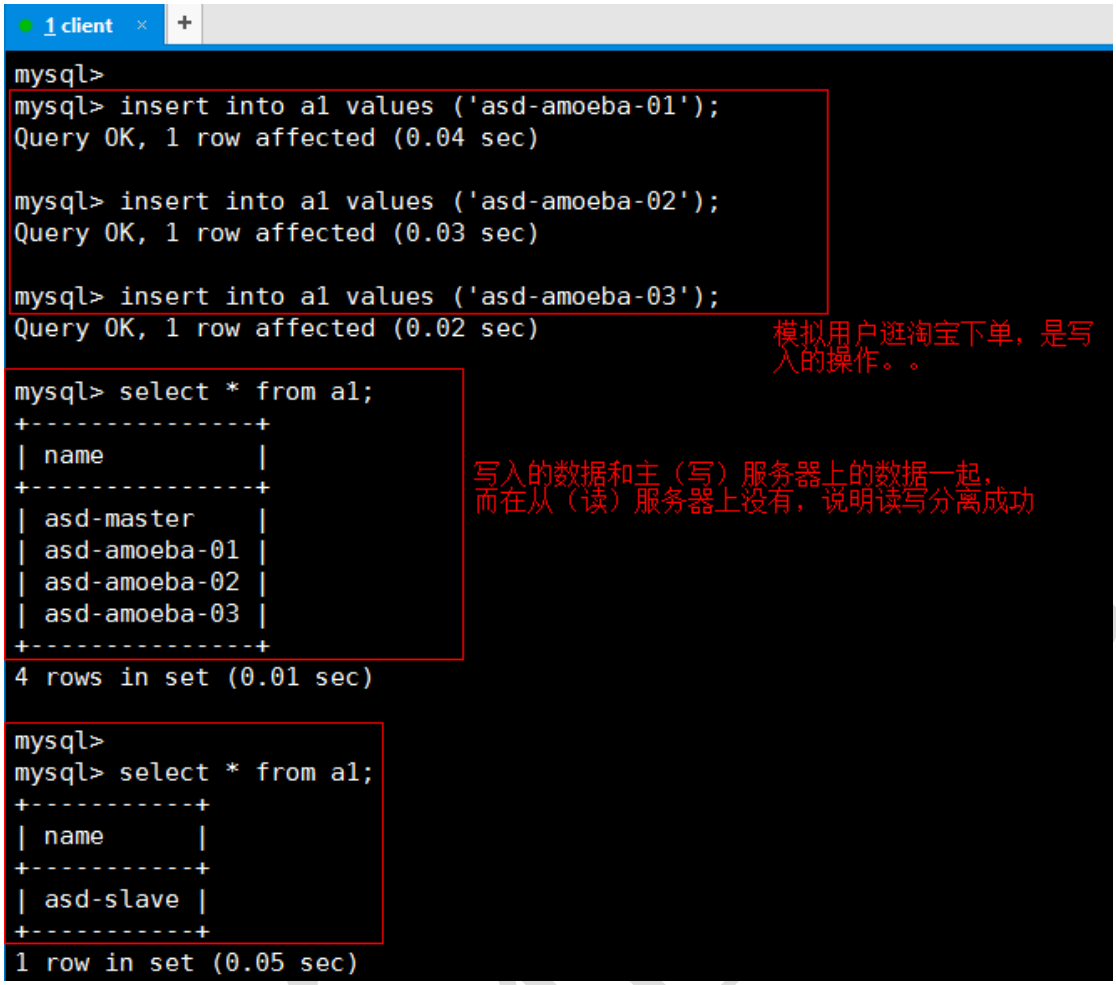
在主、从服务器上创建表 a1，在主服务器的表中插入数据



之后在客户端登录测试
读取池的效果：



写入池效果



以上测试纯粹为了实验效果，在实际生产中，主从开启，主服务器上写入的数据也会同步到从服务器中