1 mysql主从同步

1.1 相关概念：

1 主从同步介绍（什么是主从同步） 存储数据的服务结构

分为2中角色：

主数据库服务器：接收客户端连接的服务器

从数据库服务器：自动同步主数据库服务器的数据到本机

2 主从同步工作过程？

主服务器必须启用binlog日志

从服务器会有IO线程和SQL线程

IO线程的工作任务：复制master主机 binlog日志文件里的SQL命令到本机的relay-log文件里。

SQL线程的工作任务：执行本机relay-log文件里的SQL语句，实现与Master数据一致。

1.2 配置 mysql主从同步

1 拓扑结构

2 配置mysql主从同步

2.1 配置主服务器（master）具体配置如下：192.168.4.51

启用binlog日志

]# vim /etc/my.cnf

server\_id=51

log\_bin

:wq

[root@host51 ~]# systemctl restart mysqld

用户授权(给从服务器拷贝sql命令时连接使用的用户)

[root@host51 ~]# mysql -uroot -p123qqq...A

mysql> grant replication slave on \*.\* to repluser@"%" identified by "123qqq...A";

查看日志信息

mysql> show master status;

+-------------------+----------+--------------+------------------+-------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

+-------------------+----------+--------------+------------------+-------------------+

| host51-bin.000001 | 441 | | | |

+-------------------+----------+--------------+------------------+-------------------+

1 row in set (0.00 sec)

mysql>

2.2 配置从服务器（slave）192.168.4.52 具体步骤如下:

1 指定server\_id 并重启数据库服务

]# vim /etc/my.cnf

[mysqld]

server\_id=52

:wq

[root@host52 ~]# systemctl restart mysqld

[root@host52 ~]# setenforce 0

[root@host52 ~]# systemctl stop firewalld

2 确保与主数据库服务器数据一致

在主服务器51 做数据的完全备份

[root@host51 ~]# mysqldump -uroot -p123qqq...A --master-data -B bbsdb db1 db3 > /root/bak.sql

[root@host51 ~]# scp /root/bak.sql root@192.168.4.52:/root/

在从服务器52 使用备份文件恢复数据

[root@host52 ~]# mysql -uroot -p123qqq...A < /root/bak.sql

[root@host52 ~]# mysql -uroot -p123qqq...A -e 'show databases'

[root@host52 ~]# grep host51 /root/bak.sql

CHANGE MASTER TO MASTER\_LOG\_FILE='host51-bin.000001', MASTER\_LOG\_POS=441;

[root@host52 ~]#

3 数据库管理员登录服务，指定主服务器信息

mysql> change master to master\_host="192.168.4.51" , master\_user="repluser" , master\_password="123qqq...A" ,

master\_log\_file="host51-bin.000001" , master\_log\_pos=441 ;

4 启动slave进程

mysql> start slave;

5 查看进程状态

MySQL> show slave status \G

Master\_Host: 192.168.4.51

Slave\_IO\_Running:Yes

Slave\_SQL\_Running: Yes

3 测试主从同步配置

3.1 在主服务添加客户端连接使用的用户

mysql> grant select , insert , update ,delete on db3.\* to yaya@"%" identified by "123qqq...A";

mysql> grant select , insert , update ,delete on db1.\* to yaya@"%";

[root@host52 ~]# mysql -uroot -p123qqq...A -e 'select user from mysql.user'

3.2 在客户端50连接连接主服务器，存储数据

[root@host50 ~]# mysql -h192.168.4.51 -uyaya -p123qqq...A

mysql> show grants;

mysql> insert into db3.user(name,uid)values("xxx",999);

Query OK, 1 row affected (0.01 sec)

mysql> insert into db3.user(name,uid)values("xxx2",999);

Query OK, 1 row affected (0.01 sec)

mysql> insert into db3.user(name,uid)values("xxx3",999);

Query OK, 1 row affected (0.02 sec)

mysql> exit

3.3 客户端50 连接从服务器也可以看到同样的数据。

mysql> select name,uid from db3.user where name like 'x%';

+------+------+

| name | uid |

+------+------+

| xxx | 999 |

| xxx2 | 999 |

| xxx3 | 999 |

+------+------+

3 rows in set (0.00 sec)

mysql>

1.3 排错

3从服务器数据库目录下会自动创建相关的配置文件

相关的配置文件（里存储的是什么信息）

]# cd /var/lib/mysql/

master.info 记录主服务器服务器信息

主机名-relay-bin.编号 中继日志文件 （存放io线程拷贝的sql命令）

主机名-relay-bin.index 中继日志所有文件 （保存当前已有中继日志文件）

relay-log.info 日志文件 （记录中继日志信息的文件）

1.3 排错

查看报错信息 show slave status \G

Last\_IO\_Error: io线程的报错信息

Last\_SQL\_Error: sql线程的报错信息

把从数据库服务器还原，重新配置。

把相关的配置文件删除

重启数据库服务

[root@host52 mysql]# rm -rf master.info relay-log.info host52-relay-bin.00000\* host52-relay-bin.index

[root@host52 mysql]# systemctl restart mysqld

[root@host52 mysql]# mysql -uroot -p123qqq...A

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.

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.

52 ]# vim /var/lib/mysql/auto.cnf

52 ]# systemctl restart mysqld

Last\_IO\_Error: Got fatal error 1236 from master when reading data from binary log: 'Client requested master to start replication from position > file size'

51 show master status ;

52 stop slave;

change master to master\_log\_file="日志名" , master\_log\_pos=偏移量；

start slave;

show slave status \G

Last\_SQL\_Error: Could not execute Write\_rows event on table db3.user; Duplicate entry '52' for key 'PRIMARY', Error\_code: 1062; handler error HA\_ERR\_FOUND\_DUPP\_KEY; the event's master log localhost-bin.000001, end\_log\_pos 1206 老师我这么也有问题

52 drop database db3.user;

51 mysqldump -uroot -p密码 db3 user > /root/user.sql

scp /root/user.sql root@192.168.4.52:/root/

52 stop slave;

use db3;

source /root/user.sql

start slave;

show slave status \G

mysql> start slave;

ERROR 1200 (HY000): The server is not configured as slave; fix in config file or with CHANGE MASTER TO

检查 /etc/my.cnf 文件 指定 server\_id=52 吗 ？

指定server\_id 后 重启 mysqld服务 再执行 change master to 然后再执行start slave;

SLAVE\_IO\_Running:Connecting

52 cat /var/lib/mysql/master.info

[root@host52 ~]# mysql -h192.168.4.51 -urepluser -p123qqq...A

51 HE 52 firewalld 都要关闭

2 mysql主从同步模式

2.1 mysql主从同步结构类型

一主一从 51 52

一主多从

主从从结构

2.2 配置mysql一主多从同步结构： 把数据库服务器53也配置为51的从数据库服务器

51的配置 数据库服务要运行着 binlog日志是启用的 有授权用户repluser

从服务器53的配置

安装MySQL服务软件并运行服务

1 指定server\_id 并重启数据库服务

2 确保与主服务器数据一致

[root@host51 ~]# mysqldump -uroot -p123qqq...A --master-data -B bbsdb db1 db3 mysql > /root/four.sql

[root@host51 ~]# scp /root/four.sql root@192.168.4.53:/opt/

[root@host53 ~]# mysql -uroot -p123qqq...A < /opt/four.sql

[root@host53 ~]# systemctl restart mysqld

3 测试主服务器的授权用户

[root@host53 ~]# mysql -h192.168.4.51 -urepluser -p123qqq...A

4 管理员root登录 指定主服务器信息

[root@host53 ~]# grep -i "host51-bin" /opt/four.sql

CHANGE MASTER TO MASTER\_LOG\_FILE='host51-bin.000001', MASTER\_LOG\_POS=3119;

[root@localhost ~]#

mysql> change master to master\_host="192.168.4.51" , master\_user="repluser" , master\_password="123qqq...A" , master\_log\_file="host51-bin.000001" , master\_log\_pos=3119;

Query OK, 0 rows affected, 2 warnings (0.01 sec)

5 启动slave 进程

mysql> start slave ;

6 查看状态

mysql> show slave status \G

Master\_Host: 192.168.4.51

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

测试：50 连接51 存储数据

在52 和 53 服务器上也能看到同样的数据

[root@host50 ~]# mysql -h192.168.4.51 -uyaya -p123qqq...A

mysql> insert into db1.t3

mysql> insert into db1.t3 values ("yyy","yyy@163.com");

mysql> insert into db1.t3 values ("yyy","yyy@163.com");

mysql> insert into db1.t3 values ("yyy","yyy@163.com");

[root@host50 ~]# mysql -h192.168.4.52 -uyaya -p123qqq...A -e 'select \* from db1.t3'

[root@host50 ~]# mysql -h192.168.4.53 -uyaya -p123qqq...A -e 'select \* from db1.t3'

配置mysql 主从从同步结构 环境准备

把53 恢复为独立的数据库服务器

20 cd /var/lib/mysql

21 rm -rf master.info relay-log.info \*-bin.\*

22 systemctl restart mysqld

23 mysql -uroot -p123qqq...A -e 'show slave status \G'

24 mysql -uroot -p123qqq...A -e 'show databases' 只保留默认的4个库

29 mysql -uroot -p123qqq...A -e 'select user,host from mysql.user' 删除自己添加的授权用户

30\* mysql -uroot -p123qqq...A -e 'delete from mysql.user where user not in ("root","mysql.sys")'

32 mysql -uroot -p123qqq...A -e 'flush privileges'

54主机 netstat -utnlp | grep 3306

55主机 netstat -utnlp | grep 3306

\*\*\*\*\*关闭所有服务器的 selinux 和 firewalld

具体配置如下：

配置主服务器192.168.4.53

[root@host53 mysql]# vi /etc/my.cnf

[mysqld]

server\_id=53

log\_bin=master53

:wq

[root@host53 mysql]# systemctl restart mysqld

[root@localhost mysql]# mysql -uroot -p123qqq...A

mysql> grant replication slave on \*.\* to repluser2@"%" identified by "123qqq...A";

mysql> show master status;

+-----------------+----------+--------------+------------------+-------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

+-----------------+----------+--------------+------------------+-------------------+

| master53.000001 | 442 | | | |

+-----------------+----------+--------------+------------------+-------------------+

1 row in set (0.00 sec)

mysql>

配置数据库服务器192.168.4.54

把其配置为192.168.4.53 的从服务器

同时做192.168.4.55的主服务器

]# vim /etc/my.cnf

[mysqld]

server\_id=54

log\_bin=master54

log\_slave\_updates 启用级联复制功能

:wq

]# systemctl restart mysqld

]# mysql -uroot -p密码

mysql> change master to master\_host="192.168.4.53" , master\_user="repluser2" , master\_password="123qqq...A" , master\_log\_file="master53.000001" , master\_log\_pos=442;

mysql> start slave ;

mysql> show slave status \G

mysql> grant replication slave on \*.\* to repluser3@"%" identified by "123qqq...A";

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> show master status;

+-----------------+----------+--------------+------------------+-------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

+-----------------+----------+--------------+------------------+-------------------+

| master54.000001 | 442 | | | |

+-----------------+----------+--------------+------------------+-------------------+

1 row in set (0.00 sec)

mysql>

配置数据库服务器192.168.4.55 做主机192.168.4.54的从服务器

指定server\_id 并重启服务

测试54的授权用户repluser3

管理登录 指定主服务器信息

启动slave 进程

查看状态

]# vim /etc/my.cnf

[mysqld]

server\_id=55

:wq

]# systemctl restart mysqld

~]# mysql -h192.168.4.54 -urepluser3 -p123qqq...A

mysql> exit;

]# mysql -uroot -p密码

mysql> change master to master\_host="192.168.4.54" , master\_user="repluser3" , master\_password="123qqq...A" , master\_log\_file="master54.000001" ,master\_log\_pos=442;

mysql> start slave;

mysql> show slave status \G

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

测试主从从同步结构的配置

在主服务器53存储数据

mysql -uroot -p123qqq...A -e 'create database bbsdb'

mysql -uroot -p123qqq...A -e 'create table bbsdb.t1(id int)'

mysql -uroot -p123qqq...A -e 'insert into bbsdb.t1 values(119)'

mysql -uroot -p123qqq...A -e 'select \* from bbsdb.t1'

在服务器54 和 55 上能看到同样的数据

]# mysql -uroot -p123qqq...A -e 'select \* from bbsdb.t1'

2.4 配置mysql主主同步结构（也被称为互为主从）（课后练习）

把数据库服务器56 和 57 配置为主主结构

验证方式 2台服务器的IO线程的SQL线程都是Yes状态

2.5 MySQL主从同步数据复制模式

2.5.1 数据复制模式

异步复制模式（默认）

半同步复制模式

休息一下 到 16：52上课

2.5.2 配置半同步数据复制模式

查看是否允许加载模块

mysql> show variables like 'have\_dynamic\_loading';

+----------------------+-------+

| Variable\_name | Value |

+----------------------+-------+

| have\_dynamic\_loading | YES | 允许

+----------------------+-------+

1 row in set (0.02 sec)

mysql>

主服务器要启用半同步复制模式

加载master模块

启用master模块

命令行加载与启用

mysql> install plugin rpl\_semi\_sync\_master SONAME "semisync\_master.so";

mysql> select plugin\_name, plugin\_status from information\_schema.plugins where plugin\_name like '%semi%'; 查看模块是否安装成功

+----------------------+---------------+

| plugin\_name | plugin\_status |

+----------------------+---------------+

| rpl\_semi\_sync\_master | ACTIVE |

+----------------------+---------------+

1 row in set (0.00 sec)

mysql>

mysql> set global rpl\_semi\_sync\_master\_enabled=1;

mysql> show variables like "rpl\_semi\_sync\_%\_enabled"; 模块是否启用成功

+------------------------------+-------+

| Variable\_name | Value |

+------------------------------+-------+

| rpl\_semi\_sync\_master\_enabled | ON |

+------------------------------+-------+

1 row in set (0.00 sec)

永久配置/etc/my.cnf

[mysqld]

plugin-load=rpl\_semi\_sync\_master=semisync\_master.so

rpl\_semi\_sync\_master\_enabled=1

:wq

从服务器55要启用半同步复制模式

加载slave模块

启用slave模块

mysql> install plugin rpl\_semi\_sync\_slave SONAME "semisync\_slave.so";

Query OK, 0 rows affected (0.04 sec)

mysql> set global rpl\_semi\_sync\_slave\_enabled=1;

Query OK, 0 rows affected (0.00 sec)

mysql> select plugin\_name, plugin\_status from information\_schema.plugins where plugin\_name like '%semi%';

+---------------------+---------------+

| plugin\_name | plugin\_status |

+---------------------+---------------+

| rpl\_semi\_sync\_slave | ACTIVE |

+---------------------+---------------+

1 row in set (0.01 sec)

mysql> show variables like "rpl\_semi\_sync\_%\_enabled"

-> ;

+-----------------------------+-------+

| Variable\_name | Value |

+-----------------------------+-------+

| rpl\_semi\_sync\_slave\_enabled | ON |

+-----------------------------+-------+

1 row in set (0.05 sec)

mysql>

永久配置 vim /etc/my.cnf

[mysqld]

plugin-load=rpl\_semi\_sync\_slave=semisync\_slave.so

rpl\_semi\_sync\_slave\_enabled=1

:wq

192.168.4.54 主机既是主 又是 从 ，2种类型的模块都要安装和启用

安装模块

mysql> install plugin rpl\_semi\_sync\_slave SONAME "semisync\_slave.so";

mysql> install plugin rpl\_semi\_sync\_master SONAME "semisync\_master.so";

启用模块

mysql> set global rpl\_semi\_sync\_slave\_enabled=1;

mysql> set global rpl\_semi\_sync\_master\_enabled=1;

永久配置 vim /etc/my.cnf

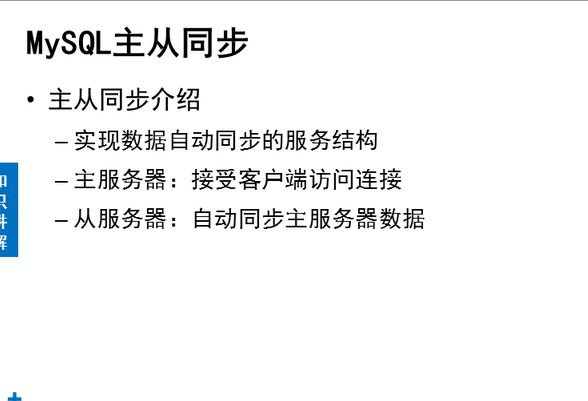
[mysqld]

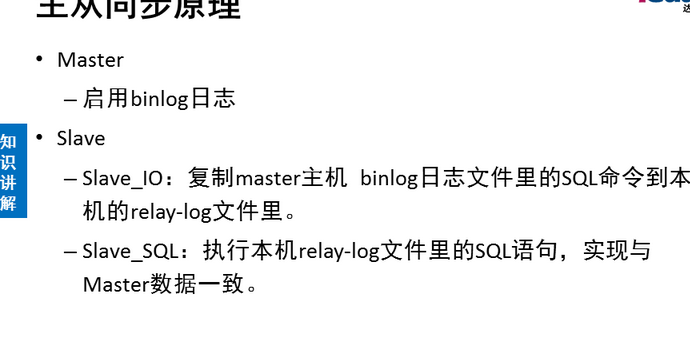
plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl\_semi\_sync\_master\_enabled=1

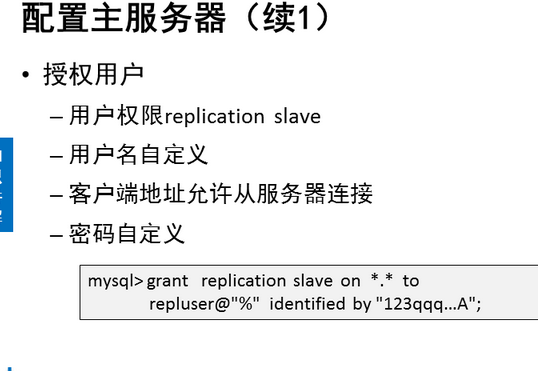
rpl\_semi\_sync\_slave\_enabled=1

:wq

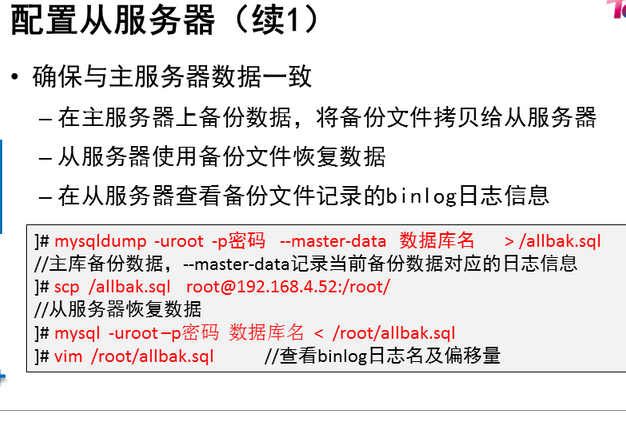


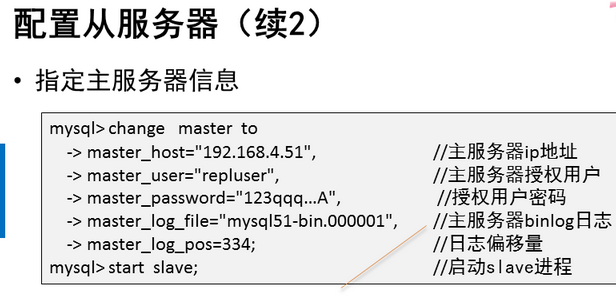


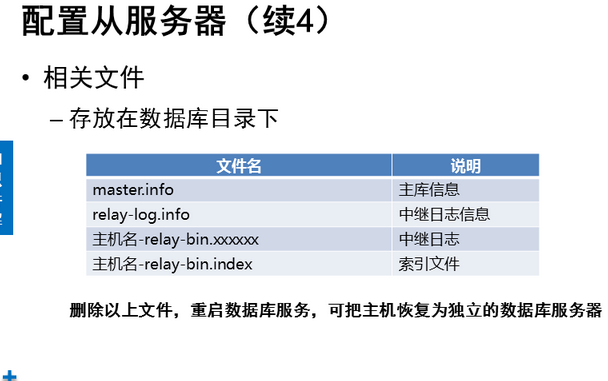


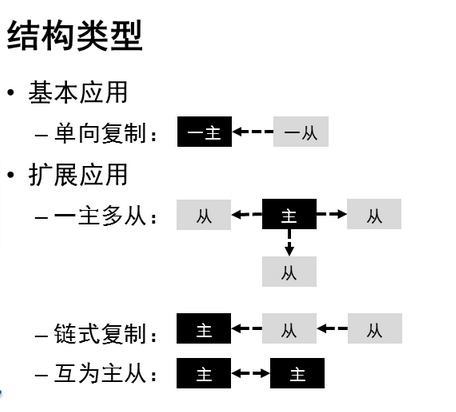


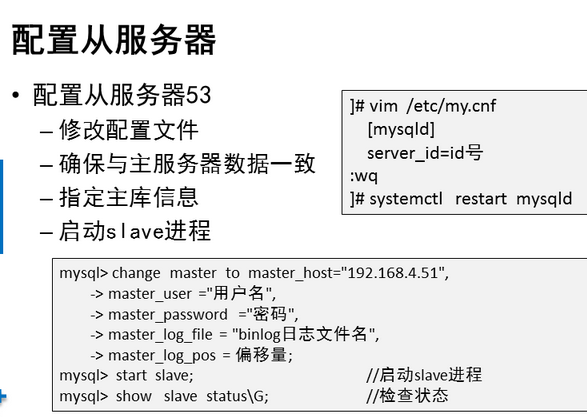


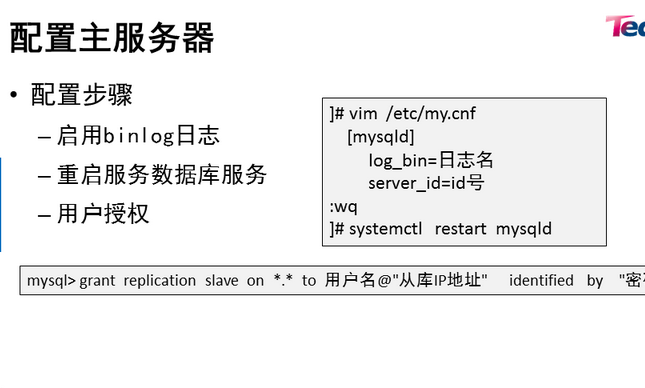


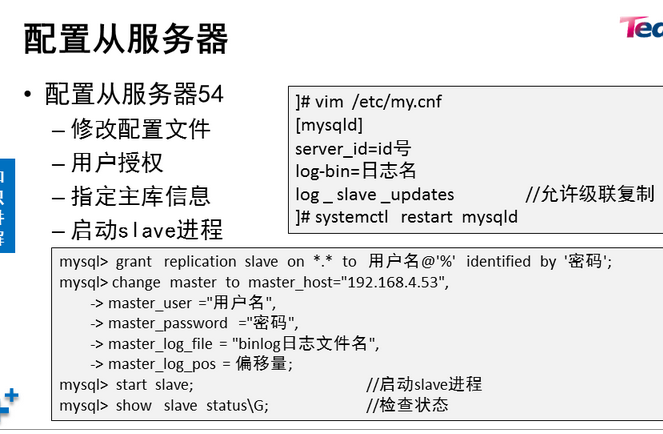


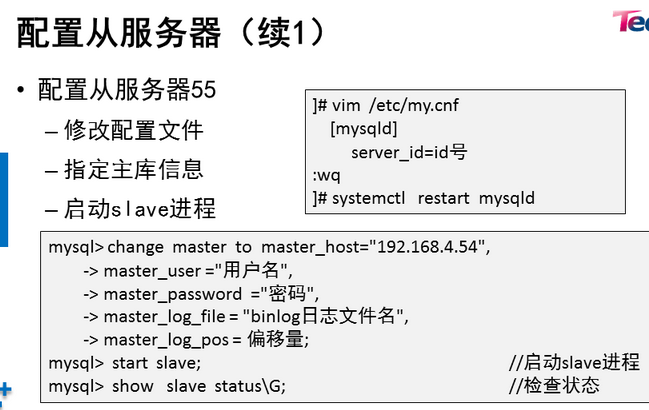


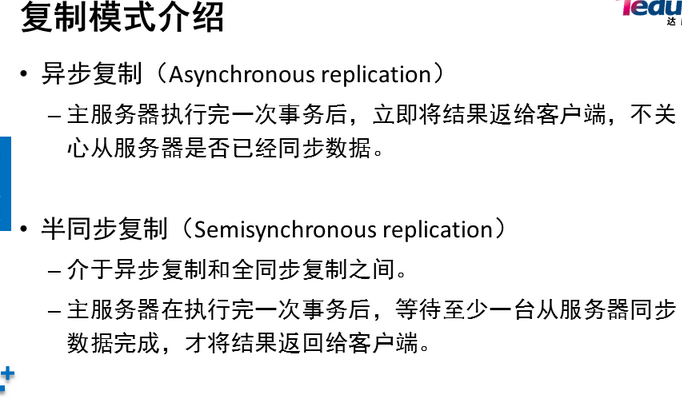


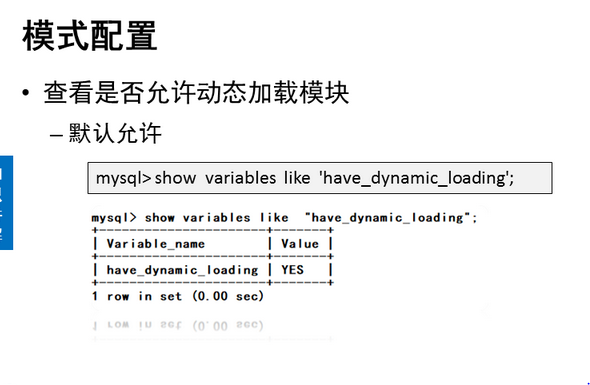


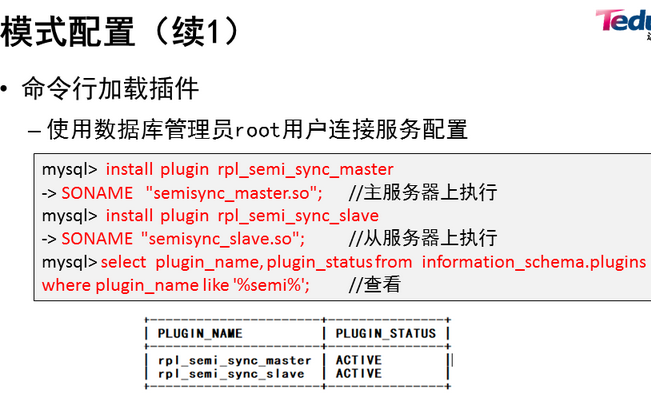


















**NSD RDBMS2 DAY01**

1. [案例1：MySQL一主一从](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS2/DAY01/CASE/01/index.html#case1)
2. [案例2：配置一主多从结构](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS2/DAY01/CASE/01/index.html#case2)
3. [案例3：配置主从从结构](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS2/DAY01/CASE/01/index.html#case3)
4. [案例4：配置半同步复制模式](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS2/DAY01/CASE/01/index.html#case4)

**1 案例1：MySQL一主一从**

**1.1 问题**

* 数据库服务器192.168.4.51配置为主数据库服务器
* 数据库服务器192.168.4.52配置为从数据库服务器
* 客户端192.168.4.50测试配置

**1.2 方案**

使用2台虚拟机，如图-1所示。其中192.168.4.51是主服务器,另一台192.168.4.52作为从服务器，通过调取主服务器上的binlog日志，在本地重做对应的库、表，实现与主服务器的数据同步。

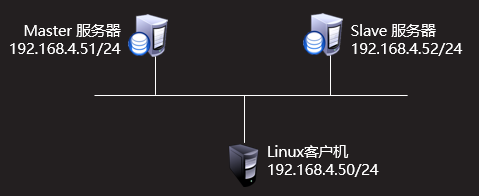


图-1

主机51和主机52分别运行MySQL数据库服务，且管理员root用户可以本机登录；主机50作为客户机 只需有命令行连接命令mysql即可。

**1.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：配置主服务器192.168.4.51

1）启用binlog日志

1. ]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=51            //server\_id
4. log-bin=master51        //日志名
5. :wq
6. ]# systemctl restart mysqld

2）用户授权

用户名自定义、客户端地址使用% 或 只指定 从服务器的地址 都可以、只给复制数据的权限即可。

1. ]# mysql -uroot -p密碼
2. mysql> grant replication slave on \*.\* to repluser@"%" identified by "123qqq...A";
3. mysql>quit;

3）查看binlog日志信息

查看日志文件名 和 偏移量位置。

mysql> show master status\G;

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

File: master51.000001 //日志名

Position: 441 //偏移量

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

步骤二：配置从服务器192.168.4.52

1）指定server\_id

Server\_id值可以自定义，但不可以与主服务器相同。

1. ]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=52 //server\_id值
4. :wq
5. ]# systemctl restart mysqld //重启服务

2）确保与主服务器数据一致（如果是使用2台新部署的数据库服务器配置主从同步，此操作可以忽略）

1. ]# mysqldump -uroot –p密码 --master-data 数据库名 > /allbak.sql //在主服务器上备份数据
2. ]# scp /allbak.sql root@192.168.4.52:/root/ //将备份文件拷贝给从服务器
3. mysql> create database 数据库名 ; //在从服务器上创建与主服务器同名的数据库
4. ]# mysql -uroot –p密码 数据库名 < /root/allbak.sql //从服务器使用备份文件恢复数据
5. ]# vim /root/allbak.sql //在从服务器查看备份文件中的binlog日志信息
6. ......
7. ......
8. CHANGE MASTER TO MASTER\_LOG\_FILE='master51.000001', MASTER\_LOG\_POS=441; //日志名与偏移量

3）指定主服务器信息

数据库管理员root本机登录，指定主服务器信息，其中日志文件名和偏移量 写allbak.sql文件记录的。

1. ]# mysql -uroot –p密码 //管理员root 本机登录
2. mysql> show slave status; //查看状态信息，还不是从服务器
3. Empty set (0.00 sec)
4. mysql> change master to //指定主服务器
5. -> master\_host=“192.168.4.51”, //主服务器ip地址
6. -> master\_user=“repluser”, //主服务器授权用户
7. -> master\_password=“123qqq…A”, //主服务器授权用户密码
8. -> master\_log\_file=“master51-bin.000001”,//主服务器日志文件
9. -> master\_log\_pos=441;          //主服务器日志偏移量
10. mysql> start slave;
11. mysql> show slave status\G; //查看状态信息
12. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
13. Slave\_IO\_State: Waiting for master to send event
14. Master\_Host: 192.168.4.51 //主服务器ip地址
15. Master\_User: repluser
16. Master\_Port: 3306
17. Connect\_Retry: 60
18. Master\_Log\_File: master51.000001
19. Read\_Master\_Log\_Pos: 437
20. Relay\_Log\_File: host52relay-bin.000002
21. Relay\_Log\_Pos: 604
22. Relay\_Master\_Log\_File: master51.000001
23. Slave\_IO\_Running: Yes        //IO线程yes状态
24. Slave\_SQL\_Running: Yes        //SQL线程yes状态
25. Replicate\_Do\_DB:
26. Replicate\_Ignore\_DB:
27. Replicate\_Do\_Table:
28. Replicate\_Ignore\_Table:
29. Replicate\_Wild\_Do\_Table:
30. Replicate\_Wild\_Ignore\_Table:
31. Last\_Errno: 0
32. Last\_Error:
33. Skip\_Counter: 0
34. Exec\_Master\_Log\_Pos: 437
35. Relay\_Log\_Space: 812
36. Until\_Condition: None
37. Until\_Log\_File:
38. Until\_Log\_Pos: 0
39. Master\_SSL\_Allowed: No
40. Master\_SSL\_CA\_File:
41. Master\_SSL\_CA\_Path:
42. Master\_SSL\_Cert:
43. Master\_SSL\_Cipher:
44. Master\_SSL\_Key:
45. Seconds\_Behind\_Master: 0
46. Master\_SSL\_Verify\_Server\_Cert: No
47. Last\_IO\_Errno: 0
48. Last\_IO\_Error:
49. Last\_SQL\_Errno: 0
50. Last\_SQL\_Error:
51. Replicate\_Ignore\_Server\_Ids:
52. Master\_Server\_Id: 50
53. Master\_UUID: 4881ee4b-8800-11e9-830a-525400001e32
54. Master\_Info\_File: /var/lib/mysql/master.info
55. SQL\_Delay: 0
56. SQL\_Remaining\_Delay: NULL
57. Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates
58. Master\_Retry\_Count: 86400
59. Master\_Bind:
60. Last\_IO\_Error\_Timestamp:
61. Last\_SQL\_Error\_Timestamp:
62. Master\_SSL\_Crl:
63. Master\_SSL\_Crlpath:
64. Retrieved\_Gtid\_Set:
65. Executed\_Gtid\_Set:
66. Auto\_Position: 0
67. Replicate\_Rewrite\_DB:
68. Channel\_Name:
69. Master\_TLS\_Version:
70. 1 row in set (0.00 sec)

步骤三：客户端测试配置

1）在主服务器添加访问数据的连接用户

授权用户对所有数据有增删改查的权限即可

1. ]# mysql –uroot –p密码
2. mysql> grant select,insert,update,delete on \*.\* to admin@"%" identified by "123qqq...A";
3. Query OK, 0 rows affected, 1 warning (0.03 sec)
4. mysql> quit

2）客户端连接主服务器访问数据

在50主机 使用主服务器51的授权用户连接

1. ]# mysql -h192.168.4.51-uadmin -p123qqq...A
2. mysql> show grants;
3. +------------------------------------------------------------+
4. | Grants for admin@% |
5. +------------------------------------------------------------+
6. | GRANT SELECT, INSERT, UPDATE, DELETE ON \*.\* TO 'admin'@'%' |
7. +------------------------------------------------------------+
8. 1 row in set (0.00 sec)
9. mysql> insert into db3.user(name,uid) values("lili",288); //db3库和user表是主从同步之前主服务器已有的。
10. Query OK, 1 row affected (0.05 sec)
11. mysql> insert into db3.user(name,uid) values("lili",288);
12. Query OK, 1 row affected (0.28 sec)
13. mysql> insert into db3.user(name,uid) values("lili",288);
14. Query OK, 1 row affected (0.05 sec)
15. mysql> select name,uid from db3.user where name="lili";
16. +------+------+
17. | name | uid |
18. +------+------+
19. | lili | 288 |
20. | lili | 288 |
21. | lili | 288 |
22. +------+------+
23. 3 rows in set (0.00 sec)

3) 客户端连接从服务器访问数据

客户端50主机使用授权用户连接从服务器可以看到和主服务器同样的数据

1. ]# mysql -h192.168.4.52 –uadmin -p123qqq…A
2. mysql> select name,uid from db3.user where name="lili";
3. +------+------+
4. | name | uid |
5. +------+------+
6. | lili | 288 |
7. | lili | 288 |
8. | lili | 288 |
9. +------+------+
10. 3 rows in set (0.00 sec)

**2 案例2：配置一主多从结构**

**2.1 问题**

* 配置192.168.4.53数据库服务器为主机192.168.4.51的从服务器
* 客户端测试配置。

**2.2 方案**

创建1台新虚拟机，配置ip地址为192.168.4.53、运行数据库服务，且数据库管理员root用户可以本机登录。具体结构，如图-2所示。

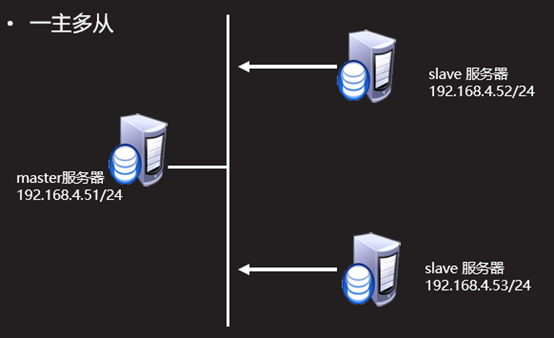


图-2

**2.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：配置从服务器192.168.4.53

1）启用binlog日志

1. ]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=53        //server\_id
4. :wq
5. ]# systemctl restart mysqld

2）确保与主服务器数据一致

在主服务器51 备份所有数据 并把备份文件拷贝给53服务器。

1. ]# mysqldump -uroot –p123qqq…A –-master-data –B db4 db3 > /root/twodb.sql
2. ]# scp /root/twodb.sql root@192.168.4.53:/root/

在53主机使用备份文件恢复数据，并查看备份文件记录的日志名和偏移量

1. ]# mysql -uroot –p123qqq…A < /root/twodb.sql
2. ]# grep mater51 /root/twodb.sql
3. CHANGE MASTER TO MASTER\_LOG\_FILE='master51.000001', MASTER\_LOG\_POS=1098; //日志名与偏移量

3）指定主服务器信息

填写备份文件里显示的日志文件名 和 偏移量位置。

1. mysql> change master to //指定主服务器
2. -> master\_host=“192.168.4.51”, //主服务器ip地址
3. -> master\_user=“repluser”, //主服务器授权用户
4. -> master\_password=“123qqq…A”, //主服务器授权用户密码
5. -> master\_log\_file=“master51-bin.000001”,//主服务器日志文件
6. -> master\_log\_pos=1098;          //主服务器日志偏移量
7. mysql> start slave;

查看状态信息

1. ]# mysql –uroot –p123qqq…A –e “show slave status\G” | grep –i yes
2. Slave\_IO\_Running: Yes        //IO线程yes状态
3. Slave\_SQL\_Running: Yes        //SQL线程yes状态
4. ]# mysql –uroot –p123qqq…A –e “show slave status\G” | grep –i “master\_host”
5. Master\_Host: 192.168.4.51 //主服务器ip地址

步骤二：客户端测试（192.168.4.50）

1）连接主服务器插入新记录

1. ]# mysql -h192.168.4.51-uadmin -p123qqq...A
2. mysql> insert into db3.user(name,uid) values("lucy",888); //db3库和user表是主从同步之前主服务器已有的。
3. Query OK, 1 row affected (0.05 sec)

2）在从服务器本机53 可以查询到新插入的数据

1. ]# mysql -uroot –p123qqq…A
2. mysql> select name,uid from db3.user;
3. +------+------+
4. | name | uid |
5. +------+------+
6. | lili | 288 |
7. | lucy | 888 |
8. +------+------+
9. 2 rows in set (0.00 sec)

**3 案例3：配置主从从结构**

**3.1 问题**

* 配置主机192.168.4.53为主服务器
* 配置主机192.168.4.54为53主机的从服务器
* 配置主机192.168.4.55为54主机的从服务器
* 客户端测试配置。

**3.2 方案**

使用3台虚拟机，分别运行mysql数据库服务，且管理员root可以本机登录；主机192.168.4.53为主服务器;主机192.168.4.54为从服务器；主机192.168.4.55为从服务器。如图-3所示。

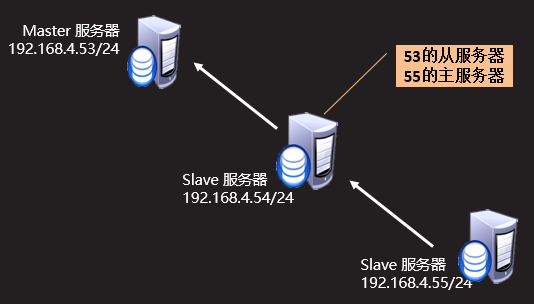


图-3

**3.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：环境准备

为了在启用binlog日志及同步之前保持主、从库的一致性，主从同步未配置之前，要保证从库上要有主库上的数据，禁用selinux，关闭防火墙服务，保证物理连接正常

1）关闭防火墙，禁用selinux，已关可忽略：

1. ]# systemctl stop firewalld
2. ]# setenforce 0

步骤二：配置主服务器192.168.4.53

2）用户授权

1. ]# mysql -uroot -p123456
2. mysql> grant replication slave on \*.\* to yaya@"%" identified by "123qqq…A“；
3. Query OK, 0 rows affected, 1 warning (0.03 sec)

3）启用binlog日志，修改/etc/my.cnf配置，重新启动MySQL服务程序

指定服务器ID号、允许日志同步：

1. ]# vim /etc/my.cnf
2. [mysqld]
3. log\_bin=db53                     //启用binlog日志，并指定文件名前缀
4. server\_id=53                 //指定服务器ID号

4）重启mysql服务：

1. ]# systemctl restart mysqld

5）确保/var/lib/mysql下面有两个文件：

1. ]# ls /var/lib/mysql/db51.\*
2. /var/lib/mysql/db53.000001 /var/lib/mysql/db53.index

6）查看主服务正在使用的日志信息

查看主服务器状态，记录下当前的日志文件名、偏移的位置（下面SLAVE发起复制时需要用到）：

1. mysql> show master status;
2. +-------------+----------+--------------+------------------+-------------------+
3. | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
4. +-------------+----------+--------------+------------------+-------------------+
5. | db53.000001 |437 | | | |
6. +-------------+----------+--------------+------------------+-------------------+
7. 1 row in set (0.00 sec)

步骤三：配置从服务器192.168.4.54

1）在服务器192.168.4.53上做用户授权（数据同步使用的连接用户）

1. ]# mysql -u root -p123456
2. mysql> grant replication slave on \*.\* to user55@”%” identified by “654321” ;

2）修改/etc/my.cnf配置，启用binlog日志，指定server\_id 和 允许级联复制

1. ]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=54
4. log-bin=db54
5. log\_slave\_updates //允许级联复制

3）配置完成后，重启mysql服务：

1. ]# systemctl restart mysqld

4）确保/var/lib/mysql下面有两个文件：

1. ]# ls /var/lib/mysql/db52.\*
2. /var/lib/mysql/db54.000001 /var/lib/mysql/db54.index

5）查看正在使用的日志信息

1. ]# mysql -uroot -p123456
2. mysql> show master status;
3. +-------------+----------+--------------+------------------+-------------------+
4. | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
5. +-------------+----------+--------------+------------------+-------------------+
6. |db54.000001 | 154 |
7. +-------------+----------+--------------+------------------+-------------------+
8. 1 row in set (0.00 sec) //查看日志文件名、偏移的位置

6）验证主服务器的的授权用户

]# mysql -h192.168.4.53 -uyaya -p123456

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: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, 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.

1. mysql> //验证成功

7）通过change master语句指定master服务器的IP地址、同步用户名/密码、起始日志文件、偏移位置（参考master上的状态输出）：

]# mysql -uroot -p123456

mysql> change master to

-> master\_host="192.168.4.53”;

-> master\_user="yaya",

-> master\_password="123456",

-> master\_log\_file="db53.000001”;

-> master\_log\_pos=437;

Query OK, 0 rows affected, 2 warnings (0.43 sec)

8）启动slave进程

mysql> start slave;

Query OK, 0 rows affected (0.03 sec)

9）查看进程状态信息，通过show slave status语句可查看从服务器状态，确认其中的IO线程、SQL线程正常运行，才能成功同步,IO线程和SQL线程必须是Yes

mysql> show slave status \G;

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

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

1. Master\_Host: 192.168.4.53 //主服务器IP地址

Master\_User: yaya

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: db53.000001

Read\_Master\_Log\_Pos: 437

Relay\_Log\_File: db54-relay-bin.000001

Relay\_Log\_Pos: 315

Relay\_Master\_Log\_File: db54.000001

1. Slave\_IO\_Running: Yes    //IO线程状态YES
2. Slave\_SQL\_Running: Yes //SQL线程状态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: 437

Relay\_Log\_Space: 521

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: 51

Master\_UUID: 81a13101-aa66-11e8-ad11-525400019e62

Master\_Info\_File: /var/lib/mysql/master.info

SQL\_Delay: 0

SQL\_Remaining\_Delay: NULL

Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates

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

Replicate\_Rewrite\_DB:

Channel\_Name:

Master\_TLS\_Version:

1 row in set (0.00 sec)

步骤四：配置从服务器192.168.4.55

1）验证主库的授权用户

1. ]# mysql -h192.168.4.54 -uuser54 -p654321
2. mysql: [Warning] Using a password on the command line interface can be insecure.
3. Welcome to the MySQL monitor. Commands end with ; or \g.
4. Your MySQL connection id is 7
5. Server version: 5.7.17-log MySQL Community Server (GPL)
6. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
7. Oracle is a registered trademark of Oracle Corporation and/or its
8. affiliates. Other names may be trademarks of their respective
9. owners.
10. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
11. mysql> //验证成功

2）指定server\_id

1. [mysqld]
2. server\_id=55

3）重新启动服务

1. ]# systemctl restart mysqld

4）管理员登录指定主库信息

1. ]# mysql -uroot -p123456
2. mysql> change master to
3. -> master\_host="192.168.4.54”;
4. -> master\_user="user55”;
5. -> master\_password="654321",
6. -> master\_log\_file=" db54.000001”;
7. -> master\_log\_pos=154;
8. Query OK, 0 rows affected, 2 warnings (0.37 sec)

5）启动slave进程

1. mysql> start slave;
2. Query OK, 0 rows affected (0.04 sec)

6）查看进程状态信息

1. mysql> show slave status\G
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Slave\_IO\_State: Waiting for master to send event
4. Master\_Host: 192.168.4.54
5. Master\_User: user55
6. Master\_Port: 3306
7. Connect\_Retry: 60
8. Master\_Log\_File: db54.000001
9. Read\_Master\_Log\_Pos: 154
10. Relay\_Log\_File: db55-relay-bin.000001
11. Relay\_Log\_Pos: 315
12. Relay\_Master\_Log\_File: db54.000001
13. Slave\_IO\_Running: Yes
14. Slave\_SQL\_Running: Yes

步骤五：客户端验证配置

在主库授权访问数据的连接用户;户端连接主库执行与权限匹配的sql操作;

授权用户连接第1台从库，可以看到主库的数据;权用户连接第2台从库，可以看到主库的数据

1）在主服务器上在主库上授权访问gamedb库的用户

1. ]# mysql -uroot -p123456
2. mysql> grant all on gamedb.\* to dada@"%" identified by "123456";
3. Query OK, 0 rows affected, 1 warning (0.03 sec)

2）客户端使用授权用户连接主库，建库、表、插入记录

1. ]# mysql -h192.168.4.53 -udada -p123456
2. Welcome to the MariaDB monitor. Commands end with ; or \g.
3. Your MySQL connection id is 7
4. Server version: 5.7.17-log MySQL Community Server (GPL)
5. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
6. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
7. MySQL [(none)]> //验证成功
8. MySQL [(none)]> create database gamedb; //创建测试库
9. Query OK, 1 row affected (0.04 sec)
10. MySQL [(none)]> create table gamedb.t1(id int); //在gamedb下创建t1表
11. Query OK, 0 rows affected (0.17 sec)
12. MySQL [(none)]> insert into gamedb.t1 values(8888); //在t1表中插入数值
13. Query OK, 1 row affected (0.22 sec)

3）客户端使用授权用户连接2台从库时，也可以看到主库上新的库表记录

1. ]# mysql -h192.168.4.54 -udada -p123456 //验证54主机的状态
2. Welcome to the MariaDB monitor. Commands end with ; or \g.
3. Your MySQL connection id is 10
4. Server version: 5.7.17-log MySQL Community Server (GPL)
5. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
6. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
7. MySQL [(none)]> select \* from gamedb.t1; //查询插入的表格
8. +------+
9. | id |
10. +------+
11. | 8888 |
12. +------+
13. 1 row in set (0.00 sec)
14. MySQL [(none)]> exit
15. [root@room9pc01 ~]# mysql -h192.168.4.55 -udada -p123456 //验证55主机的状态
16. Welcome to the MariaDB monitor. Commands end with ; or \g.
17. Your MySQL connection id is 6
18. Server version: 5.7.17 MySQL Community Server (GPL)
19. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
20. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
21. MySQL [(none)]> select \* from gamedb.t1;
22. +------+
23. | id |
24. +------+
25. | 8888 |
26. +------+
27. 1 row in set (0.00 sec)

**4 案例4：配置半同步复制模式**

**4.1 问题**

* 开启案例2 从服务器192.168.4.54 半同步复制模式
* 查看半同步复制模式是否开启

**4.2 方案**

以案例2从服务器192.168.4.54 为例演示配置，54主机既做主服务器又做从服务器，所以两种角色的半同步复制模块和功能都要启用。

**4.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：查看是否允许动态加载模块。

1）查看是否允许动态加载模块（默认允许）

1. mysql> show variables like 'have\_dynamic\_loading';
2. +----------------------+-------+
3. | Variable\_name | Value |
4. +----------------------+-------+
5. | have\_dynamic\_loading | YES |
6. +----------------------+-------+
7. 1 row in set (0.01 sec)

2）命令行加载插件

1. mysql> install plugin rpl\_semi\_sync\_master SONAME ”semisync\_master.so”; //加载master模块
2. mysql> install plugin rpl\_semi\_sync\_slave SONAME 'semisync\_slave.so'; //加载slave模块

查看模块是否安装成功：

1. mysql> select plugin\_name, plugin\_status from information\_schema.plugins where plugin\_name like '%semi%';
2. +----------------------+---------------+
3. | PLUGIN\_NAME | PLUGIN\_STATUS |
4. +----------------------+---------------+
5. | rpl\_semi\_sync\_master | ACTIVE |        //模块安装成功
6. | rpl\_semi\_sync\_slave | ACTIVE |
7. +----------------------+---------------+
8. 2 rows in set (0.00 sec)

3）启用半同步复制 （在安装完插件后，半同步复制默认是关闭的）

1. mysql> set global rpl\_semi\_sync\_master\_enabled = 1; //启用master半同步复制
2. Query OK, 0 rows affected (0.00 sec)
3. mysql> set global rpl\_semi\_sync\_slave\_enabled = 1; //启用slave半同步复制
4. Query OK, 0 rows affected (0.00 sec)

查看半同步复制模式是否启用：

1. mysql> show variables like "rpl\_semi\_sync\_%\_enabled";
2. +------------------------------+-------+
3. | Variable\_name | Value |
4. +------------------------------+-------+
5. | rpl\_semi\_sync\_master\_enabled | ON |    //模块已启用
6. | rpl\_semi\_sync\_slave\_enabled | ON |
7. +------------------------------+-------+
8. 2 rows in set (0.00 sec)

4）修改配置文件，永久启用半同步复制

1. [root@master51 ~]# vim /etc/my.cnf
2. [mysqld]
3. plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"
4. rpl-semi-sync-master-enabled = 1
5. rpl-semi-sync-slave-enabled = 1
6. ：wq

5）重启数据库服务，并查看状态信息

1. ]# mystemctl restart mysqld
2. ]# mysql -uroot -p123qqq...A
3. mysql> select plugin\_name, plugin\_status from information\_schema.plugins where plugin\_name like '%semi%';
4. +----------------------+---------------+
5. | plugin\_name | plugin\_status |
6. +----------------------+---------------+
7. | rpl\_semi\_sync\_master | ACTIVE | //模块已加载
8. | rpl\_semi\_sync\_slave | ACTIVE |
9. +----------------------+---------------+
10. 2 rows in set (0.00 sec)
11. mysql> show variables like "rpl\_semi\_sync\_%\_enabled";
12. +------------------------------+-------+
13. | Variable\_name | Value |
14. +------------------------------+-------+
15. | rpl\_semi\_sync\_master\_enabled | ON | //模式已启用
16. | rpl\_semi\_sync\_slave\_enabled | ON |
17. +------------------------------+-------+
18. 2 rows in set (0.00 sec)