

TTS 11.0 COOKBOOK

(NSD RDBMS2 DAY01)

版本编号 11.0

2019-06

达内 IT 培训集团

NSD RDBMS2 DAY01

1. 案例 1: MySQL 一主一从

- 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

- 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。

- 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

```
]# vim /etc/my.cnf
[mysqld]
server_id=51           //server_id
log-bin=master51       //日志名
:wq
]# 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。

```
]# mysql -uroot -p 密码
mysql> grant replication slave on *.* to repluser@"%" identified by "123qqq...A";
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。

```
]# vim /etc/my.cnf
[mysqld]
server_id=52 //server_id 值
:wq
]# 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）

```
]# mysqldump -uroot -p 密码 --master-data 数据库名 > /allbak.sql //在主服务器上备份数据
]# scp /allbak.sql root@192.168.4.52:/root/ //将备份文件拷贝给从服务器
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
]# 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
]# vim /root/allbak.sql //在从服务器查看备份文件中的 binlog 日志信息
.....
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。

```
]# mysql -uroot -p 密码 //管理员 root 本机登录
mysql> show slave status; //查看状态信息，还不是从服务器
Empty set (0.00 sec)

mysql> change master to //指定主服务器
-> master_host= "192.168.4.51" , //主服务器 ip 地址
-> master_user= "repluser" , //主服务器授权用户
-> master_password= "123qqq...A" , //主服务器授权用户密码
-> master_log_file= "master51-bin.000001" , //主服务器日志文件
-> master_log_pos=441; //主服务器日志偏移量

mysql> show slave status\G; //查看状态信息
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.4.51 //主服务器 ip 地址
Master_User: repluser
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: master51.000001
Read_Master_Log_Pos: 437
Relay_Log_File: host52relay-bin.000002
Relay_Log_Pos: 604
Relay_Master_Log_File: master51.000001
Slave_IO_Running: Yes //IO 线程 yes 状态
Slave_SQL_Running: Yes //SQL 线程 yes 状态
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: 812
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: 50
Master_UUID: 4881ee4b-8800-11e9-830a-525400001e32
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

```

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

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

```

]# mysql -h192.168.4.51-uadmin -p123qqq...A
mysql> show grants;
+-----+
| Grants for admin@% |
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'admin'@'%' |
+-----+
1 row in set (0.00 sec)

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。
Query OK, 1 row affected (0.05 sec)

mysql> insert into db3.user(name,uid) values("lili",288);
Query OK, 1 row affected (0.28 sec)

mysql> insert into db3.user(name,uid) values("lili",288);
Query OK, 1 row affected (0.05 sec)

mysql> select name,uid from db3.user where name="lili";
+-----+
| name | uid |
+-----+
| lili | 288 |
| lili | 288 |
| lili | 288 |
+-----+
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

```
]# mysql -h192.168.4.52 -uadmin -p123qqq...A
mysql> select name,uid from db3.user where name="lili";
+-----+-----+
| name | uid |
+-----+-----+
| lili | 288 |
| lili | 288 |
| lili | 288 |
+-----+-----+
3 rows in set (0.00 sec)
```

2. 案例 2：配置一主多从结构

• 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。

• 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-1 所示。

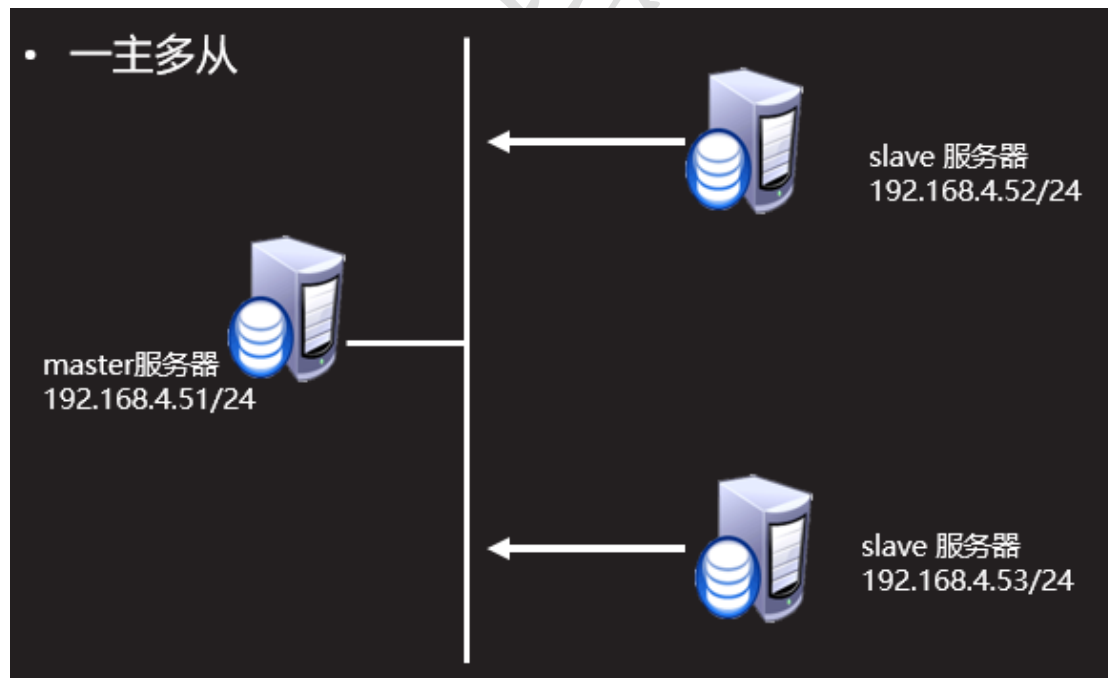


图 - 1

• 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

```
]# vim /etc/my.cnf
[mysqld]
server_id=53          //server_id
:wq
]# 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。

```
]# mysqldump -uroot -p123qqq...A --master-data -B db4 db3 > /root/twodb.sql
]# 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

```
]# mysql -uroot -p123qqq...A < /root/twodb.sql
]# grep mater51 /root/twodb.sql
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。

```
mysql> change master to //指定主服务器
-> master_host= "192.168.4.51" , //主服务器 ip 地址
-> master_user= "repluser" , //主服务器授权用户
-> master_password= "123qqq...A" , //主服务器授权用户密码
-> master_log_file= "master51-bin.000001" , //主服务器日志文件
-> master_log_pos=1098; //主服务器日志偏移量
mysql> start slave;
```

查看状态信息

```
]# mysql -uroot -p123qqq...A -e "show slave status\G" | grep -i yes
Slave_IO_Running: Yes //IO 线程 yes 状态
Slave_SQL_Running: Yes //SQL 线程 yes 状态
]# mysql -uroot -p123qqq...A -e "show slave status\G" | grep -i 192
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

```
]# mysql -h192.168.4.51-uadmin -p123qqq...A
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。
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

```
]# mysql -uroot -p123qqq..A
mysql> select name,uid from db3.user;
+-----+-----+
| name | uid |
+-----+-----+
| lili | 288 |
| lucy | 888 |
+-----+-----+
2 rows in set (0.00 sec)
```

3. 案例 3：配置主从从结构

- 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。

- 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-2 所示。

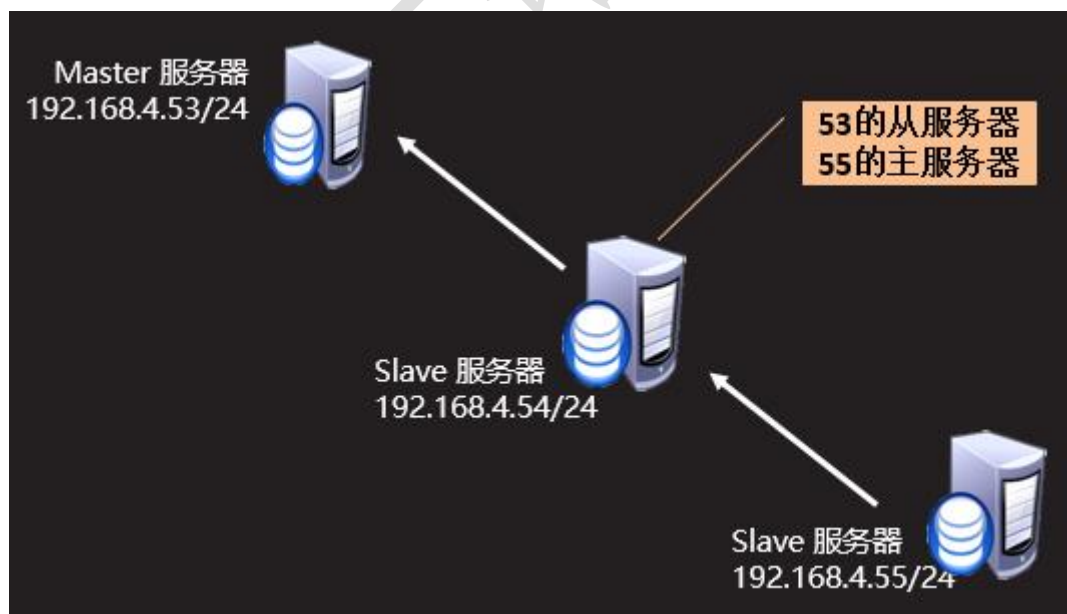


图 - 2

- 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：

```
]# systemctl stop firewalld
]# setenforce 0
```

步骤二：配置主服务器 192.168.4.53

1) 用户授权

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

2) 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：

```
]# vim /etc/my.cnf
[mysqld]
log_bin=db53                //启用 binlog 日志，并指定文件名前缀
server_id=53                //指定服务器 ID 号
```

重启 mysql 服务：

```
]# systemctl restart mysqld
```

确保/var/lib/mysql 下面有两个文件：

```
]# ls /var/lib/mysql/db51.*
/var/lib/mysql/db53.000001 /var/lib/mysql/db53.index
```

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）：

```
mysql> show master status;
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| db53.000001   | 437      |              |                  |                   |
+-----+-----+-----+-----+-----+
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）

```
]# mysql -u root -p123456
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

```
]# vim /etc/my.cnf
```

```
[mysqld]
server_id=54
log-bin=db54
log_slave_updates //允许级联复制
```

3) 配置完成后, 重启 mysql 服务:

```
]# systemctl restart mysqld
```

4) 确保/var/lib/mysql下面有两个文件:

```
]# ls /var/lib/mysql/db52.*
/var/lib/mysql/db54.000001 /var/lib/mysql/db54.index
```

5) 查看正在使用的日志信息

```
]# mysql -uroot -p123456
mysql> show master status;
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| db54.000001   | 154      |              |                  |                    |
+-----+-----+-----+-----+-----+
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.

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
        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
        Slave_IO_Running: Yes //IO 线程状态 YES
        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

```
]# mysql -h192.168.4.54 -uuser54 -p654321
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 7
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.

mysql> //验证成功
```

2) 指定 server_id

```
[mysqld]
server_id=55
```

3) 重新启动服务

```
]# systemctl restart mysqld
```

4) 管理员登录指定主库信息

```
]# mysql -uroot -p123456
mysql> change master to
-> master_host="192.168.4.54";
-> master_user="user55";
-> master_password="654321",
-> master_log_file=" db54.000001";
-> master_log_pos=154;
Query OK, 0 rows affected, 2 warnings (0.37 sec)
```

5) 启动 slave 进程

```
mysql> start slave;
Query OK, 0 rows affected (0.04 sec)
```

6) 查看进程状态信息

```
mysql> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.4.54
Master_User: user55
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: db54.000001
Read_Master_Log_Pos: 154
Relay_Log_File: db55-relay-bin.000001
Relay_Log_Pos: 315
Relay_Master_Log_File: db54.000001
Slave_IO_Running: Yes
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

```
]# mysql -uroot -p123456
mysql> grant all on gamedb.* to dada@%" identified by "123456";
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

```
]# mysql -h192.168.4.53 -udada -p123456
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> //验证成功
MySQL [(none)]> create database gamedb; //创建测试库
Query OK, 1 row affected (0.04 sec)
MySQL [(none)]> create table gamedb.t1(id int); //在 gamedb 下创建 t1 表
Query OK, 0 rows affected (0.17 sec)
MySQL [(none)]> insert into gamedb.t1 values(8888); //在 t1 表中插入数值
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

```
]# mysql -h192.168.4.54 -udada -p123456 //验证 54 主机的状态
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [(none)]> select * from gamedb.t1; //查询插入的表格
+-----+
| id |
+-----+
| 8888 |
+-----+
1 row in set (0.00 sec)
MySQL [(none)]> exit
[root@room9pc01 ~]# mysql -h192.168.4.55 -udada -p123456 //验证 55 主机的状态
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [(none)]> select * from gamedb.t1;
+-----+
| id |
+-----+
```

```
| 8888 |
+-----+
1 row in set (0.00 sec)
```

4. 配置半同步复制模式

• 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

• 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。

• 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）

```
mysql> show variables like 'have_dynamic_loading';
+-----+
| Variable_name | Value |
+-----+
| have_dynamic_loading | YES |
+-----+
1 row in set (0.01 sec)
```

2) 命令行加载插件

```
mysql> install plugin rpl_semi_sync_master SONAME "semisync_master.so"; //
加载 master 模块
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：

```
mysql> select plugin_name, plugin_status from information_schema.plugins where
plugin_name like '%semi%';
+-----+
| PLUGIN_NAME | PLUGIN_STATUS |
+-----+
| rpl_semi_sync_master | ACTIVE |
| rpl_semi_sync_slave | ACTIVE |
+-----+
2 rows in set (0.00 sec)
```

//模块安装成功

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)

```
mysql> set global rpl_semi_sync_master_enabled = 1; //启用 master 半同步复制
Query OK, 0 rows affected (0.00 sec)
mysql> set global rpl_semi_sync_slave_enabled = 1; //启用 slave 半同步复制
Query OK, 0 rows affected (0.00 sec)
```

查看半同步复制模式是否启用:

```
mysql> show variables like "rpl_semi_sync_%_enabled";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rpl_semi_sync_master_enabled | ON | //模块已启用
| rpl_semi_sync_slave_enabled | ON |
+-----+-----+
2 rows in set (0.00 sec)
```

4) 修改配置文件, 永久启用半同步复制

```
[root@master51 ~]# vim /etc/my.cnf
[mysqld]
plugin-load="rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisyn
c_slave.so"
rpl-semi-sync-master-enabled = 1
rpl-semi-sync-slave-enabled = 1
:wq
```

5) 重启数据库服务, 并查看状态信息

```
]# systemctl restart mysqld
]# mysql -uroot -p123qqq...A
mysql> select plugin_name, plugin_status from information_schema.plugins where
plugin_name like '%semi%';
+-----+-----+
| plugin_name | plugin_status |
+-----+-----+
| rpl_semi_sync_master | ACTIVE | //模块已加载
| rpl_semi_sync_slave | ACTIVE |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show variables like "rpl_semi_sync_%_enabled";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rpl_semi_sync_master_enabled | ON | //模式已启用
| rpl_semi_sync_slave_enabled | ON |
+-----+-----+
2 rows in set (0.00 sec)
```