在00\_linux下mysql安装的基础上，我们搭建mysql主从

环境

三台虚拟机：192.168.86.128（master），192.168.86.129（node1）, 192.168.86.130(node2)

将master做为主，将另外两个机器作为从服务器

服务启动

启动三台服务器的mysql

[root@master mysql]# ./bin/mysqld\_safe --defaults-file=./conf/my.cnf

设置同步用户

设置用户repl对数据库的操作权限

Master机器：

GRANT ALL PRIVILEGES ON \*.\* TO 'repl'@'192.168.86.130' IDENTIFIED BY 'repl' WITH GRANT OPTION;

Node1机器：

客户端连接mysql服务器，执行：

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'repl'@'192.168.86.128' IDENTIFIED BY 'repl' WITH GRANT OPTION;

Node2机器：

客户端连接mysql服务器，执行：

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'repl'@'192.168.86.128' IDENTIFIED BY 'repl' WITH GRANT OPTION;

注意：关闭防火墙，之后为了追求更安全和可用性可以配合相应的入站策略。

配置和重启

三台机器的配置，在[mysqld]下打开以下选项：

log-bin=mysql-bin

innodb\_flush\_log\_at\_trx\_commit = 1  
sync\_binlog=1

binlog-do-db=bookmark  
binlog-do-db=debate  
binlog-ignore-db=test  
binlog-ignore-db=mysql

replicate-do-db=bookmark  
replicate-do-db=debate

重启node1和node2

[root@node1 mysql]# ./bin/mysqld\_safe --defaults-file=./conf/my.cnf

[root@node2 mysql]# ./bin/mysqld\_safe --defaults-file=./conf/my.cnf

客户端连接node1，查看master的状态，执行命令：show master status;

mysql> show master status;

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

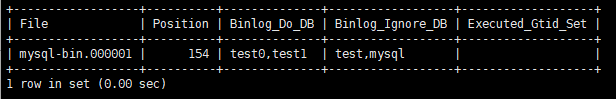
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000001 | 154 | test0,test1 | test,mysql | |

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

1 row in set (0.00 sec)



查看slave的状态，执行命令show slave status;

mysql> show slave status;

Empty set (0.00 sec)

------------------------

客户端连接node2, 查看master的状态，执行命令：show master status;

mysql> show master status;

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

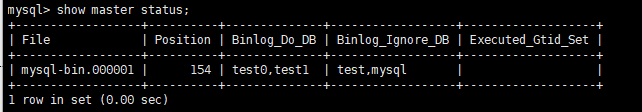
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000001 | 154 | test0,test1 | test,mysql | |

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

1 row in set (0.00 sec)



查看slave的状态，执行命令show slave status;

mysql> show slave status;

Empty set (0.00 sec)

此处empty说明slave没有连接上，可能是防火墙的原因

我们可以简单的将防火墙服务关闭。

关闭防洪墙

为了避免数据风险，最好保持master和node1、node2的数据一致，可以手动拷贝master的数据文件（需要复制的）到另外两台机器上，或者采取其他的方法。

执行以下操作：

在master上，锁定表从而使得不在对外提供服务，进而保持数据不会再变。锁表命令：

mysql> FLUSH TABLES WITH READ LOCK;

Query OK, 0 rows affected (0.02 sec)

查看主的状态，执行命令：show master status;

mysql> show master status;

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

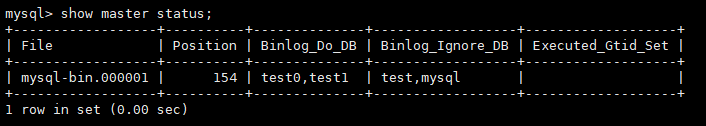
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000001 | 154 | test0,test1 | test,mysql | |

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

1 row in set (0.00 sec)



将需要同步的库的文件拷贝到两台两台服务器的data目录（如果存在同名文件则复制）

同步完数据后，启动两个从后，执行同步：

Node1上mysql客户端连接到服务器，执行命令：

mysql> CHANGE MASTER to MASTER\_HOST='192.168.86.128', MASTER\_PORT=3306, MASTER\_USER='repl', MASTER\_PASSWORD='repl', MASTER\_LOG\_FILE='master-bin.000001', MASTER\_LOG\_POS=154;

Query OK, 0 rows affected, 2 warnings (0.04 sec)

注意：MASTER\_LOG\_FILE为master端看到的File的值；MASTER\_LOG\_POS为master端看到的Posotion值。

然后执行命令，start slave;

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

node2上mysql客户端连接到服务器，执行同样的命令：

mysql> CHANGE MASTER to MASTER\_HOST='192.168.86.128', MASTER\_PORT=3306, MASTER\_USER='repl', MASTER\_PASSWORD='repl', MASTER\_LOG\_FILE='master-bin.000001', MASTER\_LOG\_POS=154;

Query OK, 0 rows affected, 2 warnings (0.04 sec)

mysql> start slave;

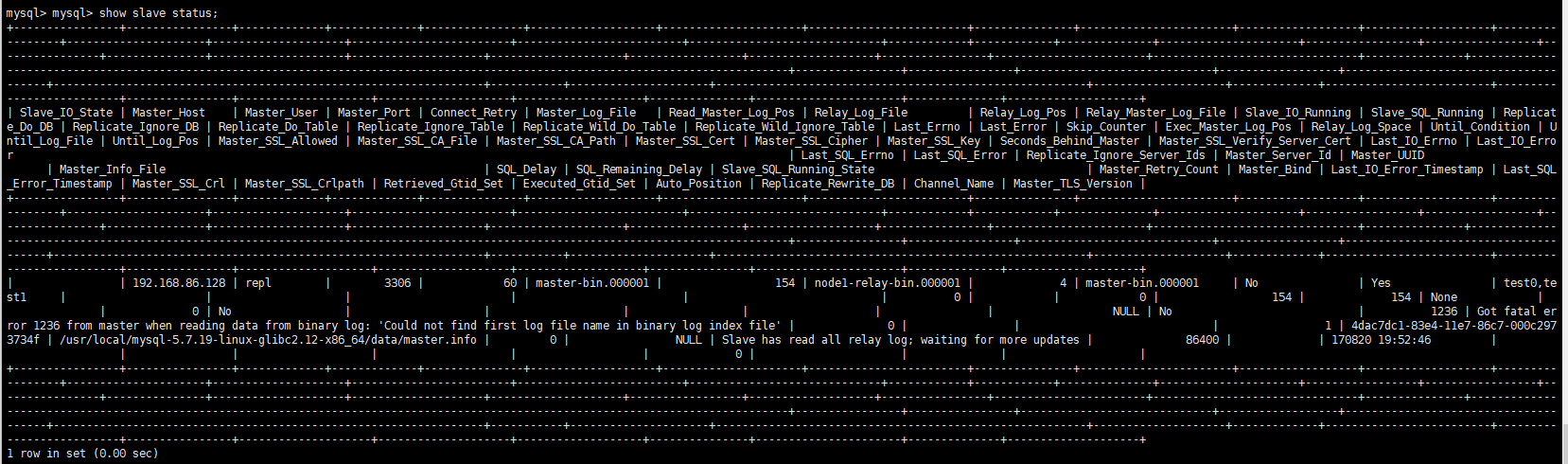
Query OK, 0 rows affected (0.00 sec)

在master上，解除锁表（如果之前执行了锁表并进行了数据同步的话）

在master端执行：unlock tables;

在node1和node2上查看slave的状态，执行show slave status;

mysql> mysql> show slave status;



不能看清，可以指定显示：

mysql> show slave status\G;

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

Slave\_IO\_State:

Master\_Host: 192.168.86.128 #主IP

Master\_User: repl #同步用户

Master\_Port: 3306 #主端口

Connect\_Retry: 60

Master\_Log\_File: master-bin.000001 #需要与master的show master status的File相同

Read\_Master\_Log\_Pos: 154 #需要与master的show master status 的Position相同

Relay\_Log\_File: node1-relay-bin.000001

Relay\_Log\_Pos: 4

Relay\_Master\_Log\_File: master-bin.000001

Slave\_IO\_Running: No #同步当前没有运行

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB: test0,test1 #需要同步的库

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: 154

Relay\_Log\_Space: 154

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: NULL

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 1236

Last\_IO\_Error: 0 #表示复制没有问题

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 1

Master\_UUID: 4dac7dc1-83e4-11e7-86c7-000c2973734f

Master\_Info\_File: /usr/local/mysql-5.7.19-linux-glibc2.12-x86\_64/data/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: 170820 19:52:46

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)

ERROR:

No query specified

测试集群的正确性：

Master上，在同步库test0和test1中创建表并插入数据

mysql> create database test0 charset=utf8;

Query OK, 1 row affected (0.00 sec)

mysql> use test0;

Database changed

mysql> create table person(

-> id int(11) unsigned auto\_increment,

-> name char(20) not null default '',

-> primary key(id)

-> )engine innodb charset=utf8;

Query OK, 0 rows affected (0.12 sec)

然后查看node1和node2,看数据是否同步。