CentOS 6.5 MySQL 5.6 MHA高可用测试笔记

主库失败测试

马飞

2018-1-27~2018-1-27

# 一、环境描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机名 | 操作系统 | IP地址 | 角色 | 数据库 |
| Manager | CentOS6.5 | 192.168.9.101 | Mha管理节点 |  |
| Master | CentOS6.5 | 192.168.9.102:3306 | Mha主节点 | MySQL5.6.22 |
| Slave | CentOS6.5 | 192.168.9.103:3306 | Mha备节点 | MySQL5.6.22 |
| Slave2 | CentOS6.5 | 192.168.9.104:3306 | Mha备节点 | MySQL5.6.22 |
|  |  | 192.168.9.120 | VIP |  |

主从复制相关用户：

|  |  |
| --- | --- |
| 用户名 | 口令 |
| mha | 123456 |
| myrpl | 123456 |

# 二、测试场景

* 主库正常关闭,mha是否进行故障转移并恢复失败主库
* 主库宕机,mha是否进行故障转移并恢复失败主库

# 三、测试日志

3.1 主库正常关闭服务，mha是否进行故障转移

### 3.1.1 实验环境

|  |  |
| --- | --- |
| 主库 | 192.168.9.102/maser |
| 从库 | 192.168.9.103/slave，192.168.9.104/slave2 |
| VIP | 192.168.9.120（在主库上） |

### 3.1.2 实验步骤

|  |  |  |
| --- | --- | --- |
| 序号 | 操作节点 | 操作内容描述 |
| 1 | 主库，从库 | 查询主库、从库UUID及状态信息 |
| 2 | 主库：192.168.9.102 | 正常关闭主库  /usr/local/mysql/mysqladmin --login-path=mysql3306 shutdown |
| 3 | 主库：192.168.9.102/master  从库：192.168.9.103/slave  192.168.9.104/slave2 | 查看mha是否进行了故转移操作  新主库：192.168.9.103/slave  从库：192.168.9.104/slave2  故障主库：192.168.9.102/master，已移出mha集群中。 |
| 4 | 失败主库：192.168.9.102 | 重建主从关系 |

### 3.1.3 操作步骤

#### （1）查看主库master

|  |
| --- |
| #查看主库UUID  mysql> show global variables like 'server\_uuid';  +---------------+--------------------------------------+  | Variable\_name | Value |  +---------------+--------------------------------------+  | server\_uuid | 92f6e6cb-f84f-11e7-ab62-000c291dadaa |  +---------------+--------------------------------------+  1 row in set (0.00 sec)  #查看当前主库的GTID  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000004  Position: 113602604  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-24  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | OFF |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | NULL | 0 |  +---------+----------+  1 row in set (0.01 sec)  #查询主库VIP  inet 192.168.9.102/24 brd 192.168.9.255 scope global eth0  inet 192.168.9.120/24 scope global secondary eth0 |

#### （2）查看从库slave

|  |
| --- |
| mysql> show global variables like 'server\_uuid';  +---------------+--------------------------------------+  | Variable\_name | Value |  +---------------+--------------------------------------+  | server\_uuid | 43790147-f874-11e7-ac51-000c2992078d |  +---------------+--------------------------------------+  1 row in set (0.00 sec)  #登陆从库查看GTID  mysql> show slave status\G  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-24  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | NULL | 0 |  +---------+----------+  1 row in set (0.00 sec) |

#### （3）查看从库slave2

|  |
| --- |
| #登陆从库slave2查看当前数据库的UUID  mysql> show global variables like 'server\_uuid';  +---------------+--------------------------------------+  | Variable\_name | Value |  +---------------+--------------------------------------+  | server\_uuid | 8d9a99f0-f89c-11e7-ad58-000c293e10f6 |  +---------------+--------------------------------------+  #登陆从库查看GTID  mysql> show slave status\G  43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-24  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | NULL | 0 |  +---------+----------+  1 row in set (0.00 sec) |

#### （4）模拟主库正在进行事务

#登陆主库：192.168.9.102/master，执行以下脚本

|  |
| --- |
| use test;  create table t01(xh int primary key);  delimiter //  create procedure p\_ins\_t01(n int)  begin  declare i int;  set i=1;  while i<=n do  start transaction;  insert into t01(xh) values(i);  commit;  set i=i+1;  end while;  end //  delimiter ;  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000004  Position: 1122  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-2,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-23  #模拟主库产生100000个事务  use test  mysql> call p\_ins\_t01(100000); |

#### （5）正常关闭主库

#正常关闭：192.168.9.102/maser

|  |
| --- |
| su - mysql  /usr/local/mysql/bin/mysqladmin --login-path=mysql3306 shutdown  180128 01:27:20 mysqld\_safe mysqld from pid file /data/data3306/tmp/mysql3306.pid ended  #上面执行中的事务会终止  mysql> call p\_ins\_t01(100000);  ERROR 1053 (08S01): Server shutdown in progress |

说明：关闭主库操作前请确保mha进程正在运行中，否则不会进行故障转移。

在主库事务执行中，正常关闭主库。

#### （6）查看mha日志

|  |
| --- |
|  |

#### （7）故障转移后主从环境检查

|  |  |
| --- | --- |
| 主库 | 192.168.9.103/slave |
| 从库 | 192.168.9.104/slave2 |
| 原主库 | 192.168.9.102/maser 已失败，移出集群 |
| VIP | 192.168.9.120（在主库slave上） |

#查看主库：192.168.9.103/slave

|  |
| --- |
| mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000005  Position: 114008419  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  1 row in set (0.00 sec)  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | OFF |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 4616 | 4616 |  +---------+----------+  1 row in set (0.00 sec)  [root@slave ~]# ip addr  inet 192.168.9.103/24 brd 192.168.9.255 scope global eth0  inet 192.168.9.120/24 scope global secondary eth0 |

#查看从库：192.168.9.104/slave2

|  |
| --- |
| mysql> show slave status\G  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 4616 | 4616 |  +---------+----------+  1 row in set (0.00 sec) |

#查看mha进程：

|  |
| --- |
| [root@manager bin]# supervisorctl status  mha:mha\_testdbapp\_3306 FATAL Exited too quickly (process log may have details) |

#### （8）重建失败主库

|  |  |
| --- | --- |
| 主库 | 192.168.9.103/slave |
| 从库 | 192.168.9.104/slave2 |
| 原主库 | 192.168.9.102/maser 已失败，移出集群 |
| VIP | 192.168.9.120（在主库slave上） |

#启动原主库：192.168.9.102,因主库是正常关闭，所以可以正常启动

|  |
| --- |
| [mysql@master dba]$ ./mysql\_startup.sh 3306 |

#登陆原主库：192.168.9.102，查询服务停止前的GTID

|  |
| --- |
| mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000006  Position: 231  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  1 row in set (0.00 sec) |

#登陆原主库：192.168.9.102，重建主从关系

|  |
| --- |
| change master to  master\_host='192.168.9.103',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1;  Query OK, 0 rows affected, 2 warnings (0.02 sec)  start slave;  Query OK, 0 rows affected (0.01 sec)  mysql> show slave status\G  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-404279,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 4616 | 4616 |  +---------+----------+  1 row in set (0.00 sec)  #将从库设置为只读  mysql> set global read\_only=on;  Query OK, 0 rows affected (0.00 sec) |

#### （9）检查新的主从同步状态

|  |  |
| --- | --- |
| 主库 | 192.168.9.103 |
| 从库 | 192.168.9.104，192.168.9.102 |
| VIP | 192.168.9.120（在主库上） |

#查看mha最新主从同步状态

|  |
| --- |
| 192.168.9.103(192.168.9.103:3306) (current master)  +--192.168.9.102(192.168.9.102:3306)  +--192.168.9.104(192.168.9.104:3306)  附件：  检查日志，发现了一个问题，新恢复的主库没有设置为只读，已将这个过程补充至（10）最后。  Sun Jan 28 03:17:45 2018 - [info] Checking slave configurations..  Sun Jan 28 03:17:45 2018 - [info] read\_only=1 is not set on slave 192.168.9.102(192.168.9.102:3306) |

### 3.1.4 实验结论

（1）主库正常停止，MHA可以进行故障转移操作，自动选择新主库，应用差异日志，从库自动从新主库接收日志。整个过程不需要手动干预，由mha自动完成。

（2）故障转移后mha-helper进程会终止，需要手动再次启动。

（3）故障主库重新加入mha集群后，需要手动设置数据库为只读。

3.2 主库宕机,mha是否会正常切换

### 3.2.1 实验环境

|  |  |
| --- | --- |
| 主库 | 192.168.9.103 |
| 从库 | 192.168.9.104，192.168.9.102 |
| VIP | 192.168.9.120（在主库上） |

### 3.2.2 实验步骤

|  |  |  |
| --- | --- | --- |
| 序号 | 操作节点 | 操作内容描述 |
| 1 | 管理节点 | 确保mha服务是启动状态。 |
| 1 | 主库：192.168.9.103/slave | 模拟主库并发执行进行大量事务。  kill -9 杀掉主库mysql进程,模拟主库异常终止。 |
| 2 | 主库：192.168.9.103/slave  从库：192.168.9.104/slave2  192.168.9.102/master | 查看mha是否进行了故转移操作  新主库：192.168.9.102/master  从库：192.168.9.104/slave2  故障主库：192.168.9.103/slave，已移出mha集群中。 |
| 3 | 失败主库：192.168.9.103 | 从mysql错误日志中查找失败前最新的GTID  重建主从关系，跳过失败主库已执行过的GTID  启动从库线程  如果主库数据文件损坏，则需要重建主库，过程省略。 |

### 3.2.3 操作步骤

#### （1）管理节点查看mha状态

|  |
| --- |
| [root@manager bin]# supervisorctl status  mha:mha\_testdbapp\_3306 FATAL Exited too quickly (process log may have details)  [root@manager bin]#  [root@manager bin]# supervisorctl status  mha:mha\_testdbapp\_3306 FATAL Exited too quickly (process log may have details)  [root@manager bin]# supervisorctl start mha:mha\_testdbapp\_3306  mha:mha\_testdbapp\_3306: started  [root@manager bin]# supervisorctl status  mha:mha\_testdbapp\_3306 RUNNING pid 38437, uptime 0:00:08 |

说明：实验前确保mha服务是运行状态。

#### （2）主库模拟并发

|  |
| --- |
| #创建过程，用于模拟产生事务  use test  delimiter //  create procedure p\_ins\_t02(n\_start int,n\_stop int)  begin  declare i int;  set i=n\_start;  while i<=n\_stop do  start transaction;  insert into t01(xh) values(i);  commit;  set i=i+1;  end while;  end //  delimiter ;  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | NULL | 0 |  +---------+----------+  1 row in set (0.00 sec)  #客户端1: 模拟主库产生100000个事务  use test  call p\_ins\_t02(1,100000);  #客户端2: 模拟主库产生100000个事务  use test  call p\_ins\_t02(100001,200000); |

#### （3）杀掉主库进程

|  |
| --- |
| #操作系统:杀掉MySQL-Server进程，以上客户端进程执行大约3秒后，执行该操作  kill -9 36439 37611  #客户端1窗口报错  mysql> call p\_ins\_t02(5001,25000);  ERROR 2013 (HY000): Lost connection to MySQL server during query  #客户端2窗口报错  mysql> call p\_ins\_t02(25001,40000);  ERROR 2013 (HY000): Lost connection to MySQL server during query |

#### （4）查看mha日志

|  |
| --- |
| Sun Jan 28 05:01:01 2018 - [warning] Got error on MySQL connect: 2003 (Can't connect to MySQL server on '192.168.9.103' (111 "拒绝连接"))  完整mha日志详见附件： |

#### （5）检查复制环境

#主库：192.168.9.103/slave

|  |
| --- |
| #主库因被杀掉，无法连接了  [mysql@slave errlog]$ mysql3306  ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/data/data3306/tmp/mysql3306.sock' (111)  #主库VIP，仍在失败主库上  [mysql@slave errlog]$ ip addr show eth0  2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000  link/ether 00:0c:29:92:07:8d brd ff:ff:ff:ff:ff:ff  inet 192.168.9.103/24 brd 192.168.9.255 scope global eth0  inet 192.168.9.120/24 scope global secondary eth0  inet6 fe80::20c:29ff:fe92:78d/64 scope link  valid\_lft forever preferred\_lft forever  **#主库错误日志**：  2018-01-28 20:21:03 37611 [Note] Slave: received end packet from server due to dump thread being killed on master. Dump threads are killed for example during master shutdown, explicitly by a user, or when the master receives a binlog send request from a duplicate server UUID <43790147-f874-11e7-ac51-000c2992078d> : Error  2018-01-28 20:21:03 37611 [Note] Slave I/O thread: Failed reading log event, reconnecting to retry, log 'mysql\_bin.000005' at position 1338871  2018-01-28 20:21:03 37611 [Warning] Storing MySQL user name or password information in the master info repository is not secure and is therefore not recommended. Please consider using the USER and PASSWORD connection options for START SLAVE; see the 'START SLAVE Syntax' in the MySQL Manual for more information.  2018-01-28 20:21:03 37611 [ERROR] Slave I/O: error reconnecting to master 'myrpl@192.168.9.102:3306' - retry-time: 60 retries: 1, Error\_code: 2003  2018-01-28 20:21:12 37611 [Note] Slave I/O thread killed during or after a reconnect done to recover from failed read  2018-01-28 20:21:12 37611 [Note] Slave I/O thread exiting, read up to log 'mysql\_bin.000005', position 1338871  2018-01-28 20:21:12 37611 [Note] Error reading relay log event: slave SQL thread was |

#从库：192.168.9.104/slave2

|  |
| --- |
| #从库状态  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  43790147-f874-11e7-ac51-000c2992078d:1-418139,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  #数据库状态  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  #表test.t01行数，实验前是0，因主库是被杀掉了，只执行了部分事务  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 105432 | 13859 |  +---------+----------+  1 row in set (0.00 sec) |

#从库：192.168.9.102/master

|  |
| --- |
| #从库状态  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-418139,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  #数据库状态  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  #表test.t01行数，实验前是0，因主库是被杀掉了，只执行了部分事务  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 105432 | 13859 |  +---------+----------+  1 row in set (0.00 sec) |

#### （6）启动失败主库

|  |
| --- |
| #启动主库  [mysql@slave dba]$ ./mysql\_startup.sh 3306  #连接主库  [mysql@slave dba]$ mysql3306  #查看主库状态  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000006  Position: 231  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-418141,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  1 row in set (0.00 sec) |

#### （7）主库启动后查看复制环境

|  |
| --- |
| **#从库：**192.168.9.104/slave2  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Slave\_IO\_State: Waiting for master to send event  Master\_Host: 192.168.9.103  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000006  Read\_Master\_Log\_Pos: 231  Relay\_Log\_File: mysql3306-relay-bin.000004  Relay\_Log\_Pos: 401  Relay\_Master\_Log\_File: mysql\_bin.000006  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 231  Relay\_Log\_Space: 1486  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: 91033306  Master\_UUID: 43790147-f874-11e7-ac51-000c2992078d  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:404280-418141  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-418141,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  Auto\_Position: 1  1 row in set (0.00 sec)  **#从库：**192.168.9.102/master  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Slave\_IO\_State: Waiting for master to send event  Master\_Host: 192.168.9.103  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000006  Read\_Master\_Log\_Pos: 231  Relay\_Log\_File: mysql3306-relay-bin.000004  Relay\_Log\_Pos: 401  Relay\_Master\_Log\_File: mysql\_bin.000006  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 231  Relay\_Log\_Space: 1486  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: 91033306  Master\_UUID: 43790147-f874-11e7-ac51-000c2992078d  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:404280-418141  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-418141,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-4640  Auto\_Position: 1 |

### 3.2.4 实验结论

（1）主库异常停止，MHA无法进行故障转移操作,从库状态为：“Slave\_IO\_State: Reconnecting after a failed master event read”，不停地在尝试连接主库，VIP仍在主库上。

（2）如果主库可以启动成功，则mha集群会自协恢复正常。

（3）如果主库因硬件故障无法正常启动，则需新建从库代替失败主库。

（3）如果主动当时不能马上启动，此时又需要及时处理。需要手工进行如下处理：

* 从两个从库中选出一个应用失败主库日志最多的从库。
* 若主库二进制日志可用，找出最新差异日志并应用。如果不可用，会发生数据丢失。
* 提升最新从库为主库。
* 其它从库重新指向新新主库，从新主库获取二进制日志。
* 失败主库可以启动后，再将失败主库加入mha集群中，详见3.2.3-(8)步骤。

# 四、重建复制环境

当主库异常关闭后，由于mha不进行故障转移，如何快速修复复制环境。这一步是基于(5)步之后进行的操作。

4.1 停止mha-helper

|  |
| --- |
| [root@manager ~]# supervisorctl stop mha:mha\_testdbapp\_3306  mha:mha\_testdbapp\_3306: stopped  [root@manager ~]# supervisorctl status  mha:mha\_testdbapp\_3306 STOPPED Jan 28 11:39 PM |

4.2 查看两个从库已执行主库的GTID大的做新主库，

选择192.168.9.102/master做为新主库

|  |
| --- |
| #删除原复制信息  mysql> stop slave;  Query OK, 0 rows affected (0.01 sec)  mysql> reset slave all;  Query OK, 0 rows affected (0.00 sec)  mysql> show slave status\G  Empty set (0.00 sec)  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000001  Position: 2564829  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-39164  1 row in set (0.00 sec) |

4.3 将从库:192.169.9.104/slave2重新指向这个新主

|  |
| --- |
| mysql> stop slave;  Query OK, 0 rows affected (0.00 sec)  mysql> reset slave all;  Query OK, 0 rows affected (0.01 sec)  change master to  master\_host='192.168.9.102',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1;  reset master;  SET GLOBAL gtid\_purged='43790147-f874-11e7-ac51-000c2992078d:1-39164';  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: 192.168.9.102  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000001  Read\_Master\_Log\_Pos: 11004949  Relay\_Log\_File: mysql3306-relay-bin.000002  Relay\_Log\_Pos: 408  Relay\_Master\_Log\_File: mysql\_bin.000001  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 11004949  Relay\_Log\_Space: 616  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: 91023306  Master\_UUID: 92f6e6cb-f84f-11e7-ab62-000c291dadaa  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  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: 43790147-f874-11e7-ac51-000c2992078d:1-39164  Auto\_Position: 1  1 row in set (0.00 sec) |

4.4 失败主库上从最新二进制文件中查找主库宕机时已同执行GTID信息

|  |
| --- |
| /usr/local/mysql/bin/mysqlbinlog -vv --base64-output=decode-rows mysql\_bin.000002>2.sql  vi 2.sql  #从这个文件中倒着看，找到主库失败前最后一个已执行的GTID值。以下是最近一个GTID值的内容：  ----------------------------------------------------------------------------------------  SET @@SESSION.GTID\_NEXT= '43790147-f874-11e7-ac51-000c2992078d:39165'/\*!\*/;  # at 9650417  #180129 1:05:35 server id 91033306 end\_log\_pos 9650489 CRC32 0x2f5faf2d Query thread\_id=2 exec\_time=0 error\_code=0  SET TIMESTAMP=1517159135/\*!\*/;  BEGIN  /\*!\*/;  # at 9650489  #180129 1:05:35 server id 91033306 end\_log\_pos 9650542 CRC32 0x364a8f80 Rows\_query  # insert into t01(xh) values(i)  # at 9650542  #180129 1:05:35 server id 91033306 end\_log\_pos 9650588 CRC32 0xfc2aea7c Table\_map: `test`.`t01` mapped to number 73  # at 9650588  #180129 1:05:35 server id 91033306 end\_log\_pos 9650628 CRC32 0xb4c9c082 Write\_rows: table id 73 flags: STMT\_END\_F  ### INSERT INTO `test`.`t01`  ### SET  ### @1=33277 /\* INT meta=0 nullable=0 is\_null=0 \*/  # at 9650628  #180129 1:05:35 server id 91033306 end\_log\_pos 9650659 CRC32 0xac073069 Xid = 166417  COMMIT/\*!\*/;  说明：最后有一个insert语句在失败主库执行成功了，但是未将最后一个事务同步到从库。因此需要将这条语句在新主库(192.168.9.102:/master)中执行一下，补齐丢失的数据。  insert into `test`.`t01`(xh) values(33277); |

4.5 查看新主库和从库同步情况

|  |
| --- |
| #查看主库：192.168.8.102/master:  mysql> insert into `test`.`t01`(xh) values(33277);  Query OK, 1 row affected (0.01 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 33277 | 33277 |  +---------+----------+  1 row in set (0.00 sec)  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000001  Position: 11005252  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-39164,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1  1 row in set (0.00 sec)  #查看从库：92.168.8.104:/slave2:  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-39164,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1 |

4.6 将失败主库加入mha集群中

|  |
| --- |
| (1)启动失败主库  [mysql@slave dba]$ ./mysql\_startup.sh 3306  [mysql@slave dba]$ 180129 01:32:11 mysqld\_safe Logging to '/data/data3306/log/errlog/mysql.err'.  180129 01:32:11 mysqld\_safe Starting mysqld daemon with databases from /data/data3306/data  #登陆失败主库  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000003  Position: 191  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-39165  1 row in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 32493 | 32493 |  +---------+----------+  1 row in set (0.03 sec)  说明：发现新问题了，原失入主库中表test.t01中的数据比在最后一个binlog日志中查询到的数据还要小？ 为什么binlog日志中数据都传到从库了，失败主库的数据文件中的数据却是旧的？ 可能与直接杀服务器进程有关，内存中的数据没有及时刷到磁盘上，个人猜测。  (2)找到失败主库中表test.t01中32493这个值对应的GTID值  ----------------------------------------------------------------------------  SET @@SESSION.GTID\_NEXT= '43790147-f874-11e7-ac51-000c2992078d:38381'/\*!\*/;  # at 9423057  #180129 1:05:35 server id 91033306 end\_log\_pos 9423129 CRC32 0x8b8d1ac0 Query thread\_id=2 exec\_time=0 error\_code=0  SET TIMESTAMP=1517159135/\*!\*/;  BEGIN  /\*!\*/;  # at 9423129  #180129 1:05:35 server id 91033306 end\_log\_pos 9423182 CRC32 0xbf61166d Rows\_query  # insert into t01(xh) values(i)  # at 9423182  #180129 1:05:35 server id 91033306 end\_log\_pos 9423228 CRC32 0x9bffc5bc Table\_map: `test`.`t01` mapped to number 73  # at 9423228  #180129 1:05:35 server id 91033306 end\_log\_pos 9423268 CRC32 0xd726ceb6 Write\_rows: table id 73 flags: STMT\_END\_F  ### INSERT INTO `test`.`t01`  ### SET  ### @1=32493 /\* INT meta=0 nullable=0 is\_null=0 \*/  # at 9423268  #180129 1:05:35 server id 91033306 end\_log\_pos 9423299 CRC32 0xf07b5273 Xid = 162497  COMMIT/\*!\*/;  (3)将失败的主库指向新主库，跳过前面的GTID往后执行  change master to  master\_host='192.168.9.102',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1;  (4)备份失败主库二进制日制后进行如下操作,主库中显示的已执行GTID值与实际表中数据对应的GTID值不一样，因此需要将主库的GTID值修改为实际已执行过的GTID值，以便从主库中将未执行的数据同步过来。  reset master;  SET GLOBAL gtid\_purged='43790147-f874-11e7-ac51-000c2992078d:1-38381,92f6e6cb-f84f-11e7-ab62-000c291dadaa:1';  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: 192.168.9.102  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000001  Read\_Master\_Log\_Pos: 11005252  Relay\_Log\_File: mysql3306-relay-bin.000002  Relay\_Log\_Pos: 711  Relay\_Master\_Log\_File: mysql\_bin.000001  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 11005252  Relay\_Log\_Space: 919  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: 91023306  Master\_UUID: 92f6e6cb-f84f-11e7-ab62-000c291dadaa  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set: 92f6e6cb-f84f-11e7-ab62-000c291dadaa:1  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:38381,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1  Auto\_Position: 1  1 row in set (0.00 sec)  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 33277 | 32494 |  +---------+----------+  1 row in set (0.01 sec)  说明：数据无法从主库同步过来，因为主库中二进制日志已经执行过，不会再发送到从库中继日志了，因此这部分数据需要手动找出来进行同步。  (5)从库：192.168.9.104/slave2中导出表test.t01的数据，插入失败的主库中  /usr/local/mysql/bin/mysqldump --login-path=mysql3306 -R --triggers --opt --single-transaction test t01>t01.sql  (6)导出的文件t01.sql中包含了当时导出时的GTID值，同时设置了关闭二进制日志后再插入数据，到最后再恢复为数据库默认值。以下内容为t01.sql文件中最前面部分内容：  SET @MYSQLDUMP\_TEMP\_LOG\_BIN = @@SESSION.SQL\_LOG\_BIN;  SET @@SESSION.SQL\_LOG\_BIN= 0;  --  -- GTID state at the beginning of the backup  --  SET @@GLOBAL.GTID\_PURGED='43790147-f874-11e7-ac51-000c2992078d:1-39164,  8d9a99f0-f89c-11e7-ad58-000c293e10f6:1-2,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1';  (7)失败主库操作  mysql>source /data/data3306/data/t01.sql  (8)查询数据  mysql> select max(xh),count(0) from test.t01;  +---------+----------+  | max(xh) | count(0) |  +---------+----------+  | 33277 | 33277 |  +---------+----------+  1 row in set (0.00 sec)  (9)数据已补齐  mysql> show variables like '%SQL\_LOG\_BIN%';  +---------------+-------+  | Variable\_name | Value |  +---------------+-------+  | sql\_log\_bin | ON |  +---------------+-------+  1 row in set (0.00 sec)  (10)重新在失败主库上gtid为主库中已执行过的GTID  mysql> reset master;  Query OK, 0 rows affected (0.01 sec)  #失败主库上的GTID设置为从库导出文件t01.sql中记录的GTID值，考虑到从库GTID也在不变变化中，因此必须设置已执行GTID为导出文件当时的记录的GTID，这样数据导入后，启动线程后才会同步之后的变化。  SET GLOBAL gtid\_purged='43790147-f874-11e7-ac51-000c2992078d:1-39164,8d9a99f0-f89c-11e7-ad58-000c293e10f6:1-2,92f6e6cb-f84f-11e7-ab62-000c291dadaa:1';  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: 192.168.9.102  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000001  Read\_Master\_Log\_Pos: 11005252  Relay\_Log\_File: mysql3306-relay-bin.000005  Relay\_Log\_Pos: 448  Relay\_Master\_Log\_File: mysql\_bin.000001  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 11005252  Relay\_Log\_Space: 1000  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: 91023306  Master\_UUID: 92f6e6cb-f84f-11e7-ab62-000c291dadaa  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set: 92f6e6cb-f84f-11e7-ab62-000c291dadaa:1  Executed\_Gtid\_Set: 43790147-f874-11e7-ac51-000c2992078d:1-39164,  8d9a99f0-f89c-11e7-ad58-000c293e10f6:1-2,  92f6e6cb-f84f-11e7-ab62-000c291dadaa:1  Auto\_Position: 1  1 row in set (0.00 sec) |

4.7 删除失败主库上的VIP，在新主库上增加VIP

|  |
| --- |
| #192.168.9.103/slave  /sbin/ip addr delete 192.168.9.120/24 dev eth0  #192.168.9.102/master  /sbin/ip addr add 192.168.9.120/24 dev eth0 |

4.8 检查主从库的read\_only变量

|  |
| --- |
| #192.168.9.102/master  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> set global read\_only=off;  Query OK, 0 rows affected (0.00 sec)  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | OFF |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  #192.168.9.103/slave  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | OFF |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  mysql> set global read\_only=on;  Query OK, 0 rows affected (0.00 sec)  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec)  #192.168.9.104/slave2  mysql> show variables like '%read\_only%';  +------------------+-------+  | Variable\_name | Value |  +------------------+-------+  | innodb\_read\_only | OFF |  | read\_only | ON |  | tx\_read\_only | OFF |  +------------------+-------+  3 rows in set (0.00 sec) |

4.9 mha复制检查

|  |
| --- |
| #192.168.9.101/manager  [root@manager bin]# /home/dba/mha/bin /check\_rpl.sh 3306 testdbapp  Mon Jan 29 02:44:28 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.  Mon Jan 29 02:44:28 2018 - [info] Reading application default configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Mon Jan 29 02:44:28 2018 - [info] Reading server configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Mon Jan 29 02:44:28 2018 - [info] MHA::MasterMonitor version 0.57.  Mon Jan 29 02:44:29 2018 - [info] GTID failover mode = 1  Mon Jan 29 02:44:29 2018 - [info] Dead Servers:  Mon Jan 29 02:44:29 2018 - [info] Alive Servers:  Mon Jan 29 02:44:29 2018 - [info] 192.168.9.102(192.168.9.102:3306)  Mon Jan 29 02:44:29 2018 - [info] 192.168.9.103(192.168.9.103:3306)  Mon Jan 29 02:44:29 2018 - [info] 192.168.9.104(192.168.9.104:3306)  Mon Jan 29 02:44:29 2018 - [info] Alive Slaves:  Mon Jan 29 02:44:29 2018 - [info] 192.168.9.103(192.168.9.103:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Mon Jan 29 02:44:29 2018 - [info] GTID ON  Mon Jan 29 02:44:29 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Mon Jan 29 02:44:29 2018 - [info] 192.168.9.104(192.168.9.104:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Mon Jan 29 02:44:29 2018 - [info] GTID ON  Mon Jan 29 02:44:29 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Mon Jan 29 02:44:29 2018 - [info] Current Alive Master: 192.168.9.102(192.168.9.102:3306)  Mon Jan 29 02:44:29 2018 - [info] Checking slave configurations..  Mon Jan 29 02:44:29 2018 - [info] Checking replication filtering settings..  Mon Jan 29 02:44:29 2018 - [info] binlog\_do\_db= , binlog\_ignore\_db=  Mon Jan 29 02:44:29 2018 - [info] Replication filtering check ok.  Mon Jan 29 02:44:29 2018 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node package checking.  Mon Jan 29 02:44:29 2018 - [info] Checking SSH publickey authentication settings on the current master..  Mon Jan 29 02:44:29 2018 - [info] HealthCheck: SSH to 192.168.9.102 is reachable.  Mon Jan 29 02:44:29 2018 - [info]  192.168.9.102(192.168.9.102:3306) (current master)  +--192.168.9.103(192.168.9.103:3306)  +--192.168.9.104(192.168.9.104:3306)  Mon Jan 29 02:44:29 2018 - [info] Checking replication health on 192.168.9.103..  Mon Jan 29 02:44:29 2018 - [info] ok.  Mon Jan 29 02:44:29 2018 - [info] Checking replication health on 192.168.9.104..  Mon Jan 29 02:44:29 2018 - [info] ok.  Mon Jan 29 02:44:29 2018 - [info] Checking master\_ip\_failover\_script status:  Mon Jan 29 02:44:29 2018 - [info] /usr/bin/master\_ip\_hard\_failover\_helper --test\_config\_path=/home/dba/mha/helper/3306\_testdbapp --command=status --ssh\_user=root --orig\_master\_host=192.168.9.102 --orig\_master\_ip=192.168.9.102 --orig\_master\_port=3306  Reading config file: /home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf  Checking the vip using the 'metal' provider on the original master '192.168.9.102'  Connecting to 'root'@'192.168.9.102'  Executing command on '192.168.9.102': /sbin/ip addr show dev eth0  Mon Jan 29 02:44:29 2018 - [info] OK.  Mon Jan 29 02:44:29 2018 - [warning] shutdown\_script is not defined.  Mon Jan 29 02:44:29 2018 - [info] Got exit code 0 (Not master dead).  MySQL Replication Health is OK. |

4.10 启动mha-helper

|  |
| --- |
| [root@manager bin]# supervisorctl start mha:mha\_testdbapp\_3306  mha:mha\_testdbapp\_3306: started  [root@manager bin]#  [root@manager bin]# supervisorctl status  mha:mha\_testdbapp\_3306 RUNNING pid 48185, uptime 0:00:07 |

# 五、快速重新开始实验

|  |
| --- |
| **#192.168.8.103：/slave**  reset master;  Query OK, 0 rows affected (0.01 sec)  mysql> truncate table test.t01;  Query OK, 0 rows affected (0.01 sec)  **#192.168.8.102:/master**  mysql> stop slave;  mysql> reset slave all;  mysql> reset master;  change master to  master\_host='192.168.9.103',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1;  mysql> start slave;  Query OK, 0 rows affected (0.01 sec)  **#192.168.8.104:/slave2**  mysql> stop slave;  mysql> reset slave all;  mysql> reset master;  change master to  master\_host='192.168.9.103',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1; |