

MySQL主从复制转MGR---实战篇

1.环境信息

```
IP  port    role    info
10.10.30.81 3306    node1   master
10.10.30.82 3306    node2   shadow
10.10.30.83 3306    node3   slave
CentOS Linux release 7.6.1810 (Core)
MySQL Ver 8.0.19 for linux-glibc2.12 on x86_64 (MySQL Community Server - GPL)
MySQL Router Ver 8.0.20 for Linux on x86_64 (MySQL Community - GPL)
MySQL Shell Ver 8.0.20 for Linux on x86_64 - for MySQL 8.0.20 (MySQL Community Server (GPL))
```

2.软件位置

在三个节点上部署好MySQL、MySQL Router、MySQL Shell。

3.搭建复制环境，开启增强半同步

所有节点配置

```
root@localhost [(none)]>set global super_read_only=0;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>create user 'repl'@'10.10.30.%' identified by 'repl';
Query OK, 0 rows affected (0.02 sec)

root@localhost [(none)]>grant replication slave on *.* to 'repl'@'10.10.30.%';
Query OK, 0 rows affected (0.02 sec)

root@localhost [(none)]>install plugin rpl_semi_sync_slave soname
'semisync_slave.so';
Query OK, 0 rows affected (0.01 sec)

root@localhost [(none)]>install plugin rpl_semi_sync_master soname
'semisync_master.so';
Query OK, 0 rows affected (0.02 sec)
```

master节点配置

```
root@localhost [(none)]>set global rpl_semi_sync_master_enabled=ON;
Query OK, 0 rows affected (0.01 sec)

root@localhost [(none)]>show global variables like '%semi%';
+-----+-----+
| variable_name | value |
+-----+-----+
| rpl_semi_sync_master_enabled | ON |
| rpl_semi_sync_master_timeout | 10000 |
| rpl_semi_sync_master_trace_level | 32 |
| rpl_semi_sync_master_wait_for_slave_count | 1 |
| rpl_semi_sync_master_wait_no_slave | ON |
+-----+-----+
```

```

| repl_semi_sync_master_wait_point | AFTER_SYNC |
| repl_semi_sync_slave_enabled    | OFF        |
| repl_semi_sync_slave_trace_level| 32         |
+-----+-----+
8 rows in set (0.00 sec)

root@localhost [(none)]>reset master;
Query OK, 0 rows affected (0.04 sec)

```

shadow节点配置

```

root@localhost [(none)]>set global repl_semi_sync_slave_enabled=ON;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>change master to
master_host='10.10.30.81',master_port=3306,master_user='repl',master_password='r
epl',master_auto_position=1,get_master_public_key=1;
Query OK, 0 rows affected, 2 warnings (0.04 sec)

root@localhost [(none)]>reset master;
Query OK, 0 rows affected (0.04 sec)

```

shadow启动复制

```

root@localhost [(none)]>start slave;
Query OK, 0 rows affected (0.03 sec)

root@localhost [(none)]>show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 10.10.30.81
      Master_User: repl
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 155
      Relay_Log_File: ms82-relay-bin.000002
      Relay_Log_Pos: 369
      Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
...
...
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more
updates
      Master_Retry_Count: 86400
...
...
1 row in set (0.00 sec)

```

master查看半同步状态

```

root@localhost [(none)]>show global status like '%semi%';
+-----+-----+
| Variable_name | value |
+-----+-----+

```

```

| Rpl_semi_sync_master_clients          | 2      |
| Rpl_semi_sync_master_net_avg_wait_time | 0      |
| Rpl_semi_sync_master_net_wait_time    | 0      |
| Rpl_semi_sync_master_net_waits        | 0      |
| Rpl_semi_sync_master_no_times          | 0      |
| Rpl_semi_sync_master_no_tx            | 0      |
| Rpl_semi_sync_master_status            | ON     |
| Rpl_semi_sync_master_timefunc_failures | 0      |
| Rpl_semi_sync_master_tx_avg_wait_time  | 0      |
| Rpl_semi_sync_master_tx_wait_time     | 0      |
| Rpl_semi_sync_master_tx_waits          | 0      |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0      |
| Rpl_semi_sync_master_wait_sessions     | 0      |
| Rpl_semi_sync_master_yes_tx            | 0      |
| Rpl_semi_sync_slave_status             | OFF    |
+-----+-----+
15 rows in set (0.00 sec)

```

按照在shadow节点上的配置和启动方法，完成slave节点的复制搭建，配好一个一主一备一从的复制集群。

4.模拟业务，使用脚本产生事务

建表

```

root@localhost [(none)]>create database kk;
Query OK, 1 row affected (0.03 sec)
root@localhost [(none)]>use kk
Database changed
root@localhost [kk]>create table k1 ( id int auto_increment primary key , dt1
varchar(20) default 'abc');
Query OK, 0 rows affected (0.05 sec)

```

开启一个session，运行脚本产生事务

```

[root@node1 ~]# while ;; do echo "insert into kk.k1(dt1)
values('duangduangduang');" | mysql -s /data/mysql/mysql3316/tmp/mysql.sock;
sleep 1;done

```

5.手动配置MGR

配置Master，将Master转为MGR

配置参数

```

root@localhost [kk]>install plugin group_replication soname
'group_replication.so';
Query OK, 0 rows affected (0.03 sec)

root@localhost [kk]>set persist binlog_checksum=NONE;
Query OK, 0 rows affected (0.02 sec)

root@localhost [kk]>set persist transaction_write_set_extraction=XXHASH64;
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>select uuid();

```

```

+-----+
| uuid() |
+-----+
| 3260d70c-966e-11ea-ba8b-0242c0a8bc51 |
+-----+
1 row in set (0.00 sec)

root@localhost [kk]>set persist group_replication_group_name='3260d70c-966e-
11ea-ba8b-0242c0a8bc51';
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>set persist
group_replication_local_address="10.10.30.81:13306";
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>set persist
group_replication_group_seeds="10.10.30.81:13306,10.10.30.82:13306,10.10.30.83:1
3306";
Query OK, 0 rows affected (0.00 sec)

#也要加上这个，具体见文末
SET persist group_replication_recovery_get_public_key = 1;

root@localhost [kk]>set persist group_replication_bootstrap_group=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>set persist group_replication_start_on_boot=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>set global group_replication_bootstrap_group=on;
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>start group_replication;
Query OK, 0 rows affected (3.36 sec)

root@localhost [kk]>set global group_replication_bootstrap_group=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [kk]>select * from performance_schema.replication_group_members;
+-----+
+-----+
| CHANNEL_NAME | MEMBER_ID | MEMBER_HOST |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+
+-----+
| group_replication_applier | 29ea8b7f-966d-11ea-937c-0242c0a8bc51 | node1 |
| 3316 | ONLINE | PRIMARY | 8.0.19 |
+-----+
+-----+
1 row in set (0.01 sec)

```

此时发现发生事务的session出现了提醒

```
[root@node1 ~]# while ;; do echo "insert into kk.k1(dt1)
values('duangduangduang');" | mysql -S /data/mysql/mysql3316/tmp/mysql.sock;
sleep 1;done
ERROR 1290 (HY000) at line 1: The MySQL server is running with the --super-read-
only option so it cannot execute this statement
ERROR 1290 (HY000) at line 1: The MySQL server is running with the --super-read-
only option so it cannot execute this statement
ERROR 1290 (HY000) at line 1: The MySQL server is running with the --super-read-
only option so it cannot execute this statement
```

6.去配置shadow , 转换为MGR

```
root@localhost [(none)]>install plugin group_replication soname
'group_replication.so';
Query OK, 0 rows affected (0.01 sec)

root@localhost [(none)]>set persist binlog_checksum=NONE;
Query OK, 0 rows affected (0.03 sec)

root@localhost [(none)]>set persist transaction_write_set_extraction=XXHASH64;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist group_replication_group_name='3260d70c-966e-
11ea-ba8b-0242c0a8bc51';
Query OK, 0 rows affected (0.01 sec)

root@localhost [(none)]>set persist
group_replication_local_address="10.10.30.82:13306";
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist
group_replication_group_seeds="10.10.30.81:13306,10.10.30.82:13306,10.10.30.83:1
3306";
Query OK, 0 rows affected (0.00 sec)

#也要加上这个, 具体见文末
SET persist group_replication_recovery_get_public_key = 1;

root@localhost [(none)]>set persist group_replication_bootstrap_group=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist group_replication_start_on_boot=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>start group_replication;
ERROR 3092 (HY000): The server is not configured properly to be an active member
of the group. Please see more details on error log.
root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
| MEMBER_PORT          | MEMBER_STATE       | MEMBER_ROLE          | MEMBER_VERSION       |
+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+
```

```

| group_replication_applier | 2cbcfaa5-966d-11ea-8707-0242c0a8bc52 | node2
| 3316 | OFFLINE |
| | |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
1 row in set (0.01 sec)

root@localhost [(none)]>stop group_replication;
Query OK, 0 rows affected (4.78 sec)

root@localhost [(none)]>change master to
master_user='repl',master_password='repl' for channel
'group_replication_recovery';
Query OK, 0 rows affected, 2 warnings (0.03 sec)

root@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (3.88 sec)

root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| group_replication_applier | 29ea8b7f-966d-11ea-937c-0242c0a8bc51 | node1
| 3316 | ONLINE | PRIMARY | 8.0.19 |
| group_replication_applier | 2cbcfaa5-966d-11ea-8707-0242c0a8bc52 | node2
| 3316 | ONLINE | SECONDARY | 8.0.19 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

7.如法炮制，改造slave

在改造之前，突然想到，现有的架构成为了：node1(master)\node2(shadow)为MGR，node3(slave)是node1(master)的从库，那么检查一下当前三个节点的情况：

```

node1:
root@localhost [kk]>select count(*) from kk.k1;
+-----+
| count(*) |
+-----+
| 456 |
+-----+
1 row in set (0.00 sec)

root@localhost [kk]>show master status ;
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB |
Executed_Gtid_Set
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

```
| mysql-bin.000002 | 154142 | | 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
node2:
root@localhost [(none)]>select count(*) from kk.k1;
+-----+
| count(*) |
+-----+
| 456 |
+-----+
1 row in set (0.00 sec)
```

```
root@localhost [(none)]>show master status ;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000002 | 109956 | | 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

##注意, node2的 IO、SQL THREAD没有运行, 但是 Executed_Gtid_Set 是跟进的噢

```
root@localhost [(none)]>show slave status\G
***** 1. row *****
Slave_IO_State:
  Master_Host: 10.10.30.81
  Master_User: repl
  Master_Port: 3316
  Connect_Retry: 60
  Master_Log_File: mysql-bin.000002
  Read_Master_Log_Pos: 74606
  Relay_Log_File: ms82-relay-bin.000004
  Relay_Log_Pos: 74820
  Relay_Master_Log_File: mysql-bin.000002
  Slave_IO_Running: No
  Slave_SQL_Running: No
...
...
  Master_Server_Id: 813316
    Master_UUID: f78a6902-9679-11ea-b136-0242c0a8bc51
    Master_Info_File: mysql.slave_master_info
...
  Retrieved_Gtid_Set: 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-121, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111
    Executed_Gtid_Set: 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111
      Auto_Position: 1
...
```

```

1 row in set (0.00 sec)

node3:
root@localhost [(none)]>select count(*) from kk.k1;
+-----+
| count(*) |
+-----+
|      456 |
+-----+
1 row in set (0.00 sec)

root@localhost [(none)]>show master status ;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 | 169340   |              |                  | 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

root@localhost [(none)]>show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 10.10.30.81
      Master_User: repl
      Master_Port: 3316
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000002
      Read_Master_Log_Pos: 154142
      Relay_Log_File: ms83-relay-bin.000004
      Relay_Log_Pos: 154356
      Relay_Master_Log_File: mysql-bin.000002
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
...
...
      Master_UUID: f78a6902-9679-11ea-b136-0242c0a8bc51
      Master_Info_File: mysql.slave_master_info
...
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
...
      Retrieved_Gtid_Set: 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111
      Executed_Gtid_Set: 3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-350, f78a6902-9679-11ea-b136-0242c0a8bc51:1-111
      Auto_Position: 1
...
1 row in set (0.00 sec)

```

转换slave


```

root@localhost [(none)]>install plugin group_replication soname
'group_replication.so';
Query OK, 0 rows affected (0.02 sec)

root@localhost [(none)]>set persist binlog_checksum=NONE;
Query OK, 0 rows affected (0.03 sec)

root@localhost [(none)]>set persist transaction_write_set_extraction=XXHASH64;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist group_replication_group_name='3260d70c-966e-
11ea-ba8b-0242c0a8bc51';
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist
group_replication_local_address="10.10.30.83:13306";
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist
group_replication_group_seeds="10.10.30.81:13306,10.10.30.82:13306,10.10.30.83:1
3306";
group_rQuery OK, 0 rows affected (0.00 sec)

#也要加上这个，具体见文末
SET persist group_replication_recovery_get_public_key = 1;

root@localhost [(none)]>set persist group_replication_bootstrap_group=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>set persist group_replication_start_on_boot=off;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>stop slave;
Query OK, 0 rows affected (0.01 sec)

root@localhost [(none)]>change master to
master_user='repl',master_password='repl' for channel
'group_replication_recovery';
Query OK, 0 rows affected, 2 warnings (0.05 sec)

plicationroot@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (4.64 sec)

root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| group_replication_applier | 29ea8b7f-966d-11ea-937c-0242c0a8bc51 | node1
|          3316 | ONLINE          | PRIMARY          | 8.0.19          |
| group_replication_applier | 2cbcfaa5-966d-11ea-8707-0242c0a8bc52 | node2
|          3316 | ONLINE          | SECONDARY         | 8.0.19          |
| group_replication_applier | 2db7ddf1-966d-11ea-a7b3-0242c0a8bc53 | node3
|          3316 | ONLINE          | SECONDARY         | 8.0.19          |

```

```

+-----+-----+-----+-----+
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

root@localhost [(none)]>

```

8.一些tips

参数文件!

手动转换为MGR与通过MySQL Shell转换的最大区别是，后者会自动通过set persist 方式将变更写到mysqld-auto.cnf文件中，而手动操作需要注意这一点。上述实验完全没编辑my.cnf，如果使用set global，在MGR三节点再次冷启动的时候，MGR的配置参数就没了，无法启动MGR。解决方法是：

1. 在配置过程中用set persist 来代替set global，持久化保存配置
2. 如果已经手快重启了全部节点清空了临时配置，那么可以使用set persist再次设定一遍，设定好后理论上可以直接生效，启动MGR。

9.sha2_password魔咒

- 通过set global 配置后，重启了一下节点，再进行set persist持久化配置后,启动MGR后，master顺利online，但是在做node2加入MGR时，一直处于RECOVERING
- 检查errlog后发现：

```

2020-05-15T14:35:46.869802+08:00 21 [System] [MY-010597] [Rep] 'CHANGE MASTER
TO FOR CHANNEL 'group_replication_recovery' executed'. Previous state
master_host='ms81', master_port= 3316, master_log_file='', master_log_pos= 4,
master_bind=''. New state master_host='ms81', master_port= 3316,
master_log_file='', master_log_pos= 4, master_bind=''.
2020-05-15T14:35:46.906422+08:00 28 [Warning] [MY-010897] [Rep] 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.
2020-05-15T14:35:46.907876+08:00 28 [ERROR] [MY-010584] [Rep] slave I/O for
channel 'group_replication_recovery': error connecting to master 'rep@ms81:3316'
- retry-time: 60 retries: 1 message: Authentication plugin
'caching_sha2_password' reported error: Authentication requires secure
connection. Error_code: MY-002061
2020-05-15T14:35:46.923832+08:00 21 [ERROR] [MY-011582] [Rep] Plugin
group_replication reported: 'There was an error when connecting to the donor
server. Please check that group_replication_recovery channel credentials and all
MEMBER_HOST column values of performance_schema.replication_group_members table
are correct and DNS resolvable.'
2020-05-15T14:35:46.923887+08:00 21 [ERROR] [MY-011583] [Rep] Plugin
group_replication reported: 'For details please check
performance_schema.replication_connection_status table and error log messages of
slave I/O for channel group_replication_recovery.'

```

检查 performance_schema.replication_connection_status

```

root@localhost [(none)]>select * from
performance_schema.replication_connection_status\G
...
...
...

```

```

***** 3. row *****
CHANNEL_NAME: group_replication_recovery
GROUP_NAME:
SOURCE_UUID:
THREAD_ID: NULL
SERVICE_STATE: OFF
COUNT_RECEIVED_HEARTBEATS: 0
LAST_HEARTBEAT_TIMESTAMP: 0000-00-00 00:00:00.000000
RECEIVED_TRANSACTION_SET:
LAST_ERROR_NUMBER: 2061
LAST_ERROR_MESSAGE: error connecting to master
'rep@ms81:3316' - retry-time: 60 retries: 1 message: Authentication plugin
'caching_sha2_password' reported error: Authentication requires secure
connection.

...
...
3 rows in set (0.01 sec)

```

退化到recovering状态，遇到连接问题，尝试在change master上增加：

```

root@localhost [(none)]>change master to
master_user='rep1',master_password='rep1',get_master_public_key=1 for channel
'group_replication_recovery';
ERROR 3139 (HY000): CHANGE MASTER with the given parameters cannot be performed
on channel 'group_replication_recovery'.

```

这就尴尬了。

- 临时解决方法

```

[root@ms82 ~]# mysql -h 10.10.30.81 -P 3316 -urepl -prepl
rep@192.168.188.81 [(none)]>exit

[root@ms82 ~]# mysql -S /data/mysql/mysql3316/tmp/mysql.sock
root@localhost [(none)]>stop group_replication;
Query OK, 0 rows affected (4.75 sec)

root@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (5.75 sec)

root@localhost [(none)]>select * from
performance_schema.replication_connection_status\G
***** 1. row *****
CHANNEL_NAME:
GROUP_NAME:
SOURCE_UUID: 29ea8b7f-966d-11ea-937c-
0242c0a8bc51
THREAD_ID: NULL
SERVICE_STATE: OFF
COUNT_RECEIVED_HEARTBEATS: 0
LAST_HEARTBEAT_TIMESTAMP: 0000-00-00 00:00:00.000000
RECEIVED_TRANSACTION_SET: 29ea8b7f-966d-11ea-937c-
0242c0a8bc51:1-530,
3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-343
LAST_ERROR_NUMBER: 0
LAST_ERROR_MESSAGE:

```

```

                LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION:
LAST_QUEUED_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
LAST_QUEUED_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION_START_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION_END_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION:
                QUEUEING_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_START_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
***** 2. row *****
                CHANNEL_NAME: group_replication_applier
                GROUP_NAME: 3260d70c-966e-11ea-ba8b-
0242c0a8bc51
                SOURCE_UUID: 3260d70c-966e-11ea-ba8b-
0242c0a8bc51
                THREAD_ID: NULL
                SERVICE_STATE: ON
COUNT_RECEIVED_HEARTBEATS: 0
                LAST_HEARTBEAT_TIMESTAMP: 0000-00-00 00:00:00.000000
                RECEIVED_TRANSACTION_SET: 29ea8b7f-966d-11ea-937c-
0242c0a8bc51:1-530,
3260d70c-966e-11ea-ba8b-0242c0a8bc51:1-781:787
                LAST_ERROR_NUMBER: 0
                LAST_ERROR_MESSAGE:
                LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION: 3260d70c-966e-11ea-ba8b-
0242c0a8bc51:787
                LAST_QUEUED_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
LAST_QUEUED_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION_START_QUEUE_TIMESTAMP: 2020-05-15 14:38:54.721851
                LAST_QUEUED_TRANSACTION_END_QUEUE_TIMESTAMP: 2020-05-15 14:38:54.721874
                QUEUEING_TRANSACTION:
                QUEUEING_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_START_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
***** 3. row *****
                CHANNEL_NAME: group_replication_recovery
                GROUP_NAME:
                SOURCE_UUID:
                THREAD_ID: NULL
                SERVICE_STATE: OFF
COUNT_RECEIVED_HEARTBEATS: 0
                LAST_HEARTBEAT_TIMESTAMP: 0000-00-00 00:00:00.000000
                RECEIVED_TRANSACTION_SET:
                LAST_ERROR_NUMBER: 0
                LAST_ERROR_MESSAGE:
                LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION:
                LAST_QUEUED_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
LAST_QUEUED_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION_START_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
                LAST_QUEUED_TRANSACTION_END_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION:
                QUEUEING_TRANSACTION_ORIGINAL_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_IMMEDIATE_COMMIT_TIMESTAMP: 0000-00-00 00:00:00.000000
                QUEUEING_TRANSACTION_START_QUEUE_TIMESTAMP: 0000-00-00 00:00:00.000000
3 rows in set (0.00 sec)

```

正规解决方法

```
SET GLOBAL group_replication_recovery_use_ssl = ON;

SET GLOBAL group_replication_recovery_get_public_key = 1; #已合并到操作中

SET GLOBAL group_replication_recovery_public_key_path = 'path to RSA public key file';
```

10.MGR冷启动

将三节点全部关掉

```
mysql > shutdown ;
```

启动node1

```
[root@node1 ~]# mysqld --defaults-file=/data/mysql/mysql3316/my3316.cnf &
[root@node1 ~]# mysql -S /data/mysql/mysql3316/tmp/mysql.sock
root@localhost [(none)]>set global group_replication_bootstrap_group=ON;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (3.16 sec)

root@localhost [(none)]>set global group_replication_bootstrap_group=OFF;
Query OK, 0 rows affected (0.00 sec)

root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| group_replication_applier | f78a6902-9679-11ea-b136-0242c0a8bc51 | node1
|          3316 | ONLINE      | PRIMARY    | 8.0.19         |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
1 rows in set (0.01 sec)
```

启动node2

```
[root@ms82 ~]# mysqld --defaults-file=/data/mysql/mysql3316/my3316.cnf &
[root@ms82 ~]# mysql -S /data/mysql/mysql3316/tmp/mysql.sock
root@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (3.45 sec)

root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+
| group_replication_applier | f78a6902-9679-11ea-b136-0242c0a8bc51 | node1
| 3316 | ONLINE      | PRIMARY      | 8.0.19      |
| group_replication_applier | faaab4c3-9679-11ea-896f-0242c0a8bc52 | node2
| 3316 | ONLINE      | SECONDARY     | 8.0.19      |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

同理, 启动node3

```
[root@ms83 ~]# mysqld --defaults-file=/data/mysql/mysql3316/my3316.cnf &
[root@ms83 ~]# mysql -S /data/mysql/mysql3316/tmp/mysql.sock
root@localhost [(none)]>start group_replication;
Query OK, 0 rows affected (3.45 sec)

root@localhost [(none)]>select * from
performance_schema.replication_group_members;
+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST          |
| MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+
| group_replication_applier | f78a6902-9679-11ea-b136-0242c0a8bc51 | node1
| 3316 | ONLINE      | PRIMARY      | 8.0.19      |
| group_replication_applier | faaab4c3-9679-11ea-896f-0242c0a8bc52 | node2
| 3316 | ONLINE      | SECONDARY     | 8.0.19      |
| group_replication_applier | fb358b40-9679-11ea-94cb-0242c0a8bc53 | node3
| 3316 | ONLINE      | SECONDARY     | 8.0.19      |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```