第二十五节：MySQL主从复制转MGR（mysql-shell）---实战篇

1.环境信息

```
IP  port    role    info
10.10.30.81 3316    node1   master
10.10.30.82 3316    node2   shadow
10.10.30.83 3316    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。

```
[root@node1 ~]# ll /usr/local
total 40
drwxr-xr-x 2 root root 4096 Apr 11  2018 bin
drwxr-xr-x 2 root root 4096 Apr 11  2018 etc
drwxr-xr-x 2 root root 4096 Apr 11  2018 games
drwxr-xr-x 2 root root 4096 Apr 11  2018 include
drwxr-xr-x 2 root root 4096 Apr 11  2018 lib
drwxr-xr-x 2 root root 4096 Apr 11  2018 lib64
drwxr-xr-x 2 root root 4096 Apr 11  2018 libexec
lrwxrwxrwx 1 root root   47 May 13 14:22 myrouter -> /opt/mysql-router-8.0.20-
linux-glibc2.12-x86_64
lrwxrwxrwx 1 root root   49 May 13 14:22 myshell -> /opt/mysql-shell-8.0.20-
linux-glibc2.12-x86-64bit
lrwxrwxrwx 1 root root   41 May 13 14:23 mysql -> /opt/mysql-8.0.19-linux-
glibc2.12-x86_64/
drwxr-xr-x 2 root root 4096 Apr 11  2018 sbin
drwxr-xr-x 5 root root 4096 Dec  4  2018 share
drwxr-xr-x 2 root root 4096 Apr 11  2018 src
```

3.搭建复制环境，并开启增强半同步

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         |
| rpl_semi_sync_master_wait_point           | AFTER_SYNC |
| rpl_semi_sync_slave_enabled               | OFF        |
| rpl_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)
```

slave节点配置

```
root@localhost [(none)]>set global rpl_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=3316,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)
```

slave 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: 3316
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 155
               Relay_Log_File: node2-relay-bin.000002
```

```
               Relay_Log_Pos: 369
       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: 155
             Relay_Log_Space: 576
             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: 813316
                 Master_UUID: 70396ba6-9661-11ea-902e-0242c0a8bc51
            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 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: 1
         Replicate_Rewrite_DB:
                 Channel_Name:
           Master_TLS_Version:
       Master_public_key_path:
        Get_master_public_key: 1
            Network_Namespace:
1 row in set (0.01 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)
```

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 , dtl
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(dtl)
values('duangduangduang');" | mysql -S /data/mysql/mysql3316/tmp/mysql.sock;
sleep 1;done
```

5.转换MGR

Master在线转换为MGR

master配置

```
root@localhost [kk]>create user 'mgr'@'10.10.30.%' identified by 'mgr';
Query OK, 0 rows affected (0.01 sec)

root@localhost [kk]>grant all privileges on *.* to 'mgr'@'10.10.30.%' with grant
option;
Query OK, 0 rows affected (0.02 sec)

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

使用mysh将master转为MGR

```
[root@node1 ~]# mysqlsh
```

```
MySQL Shell 8.0.20

Copyright (c) 2016, 2020, 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 '\?' for help; '\quit' to exit.
 MySQL  JS > \c mgr@10.10.30.81:3306
Creating a session to 'mgr@10.10.30.81:3306'
MySQL Error 2003 (HY000): Can't connect to MySQL server on '10.10.30.81' (111)
 MySQL  JS > \c mgr@10.10.30.81:3316
Creating a session to 'mgr@10.10.30.81:3316'
Please provide the password for 'mgr@10.10.30.81:3316': ***
Save password for 'mgr@10.10.30.81:3316'? [Y]es/[N]o/Ne[v]er (default No): Y
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 863
Server version: 8.0.19 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.


 MySQL  192.168.188.81:3316 ssl  JS > var cl = dba.createCluster('kk')
A new InnoDB cluster will be created on instance '10.10.30.81:3316'.

Validating instance configuration at 10.10.30.81:3316...

This instance reports its own address as node1:3316

Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'node1:33161'.
Use the localAddress option to override.

Creating InnoDB cluster 'kk' on 'node1:3316'...

Adding Seed Instance...
Cluster successfully created. Use Cluster.addInstance() to add MySQL instances.
At least 3 instances are needed for the cluster to be able to withstand up to
one server failure.
```

将shadow加入到MGR

```
 MySQL  10.10.30.81:3316 ssl  JS > cl.addInstance('mgr@10.10.30.82:3316')
Please provide the password for 'mgr@10.10.30.82:3316': ***
Save password for 'mgr@10.10.30.82:3316'? [Y]es/[N]o/Ne[v]er (default No): y
The safest and most convenient way to provision a new instance is through
automatic clone provisioning, which will completely overwrite the state of
'node2:3316' with a physical snapshot from an existing cluster member. To use
this method by default, set the 'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates
ever executed in the cluster were done with GTIDs enabled, there are no purged
transactions and the new instance contains the same GTID set as the cluster or a
subset of it. To use this method by default, set the 'recoveryMethod' option to
'incremental'.

Incremental state recovery was selected because it seems to be safely usable.

ERROR: Cannot add instance '10.10.30.82:3316' to the cluster because it has
asynchronous (master-slave) replication configured and running. Please stop the
slave threads by executing the query: 'STOP SLAVE;'
Cluster.addInstance: The instance '10.10.30.82:3316' is running asynchronous
(master-slave) replication. (RuntimeError)
```

由于复制在运行，无法转换。

shadow停止复制

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

重新用mysh将shadow加入MGR

```
 MySQL  10.10.30.81:3316 ssl  JS > cl.addInstance('mgr@10.10.30.82:3316')
The safest and most convenient way to provision a new instance is through
automatic clone provisioning, which will completely overwrite the state of
'node2:3316' with a physical snapshot from an existing cluster member. To use
this method by default, set the 'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates
ever executed in the cluster were done with GTIDs enabled, there are no purged
transactions and the new instance contains the same GTID set as the cluster or a
subset of it. To use this method by default, set the 'recoveryMethod' option to
'incremental'.

Incremental state recovery was selected because it seems to be safely usable.

NOTE: Group Replication will communicate with other members using 'node2:33161'.
Use the localAddress option to override.

Validating instance configuration at 10.10.30.82:3316...

This instance reports its own address as node2:3316

NOTE: Some configuration options need to be fixed:
+----------------+--------------+---------------+--------------------------
+


```

```
| Variable        | Current Value | Required Value | Note
|
+-----------------+---------------+----------------+----------------------------
+
| binlog_checksum | CRC32         | NONE           | Update the server variable
|
+-----------------+---------------+----------------+----------------------------
+

NOTE: Please use the dba.configureInstance() command to repair these issues.

ERROR: Instance must be configured and validated with
dba.checkInstanceConfiguration() and dba.configureInstance() before it can be
used in an InnoDB cluster.
Cluster.addInstance: Instance check failed (RuntimeError)
```

shadow 关闭binlog_checksum

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

重新用mysh将shadow加入MGR

```
 MySQL  10.10.30.81:3316 ssl  JS > cl.addInstance('mgr@10.10.30.82:3316')
The safest and most convenient way to provision a new instance is through
automatic clone provisioning, which will completely overwrite the state of
'node2:3316' with a physical snapshot from an existing cluster member. To use
this method by default, set the 'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates
ever executed in the cluster were done with GTIDs enabled, there are no purged
transactions and the new instance contains the same GTID set as the cluster or a
subset of it. To use this method by default, set the 'recoveryMethod' option to
'incremental'.

Incremental state recovery was selected because it seems to be safely usable.

NOTE: Group Replication will communicate with other members using 'node2:33161'.
Use the localAddress option to override.
Validating instance configuration at 10.10.30.82:3316...

This instance reports its own address as node2:3316

Instance configuration is suitable.
A new instance will be added to the InnoDB cluster. Depending on the amount of
data on the cluster this might take from a few seconds to several hours.

Adding instance to the cluster...

Monitoring recovery process of the new cluster member. Press ^C to stop
monitoring and let it continue in background.
Incremental state recovery is now in progress.

* Waiting for distributed recovery to finish...
NOTE: 'node2:3316' is being recovered from 'node1:3316'
* Distributed recovery has finished
```

```
The instance '10.10.30.82:3316' was successfully added to the cluster.
```

配置slave参数

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

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

加slave进MGR

```
 MySQL  10.10.30.81:3316 ssl  JS > cl.addInstance('mgr@10.10.30.83:3316')
The safest and most convenient way to provision a new instance is through
automatic clone provisioning, which will completely overwrite the state of
'node3:3316' with a physical snapshot from an existing cluster member. To use
this method by default, set the 'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates
ever executed in the cluster were done with GTIDs enabled, there are no purged
transactions and the new instance contains the same GTID set as the cluster or a
subset of it. To use this method by default, set the 'recoveryMethod' option to
'incremental'.

Incremental state recovery was selected because it seems to be safely usable.

NOTE: Group Replication will communicate with other members using 'node3:33161'.
Use the localAddress option to override.

Validating instance configuration at 10.10.30.83:3316...

This instance reports its own address as node3:3316

Instance configuration is suitable.
A new instance will be added to the InnoDB cluster. Depending on the amount of
data on the cluster this might take from a few seconds to several hours.

Adding instance to the cluster...

Monitoring recovery process of the new cluster member. Press ^C to stop
monitoring and let it continue in background.
State recovery already finished for 'node3:3316'


The instance '10.10.30.83:3316' was successfully added to the cluster.
```

通过查看master的error.log，可以发现

```
2020-05-15T12:06:22.887737+08:00 5 [Warning] [MY-010453] [Server] root@localhost
is created with an empty password ! Please consider switching off the --
initialize-insecure option.
2020-05-15T12:06:26.438849+08:00 0 [Warning] [MY-010101] [Server] Insecure
configuration for --secure-file-priv: Location is accessible to all OS users.
Consider choosing a different directory.
```

```
2020-05-15T12:06:26.439047+08:00 0 [System] [MY-010116] [Server] /opt/mysql-8.0.19-linux-glibc2.12-x86_64/bin/mysqld (mysqld 8.0.19) starting as process 132
2020-05-15T12:06:27.486314+08:00 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2020-05-15T12:06:27.533114+08:00 0 [System] [MY-010931] [Server] /opt/mysql-8.0.19-linux-glibc2.12-x86_64/bin/mysqld: ready for connections. Version: '8.0.19'  socket: '/data/mysql/mysql3316/tmp/mysql.sock'  port: 3316  MySQL Community Server - GPL.
2020-05-15T12:06:27.785050+08:00 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: '/tmp/mysqlx.sock' bind-address: '::' port: 33060
2020-05-15T12:41:33.767064+08:00 915 [ERROR] [MY-011685] [Repl] Plugin group_replication reported: 'The group name option is mandatory'
2020-05-15T12:41:33.767558+08:00 915 [ERROR] [MY-011660] [Repl] Plugin group_replication reported: 'Unable to start Group Replication on boot'
2020-05-15T12:41:33.784632+08:00 915 [Warning] [MY-011735] [Repl] Plugin group_replication reported: '[GCS] Automatically adding IPv4 localhost address to the whitelist. It is mandatory that it is added.'
2020-05-15T12:41:33.784661+08:00 915 [Warning] [MY-011735] [Repl] Plugin group_replication reported: '[GCS] Automatically adding IPv6 localhost address to the whitelist. It is mandatory that it is added.'
2020-05-15T12:41:33.793816+08:00 919 [Warning] [MY-010604] [Repl] Neither --relay-log nor --relay-log-index were used; so replication may break when this MySQL server acts as a slave and has his hostname changed!! Please use '--relay-log=node1-relay-bin' to avoid this problem.
2020-05-15T12:41:33.811301+08:00 919 [System] [MY-010597] [Repl] 'CHANGE MASTER TO FOR CHANNEL 'group_replication_applier' executed'. Previous state master_host='', master_port= 3306, master_log_file='', master_log_pos= 4, master_bind=''. New state master_host='<NULL>', master_port= 0, master_log_file='', master_log_pos= 4, master_bind=''.
2020-05-15T12:41:40.441772+08:00 915 [System] [MY-010597] [Repl] 'CHANGE MASTER TO FOR CHANNEL 'group_replication_recovery' executed'. Previous state master_host='', master_port= 3306, master_log_file='', master_log_pos= 4, master_bind=''. New state master_host='', master_port= 3306, master_log_file='', master_log_pos= 4, master_bind=''.
2020-05-15T12:45:55.135351+08:00 0 [ERROR] [MY-013129] [Server] A message intended for a client cannot be sent there as no client-session is attached. Therefore, we're sending the information to the error-log instead: MY-001158 - Got an error reading communication packets
2020-05-15T12:45:56.808568+08:00 12 [ERROR] [MY-011161] [Server] Semi-sync master failed on net_flush() before waiting for slave reply.
2020-05-15T12:49:18.526465+08:00 0 [ERROR] [MY-013129] [Server] A message intended for a client cannot be sent there as no client-session is attached. Therefore, we're sending the information to the error-log instead: MY-001158 - Got an error reading communication packets
2020-05-15T12:49:19.589392+08:00 1400 [ERROR] [MY-011161] [Server] Semi-sync master failed on net_flush() before waiting for slave reply.
2020-05-15T12:50:17.725710+08:00 0 [ERROR] [MY-013129] [Server] A message intended for a client cannot be sent there as no client-session is attached. Therefore, we're sending the information to the error-log instead: MY-001158 - Got an error reading communication packets
2020-05-15T12:50:27.883726+08:00 1462 [Warning] [MY-011153] [Server] Timeout waiting for reply of binlog (file: mysql-bin.000002, pos: 322577), semi-sync up to file mysql-bin.000002, position 322226.
2020-05-15T12:50:28.912419+08:00 11 [ERROR] [MY-011161] [Server] Semi-sync master failed on net_flush() before waiting for slave reply.
```

```
2020-05-15T12:50:36.516850+08:00 0 [ERROR] [MY-013129] [Server] A message
intended for a client cannot be sent there as no client-session is attached.
Therefore, we're sending the information to the error-log instead: MY-001158 -
Got an error reading communication packets
2020-05-15T12:50:47.085868+08:00 1479 [Warning] [MY-011153] [Server] Timeout
waiting for reply of binlog (file: mysql-bin.000002, pos: 326533), semi-sync up
to file mysql-bin.000002, position 326182.
2020-05-15T12:50:47.106412+08:00 1478 [ERROR] [MY-011161] [Server] Semi-sync
master failed on net_flush() before waiting for slave reply.
```

查看事务session

```
[root@ms81 ~]# while :; do  echo "insert into kk.k1(dtl)
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
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
```

可以推断出， 在转为MGR过程中，由于有选举动作的产生，原事务对master 地址的访问很可能因为原master角色变更而失败，这一点需要注意。

下面停止事务，并检查三节点事务状态：

```
master：
root@localhost [performance_schema]>select count(*) from kk.k1;
+----------+
| count(*) |
+----------+
|     2334 |
+----------+
1 row in set (0.00 sec)

root@localhost [performance_schema]>show master status;
+------------------+----------+--------------+------------------+----------------
----------------------------------------------------------------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
Executed_Gtid_Set
      |
+------------------+----------+--------------+------------------+----------------
----------------------------------------------------------------------------+
| mysql-bin.000002 |   653354 |              |                  | 5a7ef74f-9666-
11ea-b09c-0242c0a8bc51:1-1482,
70396ba6-9661-11ea-902e-0242c0a8bc51:1-904 |
+------------------+----------+--------------+------------------+----------------
----------------------------------------------------------------------------+
1 row in set (0.00 sec)

shadow：
```

```
root@localhost [(none)]>select count(*) from kk.k1;
+----------+
| count(*) |
+----------+
|     2334 |
+----------+
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 |   422814 |              |                  | 5a7ef74f-9666-
11ea-b09c-0242c0a8bc51:1-1482,
70396ba6-9661-11ea-902e-0242c0a8bc51:1-904 |
+------------------+----------+--------------+------------------+-----------------
-------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

slave：

```
root@localhost [(none)]>select count(*) from kk.k1;
+----------+
| count(*) |
+----------+
|     2334 |
+----------+
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 |   363306 |              |                  | 5a7ef74f-9666-
11ea-b09c-0242c0a8bc51:1-1482,
70396ba6-9661-11ea-902e-0242c0a8bc51:1-904 |
+------------------+----------+--------------+------------------+-----------------
-------------------------------------------------------------------------+
1 row in set (0.00 sec)
```