# 使用 MySQL Clone 插件为MGR集群添加节点

原创 黄山谷 2025-07-23 162

在MySQL Group Replication (MGR)架构中，当主库有数据需要增加新节点时，我们可以用备份恢复的方式新增节点。
但是 MySQL 8.0.17 之后的版本， 可以使用Clone插件新增节点无疑是 最快 ，最高效的方法。

## 一、前期准备工作

### 1. 环境检查与要求

- **版本一致性**：确保所有节点运行**MySQL 8.0.17或更高版本**，且大版本号完全一致
- **存储引擎**：确认所有表使用InnoDB引擎（Group Replication强制要求）
- **网络带宽**：建议至少1Gbps网络连接。
- **磁盘空间**：新节点磁盘空间足够，配置建议和源相同。

### 2. 权限配置

**在主库上创建克隆专用账号并授权**：

```
CREATE USER 'clone_user'@'%' IDENTIFIED WITH mysql_native_password BY 'Ihxt6gcgi2wL0Bqi';
GRANT BACKUP_ADMIN, CLONE_ADMIN ON *.* TO 'clone_user'@'%';
```

## 二、新节点初始化配置

### 1. 基础环境准备

参数文件展开

```
## my.cnf for 8.0版本
[client]
port    = 1622
socket  = /mysql/data/mysql.sock

[mysql]
prompt = "\u@mysqldb \R:\m:\s [\d]> "
no_auto_rehash
loose-skip-binary-as-hex

[mysqld]
report_host = 10.10.3.53
skip_name_resolve = ON
user    = mysql
port    = 1622
#主从复制或MGR集群中，server_id记得要不同
#另外，实例启动时会生成 auto.cnf，里面的 server_uuid 值也要不同
#server_uuid的值还可以自己手动指定，只要符合uuid的格式标准就可以
server_id = 162253
basedir = /mysql/app/mysql
datadir = /mysql/data/
socket  = /mysql/data/mysql.sock
pid_file = mysqldb.pid
character_set_server = UTF8MB4
#若你的MySQL数据库主要运行在境外，请务必根据实际情况调整本参数
default_time_zone = "+8:00"
#启用admin_port，连接数爆满等紧急情况下给管理员留个后门
admin_address = '127.0.0.1'
admin_port = 33062
#performance setttings
default_authentication_plugin=mysql_native_password
lock_wait_timeout = 3600
open_files_limit    = 65535
back_log = 1024
max_connections = 512
max_connect_errors = 1000000
```

黄山谷
关注

| 27 | 8 | 3K+ |
|---|---|---|
| 文章 | 粉丝 | 浏览量 |

获得了 10 次点赞
内容获得 4 次评论
获得了 21 次收藏

热门文章

- ORA-00054: 资源正忙, 但指定以 NOWAIT 方式获取资源 或者超时失效 2024-11-06 612浏览
- 学习 PostgreSQL 之 repmgr 2025-02-21 311浏览
- gdisk 标准分区扩容 2025-01-01 255浏览
- 学习 PostgresSQL 高可用之 patroni + etcd 2025-05-29 246浏览
- Oracle SQLcl 一款强大小工具的使用 2024-11-13 239浏览

在线实训环境入口

MySQL在线实训环境
查看详情 »

最新文章

- 使用 mariadb-binlog --flashback 参数恢复 MySQL binlog 删除的数据 16小时前 53浏览
- MySQL Innodb Cluster 配置 2025-07-28 35浏览
- 分享一个 patroni vip 的回调脚本 2025-07-03 39浏览
- 用 vip-manager 控制 pg patroni 架构中 vip 的漂移 2025-07-01 112浏览
- PostgreSQL 获取对象 DDL 插件 ddlx or pg_dbms_metadata 2025-06-17 40浏览

目录

```
table_open_cache = 1024
table_definition_cache = 1024
thread_stack = 512K
sort_buffer_size = 4M
join_buffer_size = 4M
read_buffer_size = 8M
read_rnd_buffer_size = 4M
bulk_insert_buffer_size = 64M
thread_cache_size = 768
interactive_timeout = 600
wait_timeout = 600
tmp_table_size = 32M
max_heap_table_size = 32M

#log settings
log_timestamps = SYSTEM
log_error = /mysql/data//error.log
log_error_verbosity = 3
slow_query_log = 1
log_slow_extra = 1
slow_query_log_file = /mysql/data//slow.log
long_query_time = 0.1
log_queries_not_using_indexes = 1
log_throttle_queries_not_using_indexes = 60
min_examined_row_limit = 100
log_slow_admin_statements = 1
log_slow_slave_statements = 1
log_bin = /mysql/data//mybinlog
binlog_format = ROW
sync_binlog = 1 #MGR环境中由其他节点提供容错性，可不设置双1以提高本地节点性能
binlog_cache_size = 4M
max_binlog_cache_size = 2G
max_binlog_size = 1G
binlog_rows_query_log_events = 1
binlog_expire_logs_seconds = 604800
#MySQL 8.0.22前，想启用MGR的话，需要设置binlog_checksum=NONE才行
binlog_checksum = CRC32
gtid_mode = ON
enforce_gtid_consistency = TRUE

#myisam settings
key_buffer_size = 32M
myisam_sort_buffer_size = 128M

#replication settings
relay_log_recovery = 1
slave_parallel_type = LOGICAL_CLOCK
slave_parallel_workers = 64 #可以设置为逻辑CPU数量的2倍
binlog_transaction_dependency_tracking = WRITESET
slave_preserve_commit_order = 1
slave_checkpoint_period = 2
replication_optimize_for_static_plugin_config = ON
replication_sender_observe_commit_only = ON

#mgr settings
loose-plugin_load_add = 'mysql_clone.so'
loose-plugin_load_add = 'group_replication.so'
loose-group_replication_group_name = "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaa1"
#MGR本地节点IP:PORT，请自行替换
loose-group_replication_local_address = "10.10.3.53:33061"
#MGR集群所有节点IP:PORT，请自行替换
loose-group_replication_group_seeds = "10.10.3.51:33061,10.10.3.52:33061,10.10.3.53:33061"
loose-group_replication_start_on_boot = OFF
loose-group_replication_bootstrap_group = OFF
loose-group_replication_exit_state_action = READ_ONLY
loose-group_replication_flow_control_mode = "DISABLED"
loose-group_replication_single_primary_mode = ON
loose-group_replication_communication_max_message_size = 10M
loose-group_replication_unreachable_majority_timeout = 30
loose-group_replication_member_expel_timeout = 5
loose-group_replication_autorejoin_tries = 288
loose-group_replication_enforce_update_everywhere_checks = OFF
#innodb settings
transaction_isolation = REPEATABLE-READ
innodb_buffer_pool_size = 45056M
innodb_buffer_pool_instances = 4
innodb_data_file_path = ibdata1:12M:autoextend
innodb_flush_log_at_trx_commit = 1 #MGR环境中由其他节点提供容错性，可不设置双1以提高本地节点性能
innodb_log_buffer_size = 32M
innodb_log_file_size = 1G #如果线上环境的TPS较高，建议加大至1G以上，如果压力不大可以调小
innodb_log_files_in_group = 3
loose-innodb_redo_log_capacity = 3G
```

```
innodb_max_undo_log_size = 4G
# 根据您的服务器IOPS能力适当调整
# 一般配普通SSD盘的话，可以调整到 10000 - 20000
# 配置高端PCIe SSD卡的话，则可以调整的更高，比如 50000 - 80000
innodb_io_capacity = 4000
innodb_io_capacity_max = 8000
innodb_open_files = 65535
innodb_flush_method = O_DIRECT
innodb_lru_scan_depth = 4000
innodb_lock_wait_timeout = 10
innodb_rollback_on_timeout = 1
innodb_print_all_deadlocks = 1
innodb_online_alter_log_max_size = 4G
innodb_print_ddl_logs = 1
innodb_status_file = 1
#注意: 开启 innodb_status_output & innodb_status_output_locks 后, 可能会导致log_error文件增长较
innodb_status_output = 0
innodb_status_output_locks = 1
innodb_sort_buffer_size = 67108864
innodb_adaptive_hash_index = OFF
#提高索引统计信息精确度
innodb_stats_persistent_sample_pages = 500
innodb_adaptive_hash_index = 0
loose-sql_generate_invisible_primary_key = ON

#innodb monitor settings
innodb_monitor_enable = "module_innodb"
innodb_monitor_enable = "module_server"
innodb_monitor_enable = "module_dml"
innodb_monitor_enable = "module_ddl"
innodb_monitor_enable = "module_trx"
innodb_monitor_enable = "module_os"
innodb_monitor_enable = "module_purge"
innodb_monitor_enable = "module_log"
innodb_monitor_enable = "module_lock"
innodb_monitor_enable = "module_buffer"
innodb_monitor_enable = "module_index"
innodb_monitor_enable = "module_ibuf_system"
innodb_monitor_enable = "module_buffer_page"
#innodb_monitor_enable = "module_adaptive_hash"

#pfs settings
performance_schema = 1
#performance_schema_instrument = '%memory%=on'
loose-performance_schema_instrument = '%lock%=on'

[mysqldump]
quick
```

### 2. 初始化数据目录

```
mysqld --defaults-file=/mysql/conf/my.cnf --initialize --user=mysql --basedir=/mysql/app/my
```

启动

```
mysqld_safe --defaults-file=/mysql/conf/my.cnf --user=mysql &
```

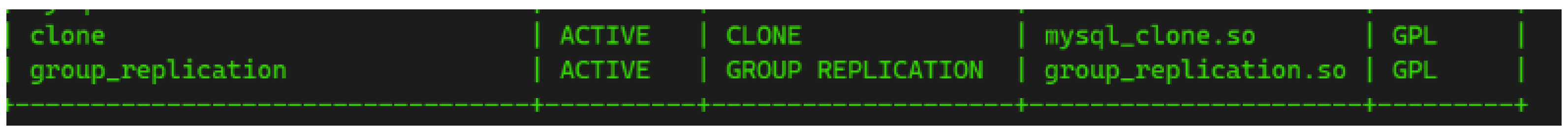
登陆、改密码

```
cat /mysql/data/error.log | grep password
cat /mysql/data/error.log | grep 'temporary password is' | grep -o 'root@localhost: .*' | c

此处通过sock登陆
mysql -uroot -p  -P 1622 -S /mysql/data/mysql.sock

alter user root@'localhost' identified with mysql_native_password by 'GKmVGe1dOZGJ8DYTb5zuF
flush privileges;
```

## 三、执行克隆操作

### 1. 在新节点安装Clone插件

```
SHOW PLUGINS;
INSTALL PLUGIN clone SONAME 'mysql_clone.so';
```



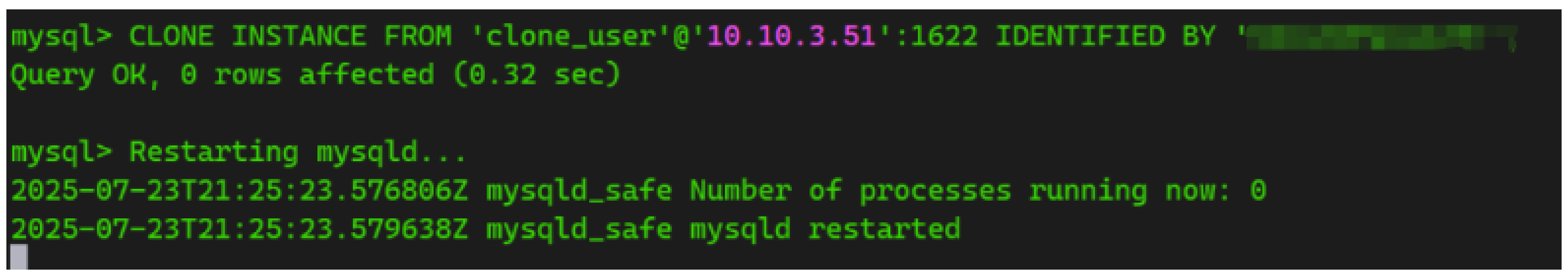
```
| clone                            | ACTIVE   | CLONE              | mysql_clone.so       | GPL     |
| group_replication                | ACTIVE   | GROUP REPLICATION  | group_replication.so | GPL     |
+----------------------------------+----------+--------------------+----------------------+---------+
```

### 2. 设置克隆源

```
SET GLOBAL clone_valid_donor_list = '10.10.3.51:1622';
```

### 3. 启动克隆过程

```
CLONE INSTANCE FROM 'clone_user'@'10.10.3.51':1622 IDENTIFIED BY 'Ihxt6gcgi2wL0Bqi';
or
--指定路径
CLONE INSTANCE FROM 'clone_user'@'10.10.3.51':1622
IDENTIFIED BY 'Ihxt6gcgi2wL0Bqi'
DATA DIRECTORY = '/mysql/data/';
```

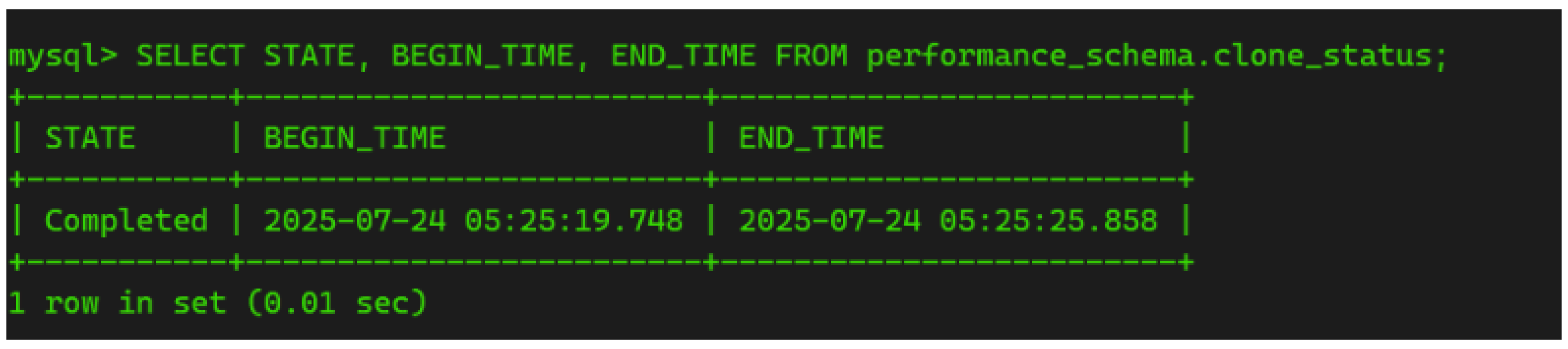


```
mysql> CLONE INSTANCE FROM 'clone_user'@'10.10.3.51':1622 IDENTIFIED BY '[redacted]
Query OK, 0 rows affected (0.32 sec)

mysql> Restarting mysqld...
2025-07-23T21:25:23.576806Z mysqld_safe Number of processes running now: 0
2025-07-23T21:25:23.579638Z mysqld_safe mysqld restarted
```

**重要说明**：

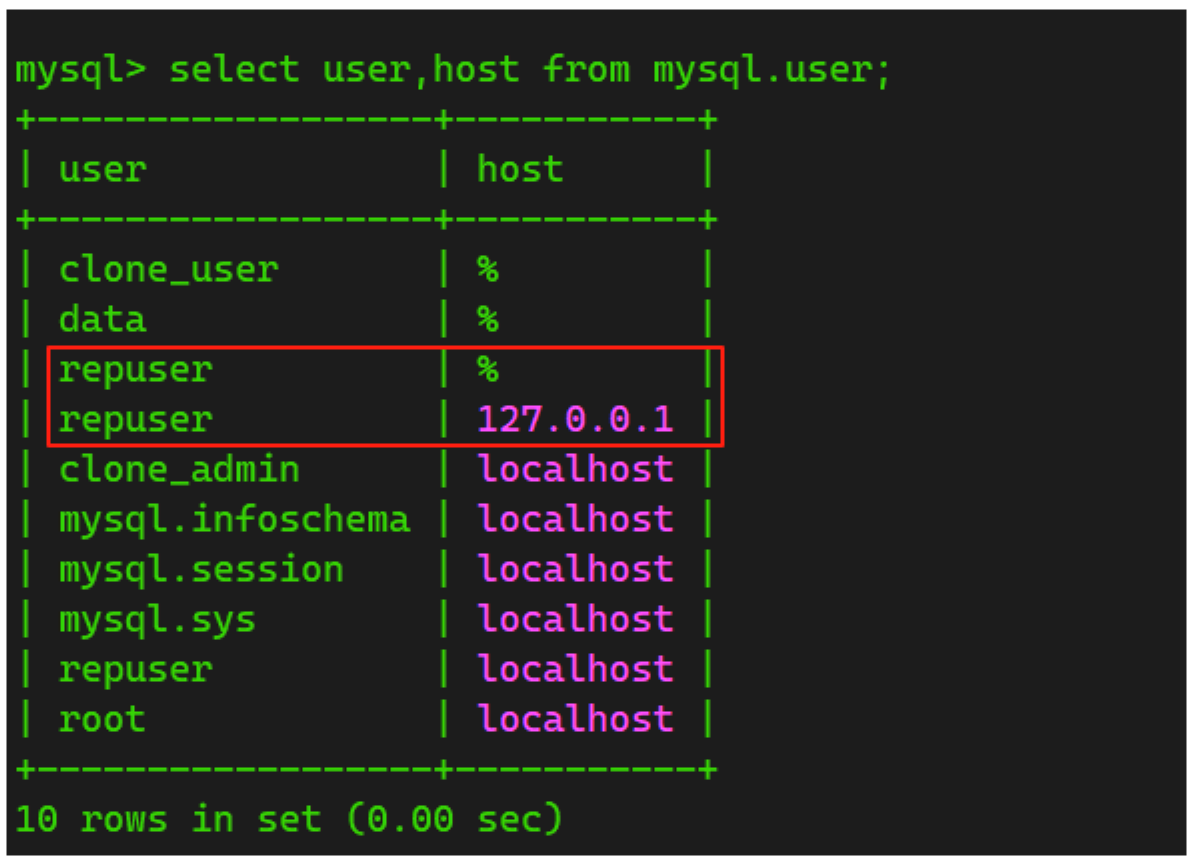
- **克隆过程会自动覆盖新节点的数据目录**
- 完成后MySQL服务会自动重启
- 可通过以下命令监控进度：

```
    SELECT STATE, BEGIN_TIME, END_TIME FROM performance_schema.clone_status;
```



```
mysql> SELECT STATE, BEGIN_TIME, END_TIME FROM performance_schema.clone_status;
+-----------+-------------------------+-------------------------+
| STATE     | BEGIN_TIME              | END_TIME                |
+-----------+-------------------------+-------------------------+
| Completed | 2025-07-24 05:25:19.748 | 2025-07-24 05:25:25.858 |
+-----------+-------------------------+-------------------------+
1 row in set (0.01 sec)
```

**可以看到用户都复制过来了**



```
mysql> select user,host from mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| clone_user       | %         |
| data             | %         |
| repuser          | %         |
| repuser          | 127.0.0.1 |
| clone_admin      | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| repuser          | localhost |
| root             | localhost |
+------------------+-----------+
10 rows in set (0.00 sec)
```

## 四、加入MGR集群

### 1. 配置白名单

在所有现有节点和新节点上执行：(可选)

```
SET GLOBAL group_replication_ip_whitelist='10.10.3.51,10.10.3.52,10.10.3.53';
or
SET GLOBAL group_replication_ip_whitelist='10.10.3.0/24';  或者使用CIDR表示法(如果IP在同一子网)
```

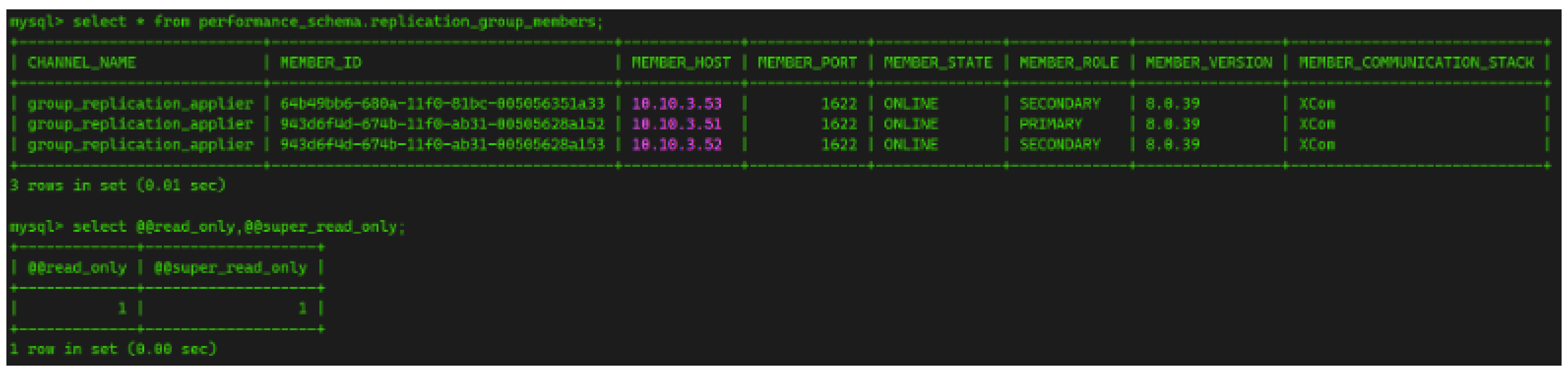
### 2、 加入集群

```
在线修改、每台机器都修改参数
1) 先全局修改下
SET GLOBAL group_replication_group_seeds='10.10.3.51:33061,10.10.3.52:33061,10.10.3.53:3306
2) 参数文件修改
loose-group_replication_group_seeds = "10.10.3.51:33061,10.10.3.52:33061,10.10.3.53:33061"
```

### 3、开启新节点复制

– 启动组复制 --这里注意防火墙策略

```
    start group_replication;
    select * from performance_schema.replication_group_members;
    select @@read_only,@@super_read_only;
```

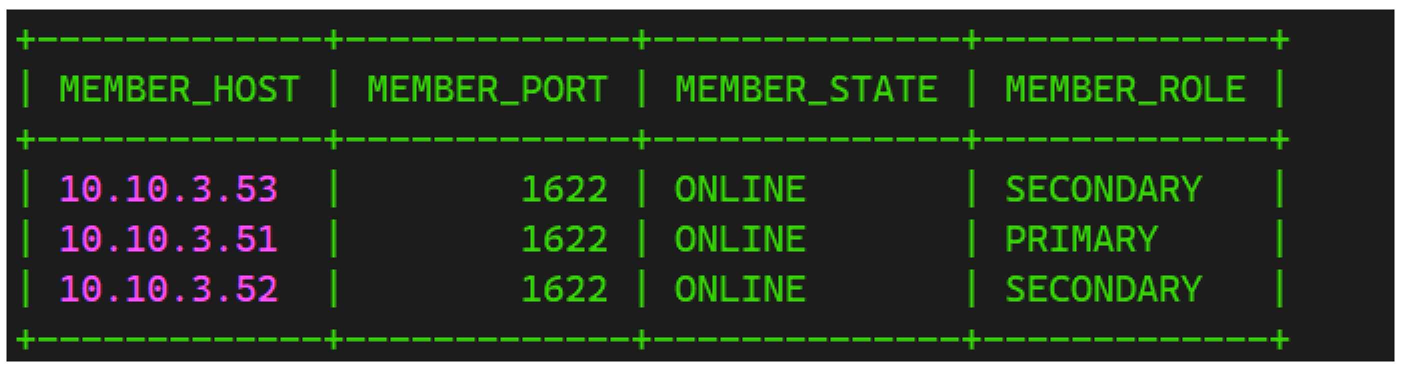


```
mysql> select * from performance_schema.replication_group_members;
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST | MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+----------------------------+
| group_replication_applier | 64b49bb6-680a-11f0-81bc-005056351a33 | 10.10.3.53  |        1622 | ONLINE       | SECONDARY   | 8.0.39         | XCom                       |
| group_replication_applier | 943d6f4d-674b-11f0-ab31-00505628a152 | 10.10.3.51  |        1622 | ONLINE       | PRIMARY     | 8.0.39         | XCom                       |
| group_replication_applier | 943d6f4d-674b-11f0-ab31-00505628a153 | 10.10.3.52  |        1622 | ONLINE       | SECONDARY   | 8.0.39         | XCom                       |
+---------------------------+--------------------------------------+-------------+-------------+--------------+-------------+----------------+----------------------------+
3 rows in set (0.01 sec)

mysql> select @@read_only,@@super_read_only;
+-------------+-------------------+
| @@read_only | @@super_read_only |
+-------------+-------------------+
|           1 |                 1 |
+-------------+-------------------+
1 row in set (0.00 sec)
```

### 4. 验证节点状态

```
SELECT
    MEMBER_HOST,
    MEMBER_PORT,
    MEMBER_STATE,
    MEMBER_ROLE
FROM performance_schema.replication_group_members;
```

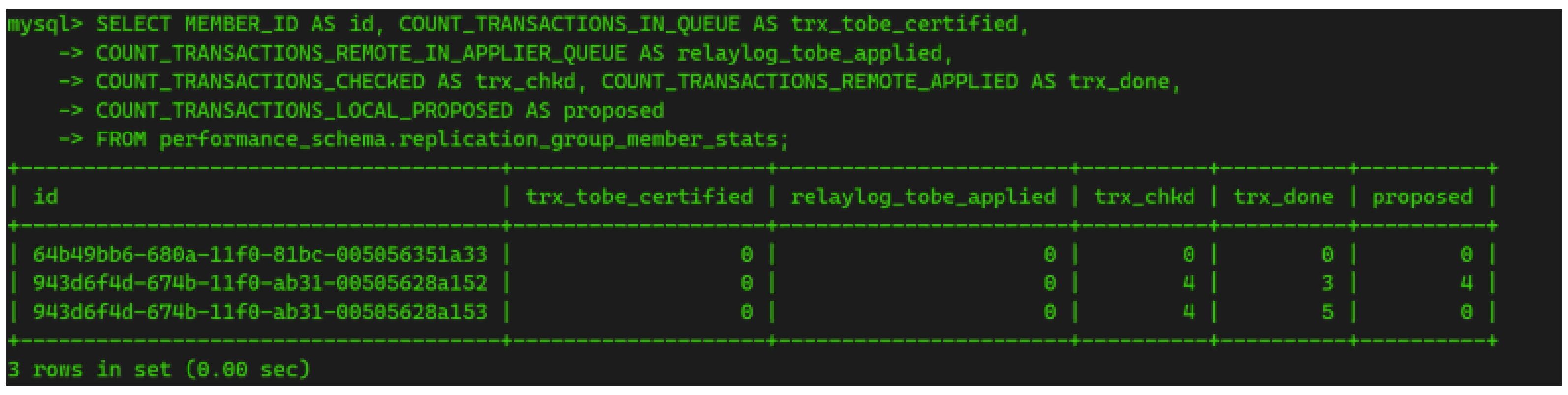
正常状态下应显示新节点为 ONLINE 状态。



```
+-------------+-------------+--------------+-------------+
| MEMBER_HOST | MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE |
+-------------+-------------+--------------+-------------+
| 10.10.3.53  |        1622 | ONLINE       | SECONDARY   |
| 10.10.3.51  |        1622 | ONLINE       | PRIMARY     |
| 10.10.3.52  |        1622 | ONLINE       | SECONDARY   |
+-------------+-------------+--------------+-------------+
```

5、监控转态

```
SELECT MEMBER_ID AS id, COUNT_TRANSACTIONS_IN_QUEUE AS trx_tobe_certified,
COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE AS relaylog_tobe_applied,
COUNT_TRANSACTIONS_CHECKED AS trx_chkd, COUNT_TRANSACTIONS_REMOTE_APPLIED AS trx_done,
COUNT_TRANSACTIONS_LOCAL_PROPOSED AS proposed
FROM performance_schema.replication_group_member_stats;
```



```
mysql> SELECT MEMBER_ID AS id, COUNT_TRANSACTIONS_IN_QUEUE AS trx_tobe_certified,
    -> COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE AS relaylog_tobe_applied,
    -> COUNT_TRANSACTIONS_CHECKED AS trx_chkd, COUNT_TRANSACTIONS_REMOTE_APPLIED AS trx_done,
    -> COUNT_TRANSACTIONS_LOCAL_PROPOSED AS proposed
    -> FROM performance_schema.replication_group_member_stats;
+--------------------------------------+--------------------+-----------------------+----------+----------+----------+
| id                                   | trx_tobe_certified | relaylog_tobe_applied | trx_chkd | trx_done | proposed |
+--------------------------------------+--------------------+-----------------------+----------+----------+----------+
| 64b49bb6-680a-11f0-81bc-005056351a33 |                  0 |                     0 |        0 |        0 |        0 |
| 943d6f4d-674b-11f0-ab31-00505628a152 |                  0 |                     0 |        4 |        3 |        4 |
| 943d6f4d-674b-11f0-ab31-00505628a153 |                  0 |                     0 |        4 |        5 |        0 |
+--------------------------------------+--------------------+-----------------------+----------+----------+----------+
3 rows in set (0.00 sec)
```

relaylog_tobe_applied 的值表示远程事务写到relay log后，等待回放的事务队列，trx_tobe_certified 表示等待被认证的事务队列大小，这二者任何一个值大于0，都表示当前有一定程度的延迟。

墨力计划 数据库实操 mysql

「喜欢这篇文章，您的关注和赞赏是给作者最好的鼓励」

关注作者 赞赏

【版权声明】本文为墨天轮用户原创内容，转载时必须标注文章的来源（墨天轮），文章链接，文章作者等基本信息，否则作者和墨天轮有权追究责任。如果您发现墨天轮中有涉嫌抄袭或者侵权的内容，欢迎发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

## 评论

分享你的看法，一起交流吧～

锁钥 LV.5

使用 MySQL Clone 插件为MGR集群添加节点

16天前 点赞 评论

## 相关阅读

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/25期）
墨天轮小助手 469次阅读 2025-07-25 15:54:18

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/17期）
墨天轮小助手 436次阅读 2025-07-17 15:31:18

Oracle19c rac for centos7图形化部署实战
飞天 397次阅读 2025-08-04 21:55:41

墨天轮「实操看我的」数据库主题征文活动启动
墨天轮编辑部 379次阅读 2025-07-22 16:11:27

Oracle AWR夺命33问，你能过几关？
陈举超 356次阅读 2025-07-26 09:12:53

Oracle RAC+ADG switchover 切换演练流程
淘气 337次阅读 2025-07-22 17:38:51

Oracle 常见的33个等待事件
Digital Observer 334次阅读 2025-07-29 16:10:47

性能优化-失败的SQL改写
布衣 274次阅读 2025-08-02 07:43:16

金仓KingbaseES v9（SQL Server 兼容版）初体验
jiayou 249次阅读 2025-07-23 08:34:14

面试:已知100多个数据库CVE漏洞编号，如何快速查询这些漏洞影响的数据库版本、详细漏洞说明等？
陈举超 235次阅读 2025-07-24 13:00:34