

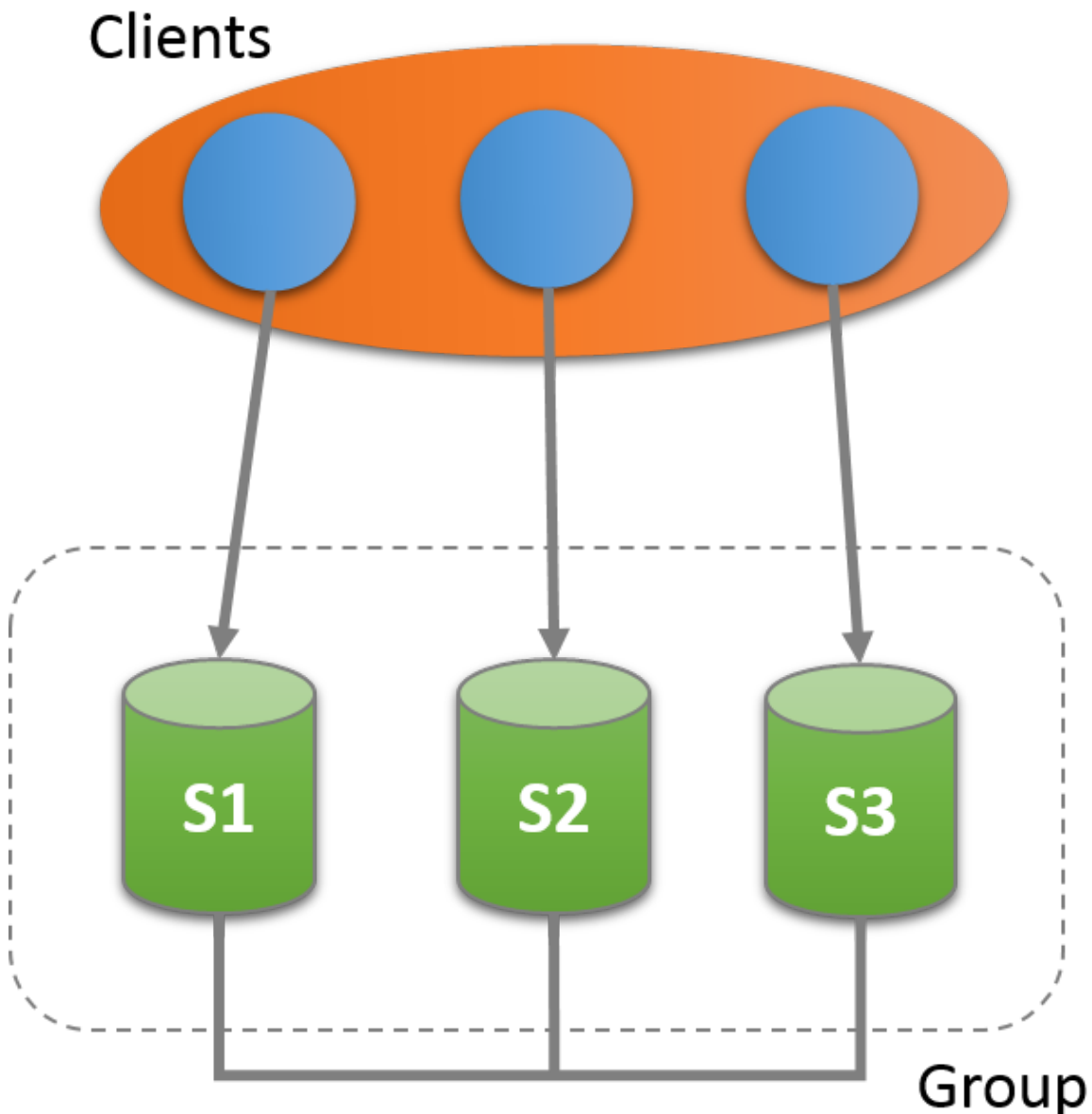
# MGR部署：单主模式---基础篇

MySQL组复制作为MySQL服务器的插件提供，并且集群中的每个MySQL server都需要配置和安装插件。

## 1.在单主模式下部署MGR：

集群中的每个MySQL实例都可以在独立的物理主机上运行，这是部署MGR的推荐方法。本节着重说明如何部署一个三节点的MGR集群，每个实例在不同的主机上运行。

如下图片展示了一个部署在不同主机上的三节点MGR集群的架构：



### 1.1初始化安装：

配置启动MySQL的步骤略过，不是本系列的重点，下面是配置文件的一个示例：

#通用的MySQL server配置本文不会指出，下面的配置只会针对于搭建MGR集群需要重点关注的配置给出相应的说明。

####server端配置：

#禁用MyISAM存储引擎的情况下，使用mysql\_upgradeMySQL升级实例时（在MySQL 8.0.16之前），可能会升级失败，可以在运行mysql\_upgrade时重新启用该存储引擎，然后在重新启动服务器时再次禁用掉。

```
disabled_storage_engines="MyISAM,BLACKHOLE,FEDERATED,ARCHIVE,MEMORY"

server_id=1

gtid_mode=ON

enforce_gtid_consistency=ON

binlog_checksum=NONE      #8.0.20及其之前版本需要开启此参数

log_bin=binlog            #8.0.3及其之前的版本需要开启

log_slave_updates=ON

binlog_format=ROW

master_info_repository=TABLE

relay_log_info_repository=TABLE

transaction_write_set_extraction=XXHASH64      #8.0.2及其之前需要开启此参数

####MGR配置部分：

plugin_load_add='group_replication.so'

group_replication_group_name="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaaa"

group_replication_start_on_boot=off      #是否需要随着MySQL Server一并启动MGR插件，在有PRIMARY节点时，建议重新设置为ON。

group_replication_local_address= "s1:33061"      #当前节点service地址

group_replication_group_seeds= "s1:33061,s2:33061,s3:33061"      #种子节点的service地址

group_replication_bootstrap_group=off      #是否使用这个Server来引导集群

loose-group_replication_enforce_update_everywhere_checks=off

loose-group_replication_ip_whitelist='0.0.0.0/0'

loose-group_replication_single_primary_mode=on

loose-group_replication_gtid_assignment_block_size=1

#为每个成员保留的连续GTID的数量，默认值为1000000

loose-group_replication_clone_threshold=50000      #超过指定事务延迟数量后触发Clone Plugin重搭备库
```

####clone插件相关:

```
plugin-load-add=mysql_clone.so
loose_clone_autotune_concurrency=ON
loose_clone_max_concurrency=16
loose_clone_max_data_bandwidth=100
loose_clone_max_network_bandwidth=100
loose_clone_ddl_timeout=300
```

####在这里为什么要用这个插件，我们后面会详细讨论。

## 2.启动集群中的所有节点:

此过程按照正常启动三个单机MySQL实例的步骤即可，具体步骤略过。

如果没有在配置文件中指定组复制插件，则实例启动时不会自动安装组复制插件，需要在实例启来时，在每个节点上执行如下命令手动安装组复制插件:

```
INSTALL PLUGIN group_replication SONAME 'group_replication.so';
```

安装成功后在每个节点执行show plugins;输出如下:

```
root@localhost:mysql.sock 11:29:16 [(none)]> show plugins;
+-----+-----+-----+-----+
| Name                               | Status | Type               | Library |
| License |
+-----+-----+-----+-----+
| group_replication                 | ACTIVE | GROUP REPLICATION |         |
| group_replication.so | GPL   |
| clone                             | ACTIVE | CLONE              |         |
| mysql_clone.so                   | GPL   |
+-----+-----+-----+-----+
-----+-----+
```

## 3.搭建MGR:

### 3.1创建用户:

```
mysql> SET SQL_LOG_BIN=0;

mysql> CREATE USER *repl_user*('@%') IDENTIFIED BY 'user_repl';

mysql> GRANT REPLICATION SLAVE ON *.* TO *rpl_user*('@%');

mysql> GRANT BACKUP_ADMIN ON *.* TO 'repl_user'('@%');

mysql> FLUSH PRIVILEGES;

mysql> SET SQL_LOG_BIN=1;
```

## 3.2配置复制通道:

```
mysql> CHANGE MASTER TO MASTER_USER='repl_user*', MASTER_PASSWORD='user_rep1'  
FOR CHANNEL 'group_replication_recovery';
```

## 3.3启动MGR集群:

### 3.3.1引导节点启动:

```
mysql> SET GLOBAL group_replication_bootstrap_group=ON;  
  
mysql> START GROUP_REPLICATION;  
  
mysql> SET GLOBAL group_replication_bootstrap_group=OFF;
```

或者通过指定复制通道用户来启动:

```
mysql> SET GLOBAL group_replication_bootstrap_group=ON;  
  
mysql> START GROUP_REPLICATION USER='repl_user*', PASSWORD='*user_rep1';  
  
mysql> SET GLOBAL group_replication_bootstrap_group=OFF;
```

检查集群是否启动且primary节点是否成功加入集群:

```
root@localhost:mysql.sock 11:43:53 [(none)]> SHOW STATUS LIKE  
'group_replication_primary_member';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| group_replication_primary_member | bfd7119e-0d2d-11eb-8c01-fa481f05d800 |  
+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT * FROM performance_schema.replication_group_members;
```

```
root@localhost:mysql.sock 11:43:46 [(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 | aeb1a9bf-0d2d-11eb-ac0f-faad9ffd2500 | mgr1 | 3306 | ONLINE | SECONDARY | 8.0.21 |  
| group_replication_applier | bfd7119e-0d2d-11eb-8c01-fa481f05d800 | mgr2 | 3306 | ONLINE | PRIMARY | 8.0.21 |  
| group_replication_applier | d6b2dae8-0d2d-11eb-9687-fae661bd6b00 | mgr3 | 3306 | ONLINE | SECONDARY | 8.0.21 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

### 3.3.2非引导节点启动:

```
SET SQL_LOG_BIN=0;  
  
CREATE USER repl_user*('@%') IDENTIFIED BY '*user_rep1';  
  
GRANT REPLICATION SLAVE ON *.* TO repl_user*('@%');  
  
GRANT BACKUP_ADMIN ON *.* TO repl_user*('@%');  
  
FLUSH PRIVILEGES;
```

```
SET SQL_LOG_BIN=1;
```

```
CHANGE MASTER TO MASTER_USER='repl_user*', MASTER_PASSWORD='*password*' FOR  
CHANNEL 'group_replication_recovery';
```

```
mysql> START GROUP_REPLICATION;
```

```
mysql> START GROUP_REPLICATION USER='*rpl_user*', PASSWORD='*password*';    #用此  
方式启动集群也可以
```

重复上述步骤，将剩下的节点加入集群中即可。

注意：在非引导节点上执行上述命令，返回结果为OK集群才算启动成功。

## 4.集群数据同步：

---

将组成员加入集群时，组复制使用分布式恢复过程对其进行数据同步。分布式恢复涉及使用group\_replication\_recovery的复制通道将事务从引导节点的二进制日志传输到新加入成员。

因此，必须设置具有正确权限的复制用户，以便组复制可以建立直接的成员到成员复制通道。如果已将组成员设置为支持将远程克隆操作用作分布式恢复的一部分（可从MySQL 8.0.17获得），则复制用户还将用作引导节点上的克隆用户，并且需要正确的权限。