# Docker ForMysql

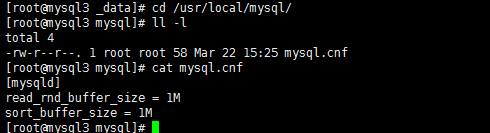
## DockerMysql单机

mkdir /usr/local/mysql

cd /usr/local/mysql

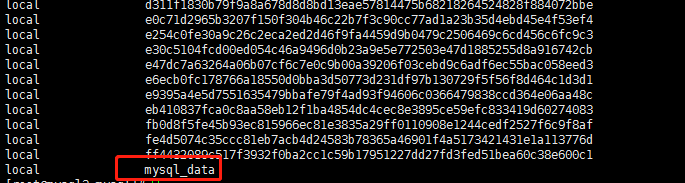
touch mysql.conf

mysql.cnf文件是我们自定义的配置文件，后面会挂载到磁盘中

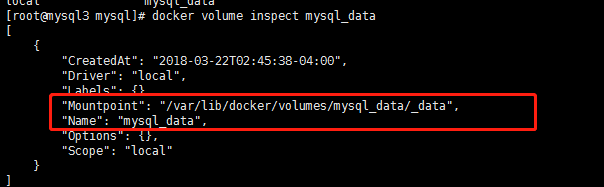


docker volume mysql\_data 创建一个mysql\_data磁盘卷，磁盘卷放置于docker下面

docker volume ls列出刚刚创建的磁盘卷



查看磁盘卷放置的目录



启动docker容器

docker run --name mysqlSingle --env MYSQL\_ROOT\_PASSWORD=123 -p3308:3306 -v /usr/local/mysql/mysql.cnf:/etc/mysql/conf.d/my.cnf -v mysql\_data:/usr/mysql/lib -d mysql:5.7

启动docker容器

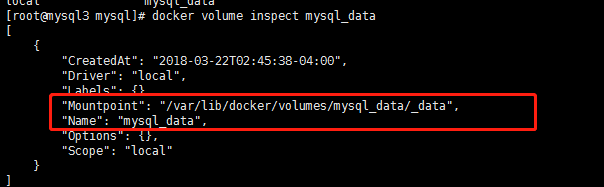
-v 是挂载外部磁盘卷

刚刚创建的mysql,.cnf挂载到/etc/mysql/conf.d/my.cnf

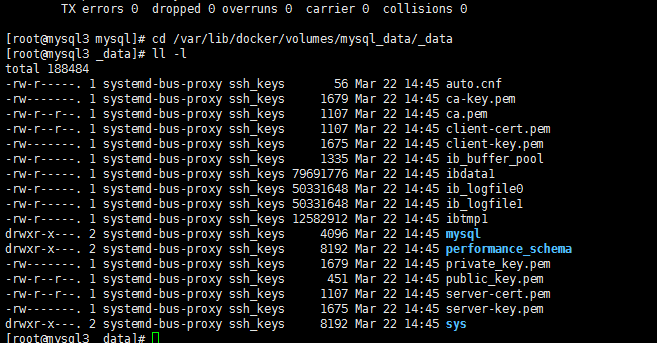
mysql\_data 接收数据的文件mysql\_data



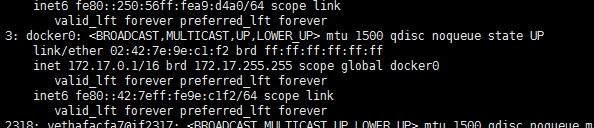
查看mysql的数据文件



所有msql数据



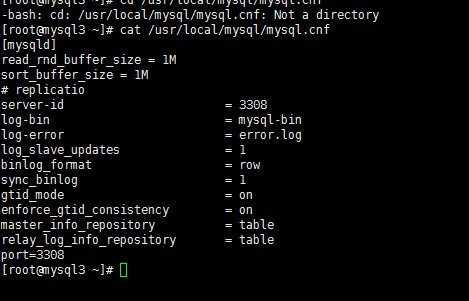
Docker网络属于桥接模式



## DockerMysql主从

### Master

|  |
| --- |
| read\_rnd\_buffer\_size = 1M  sort\_buffer\_size = 1M  # replicatio  server-id = 3308 //服务ID 确保两台机器必须不同  log-bin = mysql-bin  log-error = /var/log/mysql/error.log  log\_slave\_updates = 1  binlog\_format = row  sync\_binlog = 1  gtid\_mode = on  enforce\_gtid\_consistency = on  master\_info\_repository = table  relay\_log\_info\_repository = table  port=3308 |



启动mysql

Docker volume create mysql\_log

docker run --name mysqlSingleMgr --env MYSQL\_ROOT\_PASSWORD=123 -p3308:3308 -v /usr/local/mysql/mysqlMgr.cnf:/etc/mysql/conf.d/my.cnf -v mysql\_data:/usr/mysql/lib -v mysql\_log:/var/log/mysql -d mysql:5.7

GRANT REPLICATION SLAVE ON \*.\* TO 'mysql03'@'%' IDENTIFIED BY 'mysql03';//创建用户，赋予REPLICATION 权限

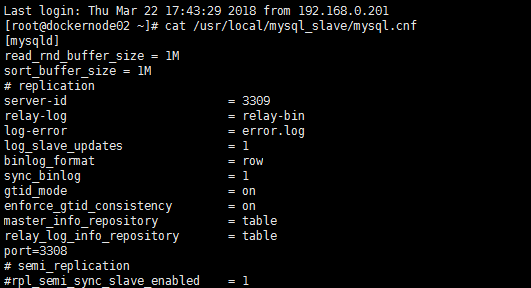
INSTALL PLUGIN rpl\_semi\_sync\_master SONAME 'semisync\_master.so';//安装集群心态检测包

SET GLOBAL rpl\_semi\_sync\_master\_enabled = 1;//设置集群心跳检测包可用

### Slave

编辑mysql.cnf

|  |
| --- |
| read\_rnd\_buffer\_size = 1M  sort\_buffer\_size = 1M  # replication  server-id = 3309  relay-log = relay-bin  log-error = /var/log/mysql/error.log  log\_slave\_updates = 1  binlog\_format = row  sync\_binlog = 1  gtid\_mode = on  enforce\_gtid\_consistency = on  master\_info\_repository = table  relay\_log\_info\_repository = table  port=3308 |



启动mysql

docker run --name mysqlSingle --env MYSQL\_ROOT\_PASSWORD=123 -p3308:3308 -v /usr/local/mysql\_slave/mysql.cnf:/etc/mysql/conf.d/my.cnf -v mysql\_data:/usr/mysql/lib -d mysql:5.7

INSTALL PLUGIN rpl\_semi\_sync\_slave SONAME 'semisync\_slave.so';

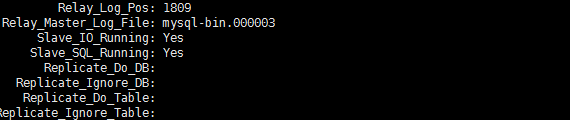
SET GLOBAL rpl\_semi\_sync\_slave\_enabled = 1;

Change master to   
master\_host='192.168.0.115',master\_port=3308,master\_user='mysql03',master\_password='mysql03'

start slave;

show slave status\G;

查看



## Docker Mysql Mgr

### Master:

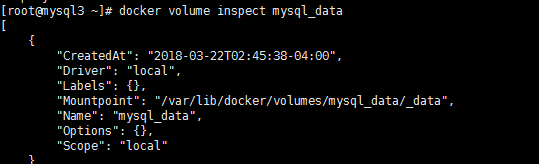
Vi /usr/local/mysql/mysqlMgrHost.cnf

|  |
| --- |
| [mysqld]  read\_rnd\_buffer\_size = 1M  sort\_buffer\_size = 1M  log-error=/var/log/mysql/mysql\_error.log  # replicatio  server-id = 3308  log-bin = mysql-bin  #log-error = error.log  log\_slave\_updates = 1  binlog\_format = row  sync\_binlog = 1  gtid\_mode = on  enforce\_gtid\_consistency = on  master\_info\_repository = table  relay\_log\_info\_repository = table  port=3308  log\_slave\_updates = ON  binlog\_checksum = NONE  binlog\_format = ROW  transaction\_isolation = READ-COMMITTED  gtid\_mode = ON  enforce\_gtid\_consistency = ON  # GR 配置项 其中loose前缀表示若Group Replication plugin未加载 mysql server仍继续启动  transaction\_write\_set\_extraction = XXHASH64  loose-group-replication-ip-whitelist="192.168.0.0/24"  loose-group\_replication\_single\_primary\_mode=true  loose-group\_replication\_enforce\_update\_everywhere\_checks=false  loose-group\_replication\_group\_name = "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa" # 组名，此处可拿select uuid();生成  loose-group\_replication\_start\_on\_boot = off # 在mysqld启动时不自动启动组复制  loose-group\_replication\_local\_address = "192.168.0.115:24901"  loose-group\_replication\_group\_seeds = "192.168.0.115:24901,192.168.0.117:24902"  loose-group\_replication\_bootstrap\_group = off |

Docker volume create mysql\_data

Docker volume create mysql\_log

mysql\_data目录



#### 启动docker

docker run --name mysqlMaster --hostname=mysql3 --network=host --env MYSQL\_ROOT\_PASSWORD=123 -p3308:3308 -v /usr/local/mysql/mysqlMgrHost.cnf:/etc/mysql/conf.d/my.cnf -v mysql\_data:/usr/mysql/lib -v mysql\_log:/var/log/mysql -d mysql:5.7

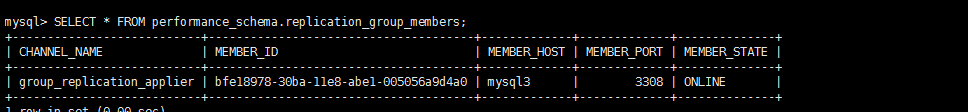
#### 进入mysql



#### Mysql配置

|  |
| --- |
| SET SQL\_LOG\_BIN=0;  CREATE USER mysql03@'%';  GRANT REPLICATION SLAVE ON \*.\* TO mysql03@'%' IDENTIFIED BY 'mysql03';  FLUSH PRIVILEGES;  SET SQL\_LOG\_BIN=1;  CHANGE MASTER TO MASTER\_USER='mysql03', MASTER\_PASSWORD='mysql03' FOR CHANNEL 'group\_replication\_recovery';  INSTALL PLUGIN group\_replication SONAME 'group\_replication.so';  SELECT \* FROM information\_schema.plugins WHERE PLUGIN\_NAME LIKE '%group%';  SET GLOBAL group\_replication\_bootstrap\_group=ON;  START GROUP\_REPLICATION;  SET GLOBAL group\_replication\_bootstrap\_group=OFF;  set global group\_replication\_allow\_local\_disjoint\_gtids\_join=1;  SELECT \* FROM performance\_schema.replication\_group\_members; |

执行select出现如图



### Slave:

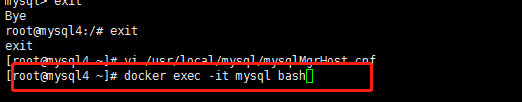
Vi /usr/local/mysql/mysqlMgrHost.cnf

|  |
| --- |
| [mysqld]  read\_rnd\_buffer\_size = 1M  sort\_buffer\_size = 1M  log-error=/var/log/mysql/mysql\_error.log  # replicatio  server-id = 3310  log-bin = mysql-bin  #log-error = error.log  log\_slave\_updates = 1  binlog\_format = row  sync\_binlog = 1  gtid\_mode = on  enforce\_gtid\_consistency = on  master\_info\_repository = table  relay\_log\_info\_repository = table  port=3308  log\_slave\_updates = ON  binlog\_checksum = NONE  binlog\_format = ROW  transaction\_isolation = READ-COMMITTED  gtid\_mode = ON  enforce\_gtid\_consistency = ON  # GR 配置项 其中loose前缀表示若Group Replication plugin未加载 mysql server仍继续启动  transaction\_write\_set\_extraction = XXHASH64  loose-group-replication-ip-whitelist="192.168.0.0/24"  loose-group\_replication\_single\_primary\_mode=true  loose-group\_replication\_enforce\_update\_everywhere\_checks=false  loose-group\_replication\_group\_name = "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa" # 组名，此处可拿select uuid();生成  loose-group\_replication\_start\_on\_boot = off # 在mysqld启动时不自动启动组复制  loose-group\_replication\_local\_address = "192.168.0.117:24902"  loose-group\_replication\_group\_seeds = "192.168.0.117:24902,192.168.0.115:24901"  loose-group\_replication\_bootstrap\_group = off |

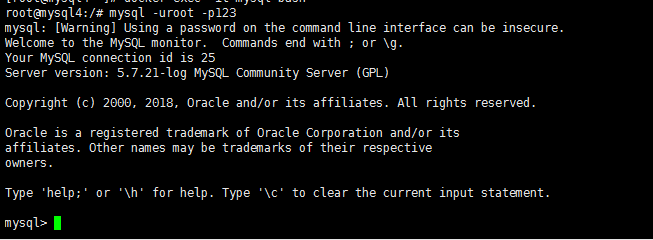
#### 启动docker

docker run --name mysql --hostname=mysql4 --network=host --env MYSQL\_ROOT\_PASSWORD=123 -p3308:3308 -v /usr/local/mysql/mysqlMgrHost.cnf:/etc/mysql/conf.d/my.cnf -v mysql\_data:/usr/mysql/lib -v mysql\_log:/var/log/mysql -d mysql:5.7

#### 进入docker



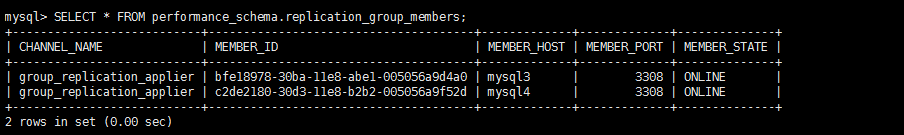
#### 进入mysql



#### Mysql配置

|  |
| --- |
| SET SQL\_LOG\_BIN=0;  CREATE USER mysql03@'%';  GRANT REPLICATION SLAVE ON \*.\* TO mysql03@'%' IDENTIFIED BY 'mysql03';  FLUSH PRIVILEGES;  SET SQL\_LOG\_BIN=1;  CHANGE MASTER TO MASTER\_USER='mysql03', MASTER\_PASSWORD='mysql03' FOR CHANNEL 'group\_replication\_recovery';  INSTALL PLUGIN group\_replication SONAME 'group\_replication.so';]  set global group\_replication\_allow\_local\_disjoint\_gtids\_join=1;  START GROUP\_REPLICATION; |

最后需要出现如图



证明配置成功

可以查看日志如图中标红所示

