

docker安装mysql搭建主从

查看镜像

docker search mysql:5.7

拉取镜像

docker pull mysql:5.7

运行mysql容器

运行mysql-master

```
docker run --name mastermysql
-p 3307:3306
-e MYSQL_ROOT_PASSWORD=123456
-v ~/opt/docker/mysql/master/data:/var/lib/mysql
-v ~/opt/docker/mysql/master/conf/my.cnf:/etc/mysql/my.cnf
-d mysql:5.7
```

运行mysql-slave

```
docker run --name slavemysql
-p 3308:3306
-e MYSQL_ROOT_PASSWORD=123456
-v ~/opt/docker/mysql/slave/data:/var/lib/mysql
-v ~/opt/docker/mysql/slave/conf/my.cnf:/etc/mysql/my.cnf
-d mysql:5.7
```

查看docker镜像运行情况

docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
def30c7dc8d4	mysql:5.7	"docker-entrypoint.s..."	39 minutes ago	Up 39 minutes	33060/tcp, 0.0.0.0:3308->3306/tcp	slavemysql
1c8b9ee38141	mysql:5.7	"docker-entrypoint.s..."	42 minutes ago	Up 42 minutes	33060/tcp, 0.0.0.0:3307->3306/tcp	mastermysql

进入master容器

docker exec -it mastermysql bash

进入mysql

root@a77371ae40bd:/# mysql -uroot -p123456

查看master状态

```
mysql> show master status;
```

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
mysql-bin.000003	154			

1 row in set (0.01 sec)

创建binlog用户，授权

创建binlog用户密码为binlog

```
CREATE USER 'binlog'@'%' IDENTIFIED BY 'binlog';
```

授权binlog用户拥有binlog库所有表的所有权限

```
GRANT ALL ON binlog.* TO 'binlog'@'%';
```

授予用户 binlog REPLICATION SLAVE 权限和 REPLICATION CLIENT 权限，用于在主从库之间同步数据。

```
GRANT REPLICATION SLAVE, REPLICATION CLIENT ON . TO 'binlog'@'%';
```

进入salve容器

```
docker exec -it slavemysql bash
```

在Slave中进入mysql，执行:

```
CHANGE MASTER TO master_host = '172.17.0.2',
master_user = 'binlog',
master_password = 'binlog',
master_port = 3306,
master_log_file = 'mysql-bin.000003',
master_log_pos = 154,
master_connect_retry = 30;
```

注:

master_host : Master 的地址，指的是容器的独立 ip, 可以通过 `docker inspect --format='{{.NetworkSettings.IPAddress}}'` 容器名称 | 容器 id 查询容器的 ip

```
→ conf docker inspect --format='{{.NetworkSettings.IPAddress}}' mastermysql
172.17.0.2
```

master_port: Master 的端口号, 指的是容器的端口号 master_user: 用于数据同步的用户
 master_password: 用于同步的用户的密码 master_log_file: 指定 Slave 从哪个日志文件开始复制数据, 即上文中提到的 File 字段的值 (主库记录的值) master_log_pos: 从哪个 Position 开始读, 即上文中提到的 Position 字段的值 (主库的pos值) master_connect_retry: 如果连接失败, 重试的时间间隔, 单位是秒, 默认是 60 秒

查询主从同步状态:

show slave status \G;

```
mysql> show slave status \G;
***** 1. row *****
      Slave_IO_State:
            Master_Host: 172.17.0.2
            Master_User: binlog
            Master_Port: 3306
            Connect_Retry: 30
            Master_Log_File: mysql-bin.000003
            Read_Master_Log_Pos: 154
            Relay_Log_File: def30c7dc8d4-relay-bin.000001
            Relay_Log_Pos: 4
            Relay_Master_Log_File: mysql-bin.000003
            Slave_IO_Running: No
            Slave_SQL_Running: No
            Replicate_Do_DB:
            Replicate_Ignore_DB: mysql,sys,information_schema,performance_schema
            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: 154
            Relay_Log_Space: 154
            Until_Condition: None
```

开启主从复制过程

start slave;

再次查询主从同步状态 show slave status \G;

```
mysql> show slave status \G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 172.17.0.2
      Master_User: binlog
      Master_Port: 3306
      Connect_Retry: 30
      Master_Log_File: mysql-bin.000003
      Read_Master_Log_Pos: 828
      Relay_Log_File: def30c7dc8d4-relay-bin.000002
      Relay_Log_Pos: 994
      Relay_Master_Log_File: mysql-bin.000003
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB: mysql,sys,information_schema,performance_schema
      Replicate_Do_Table:
```

在master库新建binlog库 mlsql_binlog表进行同步测试：

```
create database binlog;
```

```
use binlog;
```

```
CREATE TABLE mlsql_binlog (
  id int(11) unsigned NOT NULL AUTO_INCREMENT,
  log varchar(200) DEFAULT NULL,
  PRIMARY KEY ( id )
) CHARSET=utf8;
```

```
insert into mlsql_binlog ( id , log ) values(1, 'hello');
```

友情链接： <https://learnku.com/articles/30439>