



链滴

# 分布式 |mysql|docker 配置 mysql 主从模式

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1573702815662>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## javaDEMO

本网站记录了最全的各种JavaDEMO,保证下载,复制就是可用的,包括基础的, 集合的, spring的, Mybatis的等等各种,助力你从菜鸟到大牛,记得收藏哦~~

<https://www.javastudy.cloud>

## docker配置mysql主从模式

参考文档:

<https://dev.mysql.com/doc/refman/8.0/en/replication.html>

docker 配置mysql 主从模式可大致分为以下几步

1. 准备好配置文件
2. 建立docker 挂载目录
3. 分别启动docker 容器
4. 在mysql上进行主从配置
5. 验证主从的复制

## 配置文件的准备

可以在官方的my.cnf基础上更改,主要是添加server\_id  
主库新增的master.cnf配置如下:

```
[mysqld]  
# 这里的server_id 一般取机器ip的最后一位  
server_id = 1
```

```
log_bin = mysql-bin
```

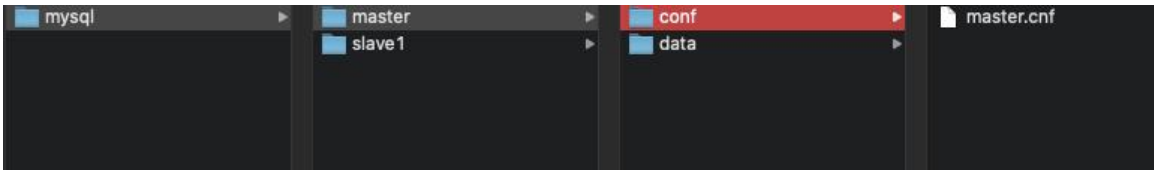
从库新增的 slave.cnf配置如下:

```
[mysqld]
server_id = 2
log_bin = mysql-bin
relay_log = /var/lib/mysql/mysql-relay-bin
log_slave_updates = 1
read_only = 1
```

## 建立docker 挂载目录

通常来说,需要挂载两个目录, 一个是配置目录, 一个是数据目录

如图所示目录结构



slave1 的目录结构也和master是一样的

## 分别启动master和slave的docker容器

启动master的docker 容器

```
docker run -d \
    --name mysql-master \
    -p 33309:3306 \
    --restart always \
    --privileged=true \
    -e MYSQL_ROOT_PASSWORD=javastudy \
    -v /Users/lixiang/soft/mount/mysql/master/conf:/etc/mysql/conf.d \
    -v /Users/lixiang/soft/mount/mysql/master/data:/var/lib/mysql \
    mysql
```

启动slave1的 docker 容器

```
docker run -d \
    --name mysql-slave1 \
    -p 33308:3306 \
    --restart always \
    --privileged=true \
    -e MYSQL_ROOT_PASSWORD=javastudy \
    -v /Users/lixiang/soft/mount/mysql/slave1/conf:/etc/mysql/conf.d \
    -v /Users/lixiang/soft/mount/mysql/slave1/data:/var/lib/mysql \
    mysql
```

docker ps 可看到如下所示:

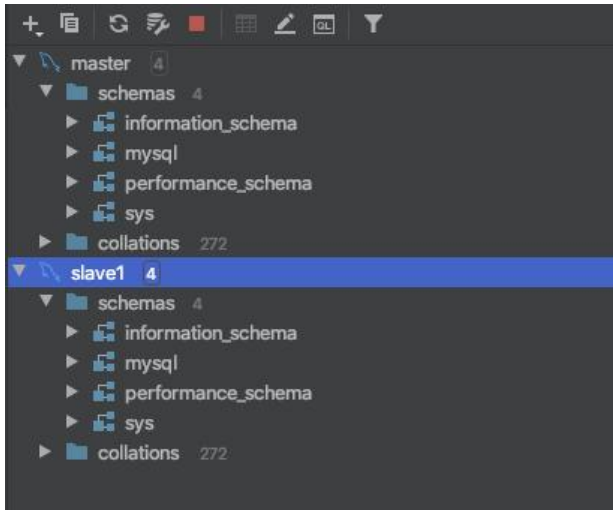
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3a65a3b36ee7	mysql	"docker-entrypoint.s..."	13 seconds ago	Up 11 seconds	3306/tcp, 0.0.0.0:33308->3306/tcp	mysql-slave1
f165d3eac29c	mysql	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	3306/tcp, 0.0.0.0:33309->3306/tcp	mysql-master

## 在mysql在进行主从配置

参考文档:

<https://dev.mysql.com/doc/refman/8.0/en/replication-howto-repuser.html>

先用可视化工具/任一客户端连接已建立好的mysql master和slave1服务器



## 在master创建用于复制的用户

注意以下命令是在master中执行

# 这里的ip只限制于docker内容访问

```
CREATE USER 'repl'@'172.17.0.%' IDENTIFIED WITH mysql_native_password BY 'studyrepl';
```

进行授权, 注意mysql 8.X 和5.X的授权语法不一致, 在网上查资料时一定要注意这点

```
GRANT REPLICATION SLAVE ON *.* TO 'repl'@'172.17.0.%';
```

## 在slave1中配置连接信息

在slave1的控制台中执行:

```
Change Master To MASTER_HOST = '172.17.0.5',  
MASTER_USER = 'repl',  
MASTER_PASSWORD = 'studyrepl',  
MASTER_LOG_FILE = 'binlog.000001',  
MASTER_LOG_POS = 0;
```

可看到以下输出:

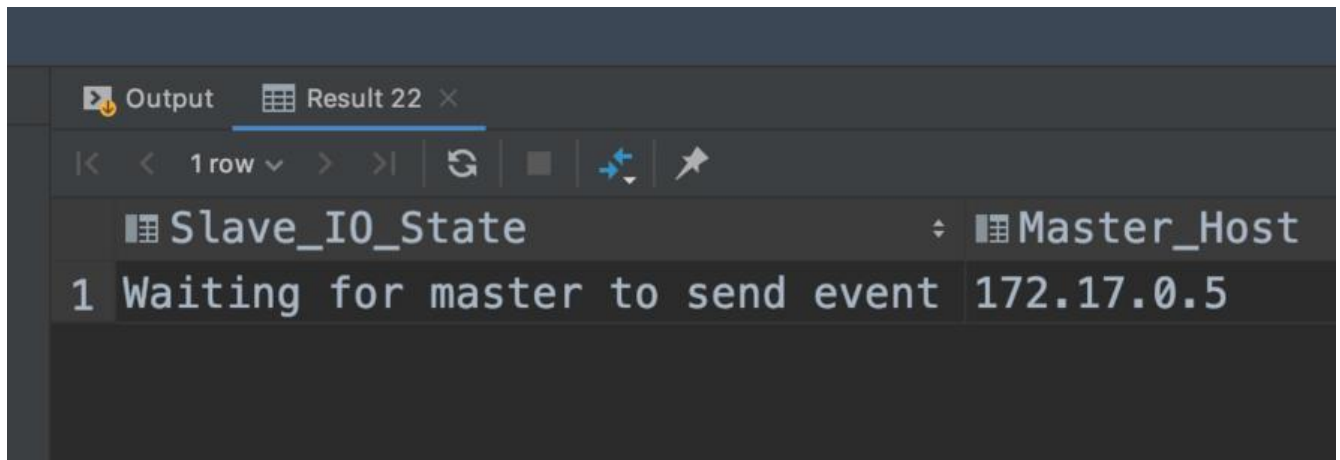
```
[2019-11-14 10:46:29] Connected  
> Change Master To MASTER_HOST = '172.17.0.5',  
MASTER_USER = 'repl',  
MASTER_PASSWORD = 'studyrepl',  
MASTER_LOG_FILE = 'binlog.000001',  
MASTER_LOG_POS = 0  
[2019-11-14 10:46:29] [HY000] [1760] Storing MySQL user name or password information in the ma  
[2019-11-14 10:46:29] completed in 123 ms
```

## 开始复制

在slave的控制台中执行:

`start slave`

然后 `show slave status` 可看到

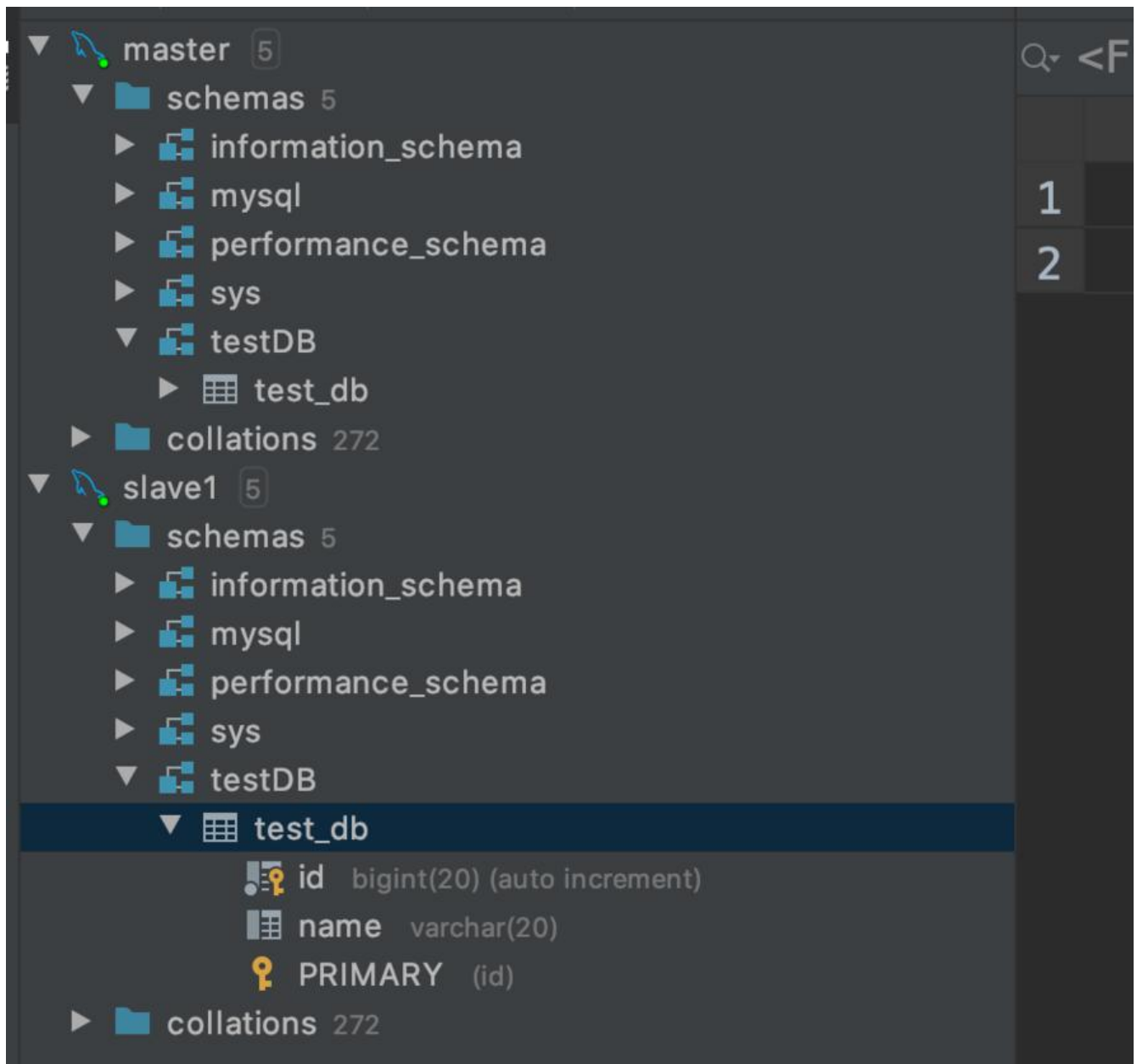


Slave_IO_State	Master_Host
1 Waiting for master to send event	172.17.0.5

## 验证主从复制

我们在master里面建个库,建张表,然后输入点数据.

其实从建库的时候就可以看到, 下面的slave也同步出现了相应的库和表



## DEMO总评

数据库主从/读写分离是数据库常用的一种扩展方案,掌握数据库主从的搭建是学习读写分离的关键,本DEMO是master/slave都是从开始就搭建好了,小伙伴们可以思考下如果master是已有的服务器,要搭配个slave要怎么操作呢?

## DEMO下载

[mysqlconfig.zip](#)