root  formssi!!

-----公共库 主从数据库-读写分离

服务器：10.207.0.161

数据库端口：3307

账号：root 密码 123456

docker镜像容器名称：mysql-master

服务器：10.207.0.161

数据库端口：3308

账号：root 密码 123456

docker镜像容器名称：mysql-slave

docker run -p 3307:3306 --net docker-mysql --ip 172.172.0.2 --restart=always --name mysql-master -v /home/mysql/common/mysql-master/config/master.cnf:/etc/mysql/conf.d/master.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

docker run -p 3308:3306 --net docker-mysql --ip 172.172.0.3 --restart=always --name mysql-slave -v /home/mysql/common/mysql-slave/config/slave.cnf:/etc/mysql/conf.d/slave.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

change master to

master\_host='172.172.0.2',

master\_port=3306,

master\_user='slave',

master\_password='123456',

get\_master\_public\_key=1,

master\_log\_file='mysql-bin.000003',

master\_log\_pos=828,

master\_connect\_retry=30;

-----oms 订单分库分表 两个库

服务器：10.207.0.161

数据库端口：3311

账号：root 密码 123456

docker镜像容器名称：mysql-oms-01

服务器：10.207.0.161

数据库端口：3312

账号：root 密码 123456

docker镜像容器名称：mysql-oms-02

docker run -p 3311:3306 --net docker-mysql --ip 172.172.0.4 --restart=always --name mysql-oms-01 -v /home/mysql/oms/mysql-oms-01/config/mysql-oms-01.cnf:/etc/mysql/conf.d/mysql-oms-01.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

docker run -p 3312:3306 --net docker-mysql --ip 172.172.0.5 --restart=always --name mysql-oms-02 -v /home/mysql/oms/mysql-oms-02/config/mysql-oms-02.cnf:/etc/mysql/conf.d/mysql-oms-02.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

-----cms 客户读多写少 读写分离

服务器：10.207.0.161

数据库端口：3313

账号：root 密码 123456

docker镜像容器名称：mysql-cms-master

服务器：10.207.0.161

数据库端口：3314

账号：root 密码 123456

docker镜像容器名称：mysql-cms-slave

docker run -p 3313:3306 --net docker-mysql --ip 172.172.0.6 --restart=always --name mysql-cms-master -v /home/mysql/cms/mysql-cms-master/config/mysql-cms-master.cnf:/etc/mysql/conf.d/mysql-cms-master.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

docker run -p 3314:3306 --net docker-mysql --ip 172.172.0.7 --restart=always --name mysql-cms-slave -v /home/mysql/cms/mysql-cms-slave/config/mysql-cms-slave.cnf:/etc/mysql/conf.d/mysql-cms-slave.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

change master to

master\_host='172.172.0.6',

master\_port=3306,

master\_user='slave',

master\_password='123456',

get\_master\_public\_key=1,

master\_log\_file='mysql-bin.000003',

master\_log\_pos=828,

master\_connect\_retry=30;

--------- 商品gms 用读写分离：读多写少

服务器：10.207.0.161

数据库端口：3315

账号：root 密码 123456

docker镜像容器名称：mysql-gms-master

服务器：10.207.0.161

数据库端口：3316

账号：root 密码 123456

docker镜像容器名称：mysql-gms-slave

docker run -p 3315:3306 --net docker-mysql --ip 172.172.0.8 --restart=always --name mysql-gms-master -v /home/mysql/gms/mysql-gms-master/config/mysql-gms-master.cnf:/etc/mysql/conf.d/mysql-gms-master.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

docker run -p 3316:3306 --net docker-mysql --ip 172.172.0.9 --restart=always --name mysql-gms-slave -v /home/mysql/gms/mysql-gms-slave/config/mysql-gms-slave.cnf:/etc/mysql/conf.d/mysql-gms-slave.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

change master to

master\_host='172.172.0.8',

master\_port=3306,

master\_user='slave',

master\_password='123456',

get\_master\_public\_key=1,

master\_log\_file='mysql-bin.000003',

master\_log\_pos=828,

master\_connect\_retry=30;

# docker部署mysql

# 拉取mysql的镜像

docker pull mysql:5.7.26

docker pull mysql:latest

## 启动容器，以该镜像启动2个容器

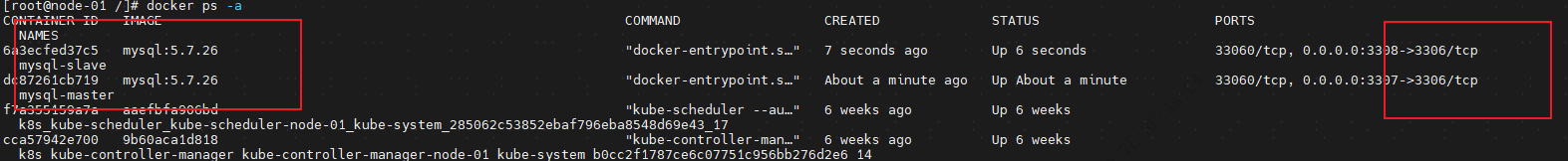
1.主库的容器

docker run -p 3307:3306 --name mysql-master -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:5.7.26

2.从库的容器

docker run -p 3308:3306 --name mysql-slave -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:5.7.26

3.使用docker ps -a检查容器是否启动成功



进入到主库和从库的容器中修改对应的数据

1.主库

docker exec -it mysql-master /bin/bash 进入到容器

往my.cnf 写入配置

echo -e "[mysqld]\nserver-id=100\nlog-bin=mysql-bin" >> etc/mysql/my.cnf

service mysql restart 重启服务

2.从库

docker exec -it mysql-slave /bin/bash 进入到容器

echo -e "[mysqld]\nserver-id=101\nlog-bin=mysql-bin" >> etc/mysql/my.cnf 修改配置文件

service mysql restart 重启服务

3.因为服务重启之后退出容器会关闭，所以需要重启这两个容器

docker restart mysql-master

docker restart mysql-slave

CTRL+D ：退出容器

# 开启数据库主从复制

1.进入到主库中，连接上mysql，创建一个便于从库登陆的用户

A：可以利用navicat进行登录操作：10.207.0.161 username：root pass：123456

B：进入到镜像容器 docker exec -it mysql-master /bin/bash 然后mysql -u root -p 123456 回车。

2. 创建一个便于从库登陆的用户

CREATE USER 'slave'@'%' IDENTIFIED BY '123456';

GRANT REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'slave'@'%';

3.重置偏移量

reset master;

4.查看主库的二进制文件读取信息

执行show master status;

图片包含 日程表

描述已自动生成

4.进入到从库中，连接主库

ip可以用docker inspect 容器名称/容器id获取

docker inspect mysql-master

夜晚的星空

中度可信度描述已自动生成

从库中执行：

change master to master\_host='172.17.0.4',

master\_user='slave',

master\_password='123456',

master\_port=3306,

master\_log\_file='mysql-bin.000001',

master\_log\_pos= 154,

master\_connect\_retry=30;

**命令说明：**  
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

master\_log\_pos：从哪个 Position 开始读，即上文中提到的 Position 字段的值

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

4.启动主从  
start slave;  
5.查看主从

在mysql-Slave中执行 show slave status \G;查看主从复制状态

文本

描述已自动生成

结果：

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

## 1、安装镜像

docker pull mysql:8.0.27

## 2.设置创建自定义网络

Docker每次重启，容器重启之后ip会变，用网段指定启动ip

docker network create --subnet=172.172.0.0/24 docker-mysql

## 3、新建挂载目录

挂载地址：/home/mysql/。。。。。。

mkdir -p /home/mysql/common/mysql-master/{config,log,data}

mkdir -p /home/mysql/common/mysql-slave/{config,log,data}

mkdir -p /home/mysql/cms/mysql-cms-master/{config,log,data}

mkdir -p /home/mysql/cms/mysql-cms-slave/{config,log,data}

mkdir -p /home/mysql/gms/mysql-gms-master/{config,log,data}

mkdir -p /home/mysql/gms/mysql-gms-slave/{config,log,data}

mkdir -p /home/mysql/oms/mysql-oms-01/{config,log,data}

mkdir -p /home/mysql/oms/mysql-oms-02/{config,log,data}

试试运行mysql:8.0容器

docker run -p 3307:3306 --name mysql-master -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql: 8.0.27

docker exec -it mysql-master bash

mysql -u root -p

## 4、创建配置文件

要想通过docker安装的mysql实现主从复制，我们需要指定运行时的配置文件

所以首先我们需要在宿主机上面建立master.cnf以及slave.cnf

其中master.cnf文件的内容为：

[mysqld]

#pid-file = /var/run/mysqld/mysqld.pid

#socket = /var/run/mysqld/mysqld.sock

datadir = /var/lib/mysql

symbolic-links=0

character-set-server = utf8

#skip-networking

innodb\_print\_all\_deadlocks = 1

max\_connections = 2000

max\_connect\_errors = 6000

open\_files\_limit = 65535

table\_open\_cache = 128

max\_allowed\_packet = 4M

binlog\_cache\_size = 1M

max\_heap\_table\_size = 8M

tmp\_table\_size = 16M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 8M

sort\_buffer\_size = 8M

join\_buffer\_size = 28M

key\_buffer\_size = 4M

thread\_cache\_size = 8

#query\_cache\_type = 1

#query\_cache\_size = 8M

#query\_cache\_limit = 2M

ft\_min\_word\_len = 4

log-bin = mysql-bin

server-id = 3307

binlog\_format = mixed

relay-log-index=slave-relay-bin.index

performance\_schema = 0

default-authentication-plugin=mysql\_native\_password

explicit\_defaults\_for\_timestamp

#lower\_case\_table\_names = 1

interactive\_timeout = 28800

wait\_timeout = 28800

# Recommended in standard MySQL setup

#sql\_mode=NO\_ENGINE\_SUBSTITUTION,NO\_AUTO\_CREATE\_USER,STRICT\_TRANS\_TABLES

[mysqldump]

quick

max\_allowed\_packet = 16M

[myisamchk]

key\_buffer\_size = 8M

sort\_buffer\_size = 8M

read\_buffer = 4M

write\_buffer = 4M

slave.cnf文件的内容为：

[mysqld]

#pid-file = /var/run/mysqld/mysqld.pid

#socket = /var/run/mysqld/mysqld.sock

datadir = /var/lib/mysql

symbolic-links=0

character-set-server = utf8

#skip-networking

innodb\_print\_all\_deadlocks = 1

max\_connections = 2000

max\_connect\_errors = 6000

open\_files\_limit = 65535

table\_open\_cache = 128

max\_allowed\_packet = 4M

binlog\_cache\_size = 1M

max\_heap\_table\_size = 8M

tmp\_table\_size = 16M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 8M

sort\_buffer\_size = 8M

join\_buffer\_size = 28M

key\_buffer\_size = 4M

thread\_cache\_size = 8

#query\_cache\_type = 1

#query\_cache\_size = 8M

#query\_cache\_limit = 2M

ft\_min\_word\_len = 4

log-bin = mysql-bin

server-id = 3308

binlog\_format = mixed

relay-log-index=slave-relay-bin.index

performance\_schema = 0

default-authentication-plugin=mysql\_native\_password

explicit\_defaults\_for\_timestamp

#lower\_case\_table\_names = 1

interactive\_timeout = 28800

wait\_timeout = 28800

# Recommended in standard MySQL setup

#sql\_mode=NO\_ENGINE\_SUBSTITUTION,NO\_AUTO\_CREATE\_USER,STRICT\_TRANS\_TABLES

[mysqldump]

quick

max\_allowed\_packet = 16M

[myisamchk]

key\_buffer\_size = 8M

sort\_buffer\_size = 8M

read\_buffer = 4M

write\_buffer = 4M

## 5、运行容器

运行master节点

docker run -p 3307:3306 --net docker-mysql --ip 172.172.0.2 --restart=always --name mysql-master -v /home/mysql/common/mysql-master/config/master.cnf:/etc/mysql/conf.d/master.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

运行slave节点

docker run -p 3308:3306 --net docker-mysql --ip 172.172.0.3 --restart=always --name mysql-slave -v /home/mysql/common/mysql-slave/config/slave.cnf:/etc/mysql/conf.d/slave.cnf -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:8.0.27

指定ip启动、开机自启动

-v是将宿主机的文件挂载到docker对应的目录中，这里我们使用的是自定义的配置文件，所以要将其挂载到docker对应的位置上

注意重启下

docker restart mysql-master

docker restart mysql-slave

## 6、配置master容器

docker exec -it mysql-master bash

mysql -u root -p

进入到master mysql以后需要分配从节点的用户

create user 'slave'@'%' identified by '123456';

grant replication slave on \*.\* to 'slave'@'%';

flush privileges;

查看master数据库状态：

show master status;

电子设备的屏幕

低可信度描述已自动生成

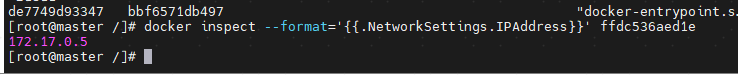
## 7、配置slave容器

首先我们需要先查一下master数据库运行的ip地址，需要根据容器的id进行查询

docker ps -a

找到对应的id，根据对应的id查询ip地址

docker inspect --format='{{.NetworkSettings.IPAddress}}' ffdc536aed1e



进入slave容器中

docker exec -it mysql-slave bash

mysql -u root -p

然后进行如下配置 注意：get\_master\_public\_key=1, 8.0版本要带这个加密

change master to

master\_host='172.17.0.5',

master\_port=3306,

master\_user='slave',

master\_password='123456',

get\_master\_public\_key=1,

master\_log\_file='mysql-bin.000003',

master\_log\_pos=828,

master\_connect\_retry=30;

## 8、启动并查看slave

start slave;

开启之后重新执行 主从配置要先停掉复制不然执行报错

#stop slave

查看slave的状态：

show slave status\G;

文本

描述已自动生成

如果服务器关机重启后，直接重启这两个数据库即可

docker restart mysql-master

docker restart mysql-slave

参考：<https://blog.csdn.net/wyg1973017714/article/details/112601802>

参考：<https://blog.csdn.net/Other_Shore/article/details/118556945>

<https://www.cnblogs.com/lvlinguang/p/15205389.html>

设置网桥：

https://blog.csdn.net/m0\_67401606/article/details/124169925