四、docker23.0.1+mysql8主从

使用docker实现同网段的两台虚拟内的容器拉取mysql8，并实现两个mysql8容器的主从同步

**一．将两台虚拟机都安装好docker23.0.1**

1. 下载docker23.0.1 版本：

wget <https://download.docker.com/linux/static/stable/x86_64/docker-23.0.1.tgz>

1. 解压docker23.0.1

tar -zxvf docker-23.0.1.tgz

3.将解压的文件全部cp到/usr/bin目录下

cp docker/\* /usr/bin

1. 在/etc/systemd/system/ 目录下，创建 docker.service 文件

给docker.service文件添加执行权限

chmod +x /etc/systemd/system/docker.service

重新加载配置文件（每次有修改docker.service文件时都要重新加载下）

systemctl daemon-reload

1. 启动docker

Systemctl start docker

1. 查看docker状态

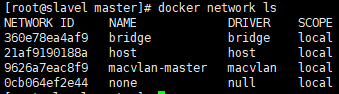
Systemctl status docker

1. **创建macvlan网络，连接两台虚拟机**

主：

Docker network create -d macvlan --subnet=192.168.110.0/24 --gateway=192.168.110.2 -o parent=ens33 macvlan-master

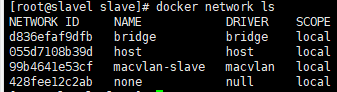
查看网络是否创建成功---docker network ls



从：

Docker network create -d macvlan --subnet=192.168.110.0/24 --gateway=192.168.110.2 -o parent=ens33 macvlan-slave

查看网络是否创建成功---docker network ls



1. **两台虚拟机分别创建MySQL容器（主和从）**
2. 拉取MySQL8的镜像

Docker pull mysql:8

1. 修改配置文件

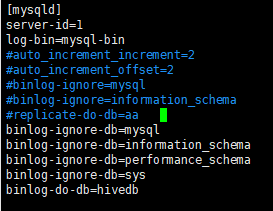
主:

mkdir -p /home/data/mysql/data/master

mkdir -p /home/data/mysql/master

cd /home/data/mysql/master

Vim master.cnf



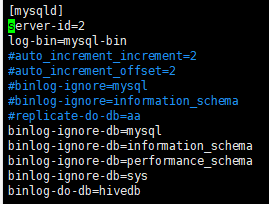
从：

Mkdir -p /home/data/mysql/data/slave

Mkdir -p /home/data/mysql/slave

Cd /home/data/mysql/slave

Vim slave.cnf



1. 创建MySQL容器

主：

Master：

docker create -p 3308:3306 --name mysqlmaster --privileged=true \

-v /home/data/mysql/data/master:/var/lib/mysql \

-v /home/data/mysql/master:/etc/mysql/conf.d \

-e MYSQL\_ROOT\_PASSWORD=123456 \

--net macvlan-master --ip 192.168.88.3 \

mysql:8

从：

docker create -p 3309:3306 --name mysqlslave --privileged=true \

-v /home/data/mysql/data/slave:/var/lib/mysql \

-v /home/data/mysql/slave:/etc/mysql/conf.d \

-e MYSQL\_ROOT\_PASSWORD=123456 \

--net macvlan-slave --ip 192.168.88.4 \

mysql:8

1. 开启MySQL容器

Docker start mysqlmaster

Docker start mysqlslave

1. 查看运行状态 docker ps -a





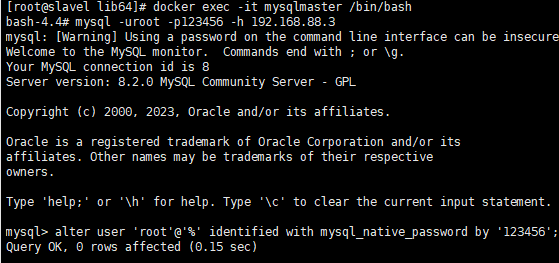
1. 容器设置远程访问：

Master：

docker exec -it mysqlmaster /bin/bash

mysql -uroot -p123456 -h 192.168.110.3

alter user ‘root’@’%’ identified with mysql\_native\_password by ‘123456’;

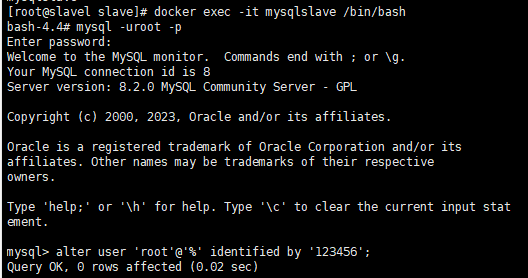


Slave：

docker exec -it mysqlslave /bin/bash

mysql -uroot -p123456 -h 192.168.110.4

alter user ‘root’@’%’ identified with mysql\_native\_password by ‘123456’;



1. **配置主从同步：**

1.进入mater容器中，赋权限以及查询master的日志文件以及位置：

set sql\_mode=(select replace(@@sql\_mode,’ONLY\_FULL\_GROUP\_BY’,’’));

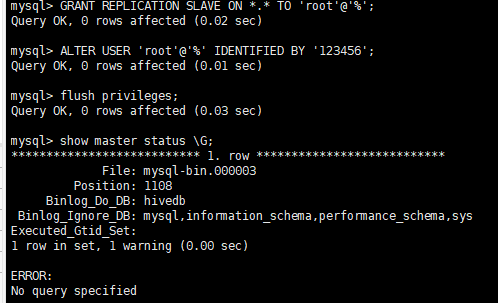
GRANT REPLICATION SLAVE ON \*.\* TO 'root'@'%';

ALTER USER 'root'@'%' IDENTIFIED BY '123456';

flush privileges;

show master status \G;





32进入slave容器中，设置slave配置

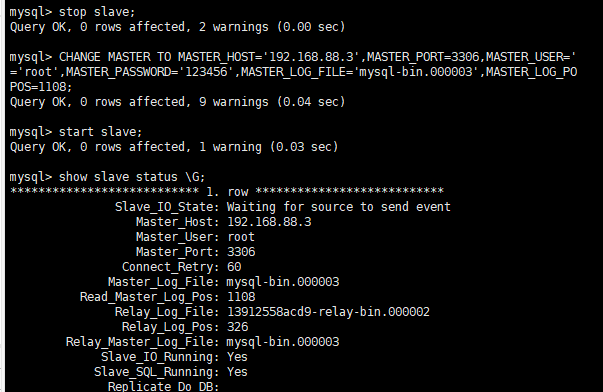
stop slave;

CHANGE MASTER TO MASTER\_HOST='192.168.88.3',MASTER\_PORT=3306,MASTER\_USER='root’

,MASTER\_PASSWORD='123456',MASTER\_LOG\_FILE='mysql-bin.000003',MASTER\_LOG\_POS=1108;

start slave;

show slave status \G;



1. **验证主从同步：**

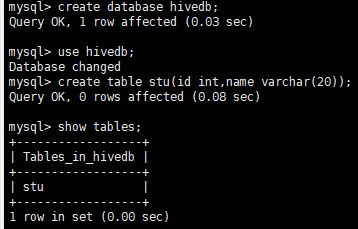
进入到master 容器中创建表格

create database hivedb;

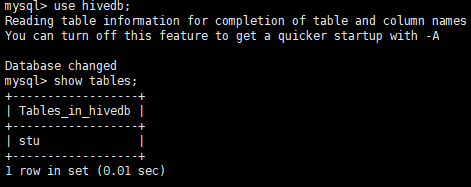
use hivedb;

create table stu(id int,name varchar(20));

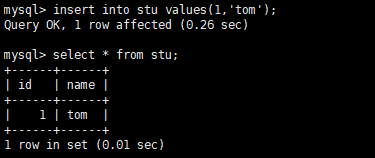
show tables;



此时在slave容器中可以查询到该表



1. 在master的表下添加一条数据后查询该表



在slave容器中查询该表时发现刚添加的数据，则主从搭建成功；

