# 题目

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mysql8容器的主从同步

# 安装docker

## 安装所需的软件包

安装gcc

yum -y install gcc

yum -y install gcc-c++

安装所需的软件包

yum install -y yum-utils

设置stable镜像仓库（采用Ali云仓库）

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

更新yum软件包索引

yum makecache fast

安装DOCKER CE

yum install docker-ce docker-ce-cli containerd.io

配置阿里云镜像加速

sudo mkdir -p /etc/docker

sudo tee /etc/docker/daemon.json <<-'EOF'

{

"registry-mirrors": ["https://pak31uuv.mirror.aliyuncs.com"]

}

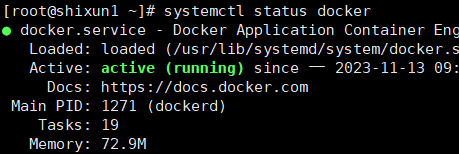
EOF

启动容器

sudo systemctl start docker

查看容器状态

命令：systemctl status docker



### 创建网桥

桥接模式命令：docker network create -d macvlan --subnet=192.168.9.0/24 --gateway=192.168.9.2 -o parent=ens33 mysql\_mac

Passthru模式命令：docker network create -d macvlan --subnet=192.168.9.0/24 --gateway=192.168.9.2 --ip-range=192.168.9.0/24 -o parent=ens33 -o macvlan\_mode=passthru my-macvlan-net

### Macvlan网络模式

Macvlan中Bridge（桥接）模式

Bridge 模式网络桥接技术。

在 Bridge 模式下，容器的网络流量通过 Docker 宿主机上的 Docker 网桥传输。

容器与其他设备通信时，流量会通过 Docker 网桥进行转发。

Passthru（透传）模式：

Passthru模式允许容器直接访问物理网络，并共享物理网络接口的 MAC 地址。这意味着容器可以像物理设备一样与物理网络中的其他设备进行通信。

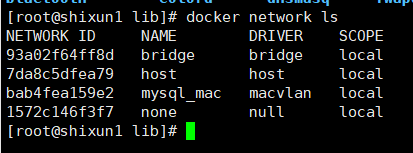
在 Passthru 模式下，容器的网络流量不经过 Linux 网桥或 Docker 网桥，而是直接通过物理网络接口传输。这使得容器能够直接与其他设备进行通信，而不需要通过任何中间设备。

但是使用Passrthu模式会使得虚拟机的网络出现问题，导致虚拟机网络失效。

创建成功后会返回一组网桥编号



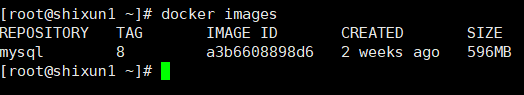
查看网桥目录，命令docker network ls



NAME为mysql\_mac的网桥为我们创建的网桥

# 拉取mysql镜像

Docker pull mysql8



# 创建mysql容器

## 创建mysql主节点的容器

命令：docker create -p 3306:3306 --name mysql-master --privileged=true \

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

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

-e MYSQL\_ROOT\_PASSWORD=000000 \

--network mysql\_mac --ip 192.168.9.3 \

mysql:8

## 创建mysql从节点的容器

命令：docker create -p 3306:3306 --name mysql-slave --privileged=true \

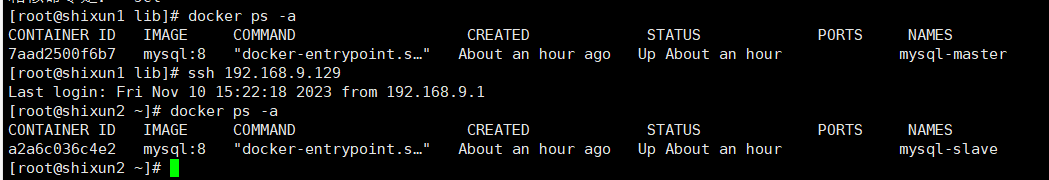
-v /home/data/mysql/data/slave:/var/lib/mysql \

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

-e MYSQL\_ROOT\_PASSWORD=000000 \

--net mysql\_mac --ip 192.168.9.4 \

## 查看容器运行状态



可以看到主从节点的状态均为UP，说明容器已经启动。

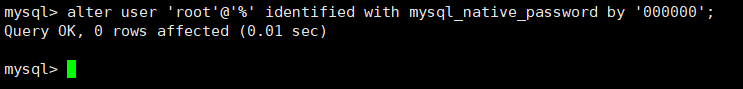
## 设置mysql远程登录访问

主节点

命令：docker exec -it mysql-master /bin/bash

mysql -uroot -p000000 -h 192.168.9.3

alter user 'root'@'%' identified with mysql\_native\_password by '000000';

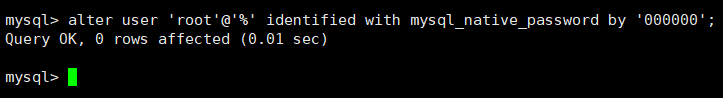


从节点

命令：docker exec -it mysql-slave /bin/bash

mysql -uroot -p000000 -h 192.168.9.4

alter user 'root'@'%' identified with mysql\_native\_password by '000000';



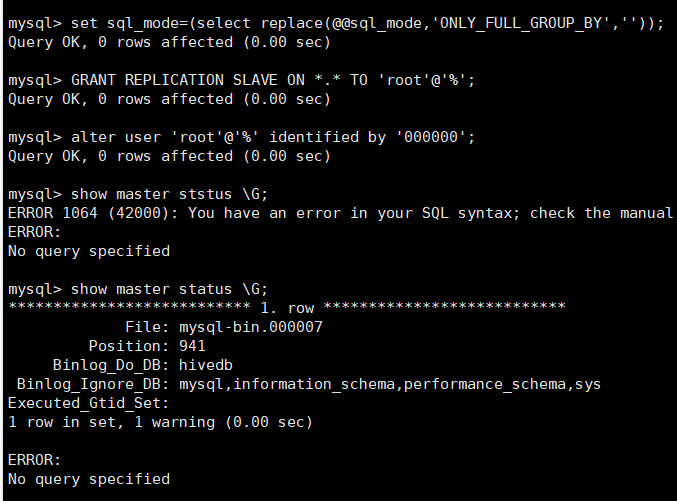
# master授权

 进入MASTER 赋权限 以及查询MASTER 的日志文件 以及位置

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

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

alter user 'root'@'%' identified by '000000';



记录下file:和Position配置从结点，Binlog\_Do\_DB: hivedb标注了主从复制的数据库 Binlog\_Ignore\_DB: mysql,information\_schema,performance\_schema,sys标注了不需要复制的数据库。

# Slave授权

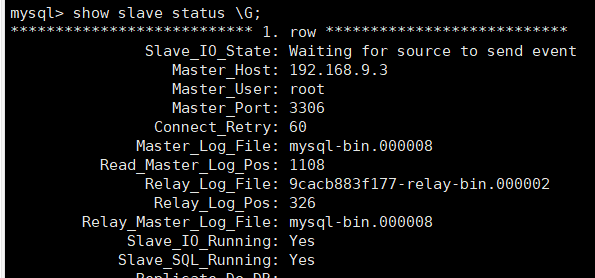
 进入SLAVE 设置SLAVE配置以及启动SLAVE

命令：stop slave;

CHANGE MASTER TO MASTER\_HOST='192.168.9.3',MASTER\_PORT=3306,MASTER\_USER='root',MASTER\_PASSWORD='000000',MASTER\_LOG\_FILE='mysql-bin.000007',MASTER\_LOG\_POS=941;

start slave;

show slave status \G;



此行为指定了二进制日志文件和位置开始复制数据。

两个都是yes说明主从配置成功，此时可以进行验证。

# 验证主从

在主容器运行以下命令

>create database hivedb；

在从容器运行

>show databases; //查看是否出现hivedb数据库

在主容器

>CREATE TABLE student (

id INT,

name VARCHAR(50),

sex VARCHAR(100)

);

在从容器查看

>show student; //是否存在student表，应存在

在主容器插入数据：INSERT INTO student (id, name, sex) VALUES (1, 'John Doe', 'Male');

在从容器查看：select \* from student;数据应同步

在从容器插入数据：INSERT INTO student (id, name, sex) VALUES (3, 'qr', 'Female');

在主容器查看数据不存在，即成功。