五．Docker23.0.1+redis6主从

1．docker安装

1下载对应版本的docker

可用下列命令行或者进入官网自行下载

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

安装包官方地址：<https://download.docker.com/linux/static/stable/x86_64/>

2. 解压

tar -zxvf docker-23.0.1-ce.tgz

3. 将解压出来的docker文件复制到 /usr/bin/ 目录下

cp docker/\* /usr/bin/

4. 创建docker.service文件

进入 /etc/systemd/system/ 目录,并创建 docker.service 文件，内容如下，这样可以将docker注册为service服务。

[Unit]

Description=Docker Application Container Engine

Documentation=https://docs.docker.com

After=network-online.target firewalld.service

Wants=network-online.target

[Service]

Type=notify

# the default is not to use systemd for cgroups because the delegate issues still

# exists and systemd currently does not support the cgroup feature set required

# for containers run by docker

ExecStart=/usr/bin/dockerd --selinux-enabled=false --insecure-registry=127.0.0.1

ExecReload=/bin/kill -s HUP $MAINPID

# Having non-zero Limit\*s causes performance problems due to accounting overhead

# in the kernel. We recommend using cgroups to do container-local accounting.

LimitNOFILE=infinity

LimitNPROC=infinity

LimitCORE=infinity

# Uncomment TasksMax if your systemd version supports it.

# Only systemd 226 and above support this version.

#TasksMax=infinity

TimeoutStartSec=0

# set delegate yes so that systemd does not reset the cgroups of docker containers

Delegate=yes

# kill only the docker process, not all processes in the cgroup

KillMode=process

# restart the docker process if it exits prematurely

Restart=on-failure

StartLimitBurst=3

StartLimitInterval=60s

[Install]

WantedBy=multi-user.target

5. 启动docker

# 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

# 启动

systemctl start docker

# 设置开机启动

systemctl enable docker.service

docker -v 可查看docker版本

2.拉取redis6镜像

docker search redis6 查找redis6版本

docker pull redis:6.2.7 拉去需求版本的镜像

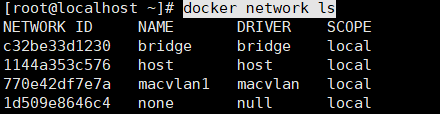
3.配置redis主从

分别在两台主机搭建macvlan，两台主机子网地址和网关相同，分别用的虚拟机网卡名

docker network create -d macvlan --subnet=192.168.1.0/24 --gateway=192.168.1.1 -o parent=etho16777736 macvlan1

docker network create -d macvlan --subnet=192.168.1.0/24 --gateway=192.168.1.1 -o parent=ens33 macvlan1

docker network ls



分别在两台主机/usr/local下建立redis文件夹

Cd /usr/local

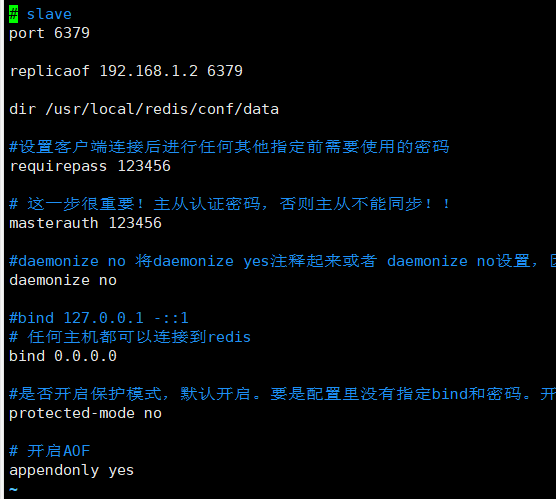
Mkdir redis/conf redis/data

进入conf文件中建立redis.conf

主：



从：



建立启动容器：

docker run --network macvlan1 --ip=192.168.1.2 \

-v /etc/redis/data:/data \

-v /etc/redis/redis.conf:/etc/redis/redis.conf \

--privileged=true \

--name redis-master \

-d redis:6.2.7 redis-server /etc/redis/redis.conf

docker run --network macvlan1 --ip=192.168.1.3 \

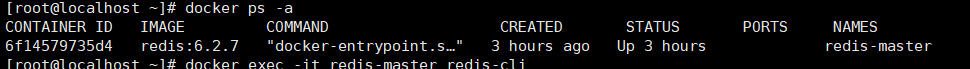
-v /etc/redis/data:/data \

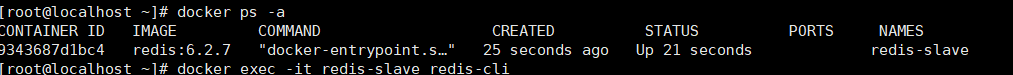
-v /etc/redis/redis.conf:/etc/redis/redis.conf \

--privileged=true \

--name redis-slave \

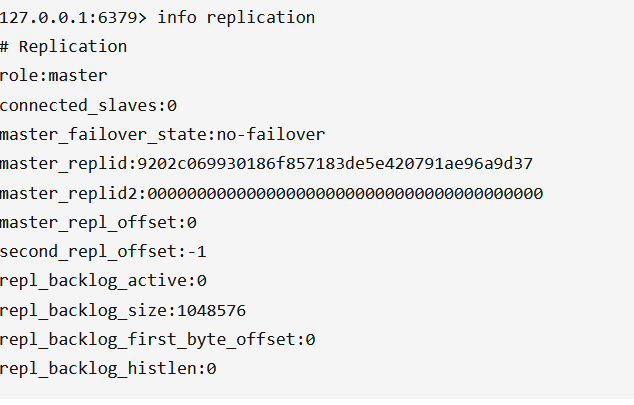
-d redis:6.2.7 redis-server /etc/redis/redis.conf





分别进入两个容器

docker exec -it redis-master redis-cli



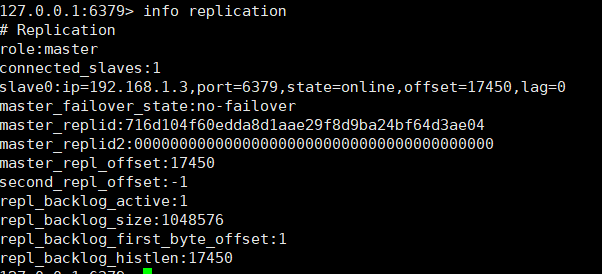
docker exec -it redis-slave redis-cli



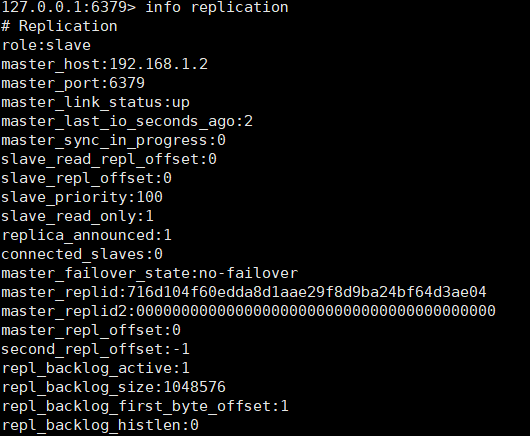
将从节点绑定到主节点

slaveof 192.168.1.2 6379

绑定后主节点：



从节点



经过测试主从同步成功。