**实验报告**

# Registry仓库部分

将192.168.100.150registry仓库恢复到 刚安装docker的快照，重新创建registry仓库

docker pull registry:2

mkdir -p /v913/registry

docker run -itd --name registry -p5000:5000 --restart always -v /v913/registry/:/var/lib/registry registry:2

# swarm部分

返回到servera：

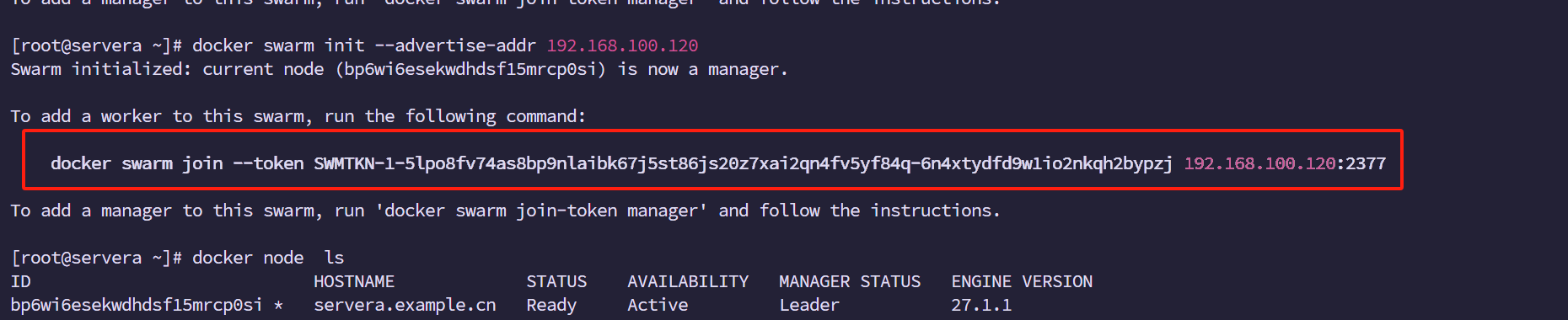
创建swarm和集群网络：

Servera：

docker swarm init --advertise-addr 192.168.100.100

serverb、c：

复制servera生成的这段黏贴到b、c让b、c加入到a集群中（这里复制自己的 别用我的）



Servera创建lnmp网卡：

docker network create --driver overlay gmq\_v913\_lnmp\_ovl --subnet 24.17.3.0/24 --gateway 24.17.3.254

# supermin部分

serverc使用supermin构建基准镜像：

yum -y install supermin

mkdir /tmp/supermin

cd /tmp/supermin/

基于这些软件包创建一个rocky9的系统环境

supermin -v --prepare yum coreutils iproute vim rpm httpd-tools -o supermin.d

supermin -v --build --format chroot supermin.d/ -o rocky9.d

cp /etc/resolv.conf rocky9.d/etc/

echo 9.4 > rocky9.d/etc/dnf/vars/releasever

导入证书文件

chroot rocky9.d/ /usr/bin/bash

rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9

打包压缩并导入成镜像

exit

tar -cf rk9.tar -C rocky9.d/ .

docker import rk9.tar gmq:v913

docker tag gmq:v913 registry:5000/v913/gmq:v913

docker push registry:5000/v913/gmq:v913

serverc：

wget <https://nginx.org/download/nginx-1.26.2.tar.gz>

vim Dockerfile #写入内容：

FROM registry:5000/v913/gmq:v913

COPY nginx-1.26.2.tar.gz /

RUN yum -y install tar httpd-tools gcc-c++ openssl-devel zlib-devel pcre-devel tar && cd / && tar -zxvf nginx-1.26.2.tar.gz && cd nginx-1.26.2 && ./configure --prefix=/usr/local/gmq --sbin-path=/usr/sbin/gmq --conf-path=/etc/nginx/nginx.conf --with-http\_ssl\_module --with-http\_stub\_status\_module && make && make install && cd / && rm -rf nginx\* && echo defaults\_page by gmq > /usr/local/gmq/html/index.html

RUN ln -s /usr/sbin/gmq /usr/bin/nginx

RUN groupadd gmq && useradd -g gmq gmq

RUN chown -R gmq:gmq /usr/local/gmq /usr/sbin/gmq /etc/nginx

RUN mkdir /www && echo "v1.0 27.1.1" > /www/index.html

USER gmq

EXPOSE 80/tcp

CMD ["/usr/sbin/gmq","-g","daemon off;"]

docker build -t gmq:nginx .

docker tag gmq:nginx registry:5000/v913/gmq:nginx

docker push registry:5000/v913/gmq:nginx

~~serverc：~~

~~openssl genrsa -out /file.key 2048~~

~~openssl req -new -key /file.key -out /file.csr -days 365~~

~~openssl x509 -req -in /file.csr -signkey /file.key -out /file.crt -days 365~~

Vim Dockerfile1 写入:

FROM registry:5000/v913/gmq:nginx

RUN echo "i'am front" > /usr/local/gmq/html/index.html

RUN mkdir /usr/local/gmq/html/cn\_name

RUN echo "耿铭琦" > /usr/local/gmq/html/cn\_name/index.html

RUN mkdir /usr/local/gmq/html/en\_name

RUN echo "gengmingqi" > /usr/local/gmq/html/en\_name/index.html

RUN mkdir /usr/local/gmq/html/you

RUN echo "rhca2013" > /usr/local/gmq/html/you/index.html

RUN echo "FILE\_UNKNOWN" > /usr/local/gmq/html/404.html

RUN mkdir /usr/local/gmq/html/Privatewebsite

RUN htpasswd -bc /etc/nginx/.htpasswd gmq gmq

RUN echo "access" > /usr/local/gmq/html/Privatewebsite/index.html

RUN mkdir /usr/local/gmq/html/ssl

RUN echo "Running on 443 port" > /usr/local/gmq/html/ssl/index.html

COPY nginx.conf /etc/nginx/nginx.conf

docker build -t gmq:front . -f Dockerfile1

docker tag gmq:front registry:5000/v913/gmq:front

docker push registry:5000/v913/gmq:front

Vim Dockerfile2 写入:

FROM registry:5000/v913/gmq:nginx

RUN echo "i'am backend v1.0" > /usr/local/gmqhtml/index.html

RUN mkdir /usr/local/gmqhtml/exam

RUN echo "v1.0 by 胡智瀚" > /usr/local/gmq/html/exam/index.html

COPY v1.0.conf /etc/nginx/nginx.conf

docker build -t gmq:backend\_v1.0 . -f Dockerfile2

Vim Dockerfile3 写入:

FROM registry:5000/v913/gmq:nginx

RUN echo "i'am backend v2.0" > /usr/local/gmq/html/index.html

RUN mkdir /usr/local/gmq/html/exam

RUN echo "v2.0 by 胡智瀚" > /usr/local/gmq/html/exam/index.html

RUN echo "v2.0 27.1.1" > /www/index.html

COPY v1.0.conf /etc/nginx/nginx.conf

docker build -t gmq:backend\_v2.0 . -f Dockerfile3

回到集群的管理节点servera 创建服务service

Servera:

docker service create --name front\_srv --replicas 3 --network gmq\_v913\_lnmp\_ovl -p 80:80 -p 443:443 registry:5000/v913/gmq:front

vim /etc/hosts

文本

描述已自动生成

验证：

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

第8要求：版本回滚

没做 好困-.-