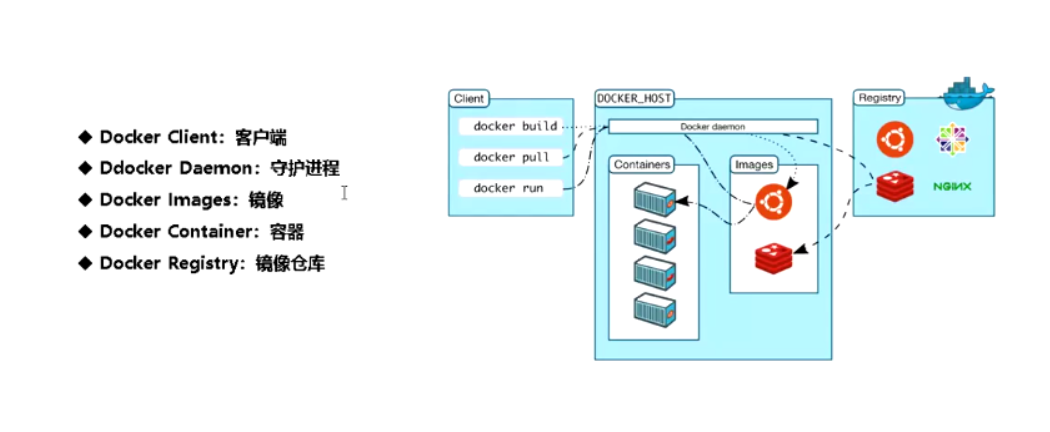
Docker 环境的配置及实部署

# 1 前言

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的[Linux](https://baike.baidu.com/item/Linux)机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口。

Docker组成：



# 2 Docker的安装

## 2.1 centos7的安装

这里只举例在centos7中的安装步骤如下：

卸载旧版本(如果安装过旧版本的话)

sudo yum remove docker \

              docker-client \

              docker-client-latest \

              docker-common \

              docker-latest \

              docker-latest-logrotate \

              docker-logrotate \

              docker-selinux \

              docker-engine-selinux \

              docker-engine

#更新 yum

yum -y update

​

#安装 docker

yum -y install docker

​

#进入 docker

vi /etc/sysconfig/docker

​

# 修改 --selinux-enabled=false

​

# /etc/sysconfig/docker

​

# Modify these options if you want to change the way the docker daemon runs

OPTIONS='--selinux-enabled=false --log-driver=journald --signature-verification=false'

if [ -z "${DOCKER\_CERT\_PATH}" ]; then

   DOCKER\_CERT\_PATH=/etc/docker

fi

​

# :wq 退出保存

# 重新启动 docker

systemctl restart docker

​

docker version

​

Client:

Version:         1.13.1

API version:     1.26

Package version: docker-1.13.1-63.git94f4240.el7.centos.x86\_64

Go version:     go1.9.4

Git commit:     94f4240/1.13.1

Built:           Fri May 18 15:44:33 2018

OS/Arch:         linux/amd64

​

Server:

Version:         1.13.1

API version:     1.26 (minimum version 1.12)

Package version: docker-1.13.1-63.git94f4240.el7.centos.x86\_64

Go version:     go1.9.4

Git commit:     94f4240/1.13.1

Built:           Fri May 18 15:44:33 2018

OS/Arch:         linux/amd64

Experimental:    false

## 2.2 开启远程API

# 编辑该文件

vi /etc/sysconfig/docker-network

​

# /etc/sysconfig/docker-network 找到 DOCKER\_NETWORK\_OPTIONS， 补全

DOCKER\_NETWORK\_OPTIONS="-H unix:///var/run/docker.sock -H 0.0.0.0:5555"

​

#然后重启docker

sudo systemctl daemon-reload

sudo service docker restart

​

​

# :wq 强制保存 在使用netstat 查看该端口

netstat -anp|grep 5555

​

#查看该应用

curl 127.0.0.1:5555/info

~~~

~~~

#关闭防火墙 因为开了防火墙无法访问

#临时关闭

systemctl stop firewalld

#禁止开机启动

systemctl disable firewalld

## 2.3 采用docker方式打包到服务器

**1.在pom.xml文件 加入 对应的插件 （ps: docker.image.prefix 是指仓库名称，建议使用dockerhub的名称，否则无法进行提交push到自己的仓库 owenwanggq）**

<properties>

 <docker.image.prefix>owenwanggq</docker.image.prefix>

</properties>

​

<plugin>

 <groupId>com.spotify</groupId>

 <artifactId>docker-maven-plugin</artifactId>

 <version>0.4.13</version>

 <configuration>

   <imageName>${docker.image.prefix}/${project.artifactId}</imageName>

   <dockerDirectory>src/main/docker</dockerDirectory>

   <!-- docker远程服务器地址 -->

<dockerHost>http://xx.xx.xx.xx:5555</dockerHost>

   <resources>

     <resource>

       <targetPath>/</targetPath>

       <directory>${project.build.directory}</directory>

       <include>${project.build.finalName}.jar</include>

     </resource>

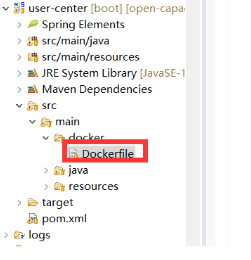
   </resources>

 </configuration>

</plugin>

采用docker方式打包到服务器

## 2.4 在各模块的src/main下面建docker包，在docker下面建dockerfile



Dockerfile 文件

FROM openjdk:8-jdk-alpine

VOLUME /tmp

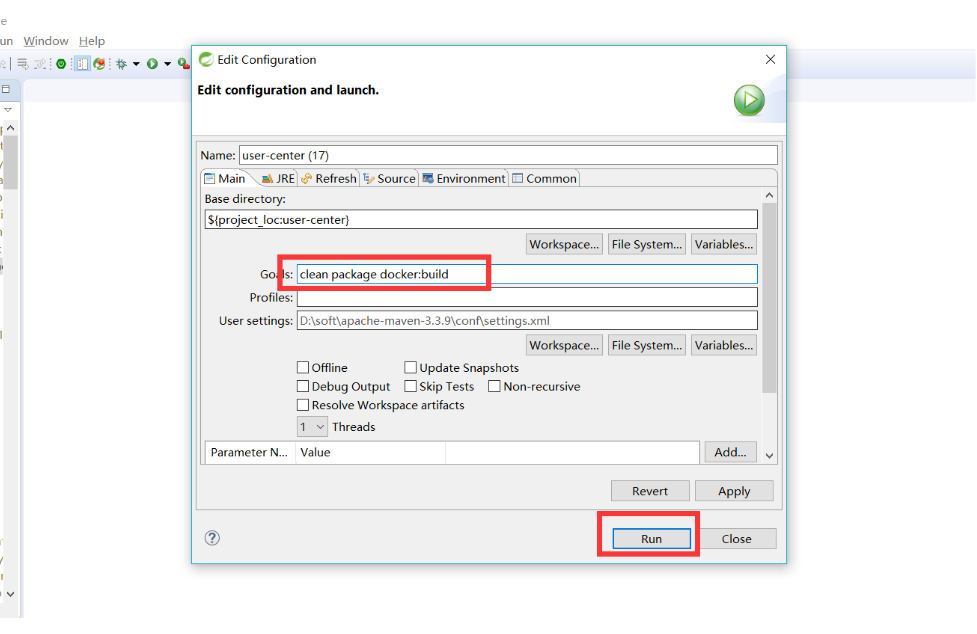
ADD user-center.jar app.jar

RUN sh -c 'touch /app.jar'

ENV JAVA\_OPTS=""

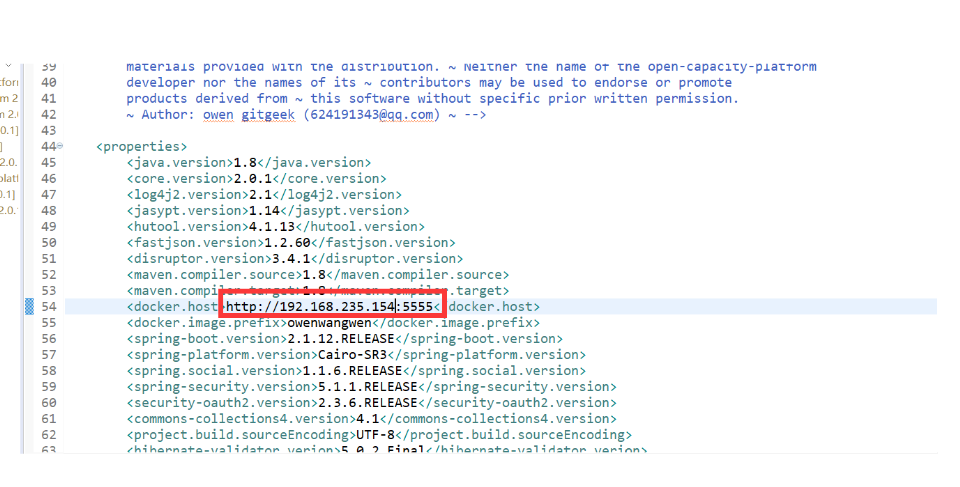
ENTRYPOINT [ "sh", "-c", "java $JAVA\_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app.jar" ]

项目打包成镜像

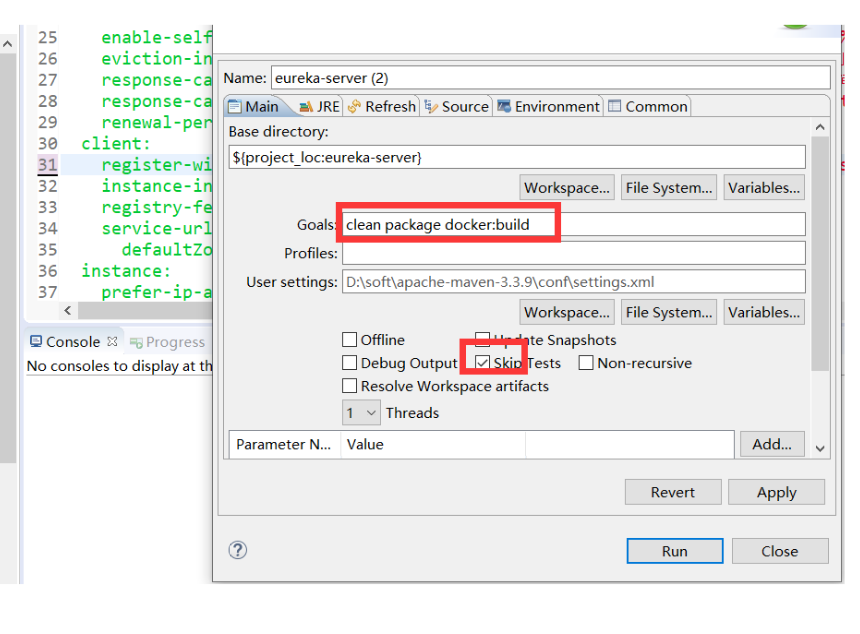


## 2.5 样例

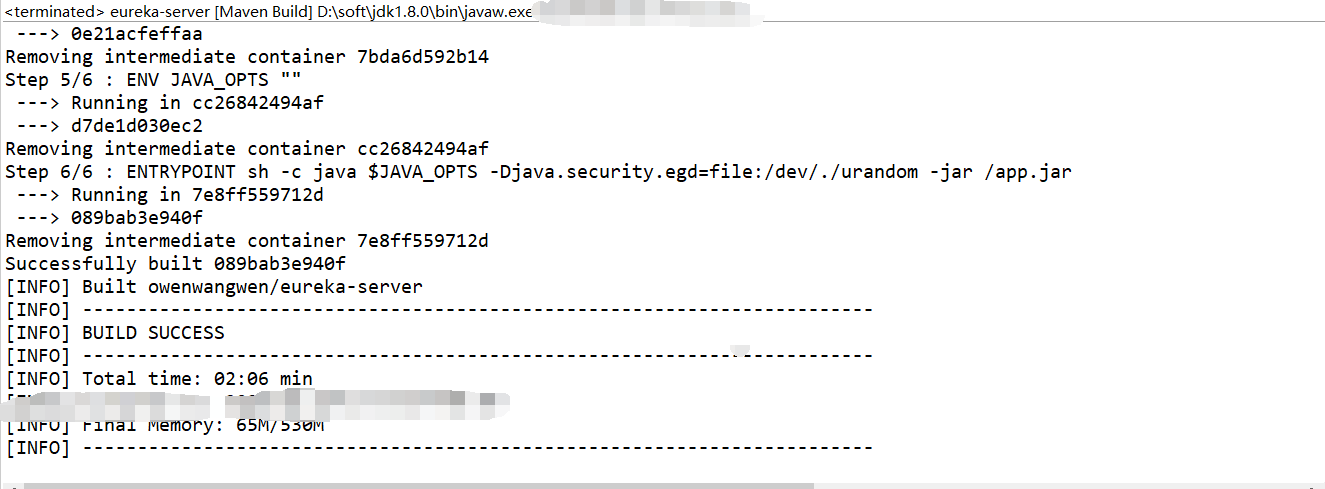
### 2.5.1 配置docker的地址



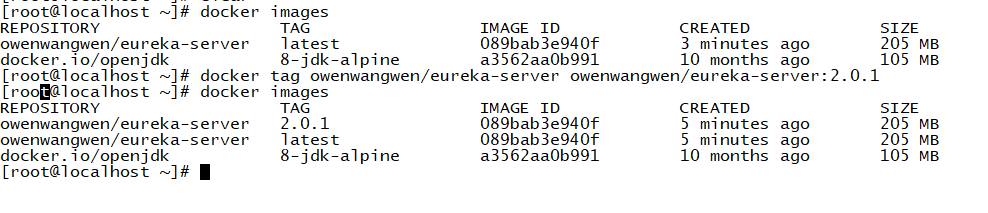
### 2.5.2 docker:build 形成镜像上传docker主机



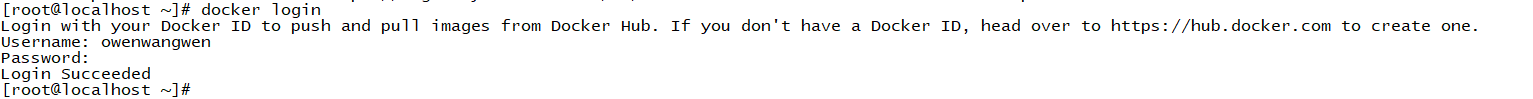
### 2.5.3 docker build 形成镜像过程



### 2.5.4登录主机查看镜像



### 2.5.5 docker login

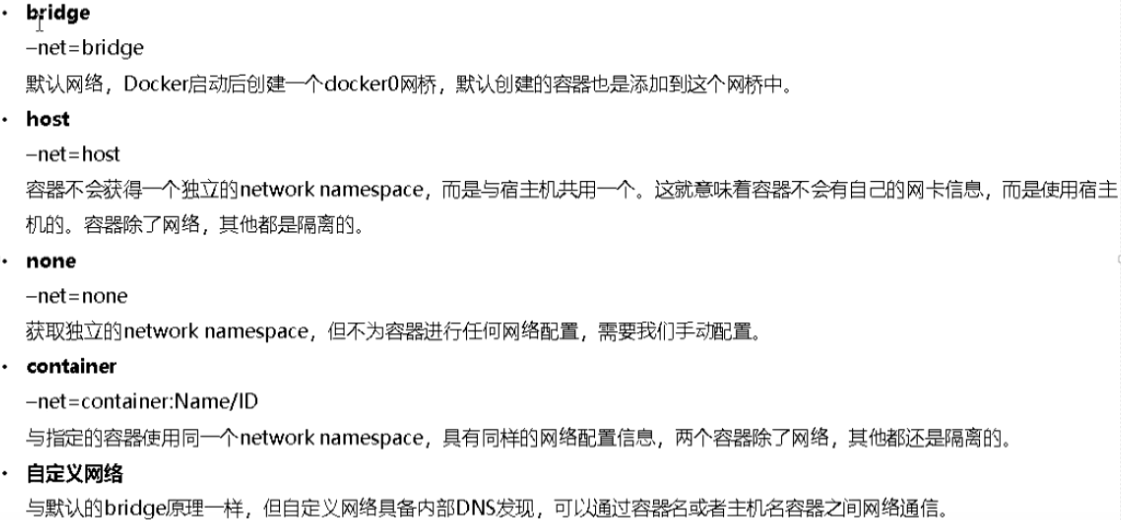


### 2.5.6 上传hubdocker

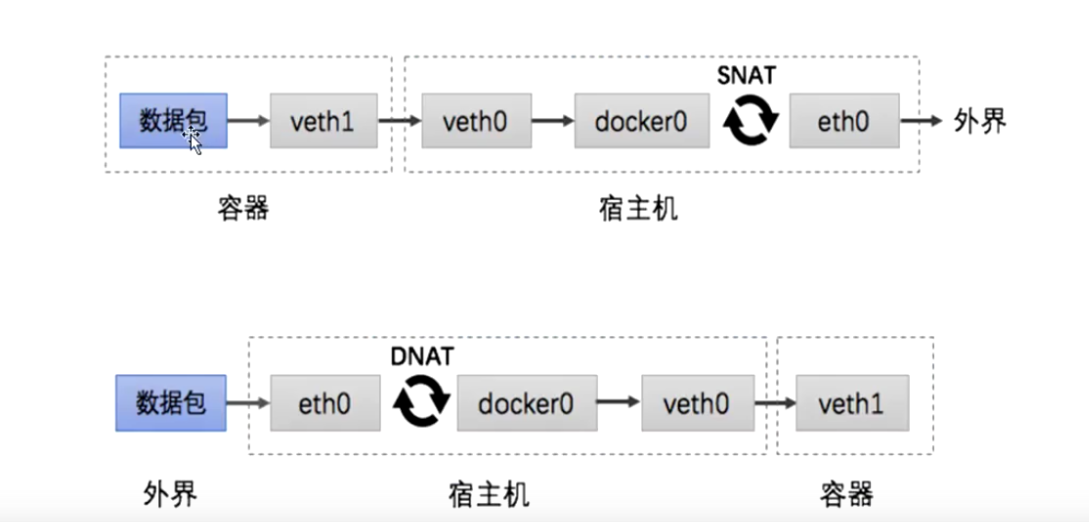
[root@localhost ~]# docker push owenwangwen/eureka-server

# 3 docker 网络模式

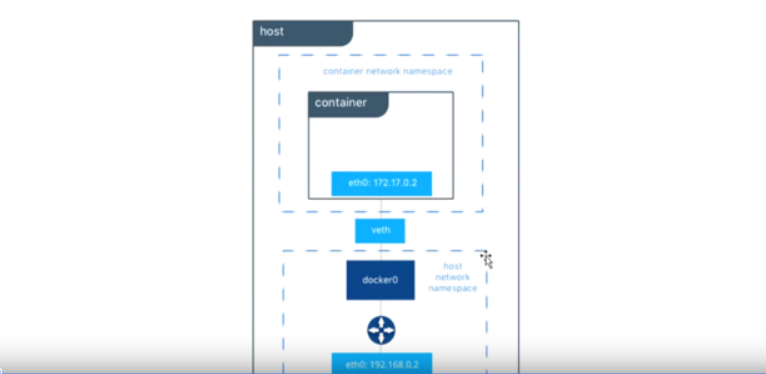
## 3.1 网络模式



## 3.2 单机docker网络原理



## 3.3 bridge,独立的网络命名空间(默认)



[root@localhost volumes]# docker run -it busybox

/# ip addr

1: lo: mtu 65536 qdisc noqueue qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

22: eth0@if23: mtu 1500 qdisc noqueue

link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff

inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0

valid\_lft forever preferred\_lft forever

#

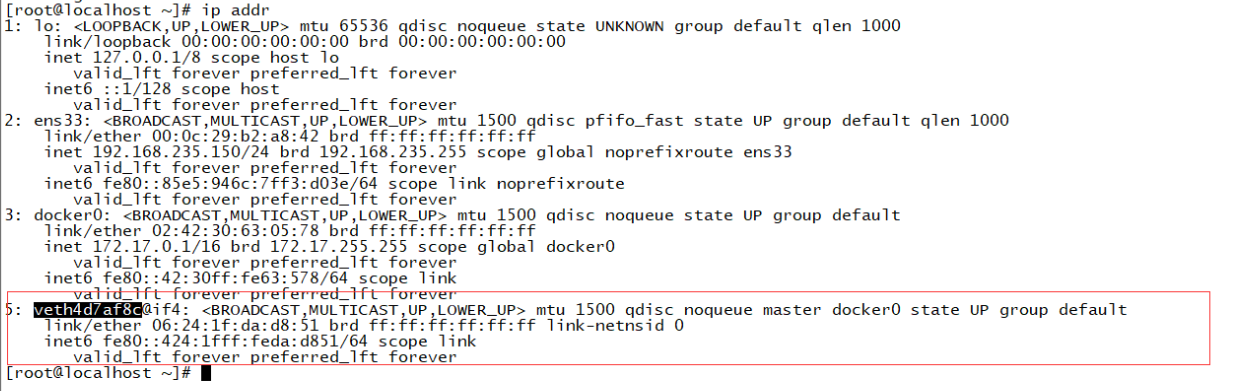
[root@localhost ~]#yum install yum list bridge-utils

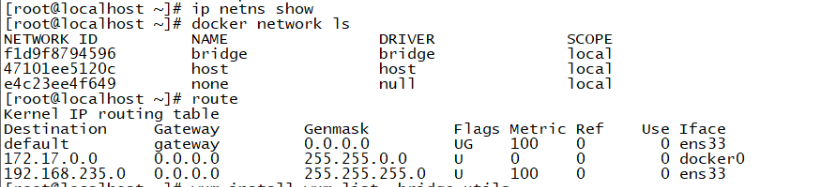
[root@localhost ~]# brctl show

bridge name bridge id STP enabled interfaces

docker0 8000.024230630578 no veth4d7af8c

[root@localhost ~]#





## 3.4 Host 使用宿主机的网络命名空间，能看到宿主机的端口

[root@localhost ~]# docker run --net=host -it busybox

/ # ip addr

1: lo: mtu 65536 qdisc noqueue qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\\_lft forever preferred\_lft forever

2: ens33: mtu 1500 qdisc pfifo\_fast qlen 1000

link/ether 00:0c:29:b2:a8:42 brd ff:ff:ff:ff:ff:ff

inet 192.168.235.150/24 brd 192.168.235.255 scope global ens33

valid\_lft forever preferred\_lft forever

inet6 fe80::85e5:946c:7ff3:d03e/64 scope link

valid\_lft forever preferred\_lft forever

3: docker0: mtu 1500 qdisc noqueue

link/ether 02:42:a7:69:a3:8b brd ff:ff:ff:ff:ff:ff

inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0

valid\_lft forever preferred\_lft forever

inet6 fe80::42:a7ff:fe69:a38b/64 scope link

valid\_lft forever preferred\_lft forever

17: vethafcb703@if16: mtu 1500 qdisc noqueue master docker0

link/ether 5a:94:a3:2e:5c:b4 brd ff:ff:ff:ff:ff:ff

inet6 fe80::5894:a3ff:fe2e:5cb4/64 scope link

valid\_lft forever preferred\\_lft forever

23: veth964ca9f@if22: mtu 1500 qdisc noqueue master docker0

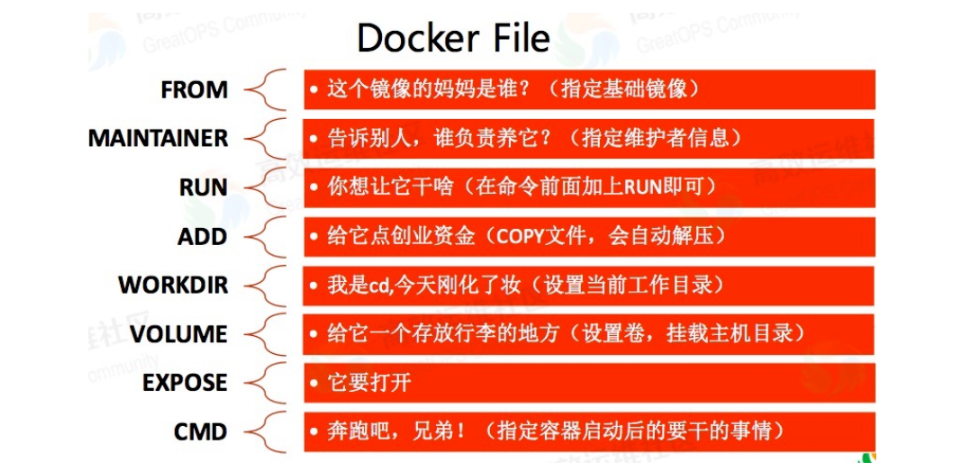
link/ether 3a:95:8a:34:70:c8 brd ff:ff:ff:ff:ff:ff

inet6 fe80::3895:8aff:fe34:70c8/64 scope link

valid\_lft forever preferred\_lft forever

/ #

# 4 DockerFile



## 4.1 Nginx

[root@localhost gitgeek\]# cat Dockerfile\_nginx

FROM centos:7

MAINTAINER gitgeek

RUN yum install -y gcc gcc-c++ make \

openssl-devel pcre-devel gd-devel \

iproute net-tools telnet wget curl && \

yum clean all && \

rm -rf /var/cache/yum/\*

RUN wget http://nginx.org/download/nginx-1.15.5.tar.gz && \

tar -zxvf nginx-1.15.5.tar.gz && \

cd nginx-1.15.5 && \

./configure --prefix=/usr/local/nginx \

--with-http\_ssl\_module \

--with-http\_stub\_status\_module && \

make -j 4 && make install && \

rm -rf /usr/local/nginx/html/\* && \

echo "ok" >> /usr/local/nginx/html/status.html && \

cd / && rm -rf nginx-1.15.5\* && \

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

ENV PATH $PATH:/usr/local/nginx/sbin

COPY nginx.conf /usr/local/nginx/conf/nginx.conf

WORKDIR /usr/local/nginx

EXPOSE 9090

CMD ["nginx" ,"-g" , "daemon off;"]

[root@localhost gitgeek\]# cat nginx.conf

user nobody ;

worker\_processes 4 ;

worker\_rlimit\_nofile 65535 ;

error\_log logs/error.log notice ;

pid /var/run/nginx.pid ;

events {

use epoll;

worker\_connections 4096 ;

}

http {

include mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"'

'$upstream\_addr';

access\_log logs/access.log main;

sendfile on;

tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 180;

fastcgi\_intercept\_errors on;

proxy\_intercept\_errors on;

#gzip on;

# 脛驴陋页脙脜脰

server{

listen 9090;

server\_name localhost;

location / {

root nlpt\_html;

index index.html index.htm;

}

location /aaz/ {

resolver 114.114.114.114;

proxy\_pass http://www.taobao.com/;

proxy\_buffers 256 4k;

proxy\_max\_temp\_file\_size 0k;

}

}

}

[root@localhost gitgeek]# docker build -t nginx:v1 -f Dockerfile\_nginx .

[root@localhost gitgeek]# docker run -d -p 9090:9090 nginx:v1

## 4.2 Tomcat

[root@localhost gitgeek]# cat Docker\_tomcat

FROM centos:7

MAINTAINER gitgeek

ENV VERSION=8.5.50

RUN yum install java-1.8.0-openjdk wget curl unzip iproute net-tools -y && \

yum clean all && \

rm -rf /var/cache/yum/\*

RUN wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v${VERSION}/bin/apache-tomcat-${VERSION}.tar.gz && \

tar -zxvf apache-tomcat-${VERSION}.tar.gz && \

mv apache-tomcat-${VERSION} /usr/local/tomcat && \

rm -rf apache-tomcat-${VERSION}.tar.gz /usr/local/tomcat/webapps/\* && \

mkdir /usr/local/tomcat/webapps/test && \

echo "ok" >> /usr/local/tomcat/webapps/test/status.html && \

sed -i '1a JAVA\_OPTS="-Djava.security.egd=file:/dev/./urandom"' /usr/local/tomcat/bin/catalina.sh && \

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

ENV PATH $PATH:/usr/local/tomcat/bin

WORKDIR /usr/local/tomcat

EXPOSE 8080

CMD ["catalina.sh","run"]

[root@localhost gitgeek]# docker build -t tomcat:v1 -f Docker\_tomcat .

[root@localhost gitgeek]# docker run -d -p 7777:8080 tomcat:v1

<http://192.168.235.150:7777/test/status.html>

## 4.3 Jenkins

[root@localhost gitgeek]# cat Dockerfile\_jenkins

FROM tomcat:v1

COPY jenkins.war /usr/local/tomcat/webapps/ROOT.war

[root@localhost gitgeek]# docker build -t jenkins:v1 -f Dockerfile\_jenkins .

Sending build context to Docker daemon 66.39MB

Step 1/2 : FROM tomcat:v1

---> e8288a9b5f6e

Step 2/2 : COPY jenkins.war /usr/local/tomcat/webapps/ROOT.war

---> 099bdecf7b52

Successfully built 099bdecf7b52

Successfully tagged jenkins:v1

[root@localhost gitgeek]# docker run -d -p 7777:8080 jenkins:v1

68b2ffccc2a6752a608694e48784483036f9da8ed6a6431021299521277371bc

[root@localhost gitgeek]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

68b2ffccc2a6 jenkins:v1 "catalina.sh run" 20 seconds ago Up 19 seconds 0.0.0.0:7777->8080/tcp quizzical\_gates

[root@localhost gitgeek]# docker exec -it 68b2ffccc2a6 bash

[root@68b2ffccc2a6 tomcat]# ls

BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work

[root@68b2ffccc2a6 tomcat]# cd webapps/

[root@68b2ffccc2a6 webapps]# ls

# 5 harbor部署

## 5.1 Docker-compose安装

curl -L [https://github.com/docker/compose/releases/download/1.24.0/docker-compose-](https://github.com/docker/compose/releases/download/1.24.0/docker-compose-)`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

## 5.2 安装harbor

wget https://storage.googleapis.com/harbor-releases/harbor-offline-installer-v1.6.1.tgz

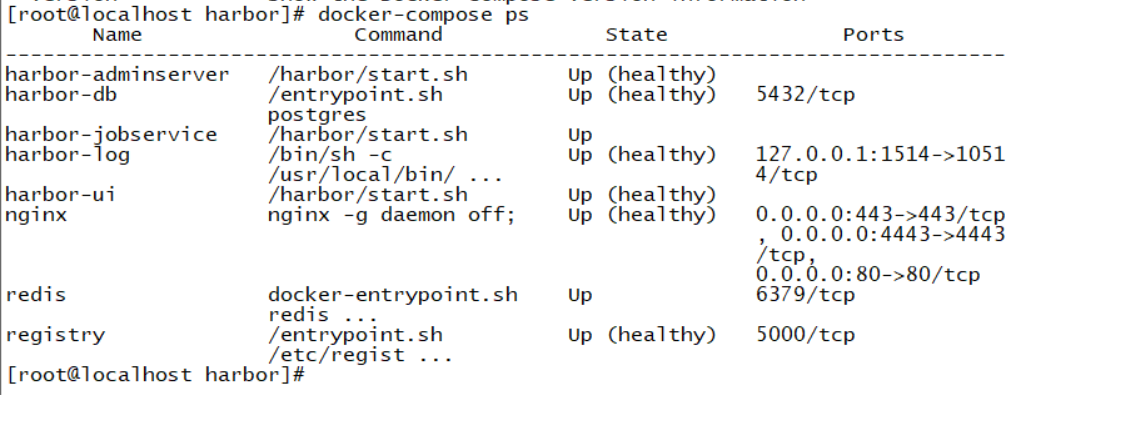
tar -zxvf harbor-offline-installer-v1.6.1.tgz

cd harbor/harbor

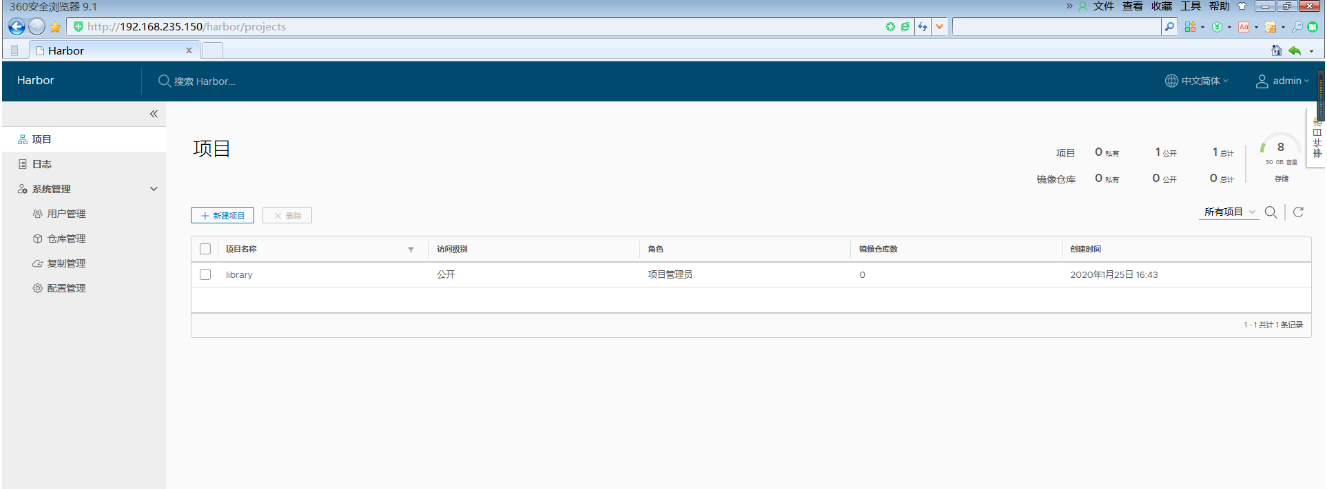
./prepare

./install.sh





## 5.3 访问harbor



## 5.4 使用harbor

[root@localhost harbor]# cat /etc/docker/daemon.json

{

"registry-mirrors": ["https://jvact13k.mirror.aliyuncs.com"],

"insecure-registries":["192.168.235.150"]

}

[root@localhost harbor]# docker login 192.168.235.150

Username: admin

Password:

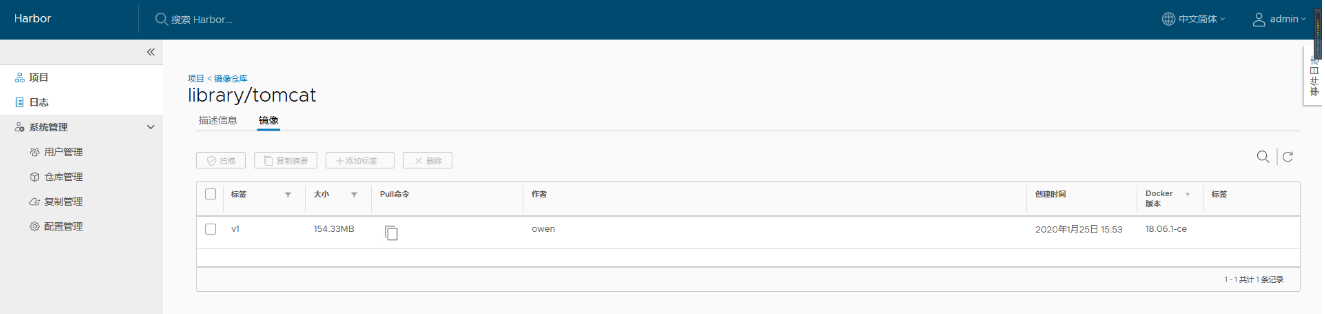
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

[root@localhost harbor]# docker tag tomcat:v1 192.168.235.150/library/tomcat:v1



# 6 使用自定义镜像

192.168.235.150

|  |  |  |
| --- | --- | --- |
| 软件 | 版本 | 备注 |
| Centos | 7.5 |  |
| Git | 1.8.31 |  |
| Harbor | 1.6.1 |  |

192.168.3.151

|  |  |  |
| --- | --- | --- |
| 软件 | 版本 | 备注 |
| Centos | 7.5 |  |
| Docker-ce | 18.06.1 |  |

192.168.3.152

|  |  |  |
| --- | --- | --- |
| 软件 | 版本 | 备注 |
| Centos | 7.5 |  |
| Jdk | 1.8 |  |
| Maven | 3.6 |  |
| Docker-ce | 18.06.1 |  |

## 6.1 168.235.150准备代码

[root@localhost harbor]# yum install -y git

[root@localhost harbor]# useradd git

[root@localhost harbor]# su - git

[git@localhost ~]$ ls

[git@localhost ocp.git]$ git --bare init

Initialized empty Git repository in /home/git/ocp.git/

[git@localhost ocp.git]$ ls

branches config description HEAD hooks info objects refs

[git@localhost ocp.git]$ cd ~ && mdkir open-capacity-platform && cd open-capacity-platform && git clone https://gitee.com/owenwangwen/open-capacity-platform.git

[git@localhost open-capacity-platform]$ vi .git/config

[core]

repositoryformatversion = 0

filemode = true

bare = false

logallrefupdates = true

[remote "origin"]

url = git@192.168.235.150:/home/git/open-capacity-platform.git

fetch = +refs/heads/\*:refs/remotes/origin/\*

[branch "master"]

remote = origin

merge = refs/heads/master

[git@localhost ~]$ cd ~ && mkdir open-capacity-platform.git

[git@localhost ~]$ ls

ocp.git open-capacity-platform open-capacity-platform.git

[git@localhost ~]$ cd open-capacity-platform.git

[git@localhost open-capacity-platform.git]$ ls

[git@localhost open-capacity-platform.git]$ git --bare init

Initialized empty Git repository in /home/git/open-capacity-platform.git/

[git@localhost open-capacity-platform.git]$

[git@localhost open-capacity-platform]$ git push origin master

git@192.168.235.150's password:

Counting objects: 17879, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (8449/8449), done.

Writing objects: 100% (17879/17879), 46.21 MiB | 10.74 MiB/s, done.

Total 17879 (delta 6137), reused 17273 (delta 5867)

To git@192.168.235.150:/home/git/open-capacity-platform.git

\* [new branch] master -> master

\*

## 6.2 168.235.151运行jenkins:v1 镜像

[root@localhost gitgeek]# cat Dockerfile\_jenkins

FROM tomcat:v1

COPY jenkins.war /usr/local/tomcat/webapps/ROOT.war

[root@localhost gitgeek]# docker build -t jenkins:v1 -f Dockerfile\_jenkins .

Sending build context to Docker daemon 66.39MB

Step 1/2 : FROM tomcat:v1

---> e8288a9b5f6e

Step 2/2 : COPY jenkins.war /usr/local/tomcat/webapps/ROOT.war

---> 099bdecf7b52

Successfully built 099bdecf7b52

Successfully tagged jenkins:v1

[root@localhost gitgeek]# docker run -d -p 8080:8080 jenkins:v1

68b2ffccc2a6752a608694e48784483036f9da8ed6a6431021299521277371bc

[root@localhost gitgeek]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

68b2ffccc2a6 jenkins:v1 "catalina.sh run" 20 seconds ago Up 19 seconds 0.0.0.0:7777->8080/tcp quizzical\_gates

[root@localhost gitgeek]# docker exec -it 68b2ffccc2a6 bash

[root@68b2ffccc2a6 tomcat]# ls

BUILDING.txt CONTRIBUTING.md LICENSE NOTICE README.md RELEASE-NOTES RUNNING.txt bin conf lib logs temp webapps work

[root@68b2ffccc2a6 tomcat]# cd webapps/

[root@68b2ffccc2a6 webapps]# ls

## 6.3 jenkins

访问<http://192.168.235.151:8080>



[root@localhost ~]# docker exec -it b6a29df78638 bash

[root@b6a29df78638 tomcat]# cat /root/.jenkins/secrets/initialAdminPassword

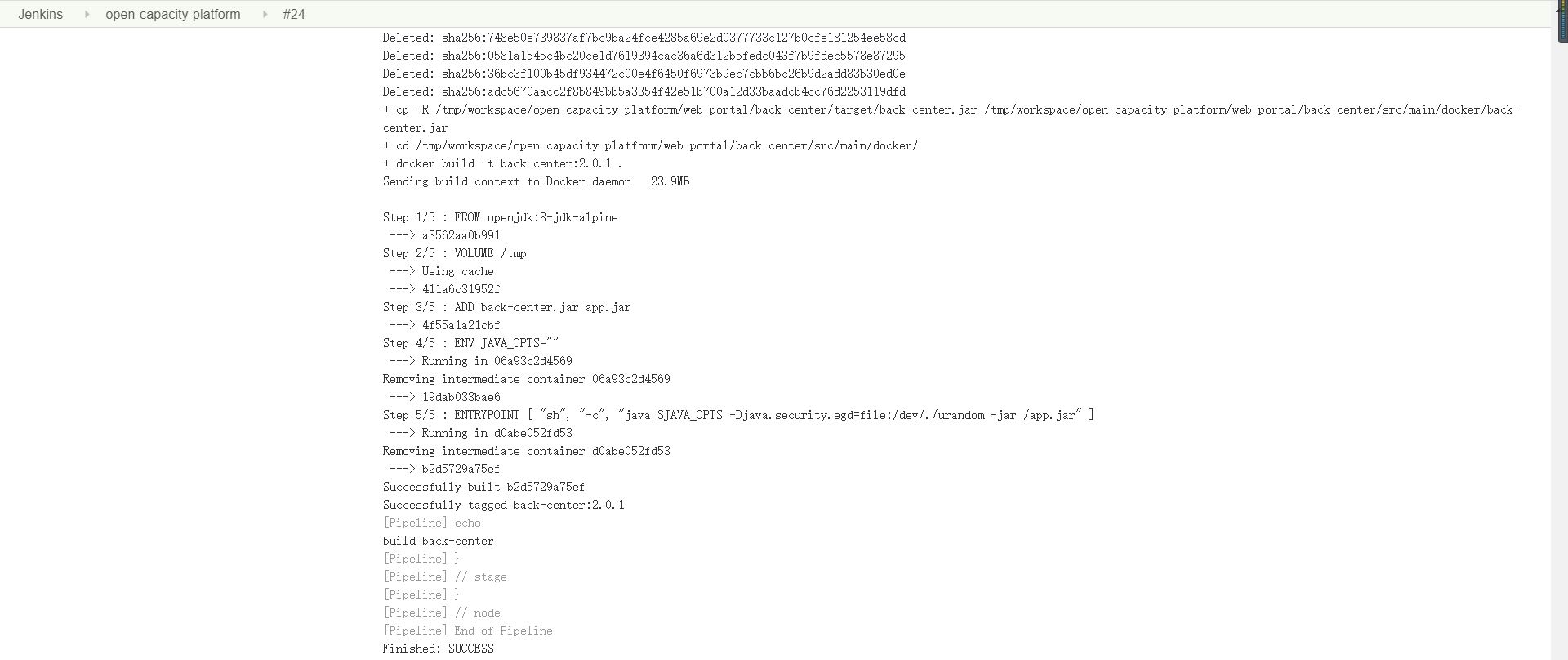
输入密码点击继续后，可能会出现 【该jenkins实例似乎已离线】错误，解决方法如下：

已经离线界面不要动，然后打开一个新的tab，输入网址http://192.168.235.151:8080/jenkins/pluginManager/advanced。这里面最底下有个【升级站点】，把其中的链接改成http的就好了，http://updates.jenkins.io/update-center.json。然后在重启http://192.168.235.151:8080/restart，刷新刚刚的界面，这样就能正常联网了。

## 6.4构建过程



## 6.5构建结果



## 6.6核查结果 https://img.kancloud.cn/55/93/559333e0c3d207979d3de0e063e06c31_1534x160.png