# Centos7 docker-engine nginx 私有仓库搭建

## 服务端docker安装配置

* 1. 官网安装地址：<https://docs.docker.com/engine/installation/linux/centos/>

Centos7 下安装docker-engine是使用centos7 systemd管理配置，配置文件不在/etc/sysconfig/docker,需修改/lib/systemd/system/docker.service

[rd@iZ25an6zbb7Z registry]$ cat /lib/systemd/system/docker.service

[Unit]

Description=Docker Application Container Engine

Documentation=http://docs.docker.com

After=network.target

Wants=docker-storage-setup.service

[Service]

Type=notify

EnvironmentFile=-/etc/sysconfig/docker #指定主配置文件路径

EnvironmentFile=-/etc/sysconfig/docker-storage

EnvironmentFile=-/etc/sysconfig/docker-network

Environment=GOTRACEBACK=crash

ExecStart=/usr/bin/docker daemon $OPTIONS $DOCKER\_STORAGE\_OPTIONS $DOCKER\_NETWORK\_OPTIONS $ADD\_REGISTRY $BLOCK\_REGISTRY $INSECURE\_REGISTRY

LimitNOFILE=1048576

LimitNPROC=1048576

LimitCORE=infinity

MountFlags=slave

TimeoutStartSec=1min

[Install]

WantedBy=multi-user.target

[rd@iZ25an6zbb7Z registry]$ cat /etc/sysconfig/docker

OPTIONS="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock"

#DOCKER\_CERT\_PATH=/etc/docker

#INSECURE\_REGISTRY='--insecure-registry 101.201.117.71:5000'

* 1. 服务端安装Docker-compose

Docker-compose是一个非常有用的Docker运行，管理的工具。你可以通过定义compose文件，使用简单的一条命令同时起多个Docker

Container运行不同的服务。Docker-compose对于开发，测试，环境保存以及CI都提供了非常大的便利。

curl -L https://github.com/docker/compose/releases/download/1.9.0/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose

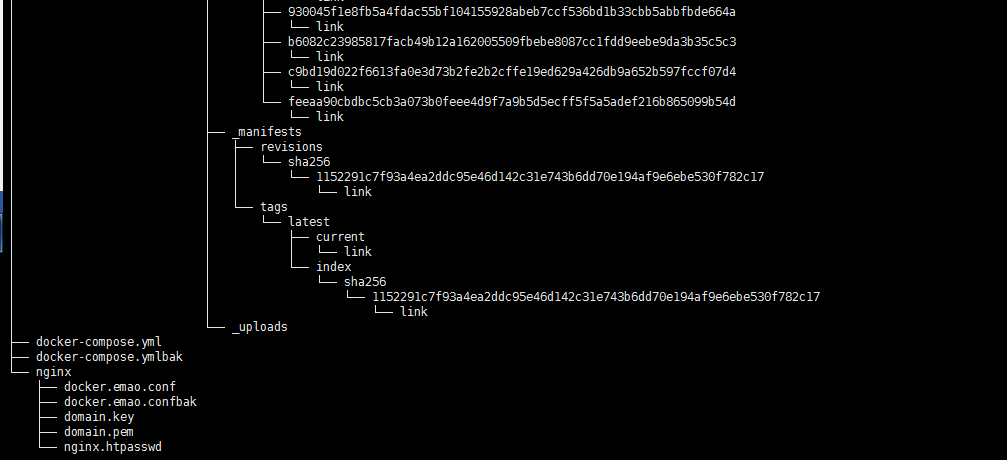
chmod +x /usr/local/bin/docker-compose

* 1. 创建一个工作目录

mkdir –p /opt/data/registry

registry目录下创建data目录(存储)，nginx目录





在registty目录下创建docker-compose.yml

[rd@iZ25an6zbb7Z registry]$ cat docker-compose.yml

nginx:

image: "nginx:1.9" #使用nginx1.9的镜像

ports:

- 443:443

links:

- registry:registry #连接一个容器

volumes:

- /opt/data/registry/nginx:/etc/nginx/conf.d #挂载到/etc/nginx/conf.d

registry:

image: registry:2.2 #使用registry:2.2镜像

ports:

-127.0.0.1:5000:5000 #只允许本地连接5000端口

environment:

REGISTRY\_STORAGE\_FILESYSTEM\_ROOTDIRECTORY: /data

volumes:

- /opt/data/registry/data:/data

#说明：

此配置文件定义了两个Container;

一个名为nginx使用镜像为：nginx:1.9；映射了443端口到宿主机上，同时连接了另一个容器registry;

并挂载了一个数据卷目录为：/data/docker/nginx挂载到容器中的/etc/nginx/conf.d 目录；

另一个名为registry使用的镜像为：registry:2.2；映射了端口5000到宿主机，并定义了镜像的存储位置：/data

【此目录为容器内的目录】；挂载一个数据卷：/data/docker/data到容器的/data目录;

阿里云oss配置

registry:

image: registry:2.2 #使用registry:2.2镜像

ports:

-127.0.0.1:5000:5000 #只允许本地连接5000端口

environment:

REGISTRY\_STORAGE\_OSS\_ACCESSKEYID=your\_oss\_accesskey\_id - REGISTRY\_STORAGE\_OSS\_ACCESSKEYSECRET=your\_oss\_accesskey\_secret - REGISTRY\_STORAGE\_OSS\_REGION=your\_oss\_region - REGISTRY\_STORAGE\_OSS\_BUCKET=your\_oss\_bucket - REGISTRY\_STORAGE\_OSS\_ENDPOINT=your\_oss\_bucket.your\_oss\_region.aliyuncs.com

## 服务端nginx配置

* 1. 创建nginx配置文件

cd /opt/data/registry/nginx

[rd@iZ25an6zbb7Z nginx]$ cat docker.emao.conf

upstream docker-registry {

server registry:5000;

}

server {

listen 443;

server\_name localhost;

# SSL

ssl on;

ssl\_certificate /etc/nginx/conf.d/domain.pem; #证书配置

ssl\_certificate\_key /etc/nginx/conf.d/domain.key; #证书配置

client\_max\_body\_size 0;

chunked\_transfer\_encoding on;

location /v2/ {

# Do not allow connections from docker 1.5 and earlier

# docker pre-1.6.0 did not properly set the user agent on ping, catch "Go \*" user agents

if ($http\_user\_agent ~ "^(docker\/1\.(3|4|5(?!\.[0-9]-dev))|Go ).\*$" ) {

return 404;

}

# To add basic authentication to v2 use auth\_basic setting plus add\_header

auth\_basic "registry.localhost";

auth\_basic\_user\_file /etc/nginx/conf.d/nginx.htpasswd;

add\_header 'Docker-Distribution-Api-Version' 'registry/2.2' always;

proxy\_pass http://docker-registry;

proxy\_set\_header Host $http\_host; # required for docker client's sake

proxy\_set\_header X-Real-IP $remote\_addr; # pass on real client's IP

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_read\_timeout 600;

}

}

* 1. 自己生成ssl证书

弄了好久出现各种问题最后放弃了从网上申请了个，欢迎大家补充

* 1. docker-compos启动

cd 到docker-compose.yml目录

docker-compose up –d(-d dubug模式 –f指定\*.yml文件)

[rd@iZ25an6zbb7Z registry]$ docker-compose up -d

Starting registry\_registry\_1

Starting registry\_nginx\_1

[rd@iZ25an6zbb7Z registry]$

初次执行会先下载nginx、registry镜像

## 客户端安装

* 1. 直接按官网安装即可不需要任何配置
  2. 将服务端证书domain.pem追加到/etc/ssl/certs/ca-certificates.crt
  3. /etc/hosts文件增加 ip 域名
  4. 测试登录

[rd@iZ2557l9sefZ certs]$ sudo docker login docker.su12.cn

Username (test): test

Password:

Login Succeeded

[rd@iZ2557l9sefZ certs]$

客户端安装已完成

## 常见错误

* 1. self-signed certs - x509: certificate signed by unknown authority
     1. centos7

从服务端拷贝证书到客户端cat /etc/pki/ca-trust/source/anchors/domain.pem |sudo tee -a /etc/ssl/certs/ca-certificates.crt

* + 1. centos6

从服务端拷贝证书到客户端

Install the ca-certificates package:

yum install ca-certificates

Enable the dynamic CA configuration feature:

update-ca-trust force-enable

Add it as a new file to /etc/pki/ca-trust/source/anchors/:

cp foo.crt /etc/pki/ca-trust/source/anchors/

Use command:

update-ca-trust extract

* + 1. centos5

从服务端拷贝证书到客户端

Append your trusted certificate to file /etc/pki/tls/certs/ca-bundle.crt

cat foo.crt >> /etc/pki/tls/certs/ca-bundle.crt

* 1. 登录报Error response from daemon: Get https://docker.su12.cn/v1/users/: dial tcp 101.201.117.71:443: getsockopt: connection refused

服务端查看443端口是否存在，nginx配置文件是否开启443启用证书

netstat -anlp|grep 443

* 1. Docker-compose up 报specify the URL with the DOCKER\_HOST environment variable
     1. Centos6 /etc/sysconfig/docker增加other\_args="-H tcp://127.0.0.1:2375 -H unix:///var/run/docker.sock"

vim /etc/sysconfig/docker

# /etc/sysconfig/docker

#

# Other arguments to pass to the docker daemon process

# These will be parsed by the sysv initscript and appended

# to the arguments list passed to docker -d

other\_args="-H tcp://127.0.0.1:2375 -H unix:///var/run/docker.sock"

同时在/etc/profile文件增加export DOCKER\_HOST=tcp://localhost:2375

* + 1. Centos7

详见服务端安装配置

* 1. 当client端docker login到官方的https://index.docker.io/v1/网站,出现x509: certificate signed by unknown authority错误时

重命名根证书mv /etc/pki/tls/certs/ca-certificates.crt /etc/pki/tls/certs/ca-certificates.crt.bak     
重启docker服务! service docker restart!

* 1. Error response from daemon: Get https://docker.su12.cn/v2/: Get https://10.45.40.247/service/token?account=admin&client\_id=docker&offline\_token=true&service=token-service: x509: cannot validate certificate for 10.45.40.247 because it doesn't contain any IP SANs
     1. /etc/sysconfig/docker文件加上

ADD\_REGISTRY='--add-registry docker.su12.cn'

DOCKER\_OPTS="--insecure-registry docker.su12.cn"

INSECURE\_REGISTRY='--insecure-registry docker.su12.cn'

## 其它安装

* 1. Linux只CentOS 7 上安装febootstrap

由于在CentOS7 上不在提供febootstrap 相关的包了，所以直接yum安装是不行的，于是我就直接去CentOS 6.8的包里找出了febootstrap ，然后在转到CentOS 7上安装，也成功了

在这里分享下我找出来的包

yum install epel-release -y

yum install -y \

https://www.dwhd.org/wp-content/uploads/2016/06/febootstrap-3.21-4.el6\_.x86\_64.rpm \

https://www.dwhd.org/wp-content/uploads/2016/06/fakechroot-libs-2.9-24.5.el6\_1.1.x86\_64.rpm \

https://www.dwhd.org/wp-content/uploads/2016/06/febootstrap-supermin-helper-3.21-4.el6\_.x86\_64.rpm \

https://www.dwhd.org/wp-content/uploads/2016/06/fakechroot-2.9-24.5.el6\_1.1.x86\_64.rpm

## 常用命令

* 1. 查看镜像

V1版本curl 127.0.0.1:5000/v1/search

V2版本curl <https://test:123456@docker.su12.cn/v2/_catalog>

* 1. 制作镜像
     1. febootstarp制作镜像
        1. febootstrap -i bash -i wget -i yum -i iputils -i iproute -i man -i vim -i openssh-server -i openssh-clients -i tar -i gzip centos6 centos6-image <http://mirrors.aliyun.com/centos/6/os/x86_64/>

#参数说明：

-i 镜像所需要安装的工具：把-i后面的参数传递给yum来实现安装，上面安装了ssh服务

centos6 是centos版本

centos6-image 是指定目录

url 指定源可以是163或者搜狐的源

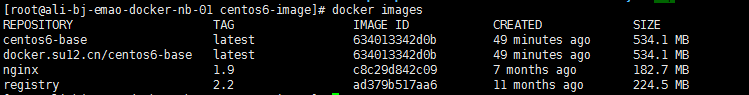
执行完后进入centos-base目录



接下来创建一个基本的镜像

cd centos6-image && tar -c .|docker import - centos6-base

查看镜像



* + 1. dockerfile文件创建镜像

Dockerfile文件的说明：

格式：

INSTRUCTION arguments

虽然指令忽略大小写，但建议用大写

# 是注释

MAINTAINER命令：

MAINTAINER命令用来指定维护者的姓名和联系方式

FROM命令：

FROM <image>/<image>:<tag>

这个是设置基本的镜像，为后续的命令使用，所以应该作为Dockerfile的第一条指令

比如：

FROM centos6-base

RUN 命令：

会上面FROM 指定的镜像里执行任何命令，然后提交（commit）结果，提交的镜像会在后面继续用到

格式：

RUN <command> (这个命令运行一个shell中 - '/bin/sh -c')

或：

RUN ["executable", "param1", "param2"]

RUN 命令等价于：

docker run image\_name command

CMD and ENTRYPOINT

命令设置在容器启动时执行命令

ENTRYPOINT，表示镜像在初始化时需要执行的命令，不可被重写覆盖，需谨记

CMD，表示镜像运行默认参数，可被重写覆盖

ENTRYPOINT/CMD都只能在文件中存在一次，并且最后一个生效 多个存在，只有最后一个生效，其它无效！

需要初始化运行多个命令，彼此之间可以使用 && 隔开，但最后一个须要为无限运行的命令，需切记！

ENTRYPOINT/CMD，一般两者可以配合使用，比如：

ENTRYPOINT ["/usr/sbin/sshd"]

CMD ["-D"]

\*\* 在Docker　daemon模式下，无论你是使用ENTRYPOINT，还是CMD，最后的命令，一定要是当前进程需要一直运行的，才能够防容器退出。\*\*\*

以下无效方式：

ENTRYPOINT service tomcat7 start #运行几秒钟之后，容器就会退出

CMD service tomcat7 start #运行几秒钟之后，容器就会退出

这样有效：

ENTRYPOINT service tomcat7 start && tail -f /var/lib/tomcat7/logs/catalina.out

# 或者

CMD service tomcat7 start && tail -f /var/lib/tomcat7/logs/catalina.out

这样也有效：

ENTRYPOINT ["/usr/sbin/sshd"]

CMD ["-D"]

USER 命令

比如指定 memcached 的运行用户，可以使用上面的 ENTRYPOINT or CMD来实现:

ENTRYPOINT ["memcached", "-u", "daemon"]

更好的方式：

ENTRYPOINT ["memcached"]

USER daemon

EXPOSE 命令

EXPOSE 命令可以设置一个端口在运行的镜像中暴露在外

在docker使用--link来链接两容器时会用到相关端口

EXPOSEd <port>

ENV命令：

用于设置环境变更

使用此dockerfile生成的image新建container，可以通过 docker inspect CONTAINER ID 看到这个环境变量

也可以通过在docker run时设置或修改环境变量

ADD 命令：

从src复制文件到container的dest路径:

ADD <src> <dest>

<src> 是相对被构建的源目录的相对路径，可以是文件或目录的路径，也可以是一个远程的文件url

<dest> 是container中的绝对路径

VOLUME 命令

VOLUME ["<mountpoint>"]

如:

VOLUME ["/data"]

创建一个挂载点用于共享目录

WORKDIR 命令

WORKDIR /path/to/workdir

配置RUN, CMD, ENTRYPOINT 命令设置当前工作路径

可以设置多次，如果是相对路径，则相对前一个 WORKDIR 命令

比如:

WORKDIR /a WORKDIR b WORKDIR c RUN pwd

其实是在 /a/b/c 下执行 pwd

完整Dockerfile文件

#Dockerfile

FROM centos6-base

MAINTAINER chenshiyang460 <chenshiyang460@emao.com>

RUN ssh-keygen -q -N "" -t dsa -f /etc/ssh/ssh\_host\_dsa\_key

RUN ssh-keygen -q -N "" -t rsa -f /etc/ssh/ssh\_host\_rsa\_key

RUN sed -ri 's/session required pam\_loginuid.so/#session required pam\_loginuid.so/g' /etc/pam.d/sshd

RUN mkdir -p /root/.ssh && chown root.root /root && chmod 700 /root/.ssh

EXPOSE 22

RUN echo 'root:centos' | chpasswd

#RUN yum install -y yum-priorities && rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm && rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

#RUN yum install tar gzip gcc vim wget -y

ENV LANG en\_US.UTF-8

ENV LC\_ALL en\_US.UTF-8

CMD /usr/sbin/sshd -D

#End

如果报错

rpmdb: /var/lib/rpm/Name: unexpected file type or format

error: cannot open Name index using db3 - Invalid argument (22)

解决方法：

rm -f /var/lib/rpm/\_\_db\*  
rpm –rebuilddb

yum clean all

Centos7搭建harbor私有仓库

Harbor项目是帮助用户迅速搭建一个企业级的registry 服务。

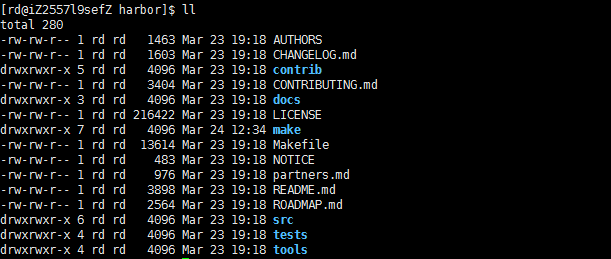
它以Docker公司开源的registry为基础，提供了管理UI, 基于角色的访问控制(Role Based Access Control)，AD/LDAP集成、以及审计日志(Audit logging) 等企业用户需求的功能，同时还原生支持中文

安装步骤

前提是已安装好docker docker-compose

* + - 1. 下载harbor

git clone <https://github.com/vmware/harbor.git>



* + - 1. 修改配置

cd /home/rd/harbor/make

vim harbor.cfg

[rd@iZ2557l9sefZ make]$ egrep -v "^#|^$" harbor.cfg

hostname = 10.45.40.247 --服务器ip

ui\_url\_protocol = https

email\_identity =

email\_server = smtp.emao.com

email\_server\_port = 25

email\_username = [statistics@memao.com](mailto:statistics@memao.com) --邮箱相关配置

email\_password = 1maoQiche@

email\_from = admin <statistics@emao.com>

email\_ssl = false

harbor\_admin\_password = Harbor12345 --默认登录密码

auth\_mode = db\_auth

ldap\_url = ldaps://ldap.mydomain.com

ldap\_basedn = ou=people,dc=mydomain,dc=com

ldap\_uid = uid

ldap\_scope = 3

db\_password = root123 --数据库密码

self\_registration = on

use\_compressed\_js = on

max\_job\_workers = 3

token\_expiration = 30

verify\_remote\_cert = on

customize\_crt = on

crt\_country = CN

crt\_state = State

crt\_location = CN

crt\_organization = emao

crt\_organizationalunit = emao

crt\_commonname = emao.com

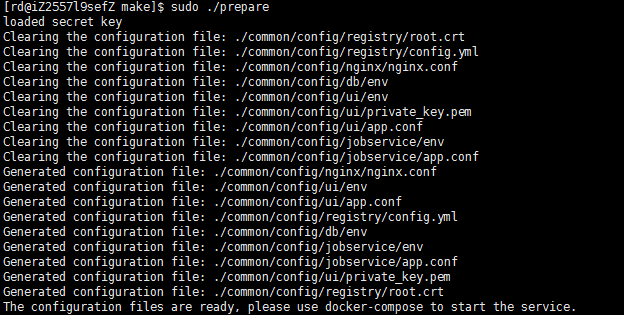
crt\_email = statistics@emao.com

project\_creation\_restriction = everyone

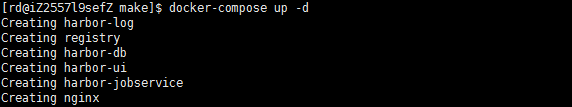
ssl\_cert = /opt/data/registry/nginx/domain.pem --https证书

ssl\_cert\_key = /opt/data/registry/nginx/domain.key

* + - 1. cd /home/rd/harbor/make 运行 ./ prepare更新配置



* + - 1. 启动



* + - 1. 使用 http://userIP/ 访问 Harbor

使用 帐号 admin, 密码为 配置文件中 harbor\_admin\_password = Harbor12345 的密码 登陆

* + - 1. docker需要上传push镜像，需要在docker中配置 --insecure-registry userIP 或者在nginx中配置 https配置完毕以后，重启docker使用 docker login userIP 登陆 Harbor

[rd@iZ256n5otzvZ docker.su12.cn:443]$ egrep -v "^#|^$" /etc/sysconfig/docker

OPTIONS="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock"

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

DOCKER\_CERT\_PATH=/etc/docker

fi

ADD\_REGISTRY='--add-registry docker.su12.cn'

DOCKER\_OPTS="--insecure-registry docker.su12.cn"

INSECURE\_REGISTRY='--insecure-registry docker.su12.cn'

[rd@iZ2557l9sefZ make]$ sudo docker login docker.su12.cn

Username (test): admin

Password:

Login Succeeded

* + - 1. 报错请参考私有仓库搭建常见错误