Docker+harbor私有仓库https

2022年2月28日 14:58

参考文章

https://blog.csdn.net/networken/article/details/107502461
https://blog.csdn.net/u013078871/article/details/112584573
https://www.cnblogs.com/cjwnb/p/13441071.html

环境ubuntu1804

1，docker和docker-compose安装

docker安装

curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun

```
root@master:~# docker version
Client: Docker Engine - Community
 Version:           20.10.12
 API version:       1.41
 Go version:        go1.16.12
 Git commit:        e91ed57
 Built:             Mon Dec 13 11:45:27 202
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true
```

docker-compose安装

curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose

```
root@master:~# docker-compose -version
docker-compose version 1.29.2, build 5becea4c
root@master:~#
```

2，安装harbor版本v1.10.10

离线包下载：

https://github.com/goharbor/harbor/releases
https://mirrors.tuna.tsinghua.edu.cn/github-release/goharbor/harbor/

在线安装

```
version=v1.10.10
mkdir -p /data/packages && cd /data/packages
wget https://github.com/goharbor/harbor/releases/download/${version}/harbor-offline-installer-${version}.tgz
tar -zxf harbor-offline-installer-${version}.tgz
```

http访问方式直接修改harbor.yml配置文件定义好主机名，注释掉关于https模式的相关参数直接安装即可，下面说https模式

首先生成证书私钥，下面所有的IP更换成自己的

```
mkdir -p /root/harbor/ssl
cd /root/harbor/ssl
```

生成颁发机构私钥

```
openssl genrsa -out ca.key 4096
```

办法机构证书

```
openssl req -x509 -new -nodes -sha512 -days 3650 \
 -subj "/C=CN/ST=Beijing/L=Beijing/O=example/OU=Personal/CN=192.168.233.130" \
 -key ca.key \
 -out ca.crt
```

#192.168.233.130，更换成自己的IP

生成服务器私钥

```
openssl genrsa -out 192.168.233.130.key 4096
```

生成服务器签名证书

```
openssl req -sha512 -new \
   -subj "/C=CN/ST=Beijing/L=Beijing/O=example/OU=Personal/CN=192.168.233.130" \
   -key 192.168.233.130.key \
   -out 192.168.233.130.csr
```

生成509v3扩展文件

```
cat > v3.ext <<-EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = IP:192.168.233.130
EOF
```

使用该文件v3.ext为harbor主机生成证书

```
openssl x509 -req -sha512 -days 3650 \
   -extfile v3.ext \
   -CA ca.crt -CAkey ca.key -CAcreateserial \
   -in 192.168.233.130.csr \
   -out 192.168.233.130.crt
```

提供证书给harbor和docker

生成后ca.crt，harbor.od.com.crt和harbor.od.com.key文件，必须将它们提供给Harbor和docker，重新配置它们

```
mkdir -p /data/cert/
cp 192.168.233.130.crt /data/cert/
cp 192.168.233.130.key /data/cert/
```

转换crt为cert，供docker使用

```
openssl x509 -inform PEM -in 192.168.233.130.crt -out 192.168.233.130.cert
```

复制服务器证书，密钥和ca文件到docker证书文件夹中，docker文件夹位于/etc/docker/certs.d/下，要新建一个文件夹

```
mkdir -p /etc/docker/certs.d/192.168.233.130/
cp 192.168.233.130.cert /etc/docker/certs.d/192.168.233.130/
cp 192.168.233.130.key /etc/docker/certs.d/192.168.233.130/
cp ca.crt /etc/docker/certs.d/192.168.233.130/
```

重启docker，编辑barbor.yml文件，修改hostname部分certificate和private_key部分，

```
# Configuration file of Harbor

# The IP address or hostname to access admin UI and registry service.
# DO NOT use localhost or 127.0.0.1, because Harbor needs to be accessed by external cl
hostname: 192.168.233.130

# http related config
http:
  # port for http, default is 80. If https enabled, this port will redirect to https po
  port: 80

# https related config
https:
  # https port for harbor, default is 443
  port: 443
  # The path of cert and key files for nginx
  certificate: /data/cert/192.168.233.130.crt
  private_key: /data/cert/192.168.233.130.key

# Uncomment external_url if you want to enable external proxy
# And when it enabled the hostname will no longer used
# external_url: https://reg.mydomain.com:8433

# The initial password of Harbor admin
# It only works in first time to install harbor
# Remember Change the admin password from UI after launching Harbor.
harbor_admin_password: Harbor12345
```

默认密码为Harbor12345

执行./install.sh开始安装

./install.sh

完整的安装日志

```
root@master:/data/packages/harbor# ./install.sh

[Step 0]: checking if docker is installed ...

Note: docker version: 20.10.12

[Step 1]: checking docker-compose is installed ...

Note: docker-compose version: 1.29.2

[Step 2]: loading Harbor images ...
Loaded image: goharbor/harbor-portal:v1.10.10
Loaded image: goharbor/registry-photon:v1.10.10
Loaded image: goharbor/clair-adapter-photon:v1.10.10
Loaded image: goharbor/chartmuseum-photon:v1.10.10
Loaded image: goharbor/notary-signer-photon:v1.10.10
Loaded image: goharbor/prepare:v1.10.10
Loaded image: goharbor/harbor-core:v1.10.10
Loaded image: goharbor/harbor-registryctl:v1.10.10
Loaded image: goharbor/redis-photon:v1.10.10
Loaded image: goharbor/harbor-log:v1.10.10
Loaded image: goharbor/harbor-db:v1.10.10
```

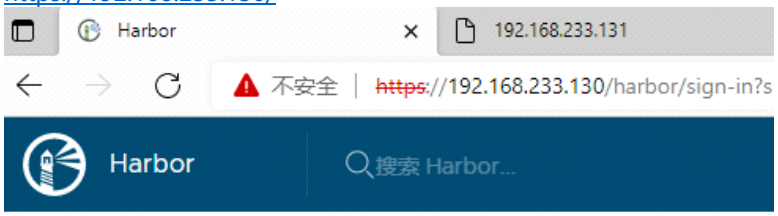
```
Loaded image: goharbor/harbor-jobservice:v1.10.10
Loaded image: goharbor/notary-server-photon:v1.10.10
Loaded image: goharbor/nginx-photon:v1.10.10
Loaded image: goharbor/clair-photon:v1.10.10


[Step 3]: preparing environment ...

[Step 4]: preparing harbor configs ...
prepare base dir is set to /data/packages/harbor
Clearing the configuration file: /config/registry/config.yml
Clearing the configuration file: /config/core/app.conf
Clearing the configuration file: /config/core/env
Clearing the configuration file: /config/nginx/nginx.conf
Clearing the configuration file: /config/db/env
Clearing the configuration file: /config/registryctl/env
Clearing the configuration file: /config/registryctl/config.yml
Clearing the configuration file: /config/jobservice/env
Clearing the configuration file: /config/jobservice/config.yml
Clearing the configuration file: /config/log/logrotate.conf
Clearing the configuration file: /config/log/rsyslog_docker.conf
Generated configuration file: /config/log/logrotate.conf
Generated configuration file: /config/log/rsyslog_docker.conf
Generated configuration file: /config/nginx/nginx.conf
Generated configuration file: /config/core/env
Generated configuration file: /config/core/app.conf
Generated configuration file: /config/registry/config.yml
Generated configuration file: /config/registryctl/env
Generated configuration file: /config/db/env
Generated configuration file: /config/jobservice/env
Generated configuration file: /config/jobservice/config.yml
loaded secret from file: /secret/keys/secretkey
Generated configuration file: /compose_location/docker-compose.yml
/usr/src/app/utils/configs.py:100: YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated, as the default Loader is unsaf
  configs = yaml.load(f)
/usr/src/app/utils/configs.py:90: YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated, as the default Loader is unsafe
  versions = yaml.load(f)
Clean up the input dir


Note: stopping existing Harbor instance ...
Stopping harbor-db     ... done
Stopping redis         ... done
Stopping harbor-portal ... done
Stopping registryctl   ... done
Stopping harbor-log    ... done
Removing harbor-db     ... done
Removing redis         ... done
Removing harbor-portal ... done
Removing registryctl   ... done
Removing harbor-log    ... done
Removing network harbor_harbor


[Step 5]: starting Harbor ...
Creating network "harbor_harbor" with the default driver
Creating harbor-log ... done
Creating harbor-portal     ... done
Creating harbor-db     ... done
Creating redis         ... done
Creating registry      ... done
Creating registryctl   ... done
Creating harbor-core   ... done
Creating harbor-jobservice ... done
Creating nginx             ... done
✔ ----Harbor has been installed and started successfully.----
root@master:/data/packages/harbor# cat install.sh
#!/bin/bash

set -e

DIR="$(cd "$(dirname "$0")" && pwd)"
```

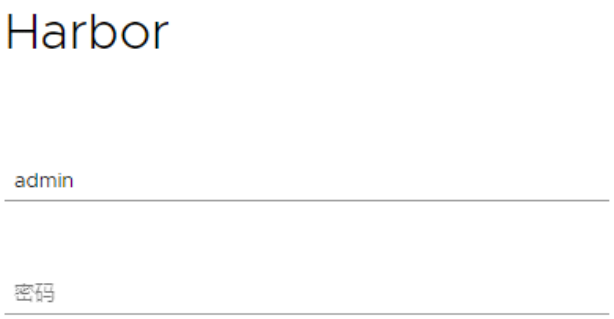
```
source $DIR/common.sh

set +o noglob

usage=$'Please set hostname and other necessary attributes in harbor.yml first. DO NOT use localhost or 127.0.0.1 for hostname, because
Please set --with-notary if needs enable Notary in Harbor, and set ui_url_protocol/ssl_cert/ssl_cert_key in harbor.yml bacause notary mu
Please set --with-clair if needs enable Clair in Harbor
Please set --with-chartmuseum if needs enable Chartmuseum in Harbor'
item=0

# notary is not enabled by default
with_notary=$false
# clair is not enabled by default
with_clair=$false
# chartmuseum is not enabled by default
with_chartmuseum=$false

while [ $# -gt 0 ]; do
      case $1 in
         --help)
         note "$usage"
         exit 0;;
         --with-notary)
         with_notary=true;;
         --with-clair)
         with_clair=true;;
         --with-chartmuseum)
         with_chartmuseum=true;;
         *)
         note "$usage"
         exit 1;;
      esac
      shift || true
done

workdir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
cd $workdir

h2 "[Step $item]: checking if docker is installed ..."; let item+=1
check_docker

h2 "[Step $item]: checking docker-compose is installed ..."; let item+=1
check_dockercompose

if [ -f harbor*.tar.gz ]
then
   h2 "[Step $item]: loading Harbor images ..."; let item+=1
   docker load -i ./harbor*.tar.gz
fi
echo ""

h2 "[Step $item]: preparing environment ...";  let item+=1
if [ -n "$host" ]
then
   sed "s/^hostname: .*/hostname: $host/g" -i ./harbor.yml
fi

h2 "[Step $item]: preparing harbor configs ...";  let item+=1
prepare_para=
if [ $with_notary ]
then
   prepare_para="${prepare_para} --with-notary"
fi
if [ $with_clair ]
then
   prepare_para="${prepare_para} --with-clair"
fi
if [ $with_chartmuseum ]
then
   prepare_para="${prepare_para} --with-chartmuseum"
fi
```

```
./prepare $prepare_para
echo ""

if [ -n "$(docker-compose ps -q)" ]
then
   note "stopping existing Harbor instance ..."
   docker-compose down -v
fi
echo ""

h2 "[Step $item]: starting Harbor ..."
docker-compose up -d

success $"----Harbor has been installed and started successfully.----"
```

部署完成之后即可访问以下地址打开web页面

https://192.168.233.130/





关于上传镜像

上传镜像必须要登录才可以

使用docker login 192.168.233.130/public登录，输入在harbor中public这个仓库中具有权限的用户名和密码

```
root@ubuntu:~/dockerfile# docker login 192.168.233.130/public
Username: user001
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@ubuntu:~/dockerfile# docker push 192.168.233.130/public/nginxnew:v1
The push refers to repository [192.168.233.130/public/nginxnew]
f665d09dc980: Pushed
762b147902c0: Pushed
235e04e3592a: Pushed
6173b6fa63db: Pushed
9a94c4a55fe4: Pushed
9a3a6af98e18: Pushed
7d0ebbe3f5d2: Pushed
v1: digest: sha256:8ac004214efeeabd5ffb9444bdbee441a51465fc1e3eeaebfd300ccca5fce43b size: 1777
root@ubuntu:~/dockerfile#
```