Private Docker registry Install on Ubuntu 14.04

（参考<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-private-docker-registry-on-ubuntu-14-04>）

centos参考（<https://www.dropbit.ch/private-docker-registry-with-nginx-on-centos-7/>）

Prerequisites

**Install docker and docker-compose and Setup registry with** **SSL and** **Authentication use nginx**

|  |
| --- |
| vim install-docker-registry.sh |
| #!/bin/bash  if [ $UID != 0 ];then 　echo "please run as root" exit  fi  host=$1  if [ "$host" == "" ]; then  　echo "please add host domain param" exit  fi  export APT\_INSTALL='sudo apt-get --assume-yes install'  export APT\_UPDATE='sudo apt-get --assume-yes update'  sudo chown $USER:$USER /etc/resolv.conf  sudo echo "nameserver 223.5.5.5" >> /etc/resolv.conf  sudo chown root:root /etc/resolv.conf  $APT\_INSTALL git  $APT\_INSTALL apt-transport-https ca-certificates  sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D  sudo echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" > /etc/apt/sources.list.d/docker.list  $APT\_UPDATE  sudo apt-get purge lxc-docker  apt-cache policy docker-engine  $APT\_UPDATE  $APT\_INSTALL linux-image-extra-$(uname -r) linux-image-extra-virtual  $APT\_UPDATE  $APT\_INSTALL docker-engine  sudo service docker start  sudo docker -v  sudo groupadd docker  sudo usermod -aG docker $USER  $APT\_INSTALL apache2-utils  #create docker-registry contents  mkdir /docker-registry  cd /docker-registry  mkdir data  mkdir nginx  #copy words below into docker-compose.yml  echo “nginx:  image: "nginx:1.9"  ports:  - 443:443  links:  - registry:registry  volumes:  - ./nginx/:/etc/nginx/conf.d  registry:  image: registry:2  ports:  - 127.0.0.1:5000:5000  environment:  REGISTRY\_STORAGE\_FILESYSTEM\_ROOTDIRECTORY: /data  volumes:  - ./data:/data” > docker-compose.yml  #save docker-compose.yml  #configure Nginx  echo “upstream docker-registry {  server registry:5000;  }  server {  listen 443;  server\_name $host;  # SSL  ssl on;  ssl\_certificate /etc/nginx/conf.d/domain.crt;  ssl\_certificate\_key /etc/nginx/conf.d/domain.key;  # disable any limits to avoid HTTP 413 for large image uploads  client\_max\_body\_size 0;  # required to avoid HTTP 411: see Issue #1486(https://github.com/docker/docker/issues/1486)  chunked\_transfer\_encoding on;  location /v2/ {  # Do not allow connections from docker 1.5 and earlier  # docker pre-1.6.0 did not properly set the user agent on ping, catch "Go \*" user agents  if ($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/registry.password;  add\_header 'Docker-Distribution-Api-Version' 'registry/2.0' 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 900;  }  }” > nginx/registry.conf  #Setting Up Authentication  cd /docker-registry/nginx  #htpasswd dependency  $APT\_INSTALL apache2-utils  #Create the first user as follows, replacing USERNAME with the username you want to use:  htpasswd -c registry.password  *vpclub*  *#*If you want to add more users in the future, just re-run the above command *#*without the -c option (the c is for create):  htpasswd registry.password USERNAME  #  cd /docker-registry/nginx  openssl genrsa -out devdockerCA.key 2048  openssl req -x509 -new -nodes -key devdockerCA.key -days 10000 -out devdockerCA.crt  openssl genrsa -out domain.key 2048  openssl req -new -key domain.key -out dev-docker-registry.com.csr  #Do not enter a challenge password.  openssl x509 -req -in dev-docker-registry.com.csr -CA devdockerCA.crt -CAkey devdockerCA.key -CAcreateserial -out domain.crt -days 10000  cd /docker-registry/nginx  mkdir /usr/local/share/ca-certificates/docker-dev-cert  cp devdockerCA.crt /usr/local/share/ca-certificates/docker-dev-cert  update-ca-certificates  service docker restart  **###Starting Docker Registry as a Service**  chown -R root: /docker-registry  **###copy words below into /etc/init/docker-registry.conf**  **echo “Description "Docker Registry"**  start on runlevel [2345]  stop on runlevel [016]  respawn  respawn limit 10 5  chdir /docker-registry  exec /usr/local/bin/docker-compose up**” > /etc/init/docker-registry.conf**  **# docker-registry service start**  service docker-registry start  #look at the current docker process  **docker ps** |

#授权

sudo chmod 755 install-docker-registry.sh

#执行安装即可

./install-docker-registry.sh

#配置客户端ssl链接

**#****Accessing Your Docker Registry from a Client Machine copy certificate**

sudo cat /docker-registry/nginx/devdockerCA.crt

-----BEGIN CERTIFICATE-----

MIID6zCCAtOgAwIBAgIJAKAbyZuhUQJiMA0GCSqGSIb3DQEBCwUAMIGLMQswCQYD

VQQGEwJaSDELMAkGA1UECAwCR0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENM

VUIxDzANBgNVBAsMBlZQQ0xVQjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIu

Y24xITAfBgkqhkiG9w0BCQEWEmNoZW4ud2VpQHZwY2x1Yi5jbjAeFw0xNjA3MTYw

ODI3NTRaFw00MzEyMDIwODI3NTRaMIGLMQswCQYDVQQGEwJaSDELMAkGA1UECAwC

R0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENMVUIxDzANBgNVBAsMBlZQQ0xV

QjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIuY24xITAfBgkqhkiG9w0BCQEW

EmNoZW4ud2VpQHZwY2x1Yi5jbjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC

ggEBAMPZRrO/Fc7PrY94uiiSl4dA4r5ko0ca5j9IbW/8wmlEwLWtMXd/+ERtsDr2

mMIkSuWxY6T5MJft2Fg4iogiaRZuS/6Vqub1jqd61A9LToCfYr2Z3jrQaC8otWI6

uN83MWcmJTgbjqONO2xwVcuWfncXIClvhWWSIYBO60kFaq6RZDFdSS1Qz6W2m+mB

U6c0B8ueUz3KE8CLGsnFVT6GXq82qsQSWM5FXeAjaJT1LlPXMk8JoQ0+JyGJYV+P

1ZcDPztD/afo9hvdJIgjN1gfdE+eEJrpnD+G3EL8sbiNKa4NuOyzEp2EnOvT30B5

1wW3VaXh2UMdvgYUUubBz3fVrTsCAwEAAaNQME4wHQYDVR0OBBYEFC88ycrUhiom

u425i3EWN+sw2HNJMB8GA1UdIwQYMBaAFC88ycrUhiomu425i3EWN+sw2HNJMAwG

A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBALV9NZ9nDdDpV3pcuRXCSzw9

39NJ7r1Merso3u91USOMokySU6nUsuRZ/A8tNg2s7HlRMMRvTqynDh0asT3rcK/C

m7d6Em+Benil6Esyb3D2zJFBAM1dsbwFAwpkbuxnQK2INJReAaOv6ijurkatz5ic

U7XDJxHgmI9QzBQRzKzPalzCFFC8nfwUHlZ8jluwRb77ctxxC3t+RXJatOmmCQ5q

KXPAQjwZMSgqN62hUbMhR98HQy/s/eo1DSiCQ6hRM/Cvx+aAqmvf/ujbd/BjbZ/c

X6Ep6ON2tOhY1YeEqLGy7qmCvRdz08Rq4PYWmlUYiXddEiA7BSi/8uZx1bQLO/k=

-----END CERTIFICATE-----

**＃配置ip　SAN　－－公网域名无需此步**

vim /etc/ssl/openssl.cnf

#add line after **[v3\_ca]**

subjectAltName = IP:hub.docker.vpclub.cn

Client Operation

**firstly:**

(如果是内网自己配的域名则需要配置)

add host config

window path [c:/window/system32/drivers/etc/hosts](file:///c:/window/system32/drivers/etc/hosts)

Linux path /etc/hosts

120.76.43.11 hub.docker.vpclub.cn

1:create dir

sudo mkdir /usr/local/share/ca-certificates/docker-dev-cert

2:create file and paste the certificate

sudo rm /usr/local/share/ca-certificates/docker-dev-cert/devdockerCA.crt

sudo nano /usr/local/share/ca-certificates/docker-dev-cert/devdockerCA.crt

3. update the certificates

sudo update-ca-certificates

4.restart docker-

sudo service docker restart

5.login

sudo docker login https://hub.docker.vpclub.cn

6. create image and push

docker tag hello-world hub.docker.vpclub.cn/test-image

sudo docker push hub.docker.vpclub.cn/test-image

＃ｒｅｇｉｓｔｒｙ日志查看

sudo tail -f /var/log/upstart/docker-registry.log