Private Docker registry Install on Ubuntu 14.04

Prerequisites

**Install docker**

#Update your apt sources

sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates

sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

#Open the /etc/apt/sources.list.d/docker.list file in your favorite editor.

#If the file doesn’t exist, create it.

vim /etc/apt/sources.list.d/docker.list

deb https://apt.dockerproject.org/repo ubuntu-trusty main

#Save and close the /etc/apt/sources.list.d/docker.list file.

#Update the APT package index.

sudo apt-get update

apt-get update

apt-get install apt-transport-https ca-certificates

apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

#Purge the old repo if it exists.

sudo apt-get purge lxc-docker

#Verify that APT is pulling from the right repository.

sudo apt-cache policy docker-engine

#Install the recommended package.

sudo apt-get install linux-image-extra-$(uname -r)

#Install both the required and optional packages.

sudo apt-get install linux-image-generic-lts-trusty

sudo reboot

sudo apt-get update

sudo apt-get install docker-engine

sudo service docker start

sudo docker run hello-world

#installed docker

**Install docker-compose**

sudo apt-get install python-pip

pip install docker-compose

Setup registry with SSL and Authentication use nginx

#create docker-registry contents

mkdir /docker-registry

cd /docker-registry

mkdir data

mkdir nginx

# please install nano first

nano docker-compose.yml

#copy words below into docker-compose.yml

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

#save docker-compose.yml

#configure Nginx

nano /docker-registry/nginx/registry.conf

#copy words below into /docker-registry/nginx/registry.conf

upstream docker-registry {

server registry:5000;

}

server {

listen 443;

server\_name hostdomain.com;

# 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;

}

}

#Setting Up Authentication

cd /docker-registry/nginx

#htpasswd dependency

sudo apt-get 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

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

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

sudo update-ca-certificates

sudo service docker restart

**###Starting Docker Registry as a Service**

sudo chown -R root: /docker-registry

sudo nano /etc/init/docker-registry.conf

**###copy words below into /etc/init/docker-registry.conf**

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

# docker-registry service start

sudo service docker-registry start

#look at the current docker process

sudo docker ps

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

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

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

MIID5zCCAs+gAwIBAgIJAP0R3XSsfmfpMA0GCSqGSIb3DQEBCwUAMIGJMQswCQYD

VQQGEwJDTjELMAkGA1UECAwCR0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENM

VUIxDzANBgNVBAsMBlZQQ0xVQjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIu

Y24xHzAdBgkqhkiG9w0BCQEWEGdpdGxhYkB2cGNsdWIuY24wHhcNMTYwNDE4MDYx

NDI5WhcNNDMwOTA0MDYxNDI5WjCBiTELMAkGA1UEBhMCQ04xCzAJBgNVBAgMAkdE

MQswCQYDVQQHDAJTWjEPMA0GA1UECgwGVlBDTFVCMQ8wDQYDVQQLDAZWUENMVUIx

HTAbBgNVBAMMFGh1Yi5kb2NrZXIudnBjbHViLmNuMR8wHQYJKoZIhvcNAQkBFhBn

aXRsYWJAdnBjbHViLmNuMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA

w5ZIwxA5SD2/cPiKcy4knRTMPgv2m4boPl4tARdB9D1Uh2vOQ08t+UTGQU32b8ZY

9XxeUDOkdG5E2kBEAt34Q/jdI50eCLIqz+ihrCwmHLeignpPcUB2zpE8+yfDreBr

gcP+ZWmZODNbUCUu3RXn1LJMs2kf4X/2RTX54lK4/+QutAtBOK31Hhkw9Va9orYH

519VsYl6NbK0MlLDk/CAOdaZ+BI8ynq0IjnWt0De+PAGMQ4NJz+l14ixUmE7HGYi

pgQYSaDeMEiSsw0HZPqyOyis/6EVQhb2zmoXiWz24I5eanN0S4xYR4VAXCnzgKr8

kFhgTtJyQU7P76swpaQfvQIDAQABo1AwTjAdBgNVHQ4EFgQUdGRf/tCVr1ftAoem

yVWn4XyTESowHwYDVR0jBBgwFoAUdGRf/tCVr1ftAoemyVWn4XyTESowDAYDVR0T

BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAIupyWrLtQAXHJ84NvjClL8eR4xYv

XSQcb1vWwyZyB8TVcwQKVKKqOaJKM6uyb+VrxdNW6fVurOYPgM8ify9+2JhvguPu

H0xiYc2XsAcCVtfFRVFyx6h+eYZHxv85L3ubHMyuPcvL1kU7BLpsaw2GoF3pcTl0

qOSi3oZUFrI8odsLp/v63VdvUf7r8jq2hX/wEQ9wFBUImlD6PReQ3DpD/+kqV+jq

WRz8aTkyHjcyrCR01ieLXGMvql05isyPOxu0NKC4PnAFmxpZKH4ua3yUd9qntjfd

xZE5c5e3LHWw6Q99M4ay+cpPXKEp/sb4ZQktPevqhhOpSYNehBzvO755aA==

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

Client Operation

**firstly:**

add host config

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

Linux path /etc/hosts

172.16.5.30 vpclubdockerregistry.com

1:create dir

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

2:create file and paste the certificate

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://vpclubdockerregistry.com

6. create image and push

docker tag test-image vpclubdockerregistry.com/test-image

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

MIID5zCCAs+gAwIBAgIJAJFNBpVCO7u2MA0GCSqGSIb3DQEBCwUAMIGJMQswCQYD

VQQGEwJDTjELMAkGA1UECAwCR0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENM

VUIxDzANBgNVBAsMBlZQQ0xVQjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIu

Y24xHzAdBgkqhkiG9w0BCQEWEGdpdGxhYkB2cGNsdWIuY24wHhcNMTYwMzI5MDg1

NzA1WhcNNDMwODE1MDg1NzA1WjCBiTELMAkGA1UEBhMCQ04xCzAJBgNVBAgMAkdE

MQswCQYDVQQHDAJTWjEPMA0GA1UECgwGVlBDTFVCMQ8wDQYDVQQLDAZWUENMVUIx

HTAbBgNVBAMMFGh1Yi5kb2NrZXIudnBjbHViLmNuMR8wHQYJKoZIhvcNAQkBFhBn

aXRsYWJAdnBjbHViLmNuMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA

uq7H+YUfdyPYfmRzz6p73E1hL7Hsh1NLOQxZ6vKeowH3htlzKPaH3+YUXx2mprke

FFZcEkm6z3zzqAt2ZyJhs4CP856nsAwYTb07V4wevI83nnoAHKDWi0agNeKxwcjv

stdjbm66tqQrI0Xdnxlh1/z6iOq5mCmdwUXQfI5zoEu7Nm9TZaLFybPEGsBsPkZ4

GRVDT7Xj361mHSWvAY/pA4yNr5KIY3AZ+vKs9XjHthcF9oiDjayBrWe4qPO3uff2

TJmNMTg1GFqi9kgJROmFujsvuYzlp0gGvGq6jWuksBnrdsXnrUE7jFRv6LJp7aEb

NGCtNabw1B7f+i+Dc0gtZQIDAQABo1AwTjAdBgNVHQ4EFgQU30Kx1E+XoppCDLXv

EhMoCZ30Ty0wHwYDVR0jBBgwFoAU30Kx1E+XoppCDLXvEhMoCZ30Ty0wDAYDVR0T

BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAQBbA+VPy4pBAczVUlyOOSsqEscni

FREoofK151KIpUSEuOz6wdjUymC8H5CRSg1ScUuD2W3ixgTf5Vd8TgMBgE1WcchE

BbpDxd6IoNR8IKITgqYThjYD4bNSBIRNevZdhe5ri1f4OSapmI1nFW4sLpQlrBQ+

xYPFoJgPHo0P+jqMC5KlFlFztAlt7AKLFa2h6bjri4Vlco5LBEydJ1N8PLlf/LrT

+Y5Cgzv7ZVZGwCIXJhlmNLy47QOhvJVGkDS2SfMuOvDqduvZuMBqHVFXt8fGFxcX

8cqC87iqFYoAM/GuAmGsJ6vfM2Lg5Oc1dtGwmFGHTMBWINk1000T7QPjFA==

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