Setting up harbor registry

Spin up VM using vagrantfile provided in the devops repo.

Login into the machine and continue installation, instructions can also be found at harbors official documentation.

Steps:

Download harbor offline installer binary.
wget
https://github.com/goharbor/harbor/releases/download/v2.6.1/harbor-offline-installer-v2.6.1.tgz

Extract the binary tarball.

tar xzvf harbor-offline-installer-v2.6.1.tgz

Change directory to /opt
cd /opt

Generate a Certificate Authority Certificate

Generate a CA certificate private key.

openssl genrsa -out ca.key 4096

Generate the CA certificate.

openssl req -x509 -new -nodes -sha512 -days 3650 -subj "/C=CN/ST=Kathmandu/L=Kathmandu/O=example/OU=Personal/CN=harbor.registry.local" -key ca.key -out ca.crt

Generate a Server Certificate

Generate a private key.

openssl genrsa -out harbor.registry.local.key 4096

Generate a certificate signing request (CSR).

openssl req -sha512 -new -subj "/C=CN/ST=Kathmandu/L=Kathmandu/O=example/OU=Personal/CN=harbor.registry.local" -key harbor.registry.local.key -out harbor.registry.local.csr

Generate an x509 v3 extension file.

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

[alt_names]
DNS.1=harbor.registry.local
DNS.2=harbor.registry
DNS.3=harbor.registry.local
EOF

Use the v3.ext file to generate a certificate for your Harbor host.

openssl x509 -req -sha512 -days 3650 -extfile v3.ext -CA ca.crt -CAkey ca.key -CAcreateserial -in harbor.registry.local.csr -out harbor.registry.local.crt

Provide the Certificates to Harbor and Docker

Create directory cert in harbor host.

mkdir -p /data/cert/

Copy the server certificate and key into the cert folder on your Harbor host.

cp harbor.registry.local.crt /data/cert/
cp harbor.registry.local.key /data/cert/

Convert yourdomain.com.crt to yourdomain.com.cert, for use by Docker.

openssl x509 -inform PEM -in harbor.registry.local.crt -out harbor.registry.local.cert

Create certs.d directory and the registry directory.

ls /etc/docker/certs.d/harbor.registry.local/

mkdir -p /etc/docker/certs.d/harbor.registry.local/

Copy the server certificate, key and CA files into the Docker certificates folder on the Harbor host.

cp harbor.registry.local.cert /etc/docker/certs.d/harbor.registry.local/

cp harbor.registry.local.key /etc/docker/certs.d/harbor.registry.local/

cp ca.crt /etc/docker/certs.d/harbor.registry.local/

Restart Docker Engine.

systemctl restart docker

Deploy Harbor

Change into the directory where the harbor binary is extracted.

cd /home/vagrant/harbor

Copy harbor.yml.tmpl to harbor.yml

cp harbor.yml.tmpl harbor.yml

Modify followings in harbor.yml file
Modify the hostname, as per our above config hostname should be

hostname: harbor.registry.local

Modify certs path, as per our above config the path should be following.

certificate: /data/cert/harbor.registry.local.crt
private_key: /data/cert/harbor.registry.local.key

Harbor installation with Trivy

./install.sh --with-trivy

Since we are doing with self signed certificates and we do not have public dns, modify /etc/hosts of client machine for point the ip dns

Browse url

https://harbor.registry.local/

Username: admin
Password: Harbor12345

To perform push pull operation for images

Login to the remote registry via docker client

docker login harbor.registry.local

In the harbor portal create a new project let's say devops

In the client terminal screen
Pull ubuntu image from dockerhub.

docker pull ubuntu

docker images

Re tag the ubuntu image to support our local registry

docker tag ubuntu harbor.registry.local/devops/ubuntutest:20.0

Push retagged image to our local registry.
docker push harbor.registry.local/devops/ubuntutest:20.04

Remove image from local repo.
docker rmi harbor.registry.local/devops/ubuntutest:20.04

Pull image from our local registry
docker pull harbor.registry.local/devops/ubuntutest:20.04