Your company has recently decided to use Docker to run containers in production. They have built some Docker images to run their own proprietary software and need a place to store and manage these images. You have been asked to build a secure, private Docker registry for use by the company. In order to verify that everything works, you have also been asked to configure a Docker workstation server to push to and pull from the registry.

To complete this lab, ensure that the following requirements are met for the registry:

1. A private Docker registry is running on the Docker registry server using version 2 of the registry image.

2. The container name for the registry should be registry.

3. The registry should always automatically restart if it stops or the Docker daemon or server restarts.

4. The registry should require authentication. Set up an initial account with the username docker and the password d0ck3rrU73z.

5. The registry should use TLS with a self-signed certificate.

6. The registry should listen on port 443.

Set up the Docker workstation server to meet the following requirements:

1. Docker is logged in to the private registry.

2. Docker is configured to accept the self-signed cert. Do not turn off certificate verification using the insecure-registries setting.

3. To confirm that everything is working, push a test image called ip-10-0-1-101:443/test-image:1 to the private registry. You can pull any image from Docker hub and tag it with ip-10-0-1-101:443/test-image:1 as a test.

4. Delete the test image locally and pull it from the registry.

[NOTE: Write the series of commands to achieve above in this file below the question scenario with documentation]

Good luck!

mkdir ~/registry

cd ~/registry

mkdir auth

docker run --entrypoint htpasswd registry:2 -Bbn docker d0ck3rrU73z > auth/htpasswd

openssl req -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key -x509 -days 365 -out certs/domain.crt

docker run -d -p 443:443 --restart=always --name registry -v /root/registry/certs:/certs -v /rot/registry/auth:/auth -e REGISTRY\_HTTP\_ADDR=0.0.0.0:443 -e REGISTRY\_HTTP\_TLS\_CERTIFICATE=/certs/domain.crt -e REGISTRY\_HTTP\_TLS\_KEY=/certs/domain.key -e REGISTRY\_AUTH=htpasswd -e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" -e REGISTRY\_AUTH\_HTPASSWD\_PATH=/auth/htpasswd registry:2

mkdir auth

docker run --entrypoint htpasswd registry:2 -Bbn docker d0ck3rrU73z > auth/htpasswd

mkdir certs

openssl req -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key -x509 -days 365 -out certs/domain.crt

docker secret create domain.crt certs/domain.crt

docker secret create domain.key certs/domain.key

docker node ls

docker node update --label-add registry=true ip- 18.219.63.100

vi /etc/ssl/openssl.cnf

vim /etc/ssl/openssl.cnf

---------------

[ v3\_ca ]

subjectAltName = IP: 18.219.63.100

------------------------------

docker service create \

--name registry \

--secret domain.crt \

--secret domain.key \

--constraint 'node.labels.registry==true' \

--mount type=bind,src=/mnt/registry,dst=/var/lib/registry \

-e REGISTRY\_HTTP\_ADDR=0.0.0.0:443 \

-e REGISTRY\_HTTP\_TLS\_CERTIFICATE=/run/secrets/domain.crt \

-e REGISTRY\_HTTP\_TLS\_KEY=/run/secrets/domain.key \

-e REGISTRY\_AUTH=htpasswd \

-e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" \

-e REGISTRY\_AUTH\_HTPASSWD\_PATH=/auth/htpasswd \

--publish published=443,target=443 \

--replicas 1 \

registry:2

docker login -udocker -pd0ck3rrU73z 18.219.63.100:443

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

Error response from daemon: Get https://172.31.1.195:443/v2/: x509: certificate signed by unknown authority

docker login -udocker -pd0ck3rrU73z 172.31.29.245:443

docker pull ubuntu:18.04

docker tag ubuntu:16.04 172.17.0.76:443/test-image:1

docker push 172.17.0.76:443/test-image:1

docker image remove ubuntu:18.04

docker image remove 172.17.0.76:443/test-image:1

docker pull 172.17.0.76:443/test-image:1

docker logout