https://www.exoscale.com/syslog/setup-private-docker-registry/

https://docs.docker.com/registry/insecure/

https://www.exoscale.com/syslog/setup-private-docker-registry/

-=====================================================================================================================================================

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.

docker run -d -p 5000:5000 --name registry registry:2

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.

docker run -d -p 5000:5000 --restart=always --name registry registry:2

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

$ mkdir auth

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

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

docker run -d -p 5000:5000 --restart=always --name registry

-v "$(pwd)"/auth:/auth e "REGISTRY\_AUTH=htpasswd" -e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" e REGISTRY\_AUTH\_HTPASSWD\_PATH=/auth/htpasswd -v "$(pwd)"/certs:/certs -e REGISTRY\_HTTP\_TLS\_CERTIFICATE=/certs/domain.crt

-e REGISTRY\_HTTP\_TLS\_KEY=/certs/domain.key registry:26. The registry should listen on port 443.

docker push host.com:443/alpine

or docker push registry:8443/username/private\_docker\_registry:2.0

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

1. Docker is logged in to the private registry.

$ docker login myregistrydomain.com:5000

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

Use self-signed certificates

Warning: Using this along with basic authentication requires to also trust the certificate into the OS cert store for some versions of docker (see below)

This is more secure than the insecure registry solution.

Generate your own certificate:

$ mkdir -p certs

$ openssl req \

-newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key \

-x509 -days 365 -out certs/domain.crt

Be sure to use the name myregistrydomain.com as a CN.

Use the result to start your registry with TLS enabled.

Instruct every Docker daemon to trust that certificate. The way to do this depends on your OS.

Linux: Copy the domain.crt file to /etc/docker/certs.d/myregistrydomain.com:5000/ca.crt on every Docker host. You do not need to restart Docker.

THEN

$ cp certs/domain.crt /usr/local/share/ca-certificates/myregistrydomain.com.crt update-ca-certificates

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.

--->> docker push ip-10-0-1-101:443/test-image:1

docker pull ubuntu:latest

docker tag test ip-10-0-1-101:443/test-image:1

$docker tag rhel-httpd registry-host:5000/myadmin/rhel-httpd

$ docker push registry-host:5000/myadmin/rhel-httpd

or

$ docker tag hello-world 159.100.243.157:5000/hello-world # Replace with your IP/domain

$ docker push 159.100.243.157:5000/hello-world

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

$ docker pull registry.local:5000/testing/test-image

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

Good luck!