ATTACK DEFENSE

by PentesterAcademy

Name	Securing Private Docker Registry
URL	https://internaltests-286313.df.r.appspot.com/challengedetails?cid=302
Type	Container Security : Security Private Docker Registry

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

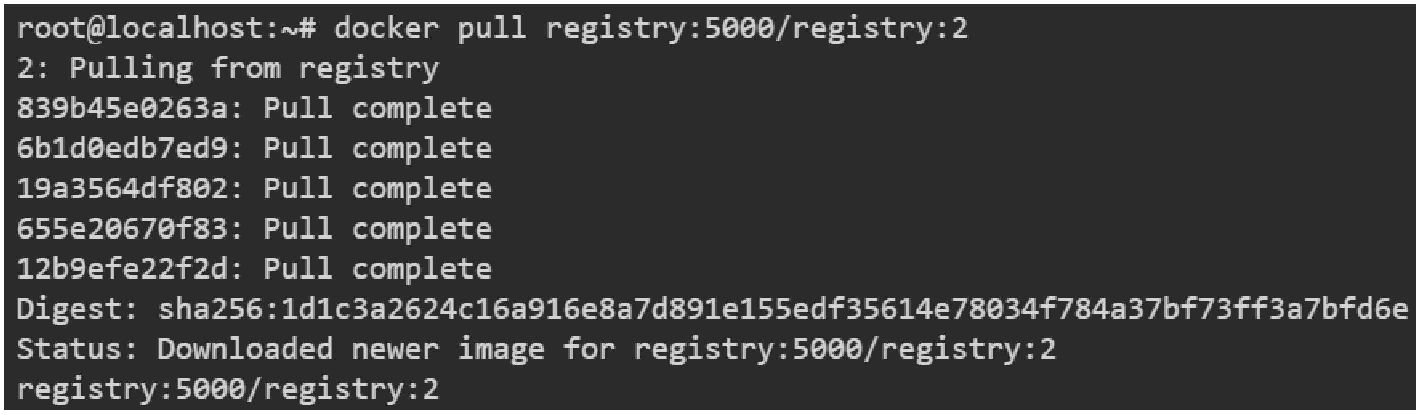
Objective: Learn how to deploy a private Docker registry, enable TLS support and basic authentication on it!

Solution:

Private Docker Registry

Step 1: Pull the registry image from the private Docker registry.

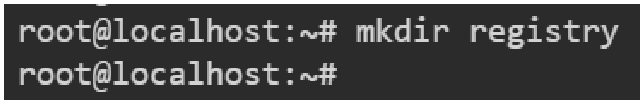
Command: docker pull registry:5000/registry:2



```
root@localhost:~# docker pull registry:5000/registry:2
2: Pulling from registry
839b45e0263a: Pull complete
6b1d0edb7ed9: Pull complete
19a3564df802: Pull complete
655e20670f83: Pull complete
12b9efe22f2d: Pull complete
Digest: sha256:1d1c3a2624c16a916e8a7d891e155edf35614e78034f784a37bf73ff3a7bfd6e
Status: Downloaded newer image for registry:5000/registry:2
registry:5000/registry:2
```

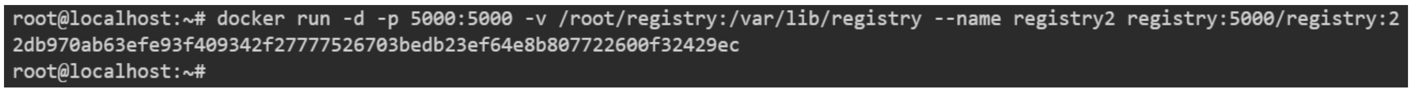
Step 2: Create a directory “registry” in the home directory of the root user (i.e. /root/registry). This directory will be mounted on the registry container.

Command: mkdir registry


```
root@localhost:~# mkdir registry
root@localhost:~#
```

Run this registry container

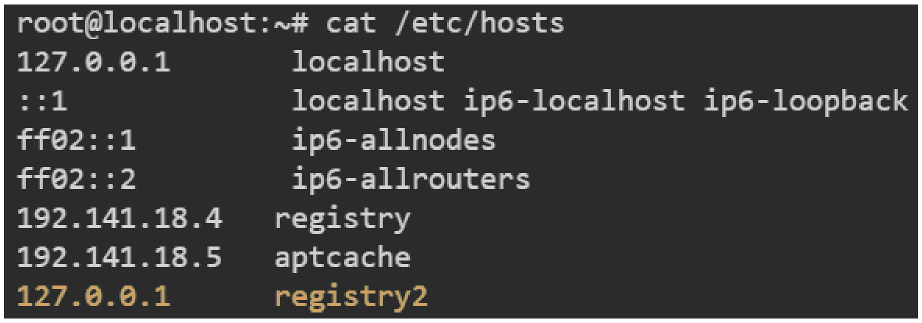
Command: docker run -d -p 5000:5000 -v /root/registry:/var/lib/registry --name registry2 registry:5000/registry:2


```
root@localhost:~# docker run -d -p 5000:5000 -v /root/registry:/var/lib/registry --name registry2 registry:5000/registry:2
2db970ab63efe93f409342f27777526703bedb23ef64e8b807722600f32429ec
root@localhost:~#
```

The port 5000 on the Docker host machine is now being used by the registry.

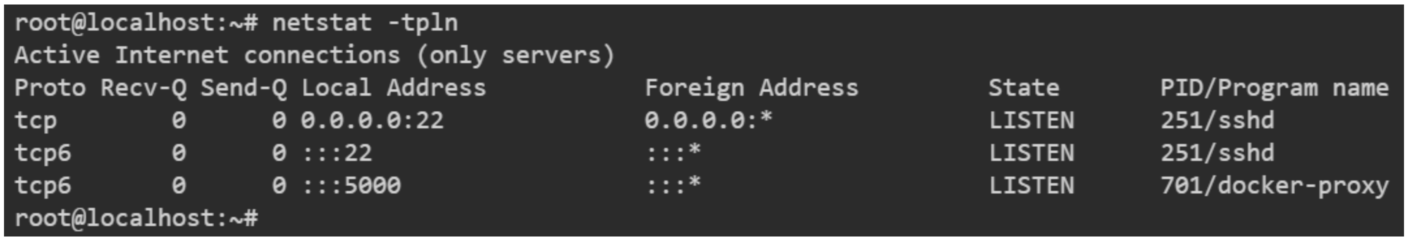
Step 3: Add an entry for registry2 in the /etc/hosts file.

Command: echo "127.0.0.1   registry2" >> /etc/hosts


```
root@localhost:~# cat /etc/hosts
127.0.0.1       localhost
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.141.18.4   registry
192.141.18.5   aptcache
127.0.0.1      registry2
```

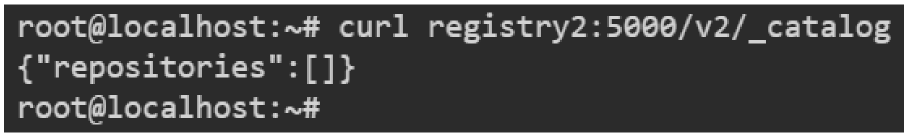
Step 4: ZCheck if the port 5000 of the local machine is open for listening.

Command: netstat -tpln


```
root@localhost:~# netstat -tpln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      251/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      251/sshd
tcp6       0      0 :::5000                 :::*                    LISTEN      701/docker-proxy
root@localhost:~#
```

Step 5: Check the images present on the registry using curl.

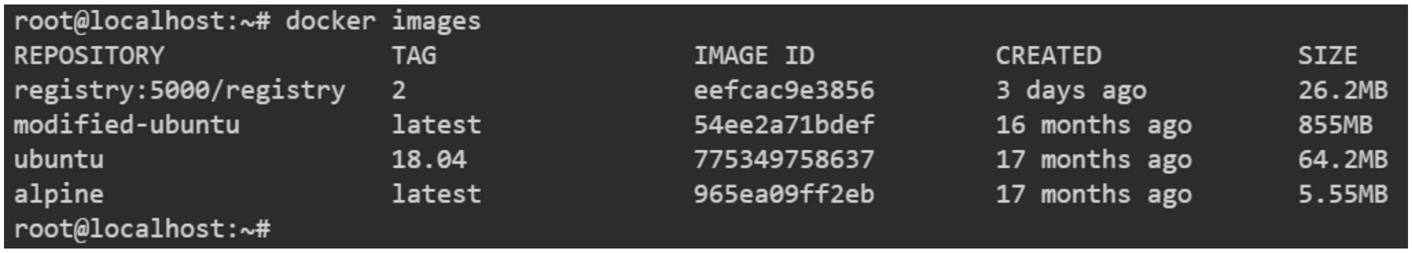
Command: curl registry2:5000/v2/_catalog


```
root@localhost:~# curl registry2:5000/v2/_catalog
{"repositories":[]}
root@localhost:~#
```

There are no images on this registry.

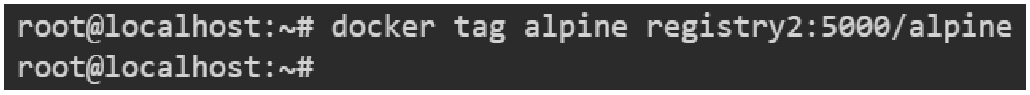
Step 7: Check the Docker images present on the Docker host.

Command: docker images

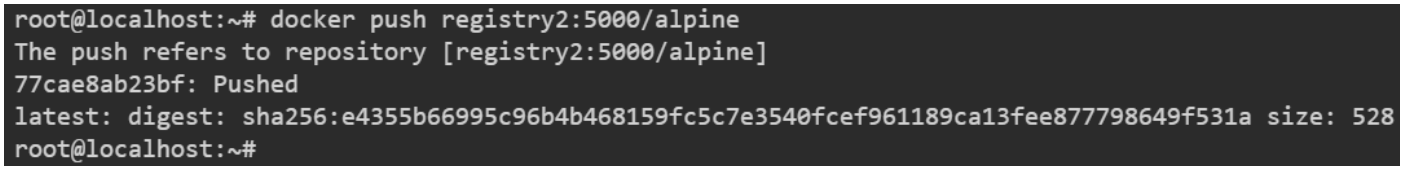

```
root@localhost:~# docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
registry:5000/registry    2                   eefcac9e3856        3 days ago          26.2MB
modified-ubuntu           latest              54ee2a71bdef        16 months ago       855MB
ubuntu                    18.04               775349758637        17 months ago       64.2MB
alpine                    latest              965ea09ff2eb        17 months ago       5.55MB
root@localhost:~#
```

Step 8: Tag alpine image on the registry2 address and push it.

Command: docker tag alpine registry2:5000/alpine

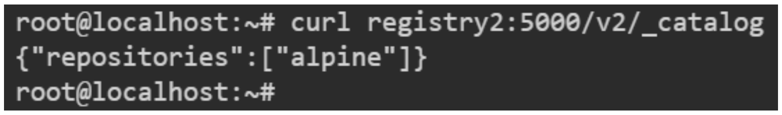

```
root@localhost:~# docker tag alpine registry2:5000/alpine
root@localhost:~#
```

Command: docker push registry2:5000/alpine


```
root@localhost:~# docker push registry2:5000/alpine
The push refers to repository [registry2:5000/alpine]
77cae8ab23bf: Pushed
latest: digest: sha256:e4355b66995c96b4b468159fc5c7e3540fcef961189ca13fee877798649f531a size: 528
root@localhost:~#
```

Step 9: Check the images present on the registry using curl.

Command: curl registry2:5000/v2/_catalog

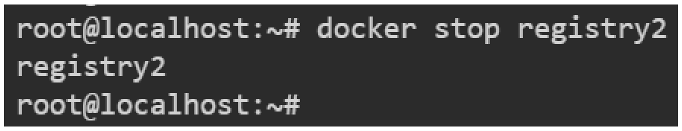

```
root@localhost:~# curl registry2:5000/v2/_catalog
{"repositories":["alpine"]}
root@localhost:~#
```

The alpine image is present on this registry.

Enabling TLS Support

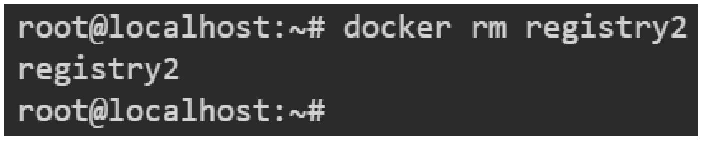
Step 1: Stop the already running registry container.

Command: docker stop registry2


```
root@localhost:~# docker stop registry2
registry2
root@localhost:~#
```

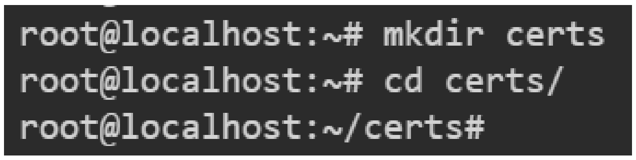
Also, remove the container.

Command: docker rm registry2


```
root@localhost:~# docker rm registry2
registry2
root@localhost:~#
```

Step 2: Create a “certs” directory in the home directory of the root user (i.e. /root/certs) and generate all certificates in that directory.

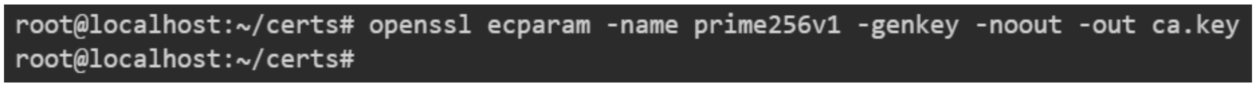
Commands:
mkdir certs
cd certs


```
root@localhost:~# mkdir certs
root@localhost:~# cd certs/
root@localhost:~/certs#
```

Generating CA Certificate

Step A: Generate CA key

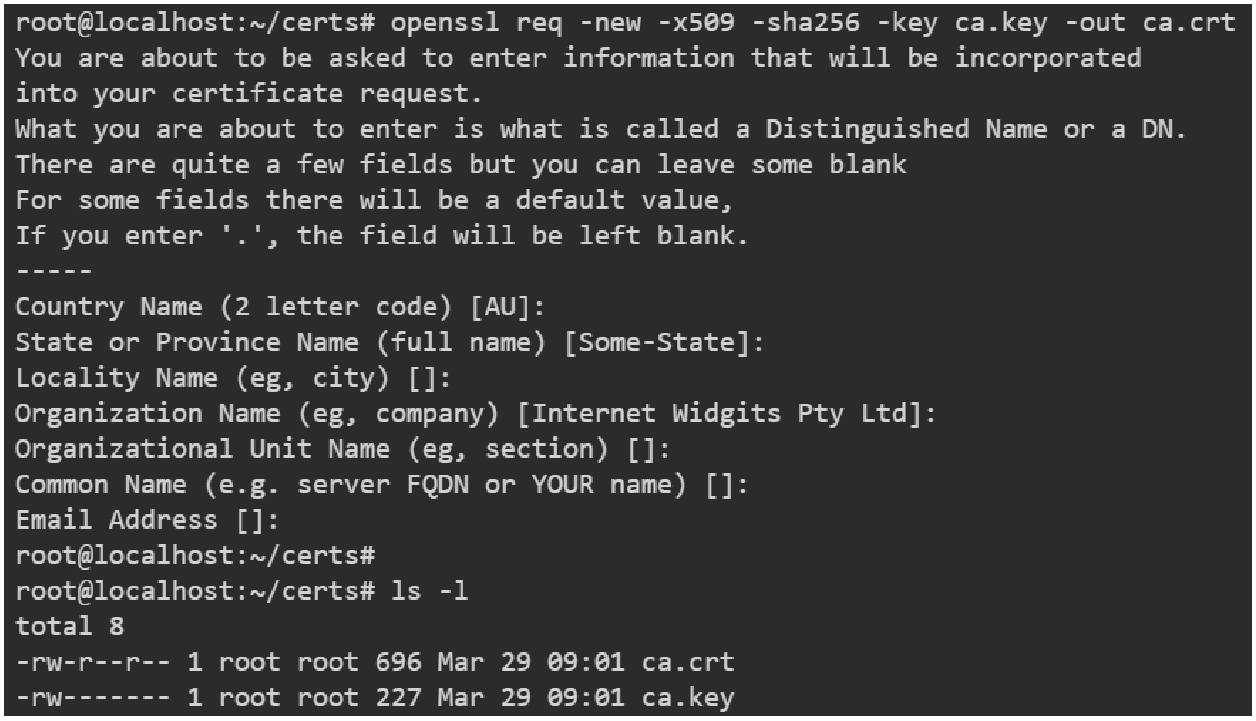
Command: openssl ecparam -name prime256v1 -genkey -noout -out ca.key


```
root@localhost:~/certs# openssl ecparam -name prime256v1 -genkey -noout -out ca.key
root@localhost:~/certs#
```

Step B: Generate CA certificate corresponding to the generated key

Command: openssl req -new -x509 -sha256 -key ca.key -out ca.crt

All the fields can be left blank in this exercise. However, it is suggested (as well as required in some cases) to fill appropriate information.

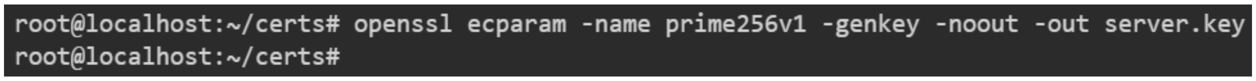


```
root@localhost:~/certs# openssl req -new -x509 -sha256 -key ca.key -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
root@localhost:~/certs#
root@localhost:~/certs# ls -l
total 8
-rw-r--r-- 1 root root 696 Mar 29 09:01 ca.crt
-rw------- 1 root root 227 Mar 29 09:01 ca.key
```

Generating Server Certificate

Step A: Generate Server key

Command: openssl ecparam -name prime256v1 -genkey -noout -out server.key

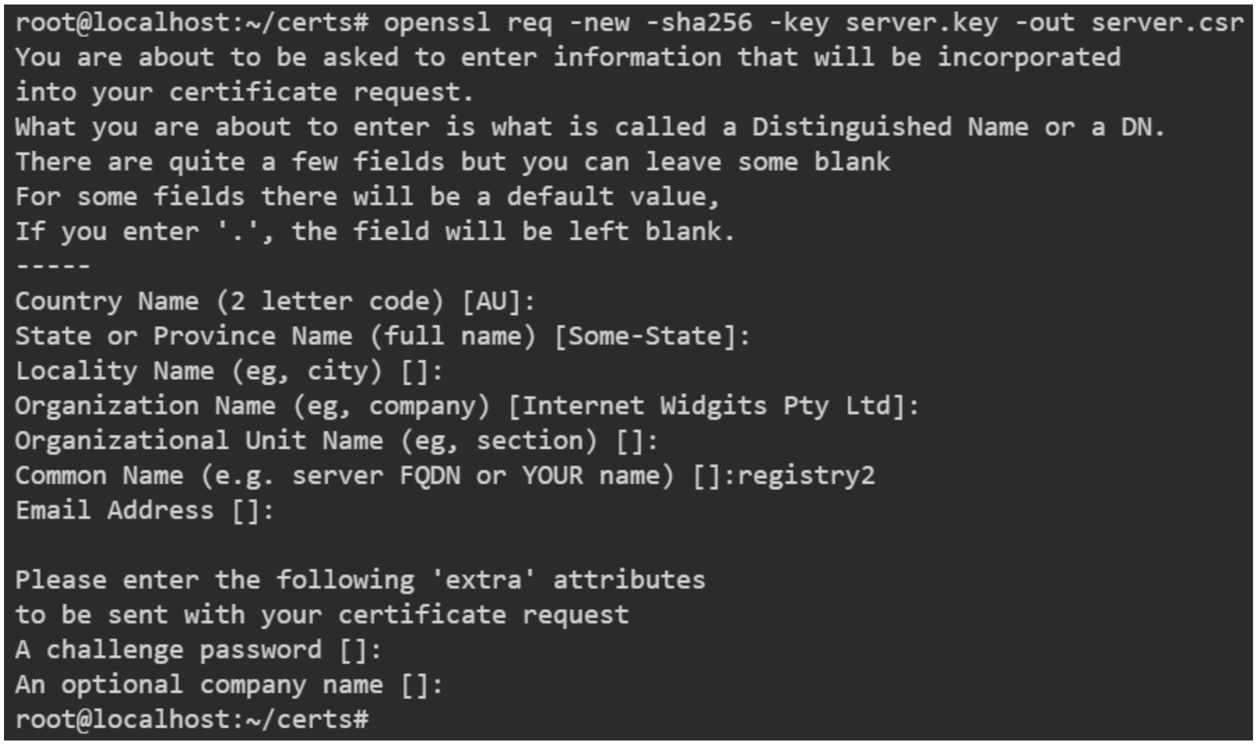


```
root@localhost:~/certs# openssl ecparam -name prime256v1 -genkey -noout -out server.key
root@localhost:~/certs#
```

Step B: Create a server certificate signing request (CSR)

Command: openssl req -new -sha256 -key server.key -out server.csr

Note: Please remember to put **registry2** in the Common Name (FQDN) field of the CSR. Otherwise the TLS connection will fail.

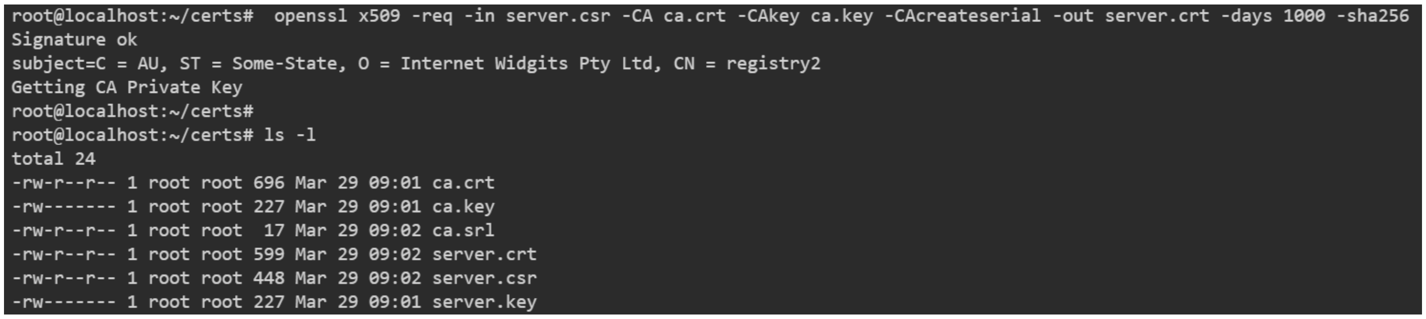


```
root@localhost:~/certs# openssl req -new -sha256 -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:registry2
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@localhost:~/certs#
```

Step C: Create a server certificate by using CA certificate/key and server certificate signing request (CSR).

Command: openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 1000 -sha256

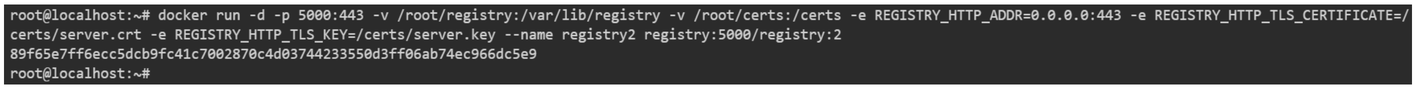


```
root@localhost:~/certs#  openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 1000 -sha256
Signature ok
subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = registry2
Getting CA Private Key
root@localhost:~/certs#
root@localhost:~/certs# ls -l
total 24
-rw-r--r-- 1 root root 696 Mar 29 09:01 ca.crt
-rw------- 1 root root 227 Mar 29 09:01 ca.key
-rw-r--r-- 1 root root  17 Mar 29 09:02 ca.srl
-rw-r--r-- 1 root root 599 Mar 29 09:02 server.crt
-rw-r--r-- 1 root root 448 Mar 29 09:02 server.csr
-rw------- 1 root root 227 Mar 29 09:01 server.key
```

Step 3: Run this registry container

Command: docker run -d -p 5000:443 -v /root/registry:/var/lib/registry -v /root/certs:/certs -e REGISTRY_HTTP_ADDR=0.0.0.0:443 -e

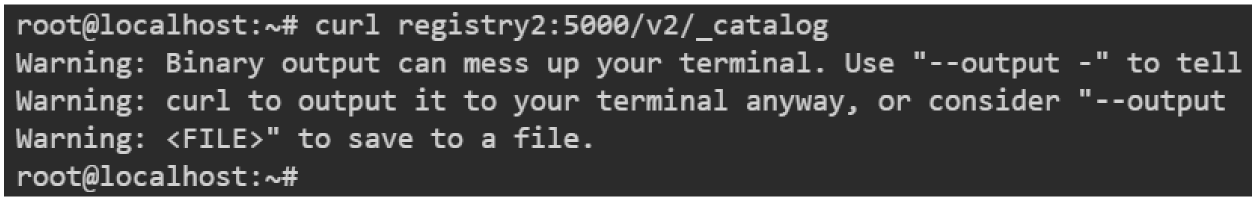
REGISTRY_HTTP_TLS_CERTIFICATE=/certs/server.crt -e
REGISTRY_HTTP_TLS_KEY=/certs/server.key --name registry2 registry:5000/registry:2



```
root@localhost:~# docker run -d -p 5000:443 -v /root/registry:/var/lib/registry -v /root/certs:/certs -e REGISTRY_HTTP_ADDR=0.0.0.0:443 -e REGISTRY_HTTP_TLS_CERTIFICATE=/
certs/server.crt -e REGISTRY_HTTP_TLS_KEY=/certs/server.key --name registry2 registry:5000/registry:2
89f65e7ff6ecc5dcb9fc41c7002870c4d03744233550d3ff06ab74ec966dc5e9
root@localhost:~#
```

Step 4: Check the images present on the registry using curl.

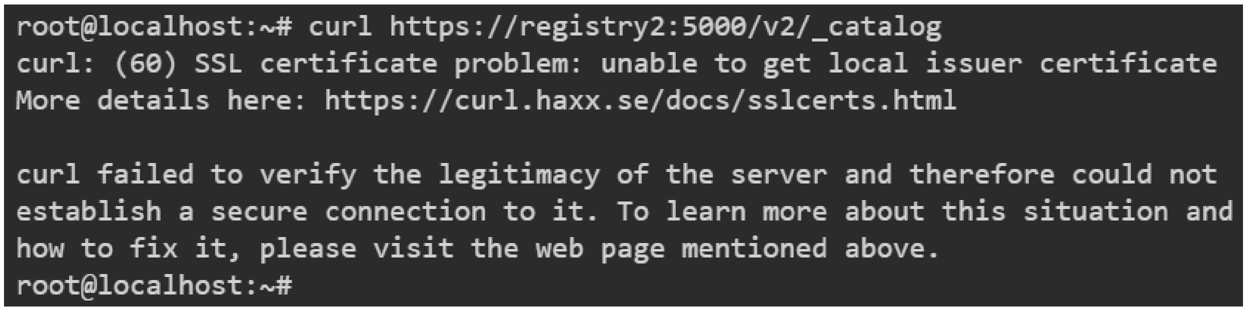
Command: curl registry2:5000/v2/_catalog



```
root@localhost:~# curl registry2:5000/v2/_catalog
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
root@localhost:~#
```

TLS is enabled on this registry, so use https in front of the URL

Command: curl https://registry2:5000/v2/_catalog

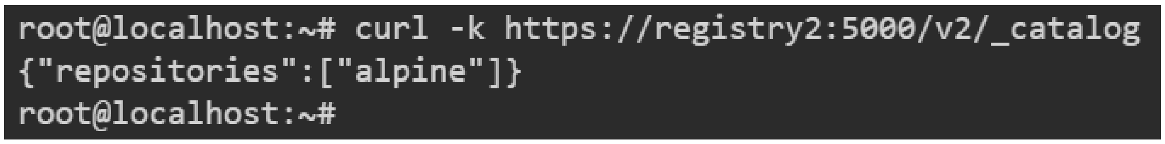


```
root@localhost:~# curl https://registry2:5000/v2/_catalog
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
root@localhost:~#
```

Curl is not able to validate the certificate of the registry. Skip the certificate validation check.

Command: curl -k https://registry2:5000/v2/_catalog

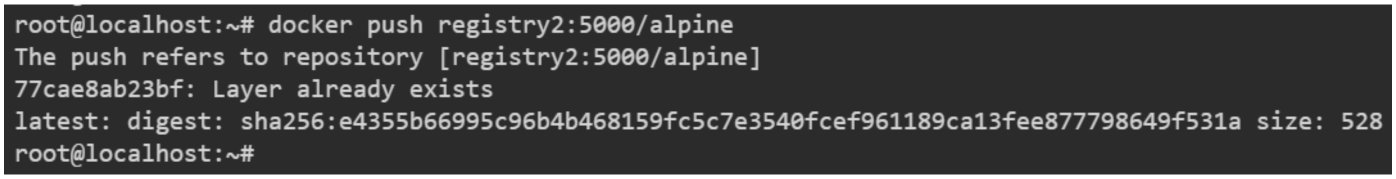


```
root@localhost:~# curl -k https://registry2:5000/v2/_catalog
{"repositories":["alpine"]}
root@localhost:~#
```

The alpine image is present on this registry.

Step 5: Try to push the already tagged alpine image to the registry.

Command: docker push registry2:5000/alpine



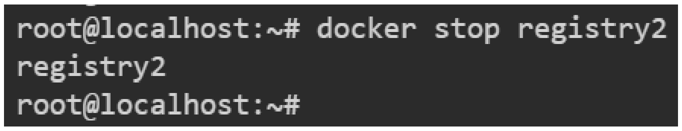
```
root@localhost:~# docker push registry2:5000/alpine
The push refers to repository [registry2:5000/alpine]
77cae8ab23bf: Layer already exists
latest: digest: sha256:e4355b66995c96b4b468159fc5c7e3540fcef961189ca13fee877798649f531a size: 528
root@localhost:~#
```

The image already exists.

Enabling Authentication

Step 1: Stop the already running registry container.

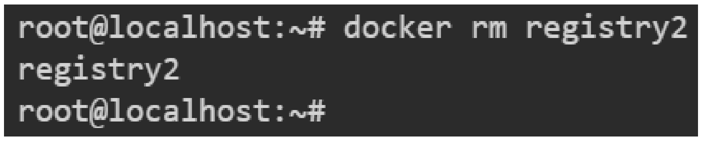
Command: docker stop registry2



```
root@localhost:~# docker stop registry2
registry2
root@localhost:~#
```

Also, remove the container.

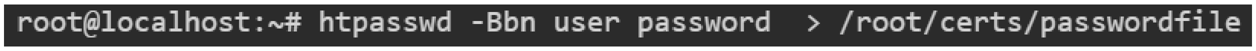
Command: docker rm registry2



```
root@localhost:~# docker rm registry2
registry2
root@localhost:~#
```

Step 2: Create password file using htpasswd.

Command: htpasswd -Bbn user password > /root/certs/passwordfile



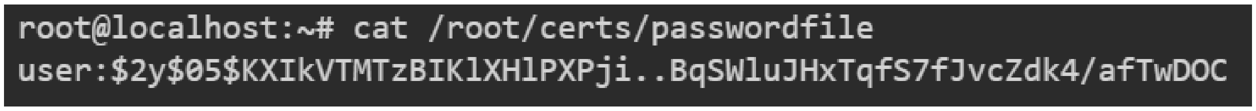
```
root@localhost:~# htpasswd -Bbn user password  > /root/certs/passwordfile
```

Credentials used:

Username: user
Password: password

Check the contents of the generated password file.

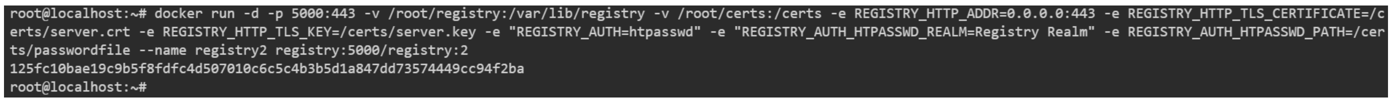
Command: cat /root/certs/passwordfile



```
root@localhost:~# cat /root/certs/passwordfile
user:$2y$05$KXIkVTMTzBIKlXHlPXPji..BqSWluJHxTqfS7fJvcZdk4/afTwDOC
```

Step 3: Run this registry container

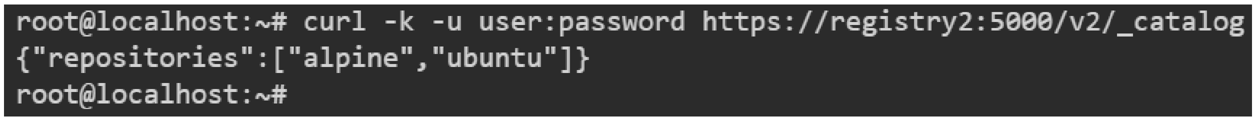
Command: docker run -d -p 5000:443 -v /root/registry:/var/lib/registry -v /root/certs:/certs -e REGISTRY_HTTP_ADDR=0.0.0.0:443 -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/server.crt -e REGISTRY_HTTP_TLS_KEY=/certs/server.key -e "REGISTRY_AUTH=htpasswd" -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" -e REGISTRY_AUTH_HTPASSWD_PATH=/certs/passwordfile --name registry2 registry:5000/registry:2



```
root@localhost:~# docker run -d -p 5000:443 -v /root/registry:/var/lib/registry -v /root/certs:/certs -e REGISTRY_HTTP_ADDR=0.0.0.0:443 -e REGISTRY_HTTP_TLS_CERTIFICATE=/c
erts/server.crt -e REGISTRY_HTTP_TLS_KEY=/certs/server.key -e "REGISTRY_AUTH=htpasswd" -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" -e REGISTRY_AUTH_HTPASSWD_PATH=/cer
ts/passwordfile --name registry2 registry:5000/registry:2
125fc10bae19c9b5f8fdfc4d507010c6c5c4b3b5d1a847dd73574449cc94f2ba
root@localhost:~#
```

Step 4: Check the images present on the registry using curl.

Command: curl -k -u user:password https://registry2:5000/v2/_catalog

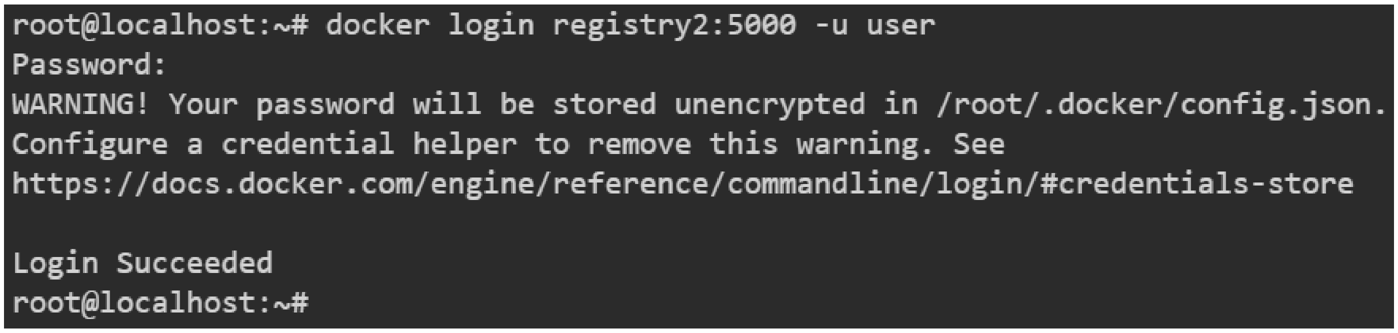


```
root@localhost:~# curl -k -u user:password https://registry2:5000/v2/_catalog
{"repositories":["alpine","ubuntu"]}
root@localhost:~#
```

Step 5: Login into the registry to use it.

Command: docker login registry2:5000 -u user

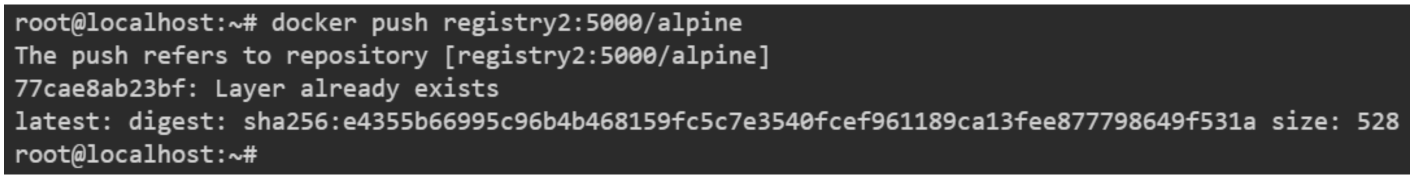
Password: password


```
root@localhost:~# docker login registry2:5000 -u user
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@localhost:~#
```

Step 6: Try to push the already tagged alpine image to the registry.

Command: docker push registry2:5000/alpine


```
root@localhost:~# docker push registry2:5000/alpine
The push refers to repository [registry2:5000/alpine]
77cae8ab23bf: Layer already exists
latest: digest: sha256:e4355b66995c96b4b468159fc5c7e3540fcef961189ca13fee877798649f531a size: 528
root@localhost:~#
```

The image can be pushed to the registry.