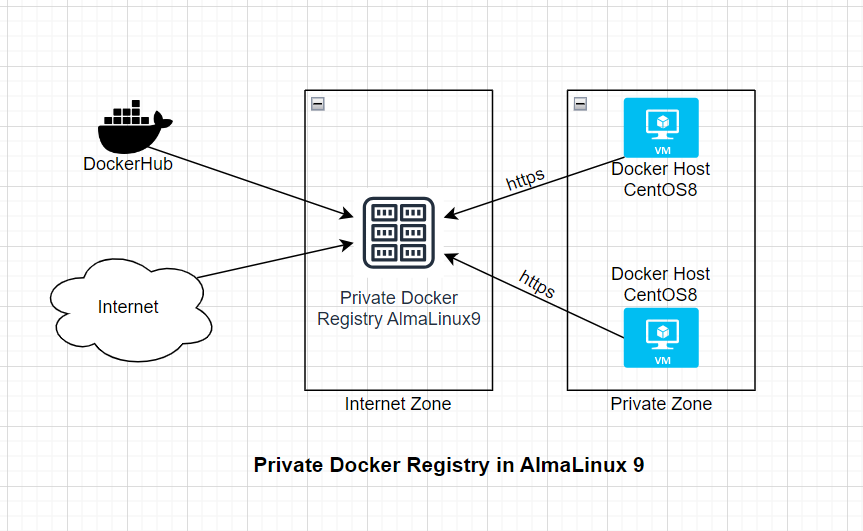
**Configuring a Docker Private Registry on AlmaLinux.**

**Working Diagram**

[](https://github.com/MdAhosanHabib/Docker_Private_Registry_AlmaLinux/blob/main/Photo/Docker_Private_Registry.PNG)

**Introduction:**

A Docker private registry is a valuable tool for managing and distributing Docker images within your organization. It allows you to store, share, and control access to container images privately. This document outlines the step-by-step process to set up a Docker private registry on an AlmaLinux system.

**Step 1: Docker Installation:**

1. Add the Docker repository: Use the dnf config-manager command to add the Docker repository for CentOS.
2. Verify the repository is added: Check the available repositories using dnf repolist -v.
3. Install Docker: Use dnf install docker-ce to install Docker.
4. Start and enable Docker: Use systemctl start docker and systemctl enable docker to start and enable the Docker service.
5. Check Docker version: Verify the Docker installation with docker --version.

**Step 2: Registry Container Setup:**

1. Create a directory for registry storage: Use mkdir /docker\_repo to create a directory for registry storage.
2. Run the Docker registry container: Start the Docker registry container with the appropriate options, such as volume mappings and environment variables.
3. Verify the container status: Check the status of the registry container using docker ps.

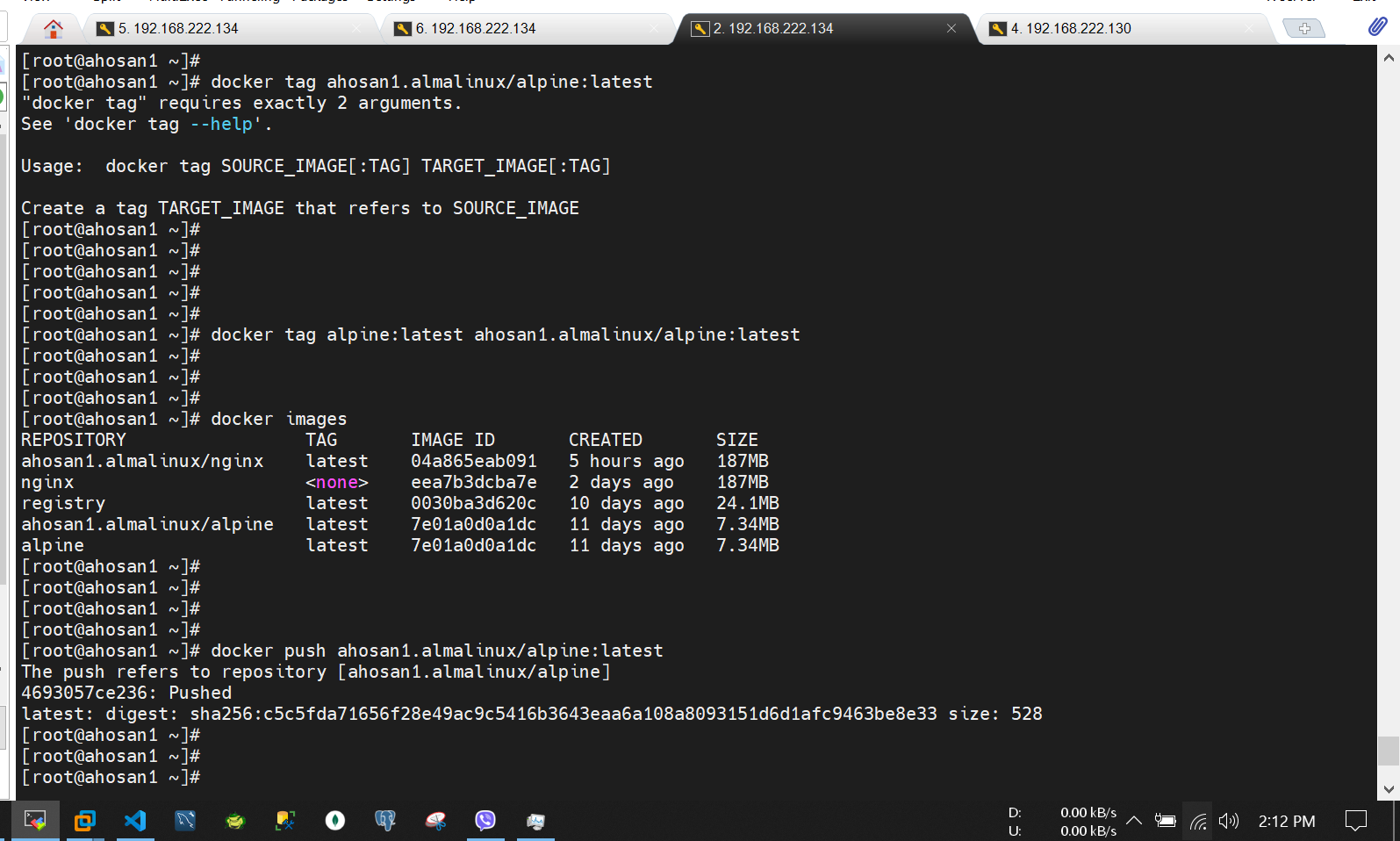
**Step 3: Install a TLS Certificate:**

1. Generate a TLS certificate: Use the openssl req command to generate a self-signed TLS certificate with the appropriate subject alternate name (SAN).
2. Save the certificate: Store the generated certificate and private key files in the /certs directory.

**Step 4: Nginx Setup:**

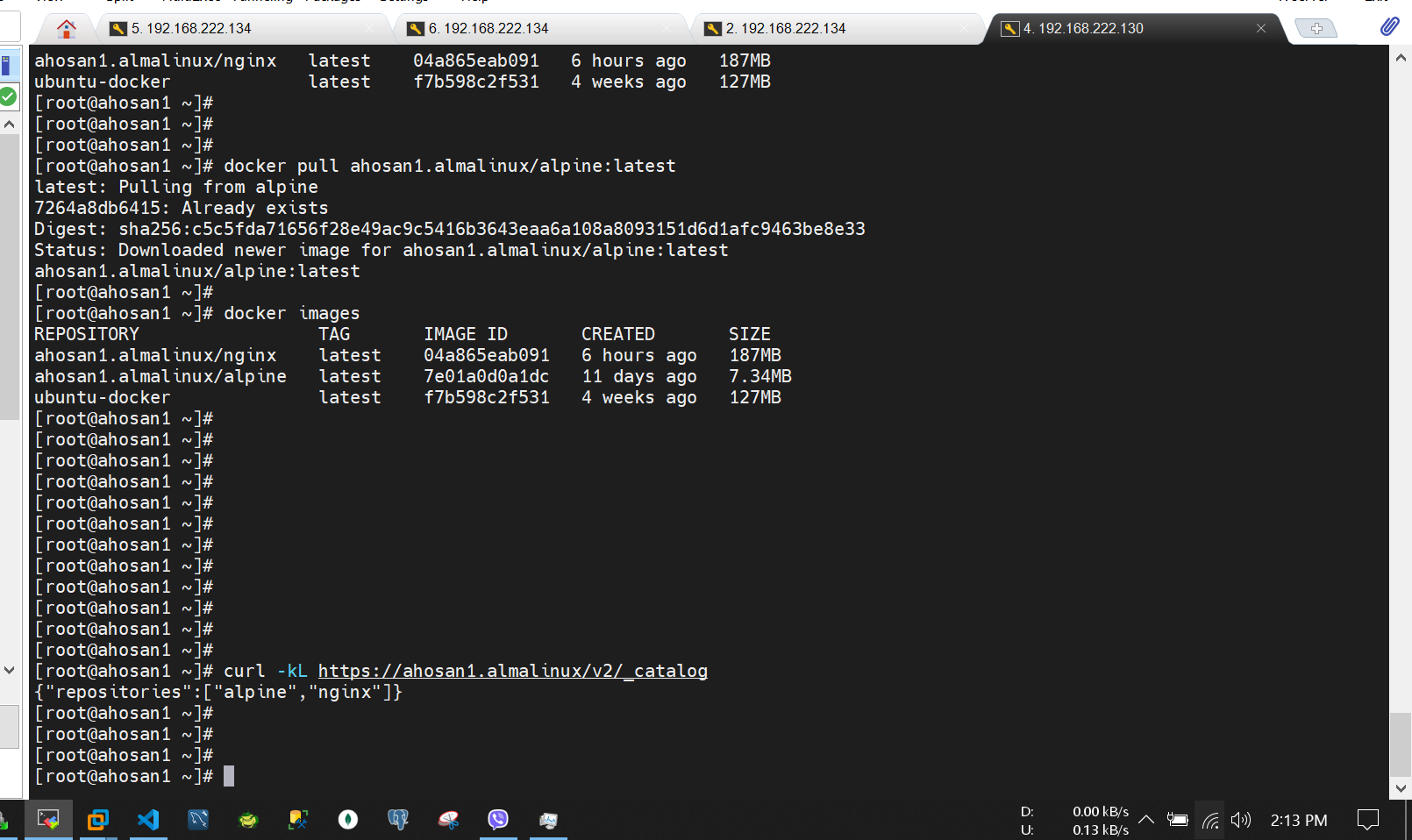
1. Install Nginx: Use dnf install nginx to install the Nginx web server.
2. Configure Nginx: Edit the Nginx configuration files to set up a reverse proxy for the Docker registry.
3. Restart Nginx: Restart the Nginx service to apply the configuration changes.

**Step 5: Push Image to Private Registry:**

[](https://github.com/MdAhosanHabib/Docker_Private_Registry_AlmaLinux/blob/main/Photo/Private_Registry.PNG)

1. Run a sample container: Start a container using the docker run command.
2. Commit the container: Use docker commit to create a new image from the running container.
3. Tag the image: Assign a new tag to the image using docker tag.
4. Push the image: Push the image to the private registry using docker push.

**Step 6: Pull & Push Images at Private Registry:**

[](https://github.com/MdAhosanHabib/Docker_Private_Registry_AlmaLinux/blob/main/Photo/Remote_host.PNG)

1. Verify the private registry: Check the repositories available in the private registry using curl.
2. Pull and tag the image: Pull an image from a public repository and tag it for the private registry.
3. Push the image to the private registry: Use docker push to push the image to the private registry.
4. Verify image availability: Check that the pushed image is available in the private registry.

**Step 7: Pull & Push Images from Remote Host:**

1. Configure remote host: Update the /etc/hosts file on a remote host to resolve the private registry hostname.
2. Configure Docker on the remote host: Configure Docker on the remote host to use the private registry as a mirror.
3. Restart Docker on the remote host: Restart the Docker service on the remote host to apply the configuration changes.
4. Pull images from the remote host: Pull images from the private registry on the remote host.

**Step 8: Delete Image from Private Registry and Push Images from Remote:**

1. Delete image from private registry: Remove an image from the private registry.
2. Restart the registry container: Restart the registry container to reflect the changes.
3. Check the registry status: Use curl to verify the repositories in the private registry.
4. Delete and push images from a remote host: Delete and push images from the private registry on a remote host.
5. Verify image availability: Confirm that the images are pushed and available in the private registry.

**Conclusion:**

Setting up a Docker private registry on AlmaLinux involves multiple steps, including Docker installation, registry container setup, TLS certificate generation, Nginx configuration, image management, and interaction between local and remote hosts. Following these steps allows you to create a secure and controlled environment for storing and distributing Docker images within your organization.

**Now we go for Hands on.**

**Step1: Docker Install on AlmaLinux**

[root@ahosan1 ~]# dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo

[root@ahosan1 ~]# dnf repolist -v

[root@ahosan1 ~]# docker --version

Docker version 24.0.5, build ced0996

[root@ahosan1 ~]# systemctl start docker

[root@ahosan1 ~]# systemctl enable docker

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.

[root@ahosan1 ~]# systemctl status docker

**Step2: Registry Container Run**

[root@ahosan1 ~]# mkdir /docker\_repo

[root@ahosan1 ~]# docker run --detach \

--restart=always \

--name registry \

--volume /docker\_repo:/docker\_repo \

--env REGISTRY\_STORAGE\_FILESYSTEM\_ROOTDIRECTORY=/docker\_repo \

--publish 5000:5000 \

registry

[root@ahosan1 ~]# netstat -tlnp | grep :5000

**Step3: Install a TLS certificate**

[root@ahosan1 ~]# mkdir /certs

[root@ahosan1 ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.222.134 ahosan1.almalinux

[root@ahosan1 ~]#

[root@ahosan1 ~]# openssl req \

-newkey rsa:4096 -nodes -sha256 -keyout /certs/ahosan1.almalinux.key \

-addext "subjectAltName = DNS:ahosan1.almalinux" \

-x509 -days 365 -out /certs/ahosan1.almalinux.crt

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /certs/ahosan1.almalinux.key -out /certs/ahosan1.almalinux.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) [XX]:BD

State or Province Name (full name) []:Dhaka

Locality Name (eg, city) [Default City]:Dhaka

Organization Name (eg, company) [Default Company Ltd]:ITCL

Organizational Unit Name (eg, section) []:DevOps

Common Name ()eg, your name or your server's hostname) []:ahosan1.almalinux

Email Address []: ahosan@itcbd.com

[root@ahosan1 ~]#

**Step4: Nginx setup**

[root@ahosan1 ~]# dnf install nginx -y

[root@ahosan1 ~]# vi /etc/nginx/nginx.conf #add this line at http section

client\_max\_body\_size 2048m;

[root@ahosan1 ~]# vi /etc/nginx/conf.d/ahosan1.almalinux.conf

server {

listen 80;

server\_name ahosan1.almalinux;

return 301 https://$host$request\_uri;

}

server {

listen 443 ssl http2;

server\_name ahosan1.almalinux;

# ssl params

ssl\_certificate /certs/ahosan1.almalinux.crt;

ssl\_certificate\_key /certs/ahosan1.almalinux.key;

ssl\_protocols TLSv1.2;

location / {

proxy\_pass http://localhost:5000;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_read\_timeout 600;

}

}

[root@ahosan1 certs]# systemctl stop firewalld.service

[root@ahosan1 certs]# systemctl disable firewalld.service

[root@ahosan1 certs]# setenforce 0

[root@ahosan1 certs]# systemctl restart nginx

[root@ahosan1 certs]# curl -kIL http://ahosan1.almalinux/v2/

**Step5: Image push to Private Repo**

[root@ahosan1 certs]# docker run --name webserver --detach nginx

[root@ahosan1 ~]# docker exec -it webserver /bin/bash

root@c4659dbc1bff:/# ip addr

bash: ip: command not found

root@c4659dbc1bff:/# ping 8.8.8.8

bash: ping: command not found

root@c4659dbc1bff:/# apt update

root@c4659dbc1bff:/# apt install iproute2 iputils-ping net-tools -y

0% [Connecting to deb.debian.org]^C

root@c4659dbc1bff:/# exit

exit

[root@ahosan1 ~]#

[root@ahosan1 ~]# vi /etc/docker/daemon.json

{

"insecure-registries" : ["https://ahosan1.almalinux"]

}

[root@ahosan1 certs]# mkdir -p /etc/docker/certs.d/ahosan1.almalinux

[root@ahosan1 certs]# cp /certs/ahosan1.almalinux.crt /etc/docker/certs.d/ahosan1.almalinux/

[root@ahosan1 certs]# systemctl restart docker

[root@ahosan1 ~]# docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest eea7b3dcba7e 2 days ago 187MB

registry latest 0030ba3d620c 9 days ago 24.1MB

[root@ahosan1 ~]# docker commit webserver nginx

sha256:04a865eab091eeca930a357934d906bf50f7611a2979f9ee2fb521f5ddf9f261

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker tag nginx ahosan1.almalinux/nginx

[root@ahosan1 ~]# docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE

ahosan1.almalinux/nginx latest eea7b3dcba7e 2 days ago 187MB

nginx latest eea7b3dcba7e 2 days ago 187MB

registry latest 0030ba3d620c 9 days ago 24.1MB

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker image ls

[root@ahosan1 certs]# docker push ahosan1.almalinux/nginx

[root@ahosan1 certs]# curl -kL https://ahosan1.almalinux/v2/\_catalog

--run in one tab

[root@ahosan1 certs]# docker logs -f registry

--run in one tab

[root@ahosan1 certs]# tail -1000f /var/log/nginx/access.log

**Step6: On remote host**

--remote host

[root@ahosan1 ~]# vi /etc/hosts

192.168.222.134 ahosan1.almalinux

--copy cert here from private docker hub

[root@ahosan1 ~]# mkdir -p /etc/docker/certs.d/ahosan1.almalinux

[root@ahosan1 ~]# vi /etc/docker/daemon.json

{

"registry-mirrors": ["https://ahosan1.almalinux"]

}

[root@ahosan1 ~]# systemctl restart docker.service

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

[root@ahosan1 ~]# docker run --name webserver\_custom -d -p 80:80 ahosan1.almalinux/nginx

Unable to find image 'ahosan1.almalinux/nginx:latest' locally

latest: Pulling from nginx

52d2b7f179e3: Pull complete

fd9f026c6310: Pull complete

055fa98b4363: Pull complete

96576293dd29: Pull complete

a7c4092be904: Pull complete

e3b6889c8954: Pull complete

da761d9a302b: Pull complete

d48f1c60ed53: Pull complete

Digest: sha256:483fdbdc7ca49f019c694630ed031c2505a5a6398f7da13d205bb26c50e01b39

Status: Downloaded newer image for ahosan1.almalinux/nginx:latest

7aff3628d882249b70854cadb03392d14d80a2b1970e9b777d816743e4f1af29

[root@ahosan1 ~]#

**Step7: Pull & Push images at Private Registry**

--private reg OS

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

{"repositories":["nginx"]}

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker pull alpine:latest

[root@ahosan1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

ahosan1.almalinux/nginx latest 04a865eab091 5 hours ago 187MB

nginx <none> eea7b3dcba7e 2 days ago 187MB

registry latest 0030ba3d620c 10 days ago 24.1MB

alpine latest 7e01a0d0a1dc 11 days ago 7.34MB

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker tag alpine:latest ahosan1.almalinux/alpine:latest

[root@ahosan1 ~]# docker push ahosan1.almalinux/alpine:latest

The push refers to repository [ahosan1.almalinux/alpine]

4693057ce236: Pushed

latest: digest: sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33 size: 528

[root@ahosan1 ~]#

--remote host OS

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

[root@ahosan1 ~]# docker pull ahosan1.almalinux/alpine:latest

latest: Pulling from alpine

7264a8db6415: Already exists

Digest: sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33

Status: Downloaded newer image for ahosan1.almalinux/alpine:latest

ahosan1.almalinux/alpine:latest

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

ahosan1.almalinux/nginx latest 04a865eab091 6 hours ago 187MB

ahosan1.almalinux/alpine latest 7e01a0d0a1dc 11 days ago 7.34MB

ubuntu-docker latest f7b598c2f531 4 weeks ago 127MB

[root@ahosan1 ~]#

**Step8: Delete from Private Reg and Push images from Remote**

--from private reg OS to delete pushed images

[root@ahosan1 ~]# rm /docker\_repo/docker/registry/v2/repositories/alpine

[root@ahosan1 ~]# docker restart registry

registry

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

{"repositories":["nginx"]}

--from remote host OS for upload images

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

{"repositories":["nginx"]}

[root@ahosan1 ~]# docker rmi -f ahosan1.almalinux/alpine

Untagged: ahosan1.almalinux/alpine:latest

Untagged: ahosan1.almalinux/alpine@sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33

Deleted: sha256:7e01a0d0a1dcd9e539f8e9bbd80106d59efbdf97293b3d38f5d7a34501526cdb

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

ubuntu-docker latest f7b598c2f531 4 weeks ago 127MB

[root@ahosan1 ~]# docker pull alpine:latest

[root@ahosan1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

alpine latest 7e01a0d0a1dc 11 days ago 7.34MB

ubuntu-docker latest f7b598c2f531 4 weeks ago 127MB

[root@ahosan1 ~]#

[root@ahosan1 ~]# docker tag alpine:latest ahosan1.almalinux/alpine:latest

[root@ahosan1 ~]# docker push ahosan1.almalinux/alpine:latest

The push refers to repository [ahosan1.almalinux/alpine]

4693057ce236: Pushed

latest: digest: sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33 size: 528

[root@ahosan1 ~]#

[root@ahosan1 ~]# curl -kL https://ahosan1.almalinux/v2/\_catalog

{"repositories":["alpine","nginx"]}

[root@ahosan1 ~]#

**Congratulations! Thank you From Ahosan Habib.**