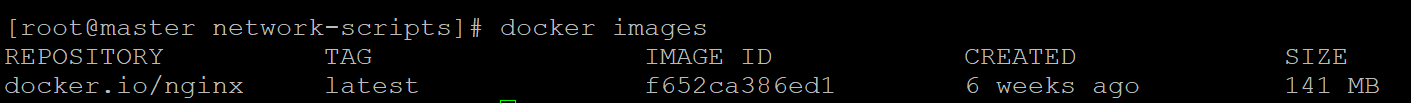
**Docker**

**Creator : Aswin K S**

**Basic Command on a container**

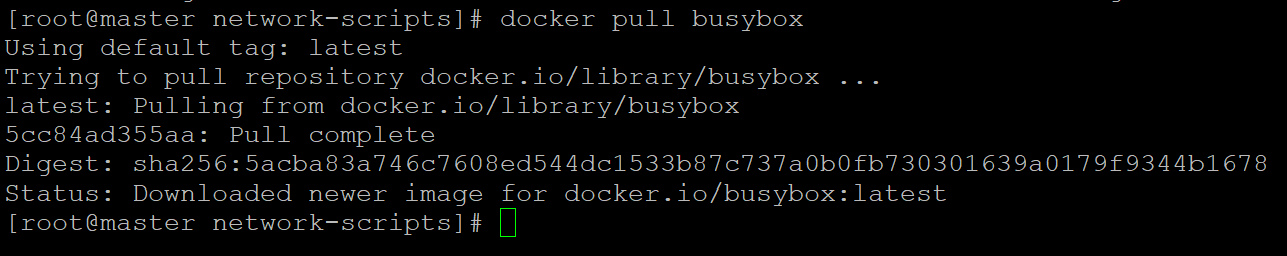
**1.To find all downloaded images**

docker images



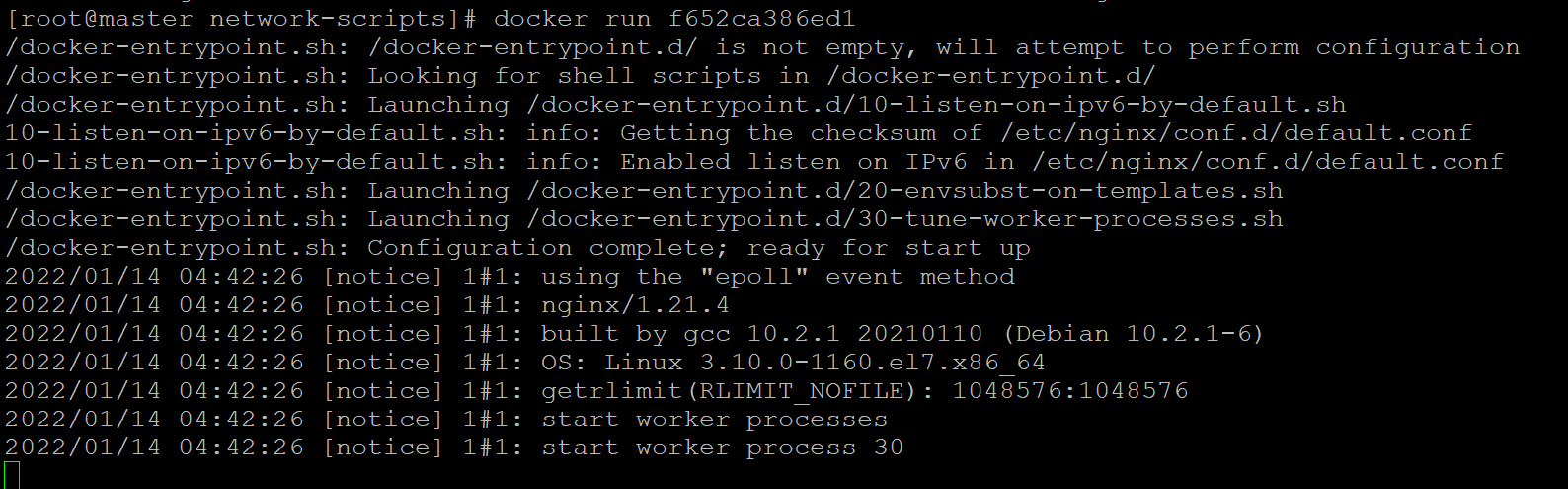
**2.To Download an image from docker hub**

Docker pull image name



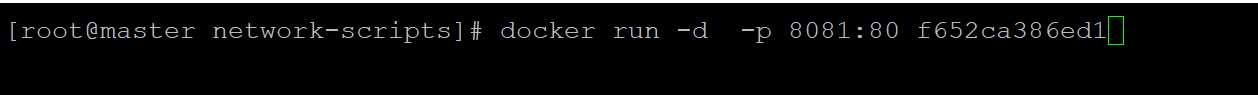
**3.To run and image as container**

docker run image name



Use -d flag to run container in background

**4.Publishing port**



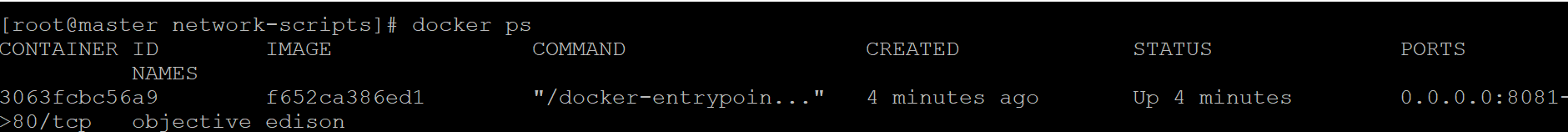
8081:80

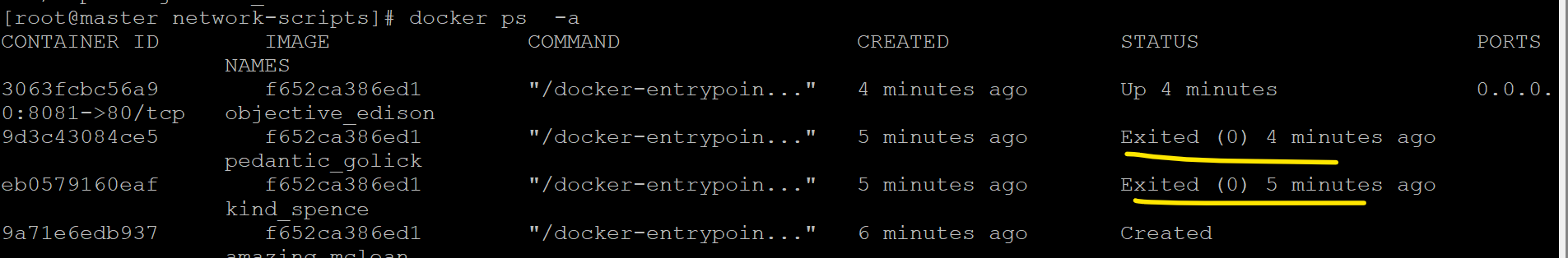
8081: host machine port and 80 is the container port

**5. To see running containers**

docker ps

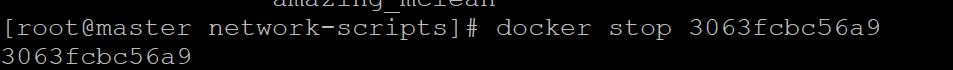
docker ps -a ( to see all container including exited ones)





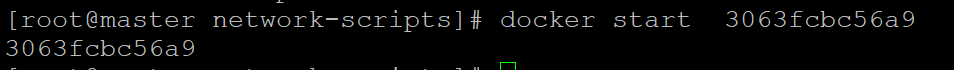
**6. To stop a container**

Docker stop CID



**7. To start a container**

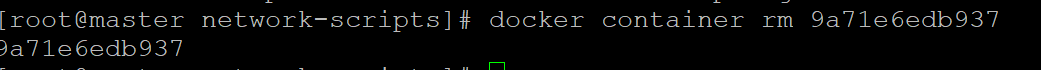
Docker start CID



**8. To kill a container**

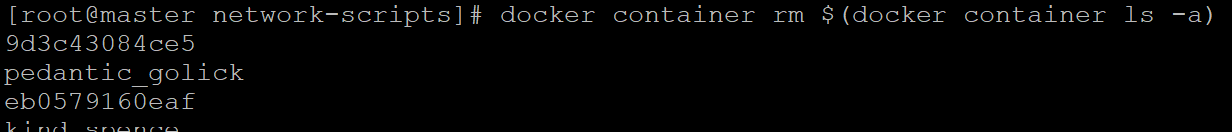
Docker kill CID

To Remove a container ( only stopped containers can be removed)



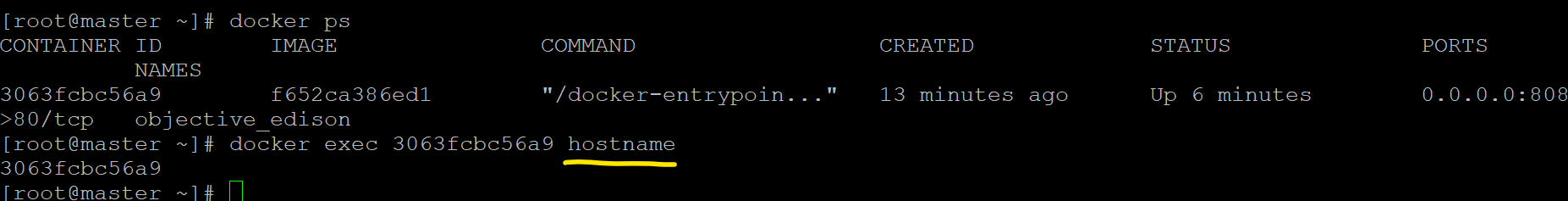
Note: To do bulk removal

docker container rm $(docker container ls -a)



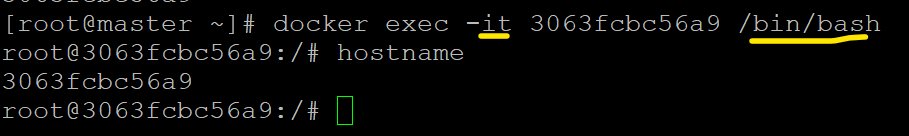
**9. To run command inside a running container**

docker exec 3063fcbc56a9(CID) hostname(Command)



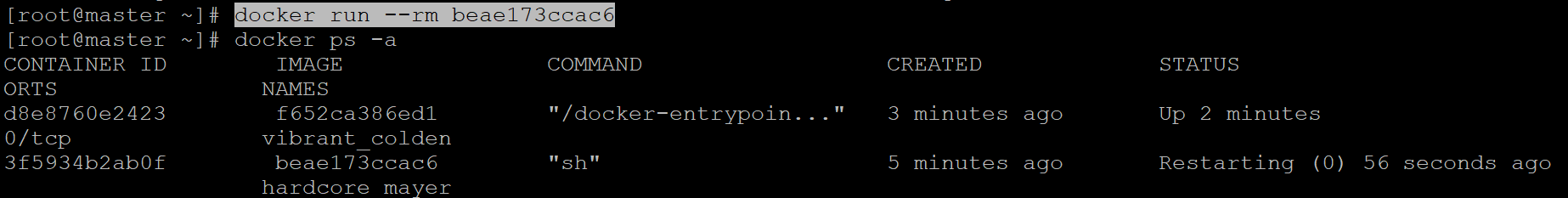
**10. To log in to a running container**

docker exec -it 3063fcbc56a9 /bin/bash



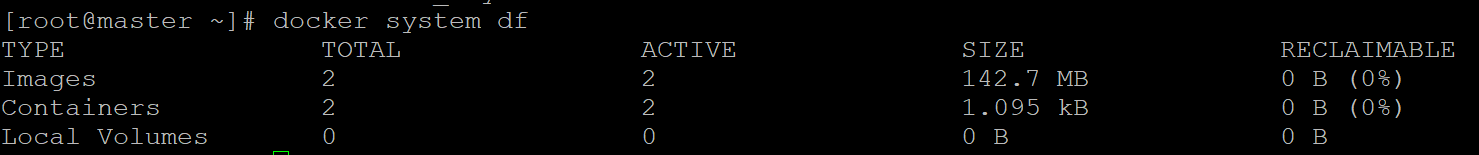
**11. To remove a container after container exit**

docker run --rm beae173ccac6



**11. To see docker system usage**

docker system df

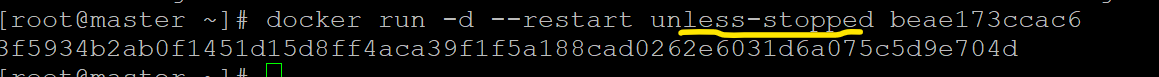


**Restart Policies**

Restart policy determine whether a container should restart itself if it shut down unexpectedly.

Flags:

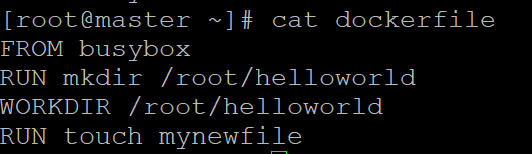
No, on-failure, unless-stopped

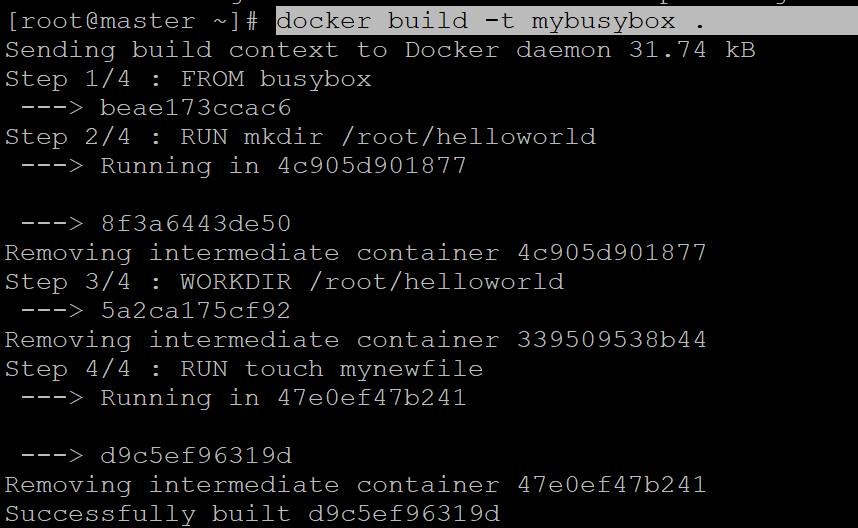


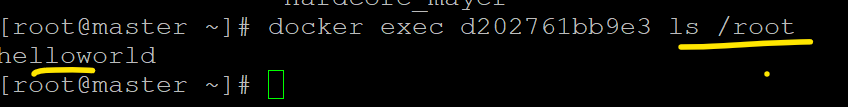
**Working with Docker images**

**1.Creating a docker file to pull a busy box image and create a new folder in /root/helloworld**

Docker file



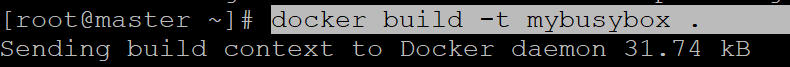




**Tagging Docker images**

1.While building an image

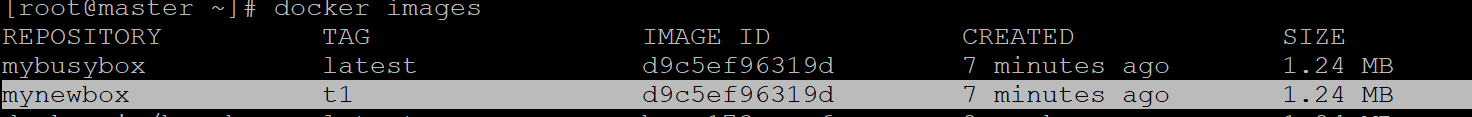
Docker build -t <tag name> .



2.After image is build

Docker tag container ID <tag name>:<version number>

docker tag d9c5ef96319d mynewbox:t1

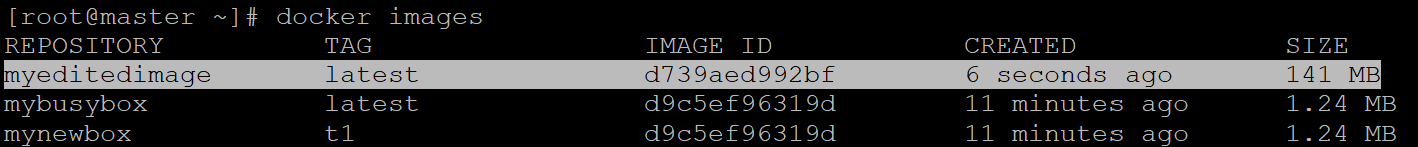


**Committing a container to create new modified image**

Creating a new image from a modified container

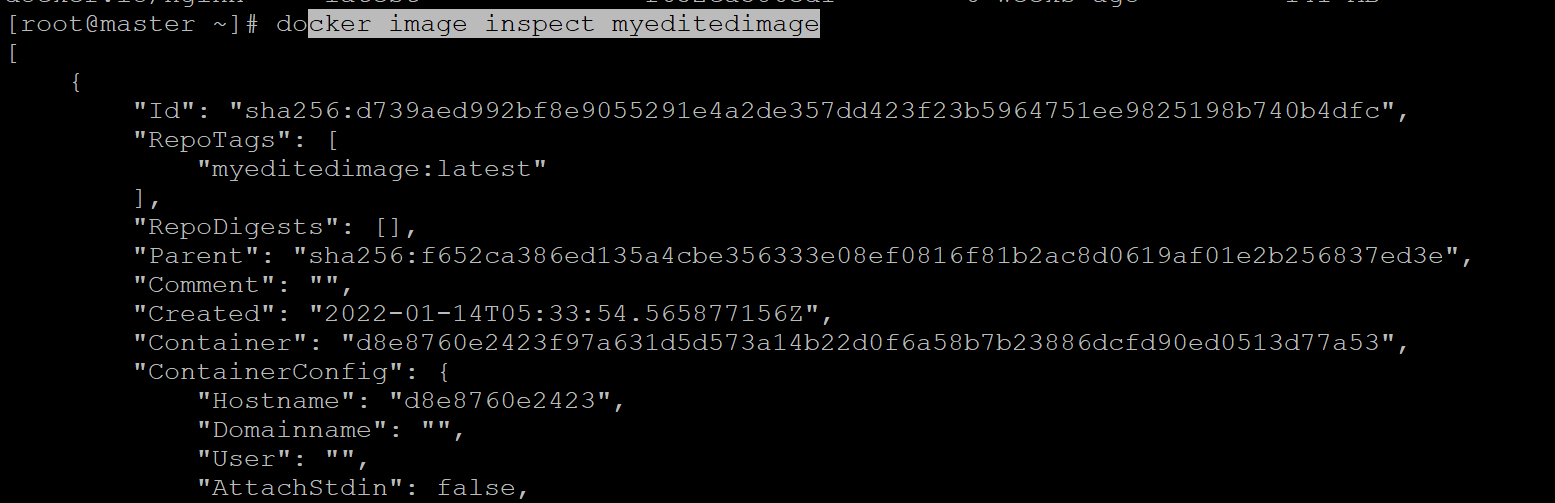
docker container commit CID myeditedimage (new image ID)





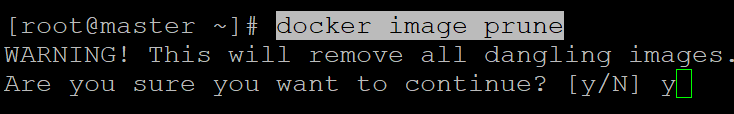
**Docker inspect**

To get all details of a docker image



**Image Prune**

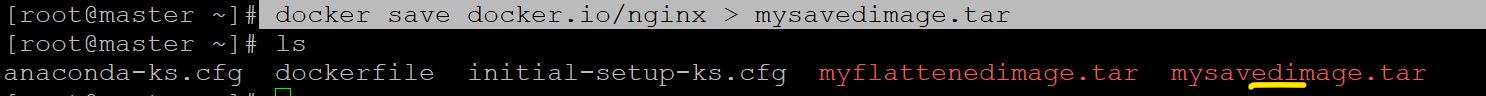
Deleting unused and unwanted docker images (Caution)



**Docker Save and load**

To save an image as a tar file and load it again. this will save all the history of the image also

Save : docker save docker.io/nginx > mysavedimage.tar



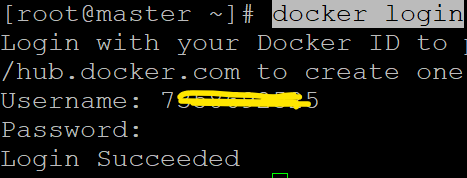
Load : docker load < mysavedimage.tar



**Docker Registry**

1. Push an image to docker registry(public)

Log in to docker account

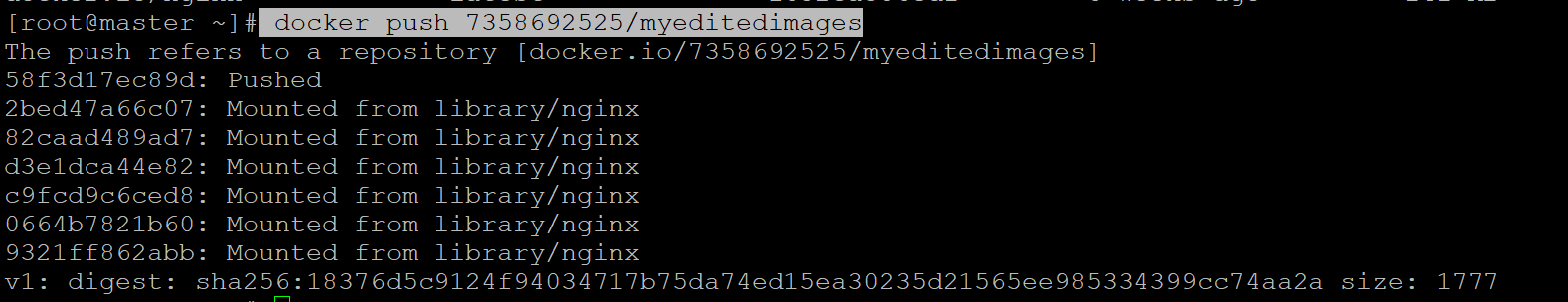


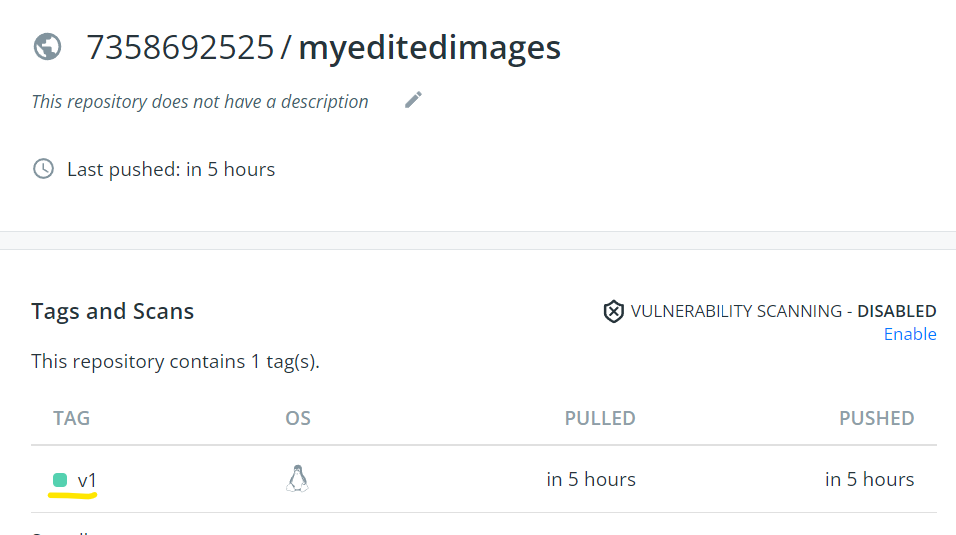
Tag the image with userid/repo name

docker tag myeditedimage 7358692525/myeditedimages:v1

push image

docker push 7358692525/myeditedimages





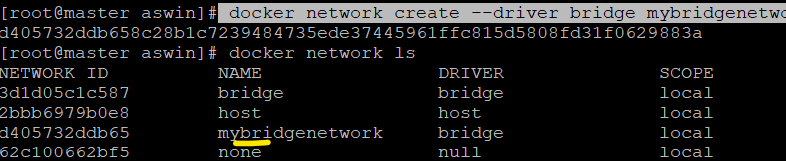
**Docker Networking**

Types of Networks

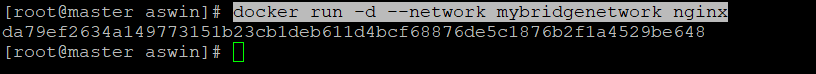
1. Bridge : Default, provide isolation between containers
2. Host: No isolation, no port mapping required, directly bind to host port
3. Overlay:
4. None : No networking

**Creating a bridge network**

docker network create --driver bridge mybridgenetwork



Adding a new container to our network

docker run -d --network mybridgenetwork nginx

**Docker Storage**

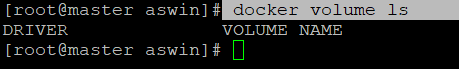
**Docker volumes**

3 types of Volume

1. Volume
2. Bind mount
3. Tmpfs mounr

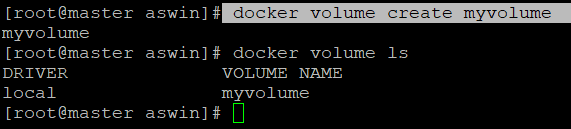
**To see docker volume**

docker volume ls

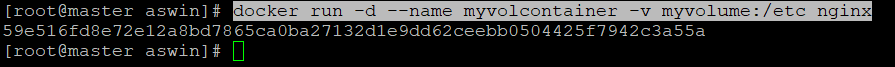


**To create a new volume**

docker volume create myvolume



**Add a container to our Volume**

****

Note: /etc of container will be mapped to the volume and all data in /etc of container reside in our volume.



/etc data is mapped in our host at location: /var/lib/docker/volumes/myvolume/\_data

1. **Bind Mounts**

A directory of host is mapped to a directory inside the container.

docker run -d -it --name bindcontainers -v "$(pwd)"/target:/home nginx



Deleting a docker volume

docker volume rm myvolume