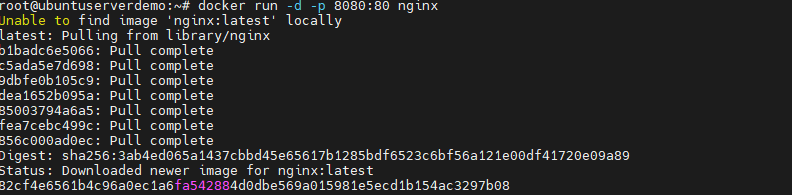
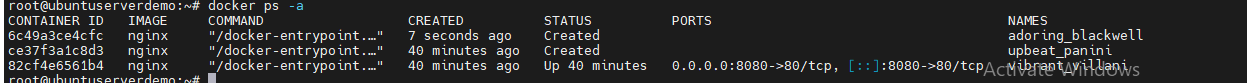
1. **To install ngnix in docker engine command used is= docker run -d -p 8080:80 nginx**
2. **Check Image** – Docker looks for the nginx image locally.
3. If found → proceed.
4. If not found → pull from **Docker Hub**.
5. **Create Container** – Docker creates an isolated container based on the nginx image.
6. **Network Binding** – It connects **host port 8080** → **container port 80**.
7. **Start Process** – It runs the Nginx web server (default process in the nginx image).
8. **Run in Background** – Because of -d, it doesn’t block your terminal



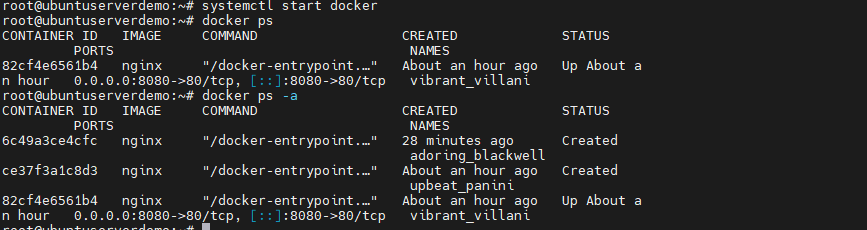
1. **To check list of container running : docker ps –a (here we have 3 container running)**



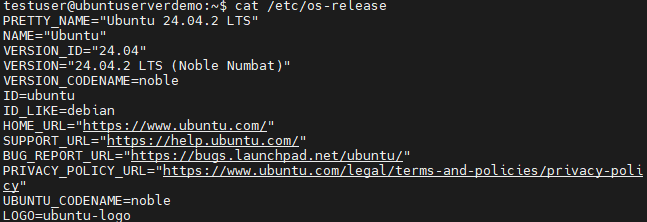
1. **To create a hu.docker.com account.**

Signto url : https:hub.docker.com with mail id and password

1. **To check docker is running and list of container running : docker ps –a , docker ps**



1. **To check the OS release version – cat /etc/os-release**



1. **How to enable user to the docker group .**
2. Check if docker group exist :

Getent group docker

1. Create docker group

Sudo groupadd docker

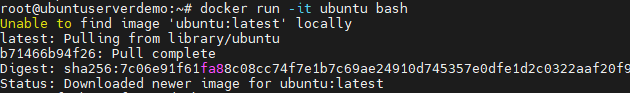
1. Add user to the docker group:

Sudo usermode –aG docker testuser

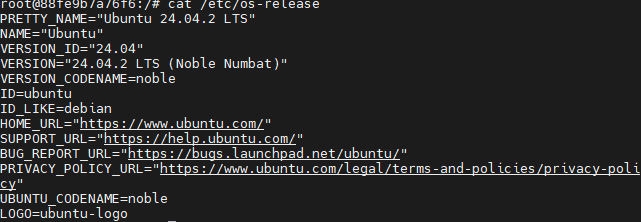
1. Log out and log back in so that group membership will be added
2. Verify your user is the docker group.

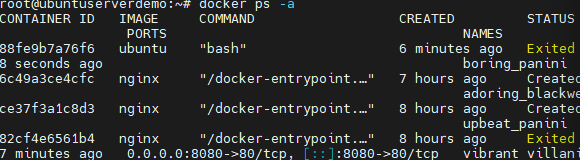
Groups

1. **To start the Ubuntu container : docker run –it Ubuntu bash**
2. Docker run- create a container
3. –it – interactive (-i) + terminal mode (-t) we can type command
4. Ubunutu- uses the default latest image from docker hub
5. Bash- command to run inside the container

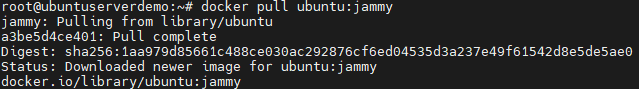


1. Now we are inside the consider root@88fe9b7a76f6: to check os realease – cat/etc/os-release

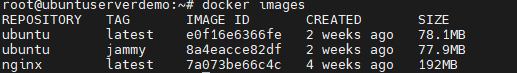




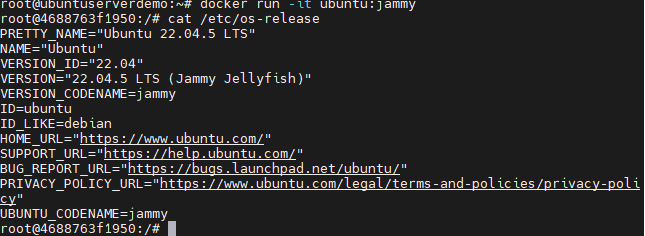
1. **IF we need only to download the Ubuntu machines and also if we need particular image version - we use the command docker pull ubuntu:jammy**



1. And to check all images we use – docker images



1. **IF we need to run the download images we have to use : docker run –it ubuntu:jammy**



1. **If we need to go inside a container which is running : command used : docker exec –it 88fe9b7a76f6 bash ( docker exec –it <container\_id> bash**

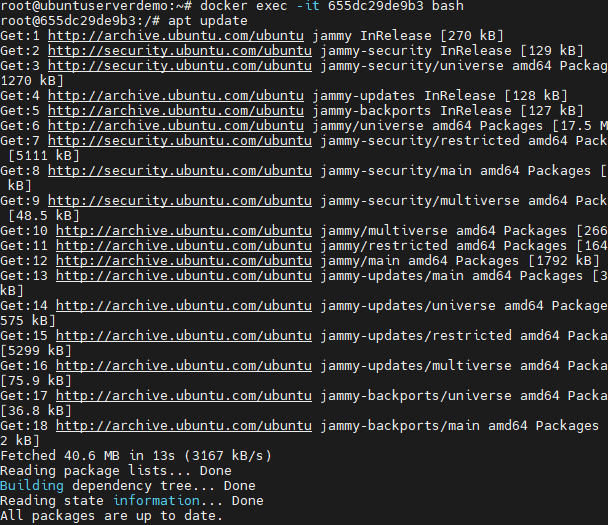


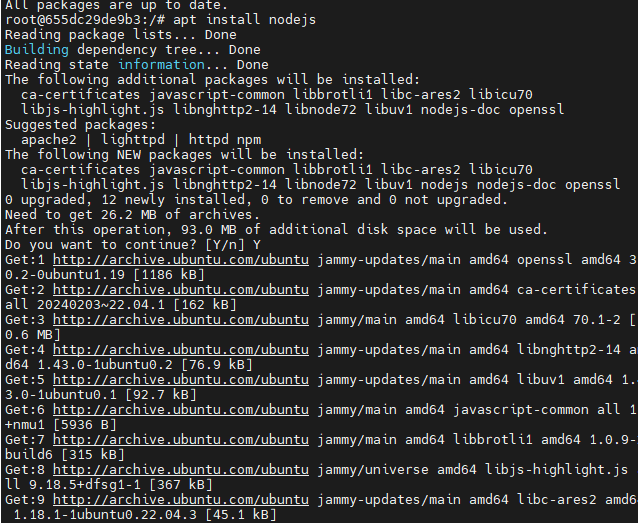
1. **If we need to remove a container : command used : docker rm <container Id>**



1. **If we need to start a container : command used: docker start <container id>**



1. **Once we enter to the container – we have updated the apt update to get the latest update on containers.** 
2. **After update we have installed nodejs : apt install nodejs**





1. **IF we have container (running Ubuntu system with node js installed, we have to save it as an image so that we can use later ): command used: docker commit : take a snapshot permanent**

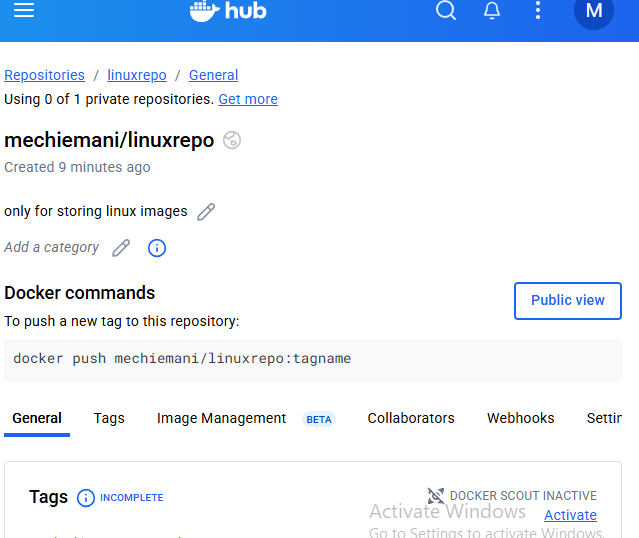
**docker commit -m "Installed nodejs" -a "Mani" 655dc29de9b3 mechiemani/ubuntu-nodejs**

1. Docker commit- command to create new image from a container
2. –m “Installed node js” –commit message describing the changes
3. –a “Mani” – author name of the image
4. 655dc29de93 –container id name
5. Mechiemani/Ubuntu-nodejs – name of the new docker image I have created

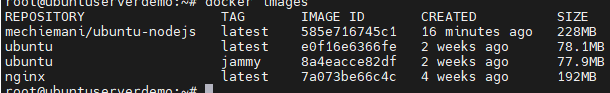


1. **After creating the share id , we have to got to docker hub : create a new repository**

**As a public with name and short description**

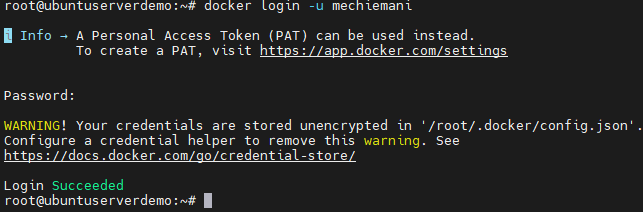


58.currently in my docker machine I have docker images as below.



59.

After commit , we have docker the docker hub login account of users.command used = docker login –u mechiemani and enter the password of account., after it shows login succeed.



60. Tag the image for later usage



61. **docker push : root@ubuntuserverdemo:~# docker push mechiemani/ubuntu-nodejs**

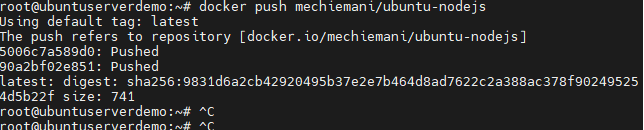
**Using default tag: latest**

**The push refers to repository [docker.io/mechiemani/ubuntu-nodejs]**

**5006c7a589d0: Pushed**

**90a2bf02e851: Pushed**

**latest: digest: sha256:9831d6a2cb42920495b37e2e7b464d8ad7622c2a388ac378f902495254d5b22f size: 741**



62. Docker pushed to a repository

