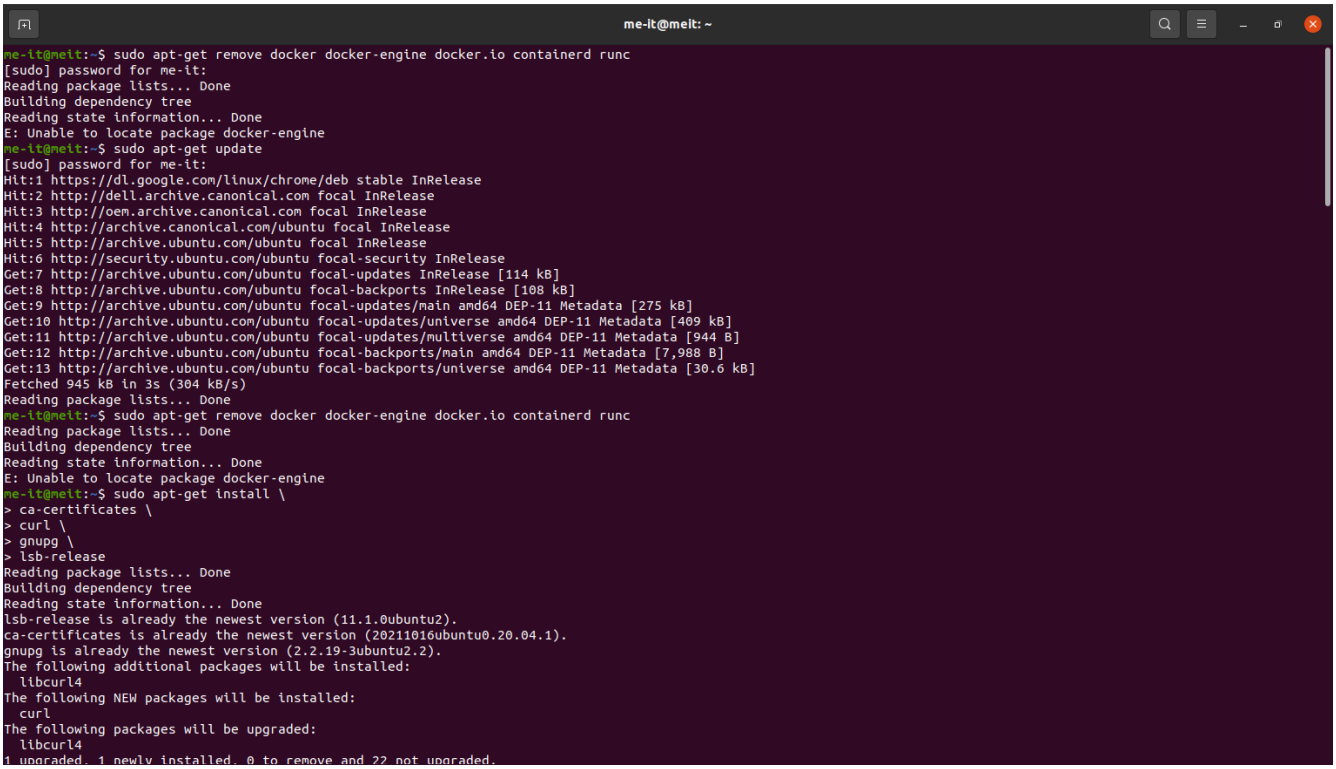
Web Application Development

Assignment: 02 – (B)

Step 1 — Removing previously installed Docker and updating the packages.

a. sudo apt-get remove docker docker-engine docker.io containerd runc: Uninstall any older versions before attempting to install a new version
b. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS
   sudo apt-get update
   sudo apt-get install \
   ca-certificates \
   curl \
   gnupg \
   lsb-release



Step 2 — Installing Docker

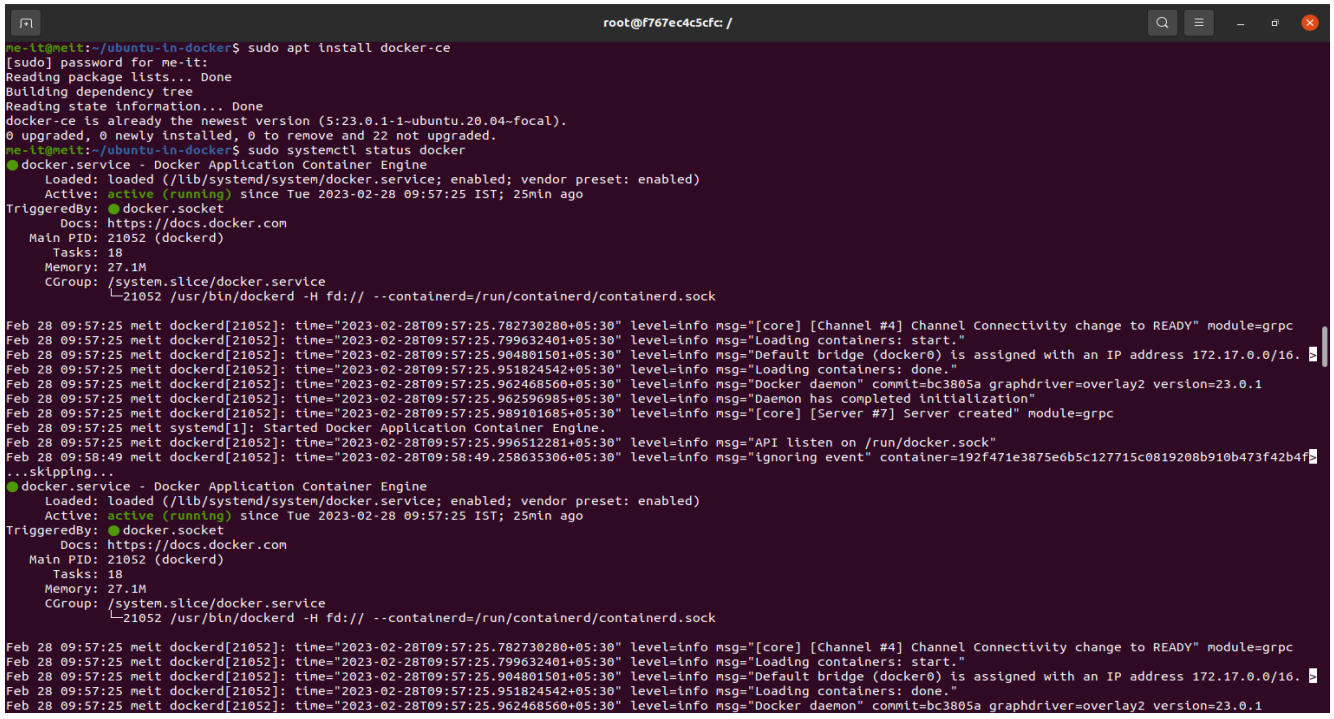
a. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin: To install the latest version

```
me-it@meit:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [24.7 kB]
Hit:3 http://dell.archive.canonical.com focal InRelease
Hit:4 http://archive.canonical.com/ubuntu focal InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://oem.archive.canonical.com focal InRelease
Hit:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:9 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:10 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 82.4 kB in 3s (28.3 kB/s)
Reading package lists... Done
me-it@meit:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin docker-scan-plugin pigz
  slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 22 not upgraded.
Need to get 112 MB of archives.
After this operation, 401 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.6.18-1 [28.2 MB]
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [74.3 kB]
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.10.2-1~ubuntu.20.04~focal [25.9 MB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:23.0.1-1~ubuntu.20.04~focal [13.2 MB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:23.0.1-1~ubuntu.20.04~focal [22.0 MB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:23.0.1-1~ubuntu.20.04~focal [8,765 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-compose-plugin amd64 2.16.0-1~ubuntu.20.04~focal [10.2 MB]
Get:9 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.23.0~ubuntu-focal [3,622 kB]
Fetched 112 MB in 1min 4s (1,751 kB/s)
Selecting previously unselected package pigz.
dpkg: warning: files list file for package 'libpaper1:amd64' missing; assuming package has no files currently installed
(Reading database ... 258584 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
```

b. apt-cache policy docker-ce: to install from the Docker repo instead of the default Ubuntu repo

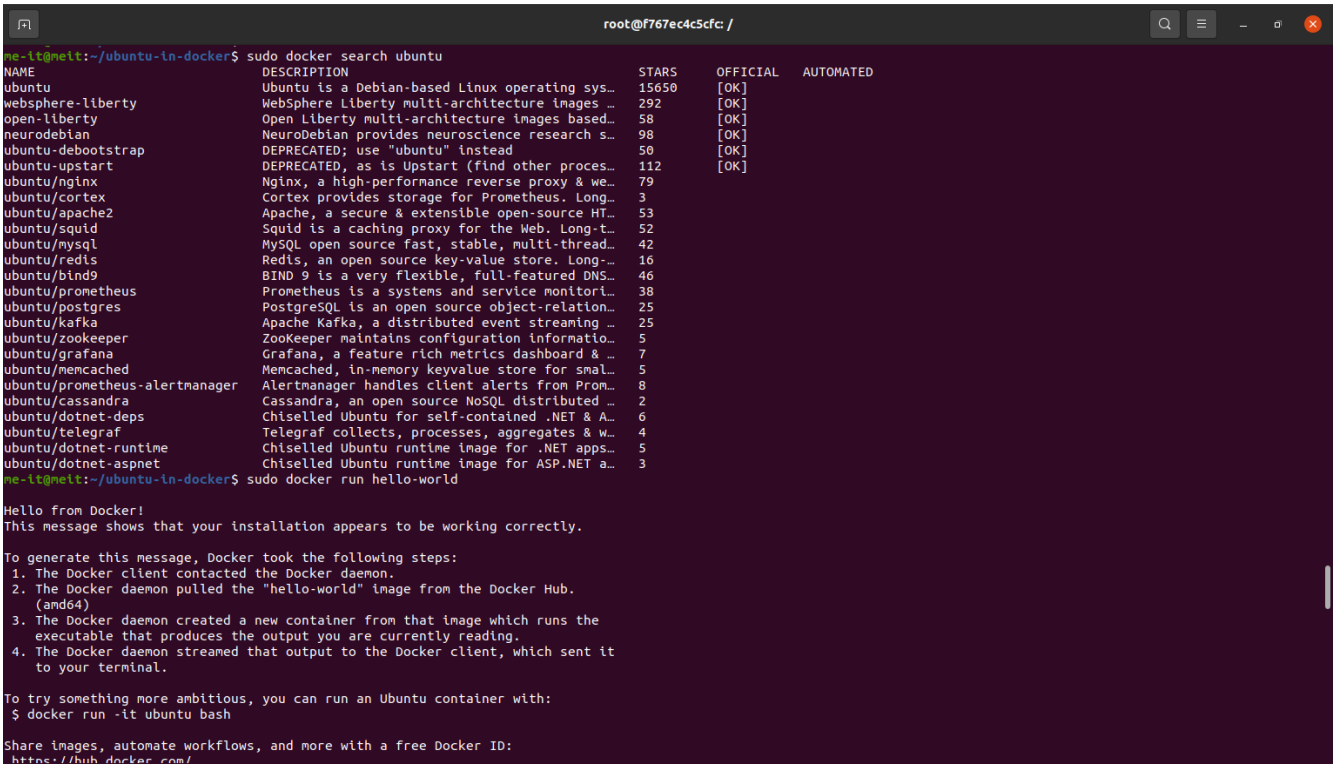
```
me-it@meit:~/ubuntu-in-docker$ apt-cache policy docker-ce
docker-ce:
  Installed: 5:23.0.1-1~ubuntu.20.04~focal
  Candidate: 5:23.0.1-1~ubuntu.20.04~focal
  Version table:
 *** 5:23.0.1-1~ubuntu.20.04~focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
        100 /var/lib/dpkg/status
     5:23.0.0-1~ubuntu.20.04~focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.23~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.22~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.21~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.20~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.19~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.18~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.17~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.16~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.15~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.14~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.13~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.12~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.11~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.10~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.9~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.8~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

c. sudo systemctl status docker: Docker installed, the daemon started, and the process enabled to start on boot. Check that it's running.



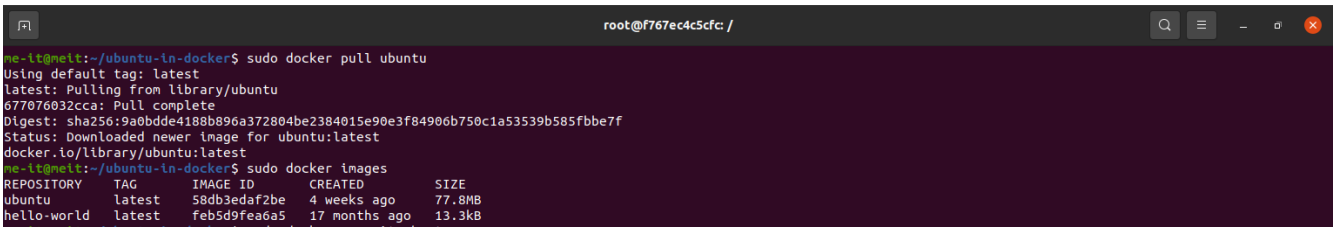
Step 3 — Working with Docker Images:

a. docker search ubuntu: You can search for images available on Docker Hub by using the docker command with the search subcommand.
b. docker run hello-world: To check whether you can access and download images from Docker Hub.



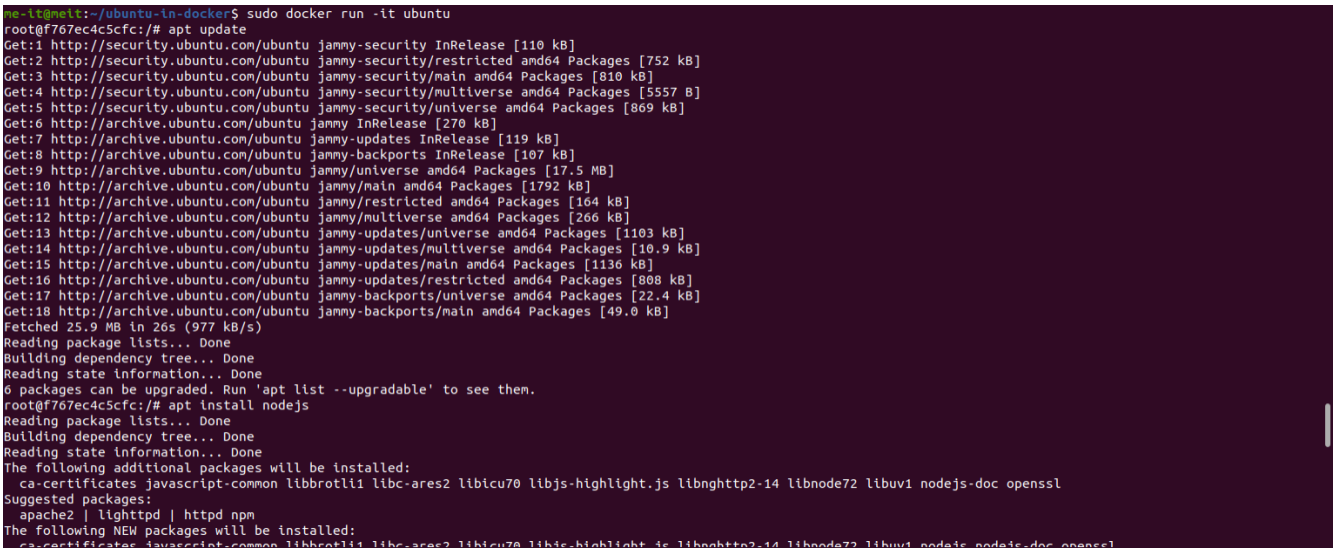
c. docker pull ubuntu: to download the official ubuntu image to your computer.

d. docker images: To see the images that have been downloaded to the computer.

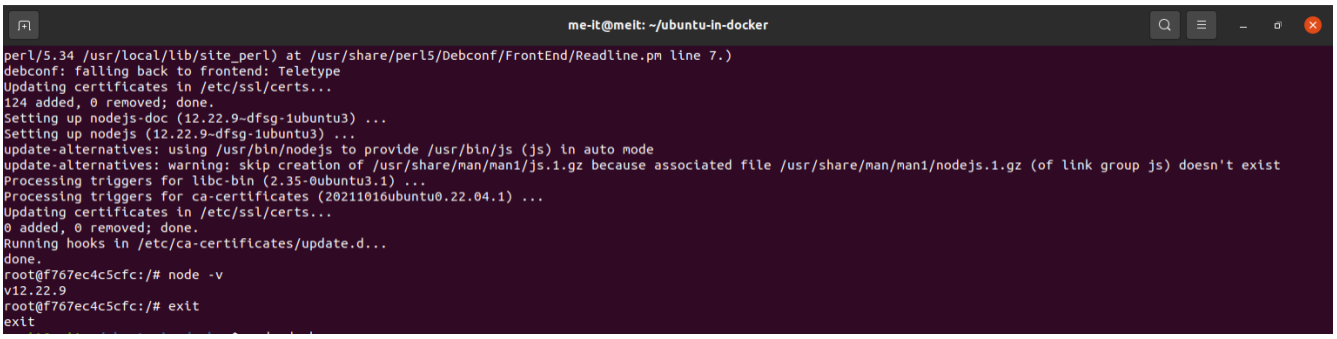


Step 4 — Running a Docker Container

a. docker run -it ubuntu: to run a container using the latest image of Ubuntu. The combination of the -i and -t switches gives you interactive shell access into the container
b. apt update: update the package database inside the container.
c. apt install nodejs: install Node.js



d. node -v: When the installation finishes, verify that Node.js is installed



Step 5 — Managing Docker Containers

a. docker ps: to see active (running) and inactive containers on the computer. In this case, active.
b. docker ps -a: To view all containers — active and inactive, run docker ps with the -a switch
c. docker ps -l: To view the latest container you created, pass it the -l switch:
d. docker start ID: To start a stopped container, use docker start, followed by the container ID or the container's name.
e. docker stop container ID or name: To stop a running container, use docker stop, followed by the container ID or name

```
me-it@meit:~/ubuntu-in-docker$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
me-it@meit:~/ubuntu-in-docker$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED          STATUS                      PORTS     NAMES
f767ec4c5cfc   ubuntu        "/bin/bash"   7 minutes ago    Exited (0) About a minute ago           crazy_khorana
dcfbaa61a2e7   hello-world   "/hello"      8 minutes ago    Exited (0) 8 minutes ago                affectionate_lovelace
192f471e3875   hello-world   "/hello"      47 minutes ago   Exited (0) 47 minutes ago               funny_yalow
me-it@meit:~/ubuntu-in-docker$ sudo docker ps -l
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                      PORTS     NAMES
f767ec4c5cfc   ubuntu    "/bin/bash"   7 minutes ago   Exited (0) About a minute ago           crazy_khorana
me-it@meit:~/ubuntu-in-docker$ sudo docker start f767ec4c5cfc
f767ec4c5cfc
me-it@meit:~/ubuntu-in-docker$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
f767ec4c5cfc   ubuntu    "/bin/bash"   9 minutes ago   Up 29 seconds            crazy_khorana
me-it@meit:~/ubuntu-in-docker$ sudo docker stop crazy_khorana
crazy_khorana
```

Step 6 — Committing Changes in a Container to a Docker Image

a. docker commit -m "added Node.js" -a "savani" d9b100f2f636 savani/ubuntu-nodejs: commit the changes to a new Docker image instance using the command.
b. docker images: Listing the Docker images again will show the new image, as well as the old one that it was derived.

```
me-it@meit:~/ubuntu-in-docker$ sudo docker commit -m "added Node.js" -a "Savani" f767ec4c5cfc Savani/ubuntu-nodejs
invalid reference format: repository name must be lowercase
me-it@meit:~/ubuntu-in-docker$ sudo docker commit -m "added Node.js" -a "savani" f767ec4c5cfc savani/ubuntu-nodejs
sha256:7fb660f6112b93b50fe699840beb0eb6c9997f1ff288677ee847bb8b3709f5b5
me-it@meit:~/ubuntu-in-docker$ sudo docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
savani/ubuntu-nodejs    latest    7fb660f6112b   36 seconds ago   203MB
ubuntu                  latest    58db3edaf2be   4 weeks ago      77.8MB
hello-world             latest    feb5d9fea6a5   17 months ago    13.3kB
me-it@meit:~/ubuntu-in-docker$
```

Step 7 — Pushing Docker Images to a Docker Repository

a. docker login -u docker-registry-username: To push your image, first log into Docker Hub.
b. docker tag sammy/ubuntu-nodejs docker-registry-username/ubuntu-nodejs : If your Docker registry username is different from the local username you used to create the image, you will have to tag your image with your registry username.
c. docker push savani/ubuntu-nodejs: To push the ubuntu-nodejs image to the savani repository.

```
me-it@meit:~/ubuntu-in-docker$ sudo docker login -u docker-registry-username
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
me-it@meit:~/ubuntu-in-docker$ docker tag savani/ubuntu-nodejs docker-registry-username/ubuntu-nodejs
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/savani/ubuntu
-nodejs/tag?repo=docker-registry-username%2Fubuntu-nodejs&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
me-it@meit:~/ubuntu-in-docker$ sudo docker tag savani/ubuntu-nodejs docker-registry-username/ubuntu-nodejs
me-it@meit:~/ubuntu-in-docker$ sudo docker push savani/ubuntu-nodejs
Using default tag: latest
The push refers to repository [docker.io/savani/ubuntu-nodejs]
2fcf4c7e44f4: Preparing
c5ff2d88f679: Preparing
denied: requested access to the resource is denied
me-it@meit:~/ubuntu-in-docker$
```