BRAC
UNIVERSITY

Inspiring Excellence

CSE-491 Cloud Computing
Assignment- 3

Light Container

Submitted By:

Mehedi Hasan

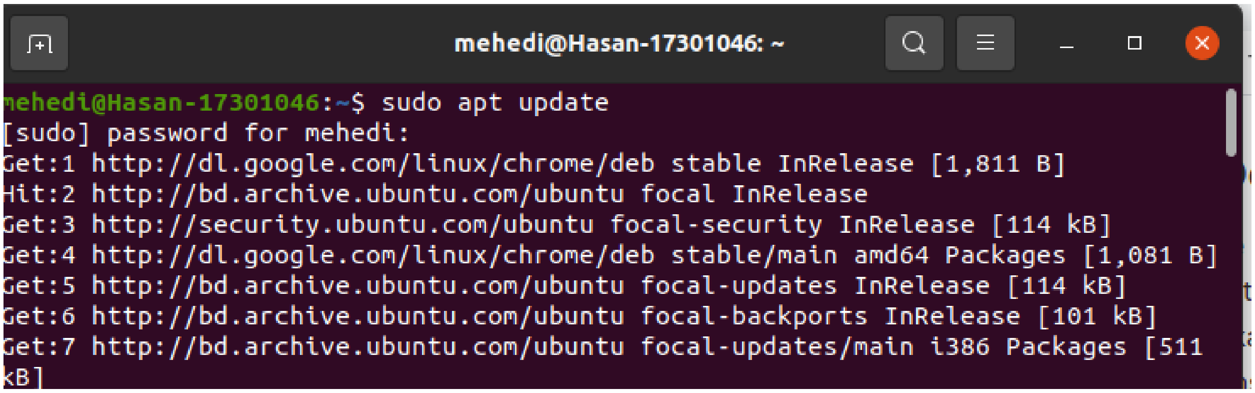
Id: 17301046

Submitted To,

Jannatun Noor Mukta

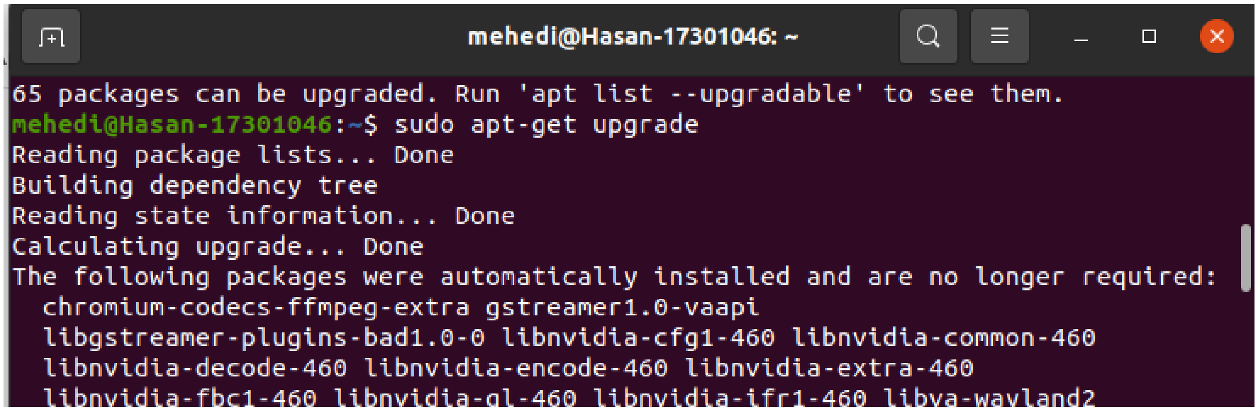
1. Install docker

Before installing any new packages it's important to update and upgrade existing packages. That will help the new package to run smoothly. So here i am updating and upgrading my packages



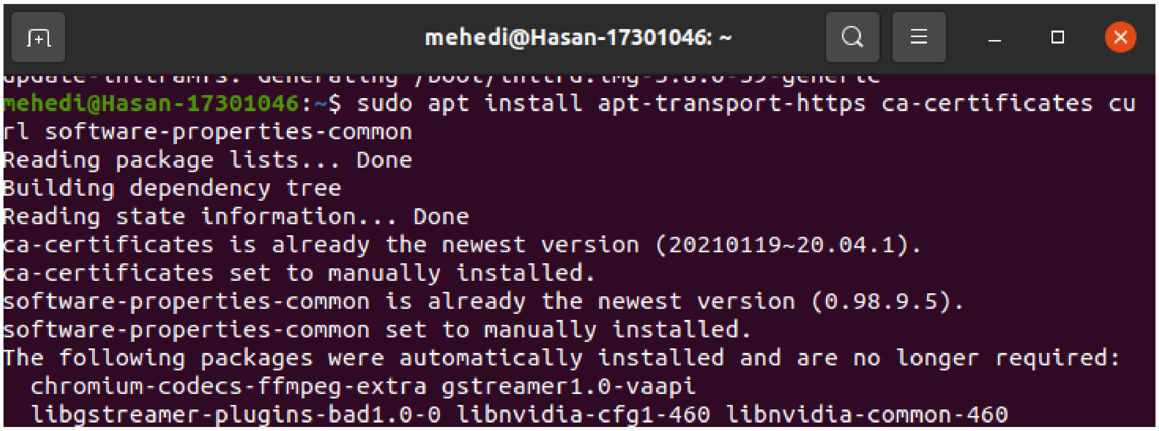
```
mehedi@Hasan-17301046:~$ sudo apt update
[sudo] password for mehedi:
Get:1 http://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Hit:2 http://bd.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,081 B]
Get:5 http://bd.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://bd.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:7 http://bd.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [511
kB]
```

Here upgrading the packages



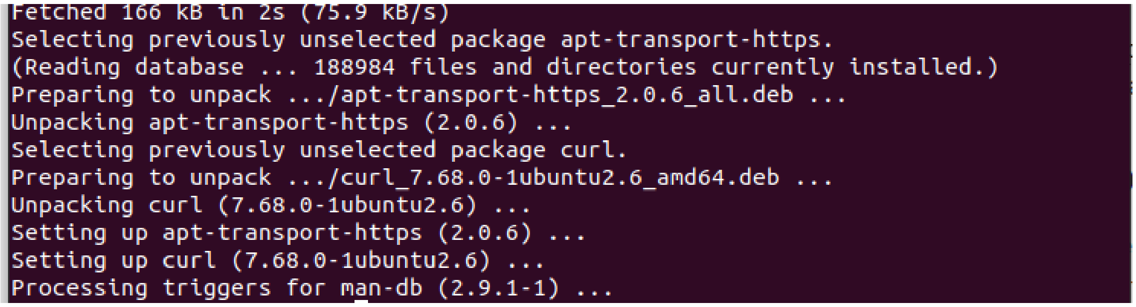
```
65 packages can be upgraded. Run 'apt list --upgradable' to see them.
mehedi@Hasan-17301046:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libnvidia-cfg1-460 libnvidia-common-460
  libnvidia-decode-460 libnvidia-encode-460 libnvidia-extra-460
  libnvidia-fbc1-460 libnvidia-gl-460 libnvidia-ifr1-460 libva-wayland2
```

Before installing Docker i need to install some of the packages which will help me to run docker smoothly.



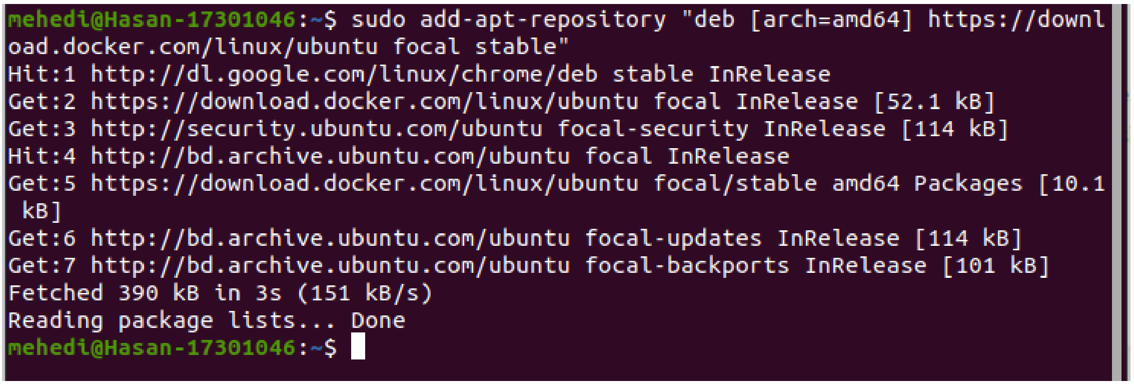
```
update-initramfs: Generating /boot/initrd.img-5.8.0-55-generic
mehedi@Hasan-17301046:~$ sudo apt install apt-transport-https ca-certificates cu
rl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20210119~20.04.1).
ca-certificates set to manually installed.
software-properties-common is already the newest version (0.98.9.5).
software-properties-common set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libnvidia-cfg1-460 libnvidia-common-460
```

Continue Package installing



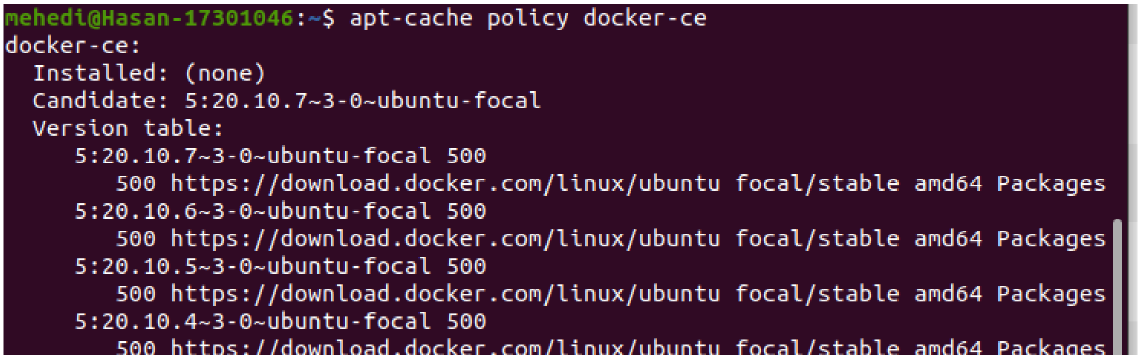
```
Fetched 166 kB in 2s (75.9 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 188984 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.6_all.deb ...
Unpacking apt-transport-https (2.0.6) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.6_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.6) ...
Setting up apt-transport-https (2.0.6) ...
Setting up curl (7.68.0-1ubuntu2.6) ...
Processing triggers for man-db (2.9.1-1) ...
```

After installing the packages we need to add a Docker repository in the APT sources file.



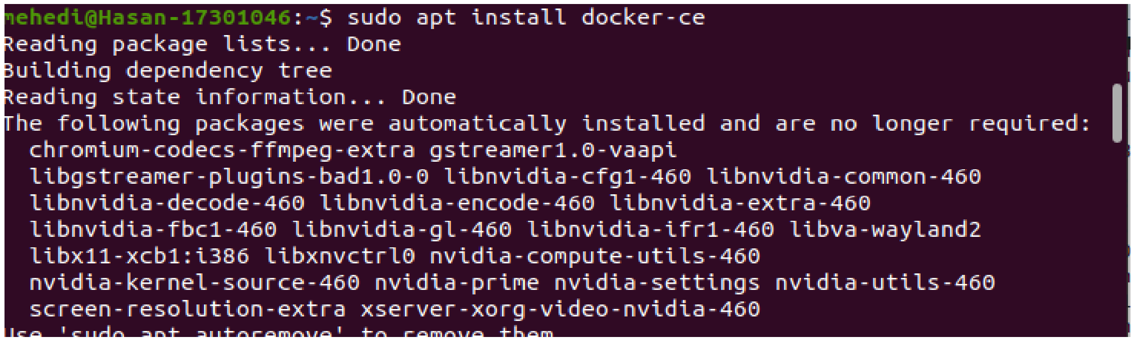
```
mehedi@Hasan-17301046:~$ sudo add-apt-repository "deb [arch=amd64] https://downl
oad.docker.com/linux/ubuntu focal stable"
Hit:1 http://dl.google.com/linux/chrome/deb stable InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [52.1 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:4 http://bd.archive.ubuntu.com/ubuntu focal InRelease
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [10.1
 kB]
Get:6 http://bd.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:7 http://bd.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Fetched 390 kB in 3s (151 kB/s)
Reading package lists... Done
mehedi@Hasan-17301046:~$ 
```

If a repository is added successfully after that I need to update my packages and add a caching policy. Which will install docker from the docker repository that I added in my previous command.



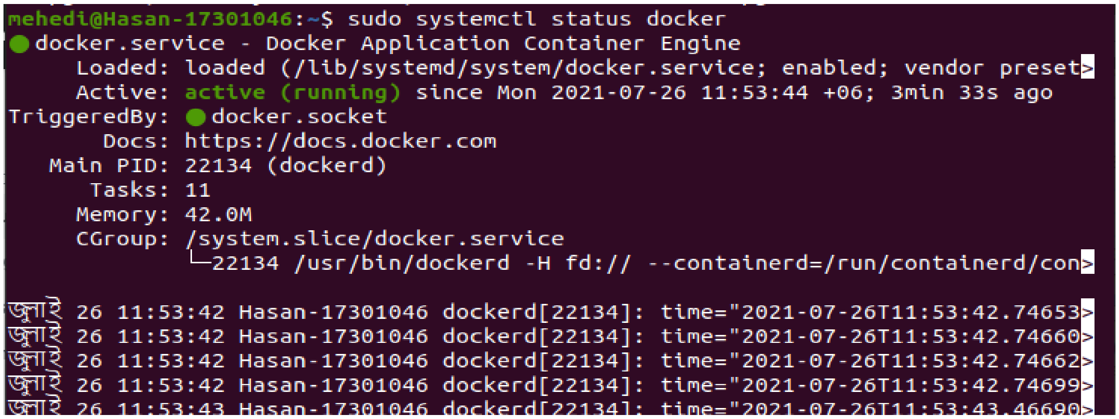
```
mehedi@Hasan-17301046:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.7~3-0~ubuntu-focal
  Version table:
     5:20.10.7~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.6~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.5~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.4~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

Docker repository connected and now its time to install docker.



```
mehedi@Hasan-17301046:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libnvidia-cfg1-460 libnvidia-common-460
  libnvidia-decode-460 libnvidia-encode-460 libnvidia-extra-460
  libnvidia-fbc1-460 libnvidia-gl-460 libnvidia-ifr1-460 libva-wayland2
  libx11-xcb1:i386 libxnvctrl0 nvidia-compute-utils-460
  nvidia-kernel-source-460 nvidia-prime nvidia-settings nvidia-utils-460
  screen-resolution-extra xserver-xorg-video-nvidia-460
Use 'sudo apt autoremove' to remove them.
```

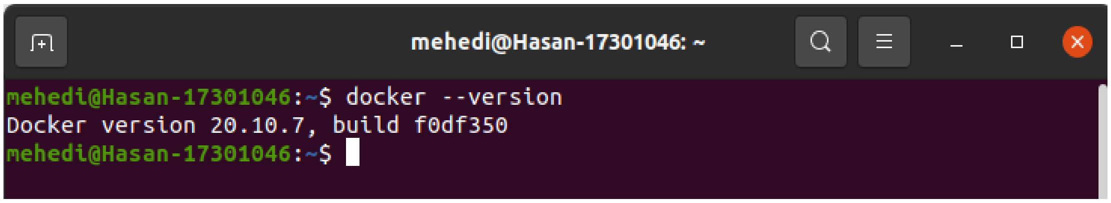
Docker installation done Successfully. Now its time to check the status of docker. Sudo systemctl will show me the docker status. If it is installed successfully it will show Active with a green logo. If not it will show inactive.



```
mehedi@Hasan-17301046:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Mon 2021-07-26 11:53:44 +06; 3min 33s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 22134 (dockerd)
      Tasks: 11
     Memory: 42.0M
     CGroup: /system.slice/docker.service
             └─22134 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>

জুলাই 26 11:53:42 Hasan-17301046 dockerd[22134]: time="2021-07-26T11:53:42.74653>
জুলাই 26 11:53:42 Hasan-17301046 dockerd[22134]: time="2021-07-26T11:53:42.74660>
জুলাই 26 11:53:42 Hasan-17301046 dockerd[22134]: time="2021-07-26T11:53:42.74662>
জুলাই 26 11:53:42 Hasan-17301046 dockerd[22134]: time="2021-07-26T11:53:42.74699>
জুলাই 26 11:53:43 Hasan-17301046 dockerd[22134]: time="2021-07-26T11:53:43.46690>
```

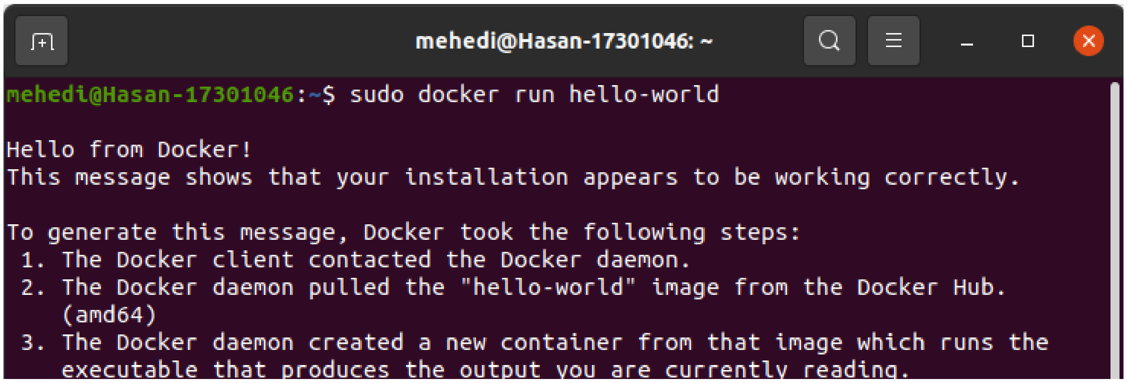
Part-2 : Show outputs of basic Docker commands (i.e pull, search, run, build, commit, rm, rmi, etc.. find more from google)

Docker Installation done. Now I can execute some of the basic docker commands. First i can check the docker version. By executing docker –version



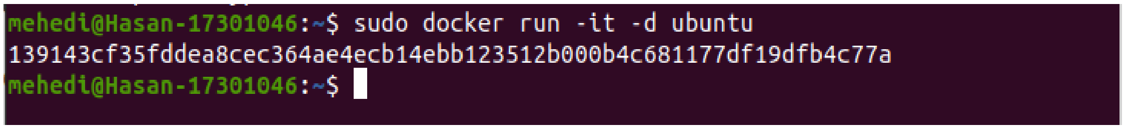
```
mehedi@Hasan-17301046:~$ docker --version
Docker version 20.10.7, build f0df350
mehedi@Hasan-17301046:~$ 
```

Docker run hello-world. It will directly installed from the docker repository and show me the output of hello docker. If my docker installation is right it will show me the output.



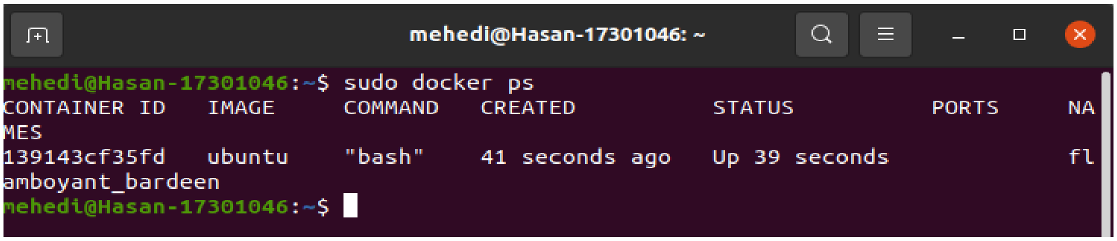
```
mehedi@Hasan-17301046:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
```

Docker run will run a container. Here i am running ubuntu container in the docker



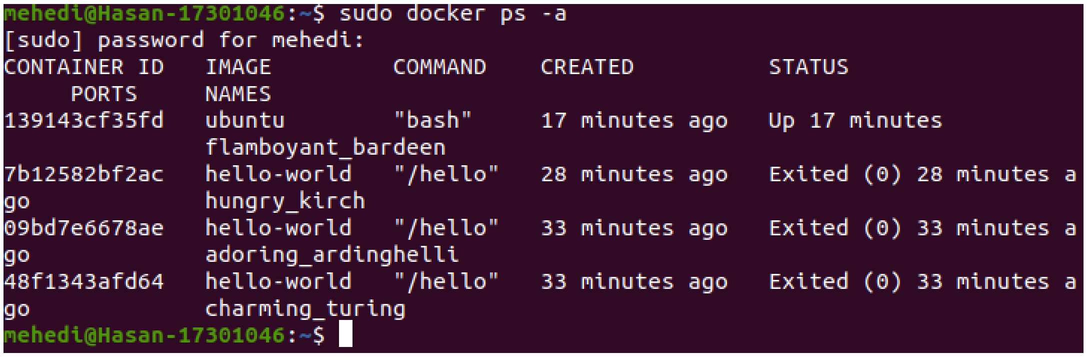
```
mehedi@Hasan-17301046:~$ sudo docker run -it -d ubuntu
139143cf35fddea8cec364ae4ecb14ebb123512b000b4c681177df19dfb4c77a
mehedi@Hasan-17301046:~$ 
```

Sudo docker ps will show me the running container in my docker.



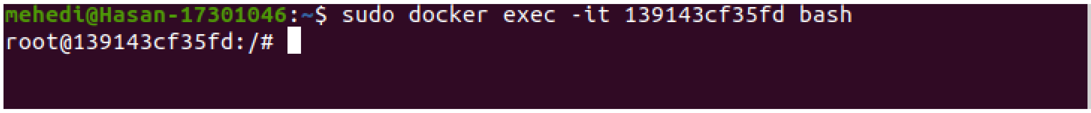
```
mehedi@Hasan-17301046:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED          STATUS          PORTS     NA
MES
139143cf35fd   ubuntu    "bash"    41 seconds ago   Up 39 seconds             fl
amboyant_bardeen
mehedi@Hasan-17301046:~$ 
```

Sudo docker ps -a will show me all the available container in my docker



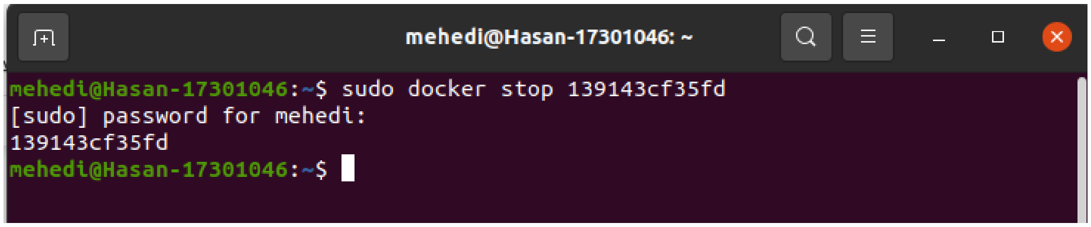
```
mehedi@Hasan-17301046:~$ sudo docker ps -a
[sudo] password for mehedi:
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS
     PORTS     NAMES
139143cf35fd   ubuntu        "bash"     17 minutes ago   Up 17 minutes
               flamboyant_bardeen
7b12582bf2ac   hello-world   "/hello"   28 minutes ago   Exited (0) 28 minutes a
go             hungry_kirch
09bd7e6678ae   hello-world   "/hello"   33 minutes ago   Exited (0) 33 minutes a
go             adoring_ardinghelli
48f1343afd64   hello-world   "/hello"   33 minutes ago   Exited (0) 33 minutes a
go             charming_turing
mehedi@Hasan-17301046:~$ 
```

If i want to use any container as a root i can, just need to **exec** with the **container id**



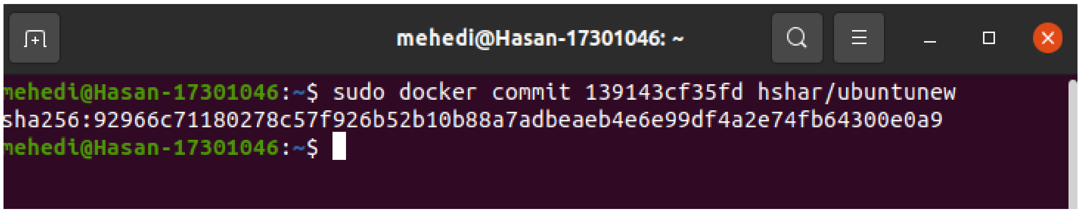
```
mehedi@Hasan-17301046:~$ sudo docker exec -it 139143cf35fd bash
root@139143cf35fd:/# 
```

If i want i can stop a docker container. By docker stop **container-id**



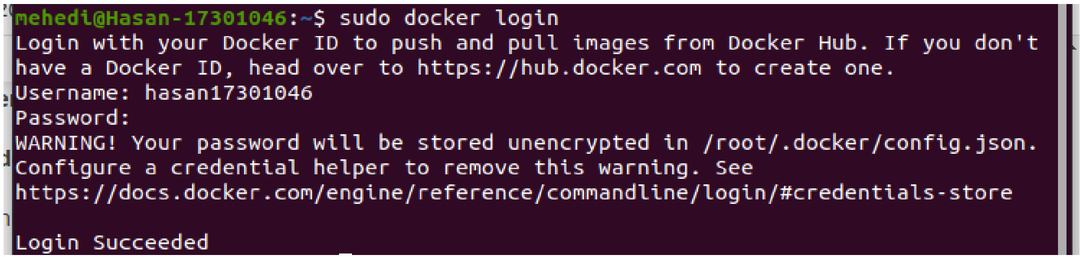
```
mehedi@Hasan-17301046:~$ sudo docker stop 139143cf35fd
[sudo] password for mehedi:
139143cf35fd
mehedi@Hasan-17301046:~$ 
```

Docker commit will create new images of an edited container. **Commit container-id**



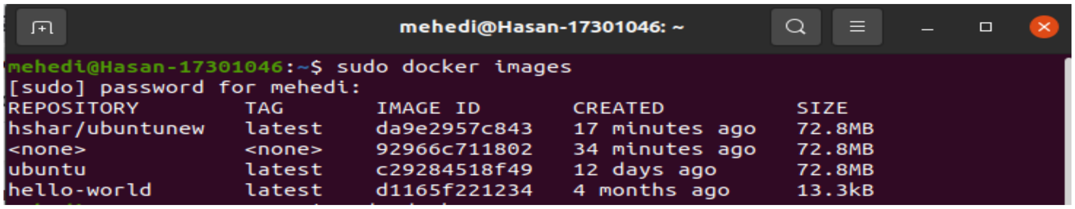
```
mehedi@Hasan-17301046:~$ sudo docker commit 139143cf35fd hshar/ubuntunew
sha256:92966c71180278c57f926b52b10b88a7adbeaeb4e6e99df4a2e74fb64300e0a9
mehedi@Hasan-17301046:~$ 
```

If i want i can login my docker account in the cli. Docker login will give me the options to login



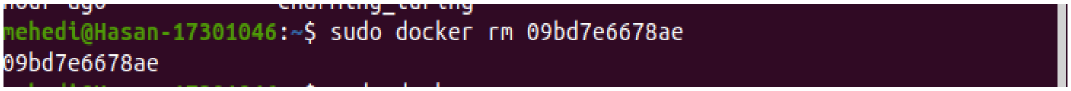
```
mehedi@Hasan-17301046:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: hasan17301046
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Docker Images will show me the available images in my Docker.



```
mehedi@Hasan-17301046:~$ sudo docker images
[sudo] password for mehedi:
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
hshar/ubuntunew     latest    da9e2957c843   17 minutes ago   72.8MB
<none>              <none>    92966c711802   34 minutes ago   72.8MB
ubuntu              latest    c29284518f49   12 days ago      72.8MB
hello-world         latest    d1165f221234   4 months ago     13.3kB
```

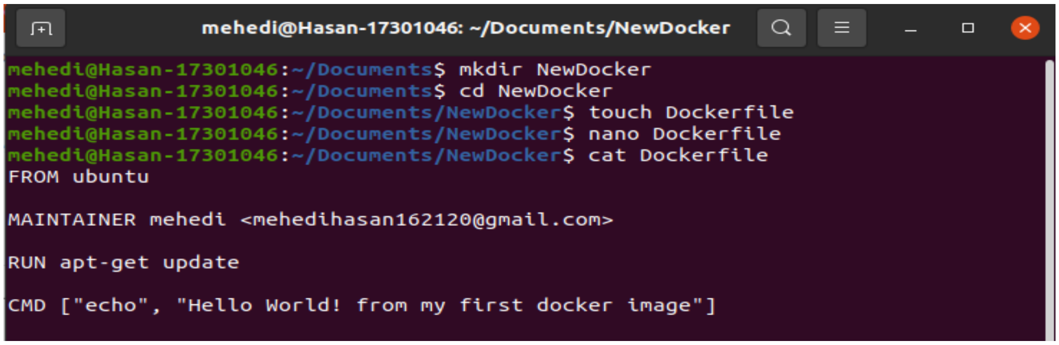
If i want i can remove a container



```
mehedi@Hasan-17301046:~$ sudo docker rm 09bd7e6678ae
09bd7e6678ae
```

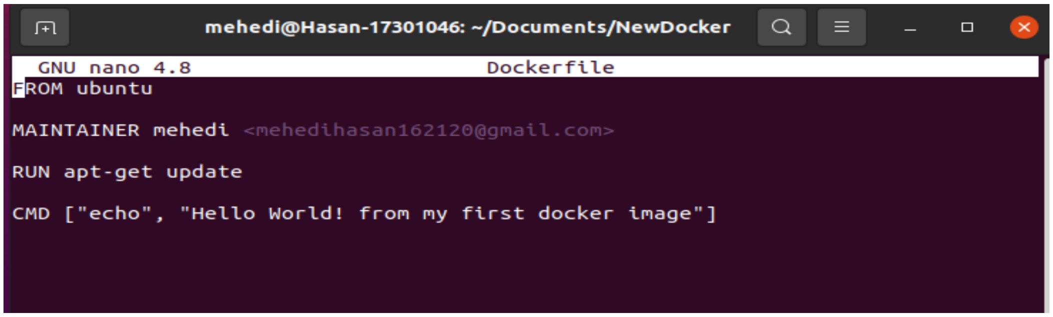
Part-3: Create a Docker image using Dockerfile.

First i need to create a directory and in that directory i need to create a file name **dockerfile**



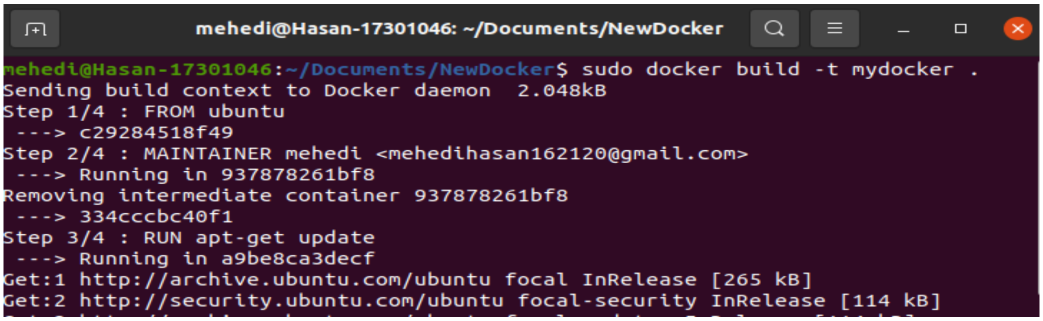
```
mehedi@Hasan-17301046:~/Documents$ mkdir NewDocker
mehedi@Hasan-17301046:~/Documents$ cd NewDocker
mehedi@Hasan-17301046:~/Documents/NewDocker$ touch Dockerfile
mehedi@Hasan-17301046:~/Documents/NewDocker$ nano Dockerfile
mehedi@Hasan-17301046:~/Documents/NewDocker$ cat Dockerfile
FROM ubuntu

MAINTAINER mehedi <mehedihasan162120@gmail.com>

RUN apt-get update

CMD ["echo", "Hello World! from my first docker image"]
```

Nano dockerfile will give me the editor option in cli. Here i can save any command and can run that command in my docker.



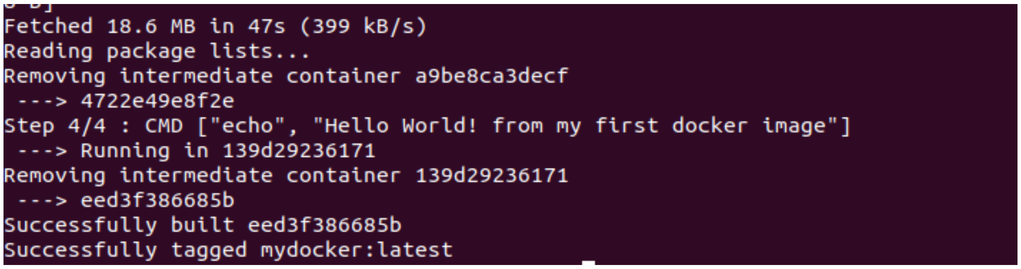
```
  GNU nano 4.8                           Dockerfile
FROM ubuntu

MAINTAINER mehedi <mehedihasan162120@gmail.com>

RUN apt-get update

CMD ["echo", "Hello World! from my first docker image"]
```

Docker build will build that docker



```
mehedi@Hasan-17301046:~/Documents/NewDocker$ sudo docker build -t mydocker .
Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM ubuntu
 ---> c29284518f49
Step 2/4 : MAINTAINER mehedi <mehedihasan162120@gmail.com>
 ---> Running in 937878261bf8
Removing intermediate container 937878261bf8
 ---> 334cccbc40f1
Step 3/4 : RUN apt-get update
 ---> Running in a9be8ca3decf
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
```

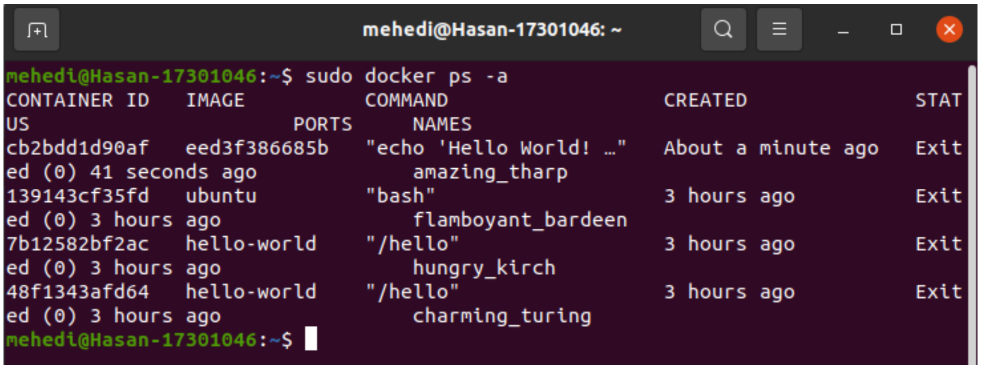
Docker image build done



```
Fetched 18.6 MB in 47s (399 kB/s)
Reading package lists...
Removing intermediate container a9be8ca3decf
 ---> 4722e49e8f2e
Step 4/4 : CMD ["echo", "Hello World! from my first docker image"]
 ---> Running in 139d29236171
Removing intermediate container 139d29236171
 ---> eed3f386685b
Successfully built eed3f386685b
Successfully tagged mydocker:latest
```

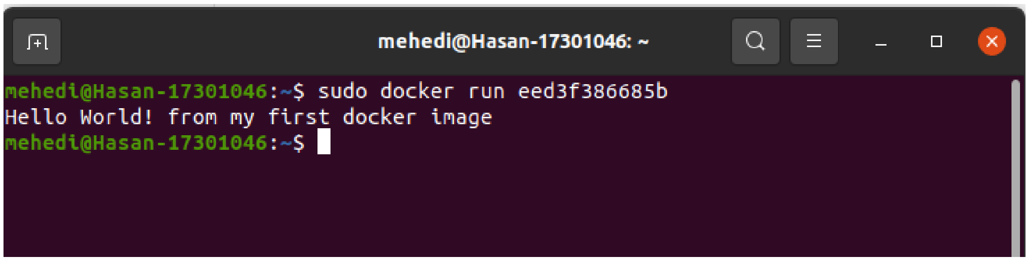
Part-4: Run a container as a single task, show outputs, show status of the all containers (using docker ps -a)

Docker ps -a will show me all the available containers in my docker. From that list i can choose anyone to run.



```
mehedi@Hasan-17301046:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STAT
US                       PORTS     NAMES
cb2bdd1d90af   eed3f386685b   "echo 'Hello World! …"   About a minute ago   Exit
ed (0) 41 seconds ago              amazing_tharp
139143cf35fd   ubuntu         "bash"                   3 hours ago          Exit
ed (0) 3 hours ago                 flamboyant_bardeen
7b12582bf2ac   hello-world    "/hello"                 3 hours ago          Exit
ed (0) 3 hours ago                 hungry_kirch
48f1343afd64   hello-world    "/hello"                 3 hours ago          Exit
ed (0) 3 hours ago                 charming_turing
mehedi@Hasan-17301046:~$
```

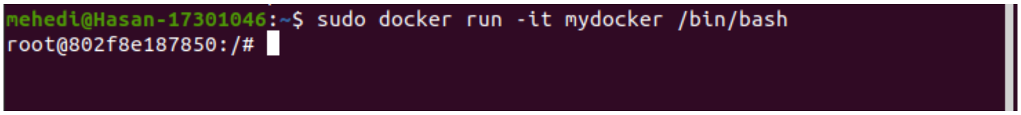
Docker run **container-id** it will run the container. Here I am run the hello-world container.



```
mehedi@Hasan-17301046:~$ sudo docker run eed3f386685b
Hello World! from my first docker image
mehedi@Hasan-17301046:~$
```

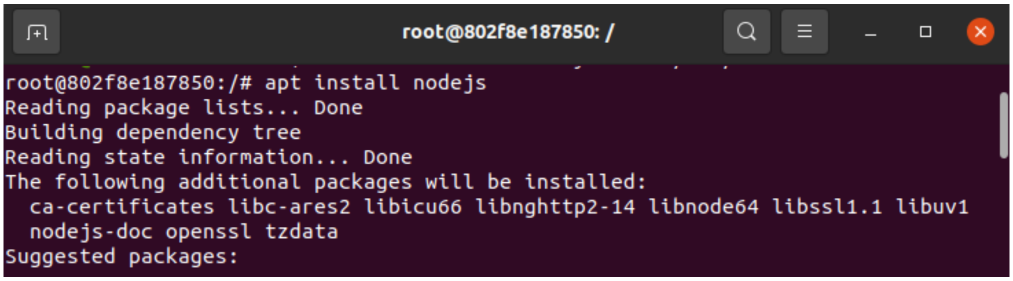
Task-5: Run a container in iterative mode and install different packages in the container. Show each step.

First I run docker, which is mydocker that I created some time ago. In that container I need to run in root user mode. As i need to install some of the packages. So root user is required.



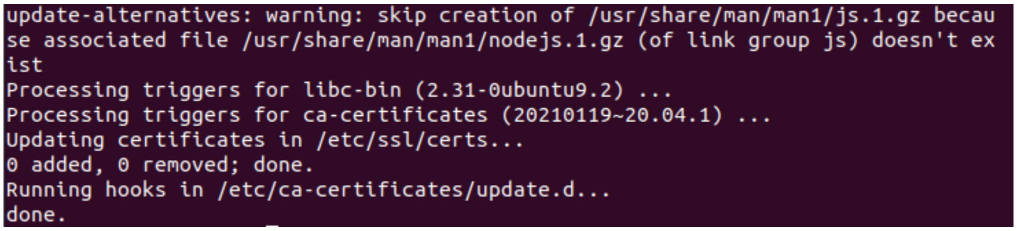
```
mehedi@Hasan-17301046:~$ sudo docker run -it mydocker /bin/bash
root@802f8e187850:/#
```

Now in that docker i installed **node.js** packages



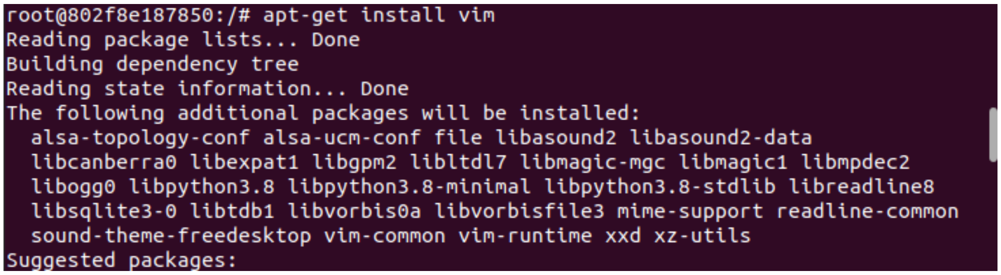
```
root@802f8e187850:/# apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates libc-ares2 libicu66 libnghttp2-14 libnode64 libssl1.1 libuv1
  nodejs-doc openssl tzdata
Suggested packages:
```

Node js Installation Done successfully.



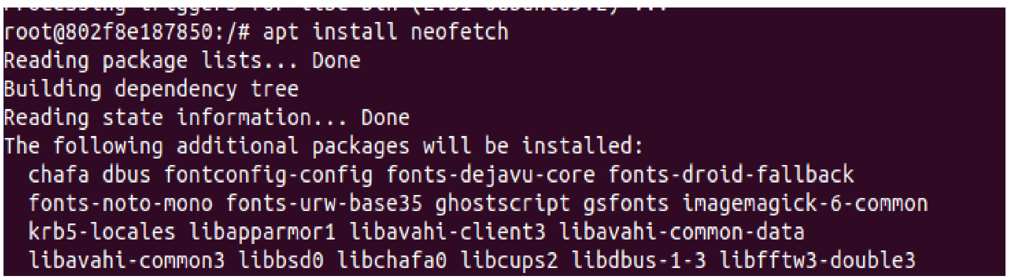
```
update-alternatives: warning: skip creation of /usr/share/man/man1/js.1.gz becau
se associated file /usr/share/man/man1/nodejs.1.gz (of link group js) doesn't ex
ist
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for ca-certificates (20210119~20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

Now installing Vim packages in my docker container



```
root@802f8e187850:/# apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf file libasound2 libasound2-data
  libcanberra0 libexpat1 libgpm2 libltdl7 libmagic-mgc libmagic1 libmpdec2
  libogg0 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libreadline8
  libsqlite3-0 libtdb1 libvorbis0a libvorbisfile3 mime-support readline-common
  sound-theme-freedesktop vim-common vim-runtime xxd xz-utils
Suggested packages:
```

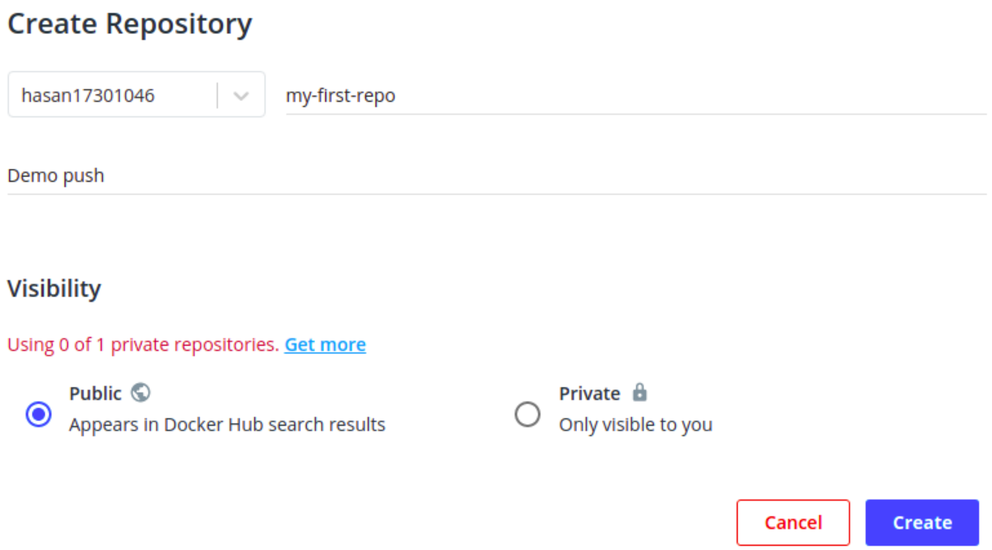
And finally installing neofetch packages in my container.



```
root@802f8e187850:/# apt install neofetch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  chafa dbus fontconfig-config fonts-dejavu-core fonts-droid-fallback
  fonts-noto-mono fonts-urw-base35 ghostscript gsfonts imagemagick-6-common
  krb5-locales libapparmor1 libavahi-client3 libavahi-common-data
  libavahi-common3 libbsd0 libchafa0 libcups2 libdbus-1-3 libfftw3-double3
```

Task-6: Push your own image into Docker public registry/Hub.

First i need to create an account in hub.docker.com with my email and password. After that I need to create a new repository. And in that repository i give a name which is my-first repo



Create Repository

hasan17301046 | my-first-repo

Demo push

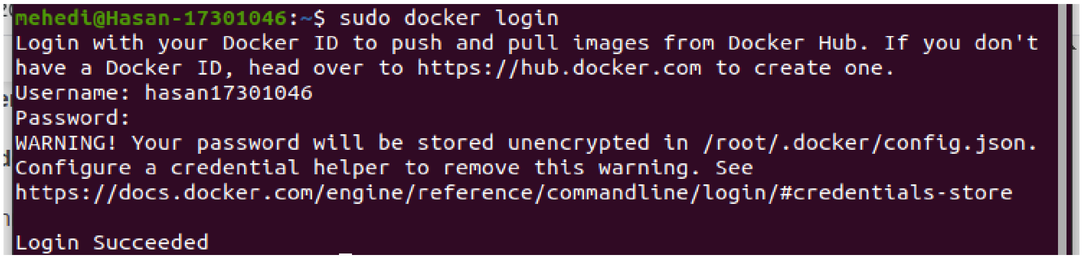
Visibility

Using 0 of 1 private repositories. Get more

● Public
Appears in Docker Hub search results

○ Private
Only visible to you

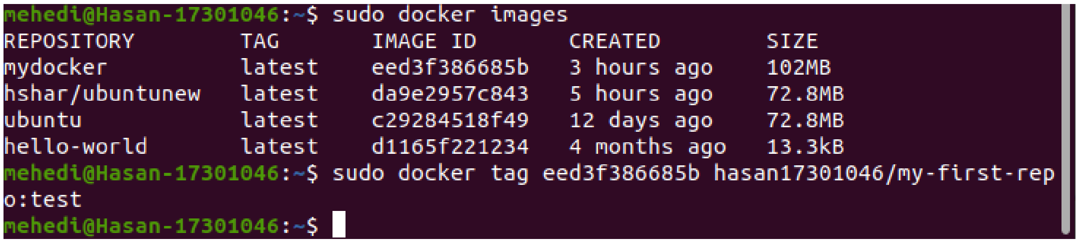
Now in CLI i need to run docker login. Which will login my docker account.



```
mehedi@Hasan-17301046:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: hasan17301046
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
```

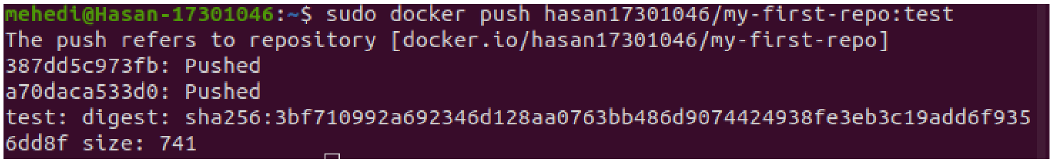
If login is successful I can see the successful msg in cli. Now i need to push my docker image in my public repository.

Docker images will show me all the available images there. And **docker tag “image-id” reponame:”name” will tag my docker image to my docker repo.**



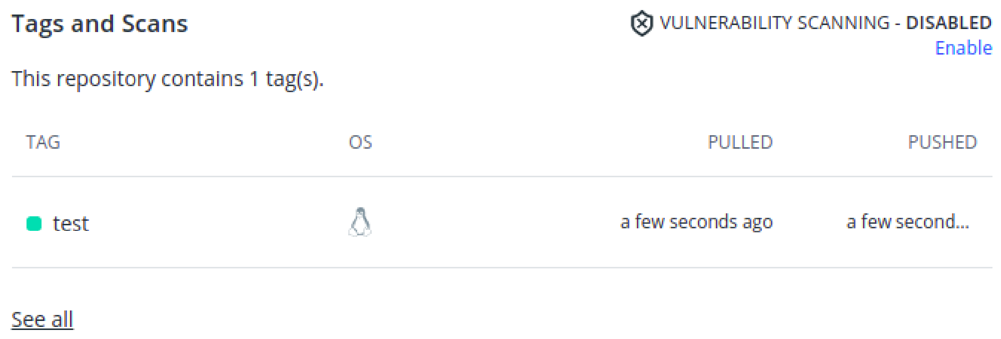
```
mehedi@Hasan-17301046:~$ sudo docker images
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
mydocker            latest    eed3f386685b   3 hours ago     102MB
hshar/ubuntunew     latest    da9e2957c843   5 hours ago     72.8MB
ubuntu              latest    c29284518f49   12 days ago     72.8MB
hello-world         latest    d1165f221234   4 months ago    13.3kB
mehedi@Hasan-17301046:~$ sudo docker tag eed3f386685b hasan17301046/my-first-rep
o:test
mehedi@Hasan-17301046:~$ 
```

Docker Push will push my image in the repo



```
mehedi@Hasan-17301046:~$ sudo docker push hasan17301046/my-first-repo:test
The push refers to repository [docker.io/hasan17301046/my-first-repo]
387dd5c973fb: Pushed
a70daca533d0: Pushed
test: digest: sha256:3bf710992a692346d128aa0763bb486d9074424938fe3eb3c19add6f935
6dd8f size: 741
```

Here in my hub.docker.com I can see that my test docker image is successfully pushed.



Tags and Scans

VULNERABILITY SCANNING - **DISABLED**
Enable

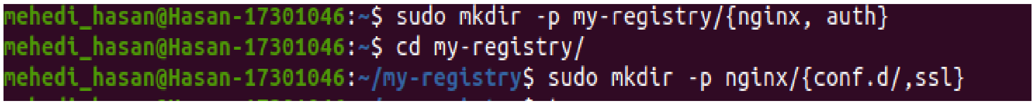
This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
🟩 test		a few seconds ago	a few second...

See all

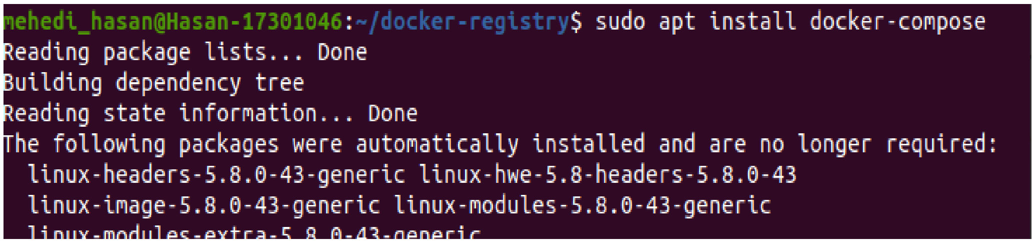
Task-7: How to make your own private registry? Show steps.

First I need to create some directory. So I will execute mkdir to create some of the folders.



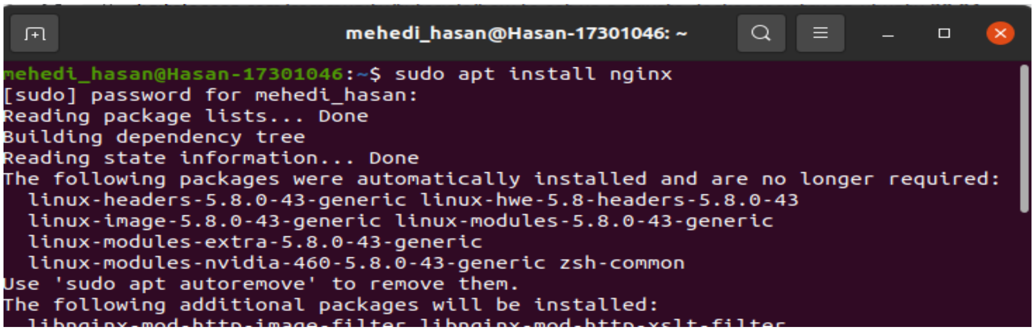
```
mehedi_hasan@Hasan-17301046:~$ sudo mkdir -p my-registry/{nginx, auth}
mehedi_hasan@Hasan-17301046:~$ cd my-registry/
mehedi_hasan@Hasan-17301046:~/my-registry$ sudo mkdir -p nginx/{conf.d/,ssl}
```

Now i need to edit docker-compose. For that i need to install **docker-compose**



```
mehedi_hasan@Hasan-17301046:~/docker-registry$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43
  linux-image-5.8.0-43-generic linux-modules-5.8.0-43-generic
  linux-modules-extra-5.8.0-43-generic
```

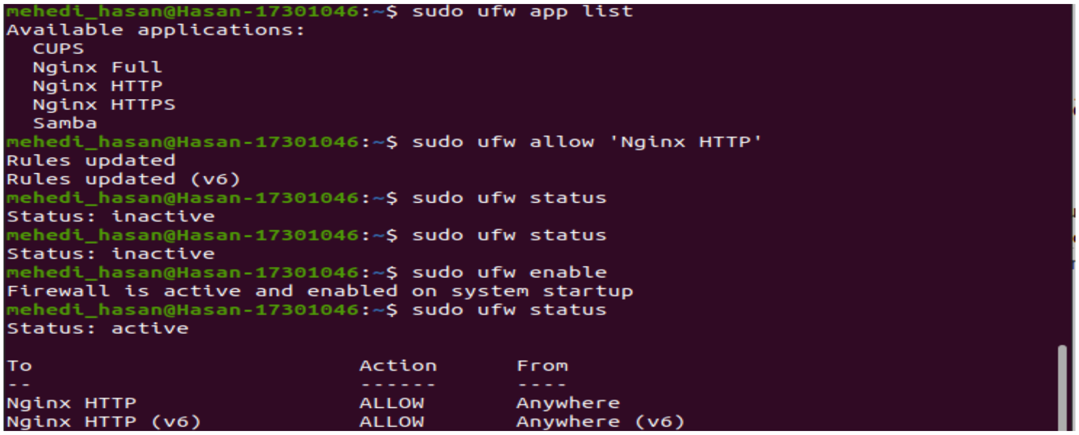
After that I need to install nginx packages. It is the https packages. By using nginx I can manage some of the ports. That port I need to execute or run my docker container.



mehedi_hasan@Hasan-17301046: ~

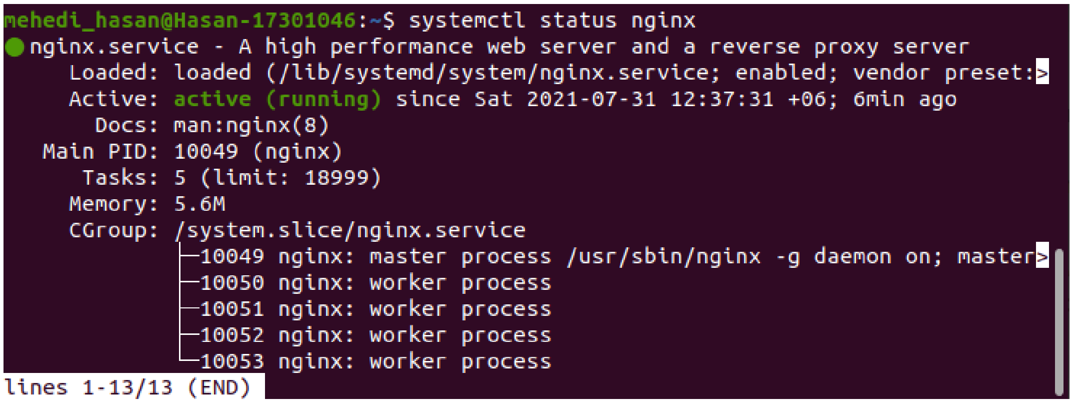
```
mehedi_hasan@Hasan-17301046:~$ sudo apt install nginx
[sudo] password for mehedi_hasan:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43
  linux-image-5.8.0-43-generic linux-modules-5.8.0-43-generic
  linux-modules-extra-5.8.0-43-generic
  linux-modules-nvidia-460-5.8.0-43-generic zsh-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
```

After installing the nginx I need to activate it.



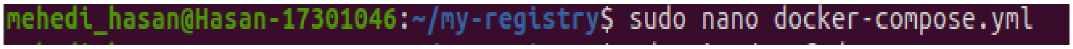
```
mehedi_hasan@Hasan-17301046:~$ sudo ufw app list
Available applications:
  CUPS
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  Samba
mehedi_hasan@Hasan-17301046:~$ sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
mehedi_hasan@Hasan-17301046:~$ sudo ufw status
Status: inactive
mehedi_hasan@Hasan-17301046:~$ sudo ufw status
Status: inactive
mehedi_hasan@Hasan-17301046:~$ sudo ufw enable
Firewall is active and enabled on system startup
mehedi_hasan@Hasan-17301046:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Nginx HTTP                 ALLOW       Anywhere
Nginx HTTP (v6)            ALLOW       Anywhere (v6)
```

Systemctl status nginx will show me the status of nginx



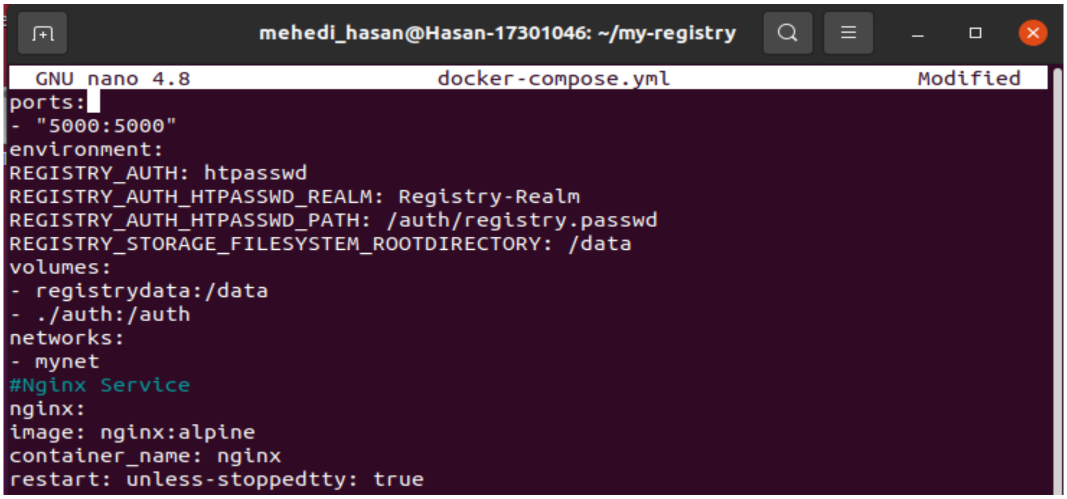
```
mehedi_hasan@Hasan-17301046:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:>
     Active: active (running) since Sat 2021-07-31 12:37:31 +06; 6min ago
       Docs: man:nginx(8)
   Main PID: 10049 (nginx)
      Tasks: 5 (limit: 18999)
     Memory: 5.6M
     CGroup: /system.slice/nginx.service
             ├─10049 nginx: master process /usr/sbin/nginx -g daemon on; master>
             ├─10050 nginx: worker process
             ├─10051 nginx: worker process
             ├─10052 nginx: worker process
             └─10053 nginx: worker process
lines 1-13/13 (END)
```

Now i need to edit the docker-compose.yml



```
mehedi_hasan@Hasan-17301046:~/my-registry$ sudo nano docker-compose.yml
```

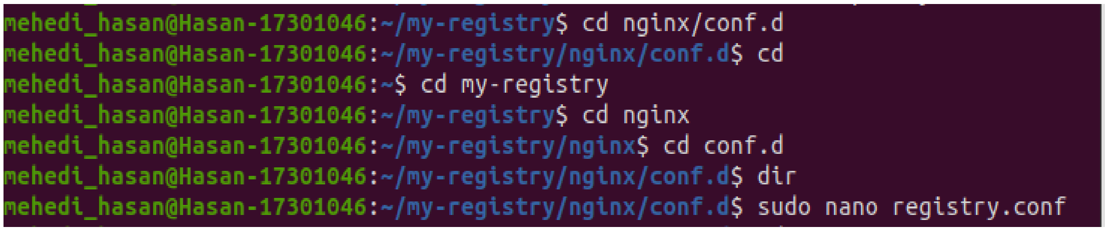
This full ports and nginx i need to add in my docker-compose



```
mehedi_hasan@Hasan-17301046: ~/my-registry
  GNU nano 4.8                    docker-compose.yml                    Modified
ports:
- "5000:5000"
environment:
REGISTRY_AUTH: htpasswd
REGISTRY_AUTH_HTPASSWD_REALM: Registry-Realm
REGISTRY_AUTH_HTPASSWD_PATH: /auth/registry.passwd
REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY: /data
volumes:
- registrydata:/data
- ./auth:/auth
networks:
- mynet
#Nginx Service
nginx:
image: nginx:alpine
container_name: nginx
restart: unless-stoppedtty: true
```

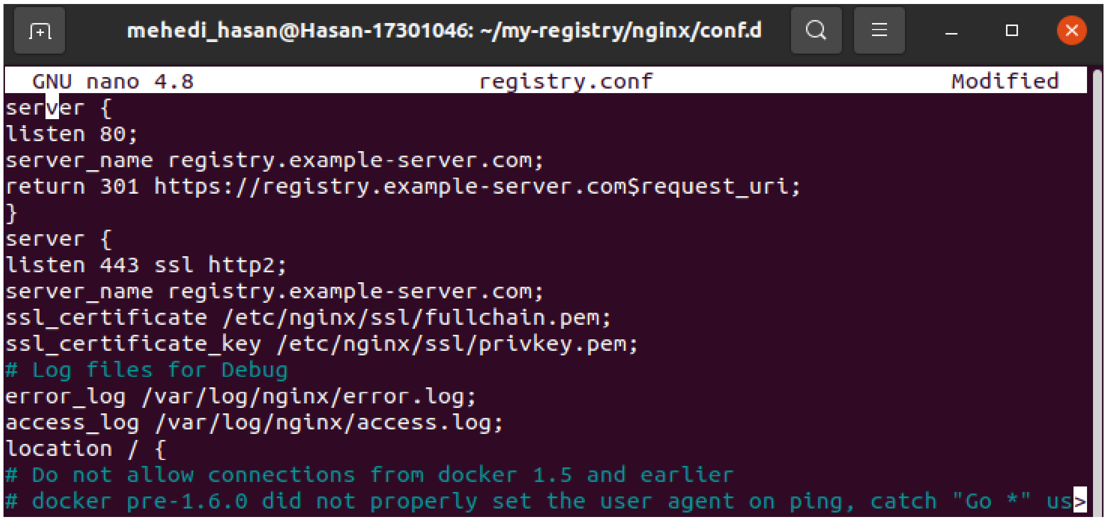
Now ctrl x and save that to my docker-compose

Now need to edit **registry.conf** also



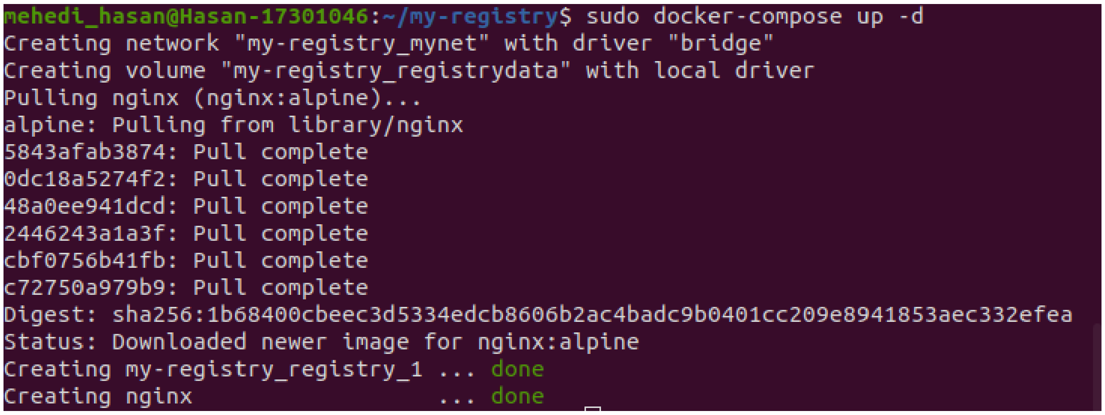
```
mehedi_hasan@Hasan-17301046:~/my-registry$ cd nginx/conf.d
mehedi_hasan@Hasan-17301046:~/my-registry/nginx/conf.d$ cd
mehedi_hasan@Hasan-17301046:~$ cd my-registry
mehedi_hasan@Hasan-17301046:~/my-registry$ cd nginx
mehedi_hasan@Hasan-17301046:~/my-registry/nginx$ cd conf.d
mehedi_hasan@Hasan-17301046:~/my-registry/nginx/conf.d$ dir
mehedi_hasan@Hasan-17301046:~/my-registry/nginx/conf.d$ sudo nano registry.conf
```

In the registry i need to assign server port for the registry



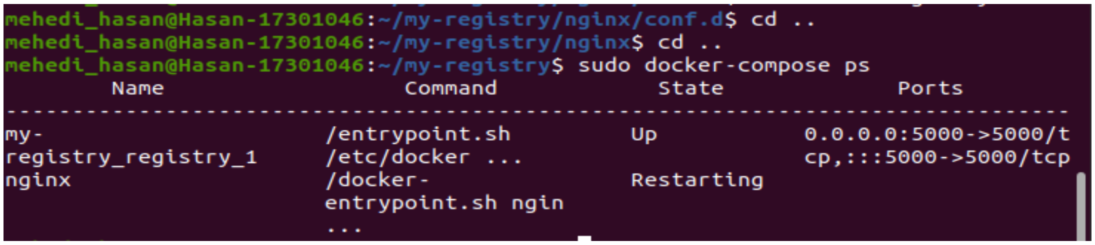
```
  GNU nano 4.8                    registry.conf                         Modified
server {
listen 80;
server_name registry.example-server.com;
return 301 https://registry.example-server.com$request_uri;
}
server {
listen 443 ssl http2;
server_name registry.example-server.com;
ssl_certificate /etc/nginx/ssl/fullchain.pem;
ssl_certificate_key /etc/nginx/ssl/privkey.pem;
# Log files for Debug
error_log /var/log/nginx/error.log;
access_log /var/log/nginx/access.log;
location / {
# Do not allow connections from docker 1.5 and earlier
# docker pre-1.6.0 did not properly set the user agent on ping, catch "Go *" us>
```

After edited the docker-compose i need to up it to locally



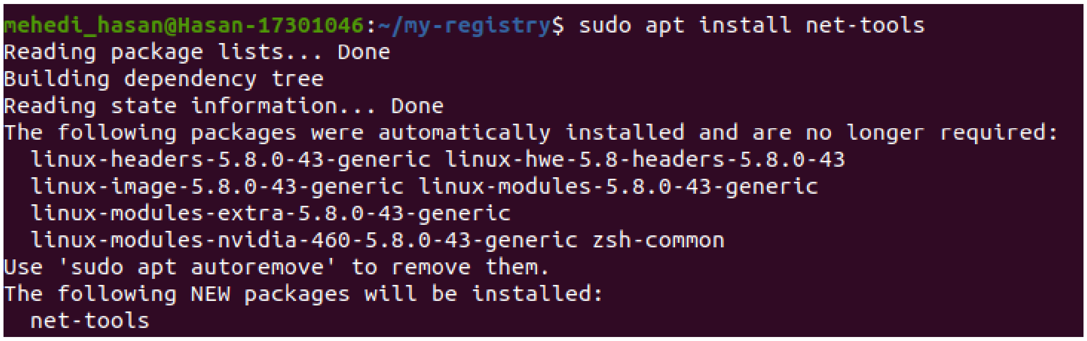
```
mehedi_hasan@Hasan-17301046:~/my-registry$ sudo docker-compose up -d
Creating network "my-registry_mynet" with driver "bridge"
Creating volume "my-registry_registrydata" with local driver
Pulling nginx (nginx:alpine)...
alpine: Pulling from library/nginx
5843afab3874: Pull complete
0dc18a5274f2: Pull complete
48a0ee941dcd: Pull complete
2446243a1a3f: Pull complete
cbf0756b41fb: Pull complete
c72750a979b9: Pull complete
Digest: sha256:1b68400cbeec3d5334edcb8606b2ac4badc9b0401cc209e8941853aec332efea
Status: Downloaded newer image for nginx:alpine
Creating my-registry_registry_1 ... done
Creating nginx                  ... done
```

In **docker-compose ps** i can see the port that are running right now



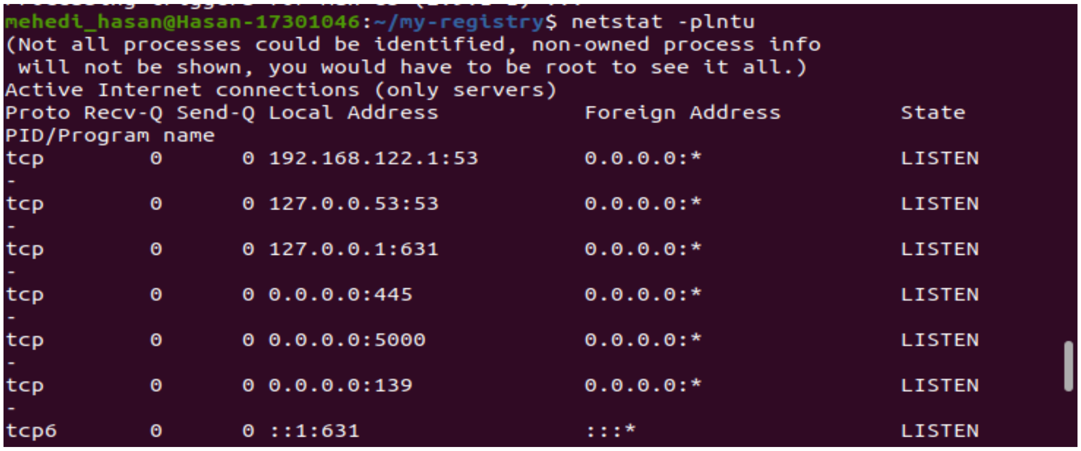
```
mehedi_hasan@Hasan-17301046:~/my-registry/nginx/conf.d$ cd ..
mehedi_hasan@Hasan-17301046:~/my-registry/nginx$ cd ..
mehedi_hasan@Hasan-17301046:~/my-registry$ sudo docker-compose ps
        Name                      Command               State              Ports
--------------------------------------------------------------------------------
my-                        /entrypoint.sh          Up           0.0.0.0:5000->5000/t
registry_registry_1        /etc/docker ...                      cp,:::5000->5000/tcp
nginx                      /docker-                Restarting
                           entrypoint.sh ngin
                           ...
```

Now to check the port and registry i need to install net-tools



```
mehedi_hasan@Hasan-17301046:~/my-registry$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43
  linux-image-5.8.0-43-generic linux-modules-5.8.0-43-generic
  linux-modules-extra-5.8.0-43-generic
  linux-modules-nvidia-460-5.8.0-43-generic zsh-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
```

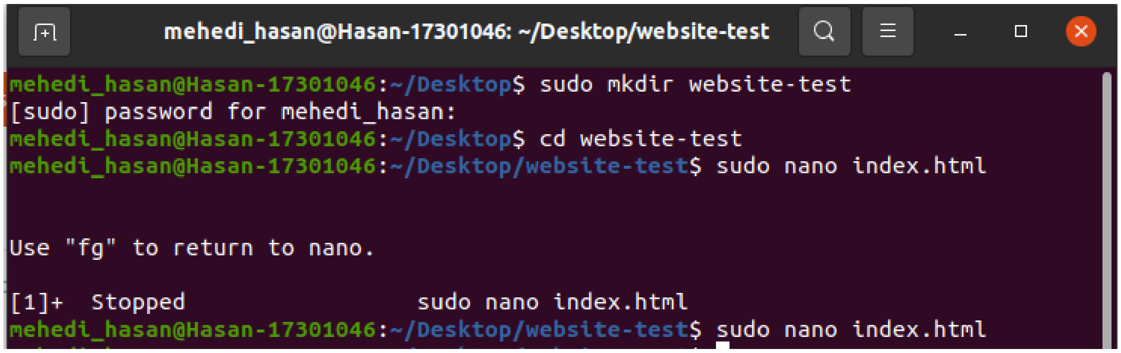
And after that **netstat -plntu** will show me the private registry



```
mehedi_hasan@Hasan-17301046:~/my-registry$ netstat -plntu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN
-
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

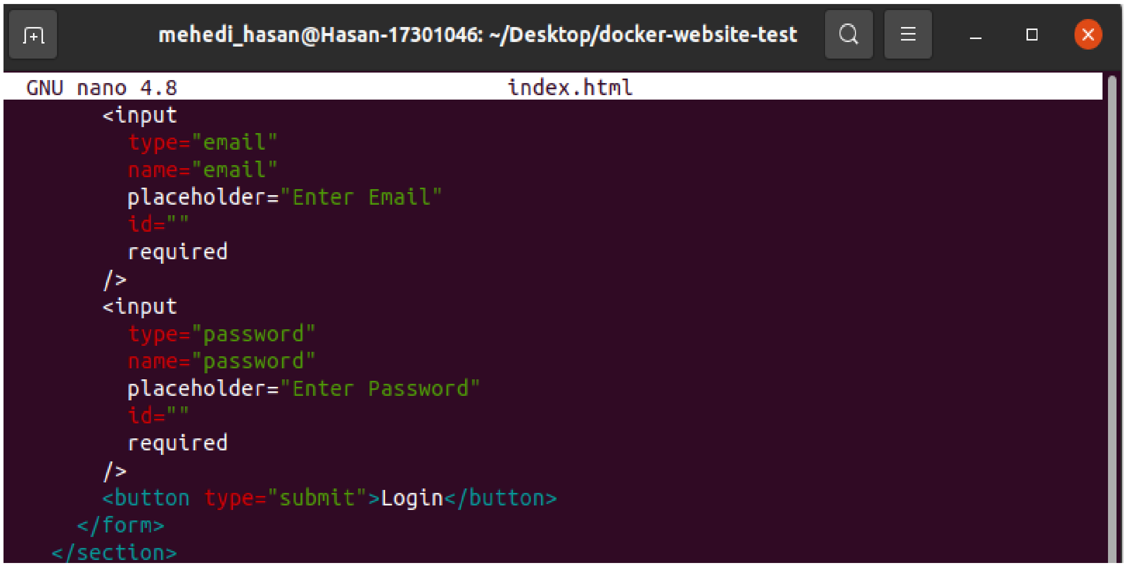
Task-8: Create a small website or app with minimal functionality (Could be a simple HTML website that has a button which opens a static image/file) inside Docker container. Then run the application (inside the container) in the background of your HOST machine in any port. Browse the website from your host machine.

First I need to create a directory to create and execute a web server.
And nano index.html will give me the option to add my html file

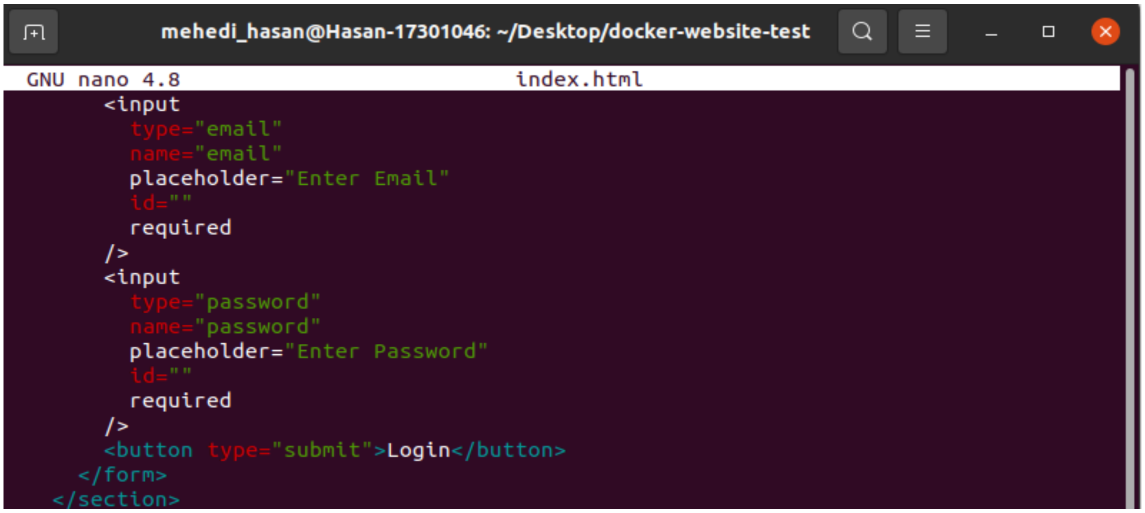


```
mehedi_hasan@Hasan-17301046:~/Desktop$ sudo mkdir website-test
[sudo] password for mehedi_hasan:
mehedi_hasan@Hasan-17301046:~/Desktop$ cd website-test
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ sudo nano index.html


Use "fg" to return to nano.

[1]+  Stopped                 sudo nano index.html
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ sudo nano index.html
```

In index.html i add my html code for web server

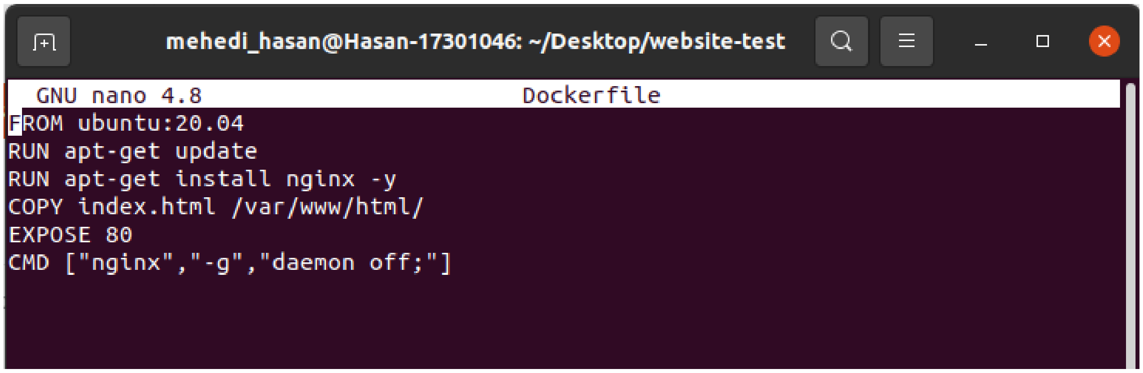


```
  GNU nano 4.8                     index.html
    <input
     type="email"
     name="email"
     placeholder="Enter Email"
     id=""
     required
    />
    <input
     type="password"
     name="password"
     placeholder="Enter Password"
     id=""
     required
    />
    <button type="submit">Login</button>
   </form>
  </section>
```



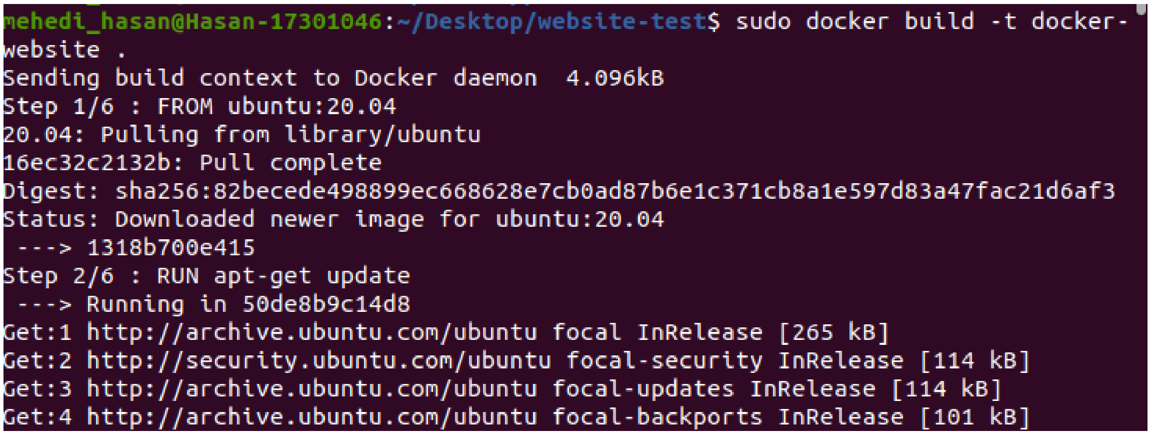
```
  GNU nano 4.8                                    index.html
        <input
          type="email"
          name="email"
          placeholder="Enter Email"
          id=""
          required
        />
        <input
          type="password"
          name="password"
          placeholder="Enter Password"
          id=""
          required
        />
        <button type="submit">Login</button>
      </form>
    </section>
```

Now i need to edit the **Dockerfile** to run my web server and some command . Here i assign the **port and nginx** for **HTTP**



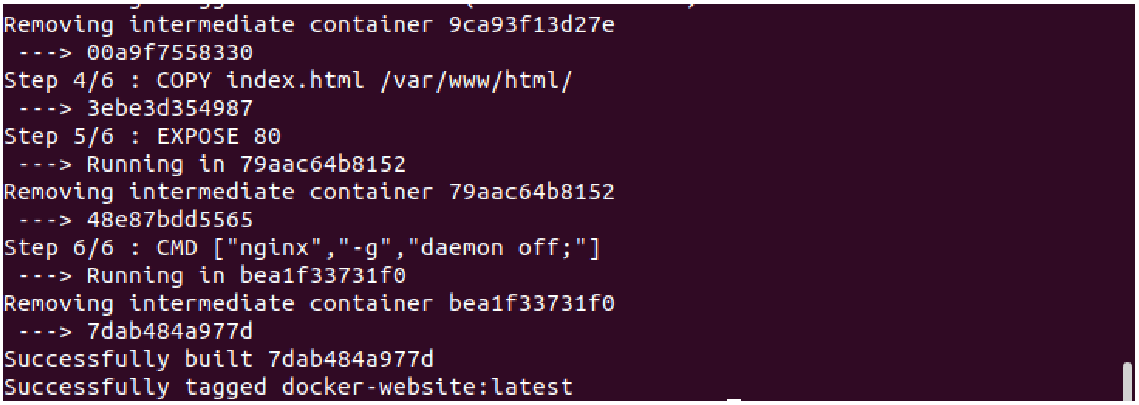
```
  GNU nano 4.8                                    Dockerfile
FROM ubuntu:20.04
RUN apt-get update
RUN apt-get install nginx -y
COPY index.html /var/www/html/
EXPOSE 80
CMD ["nginx","-g","daemon off;"]
```

Now I need to **build my docker-website.** It will build my website.



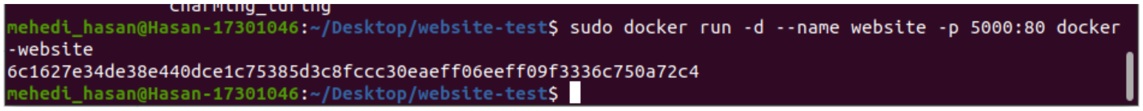
```
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ sudo docker build -t docker-
website .
Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM ubuntu:20.04
20.04: Pulling from library/ubuntu
16ec32c2132b: Pull complete
Digest: sha256:82becede498899ec668628e7cb0ad87b6e1c371cb8a1e597d83a47fac21d6af3
Status: Downloaded newer image for ubuntu:20.04
 ---> 1318b700e415
Step 2/6 : RUN apt-get update
 ---> Running in 50de8b9c14d8
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
```

Docker build processing running



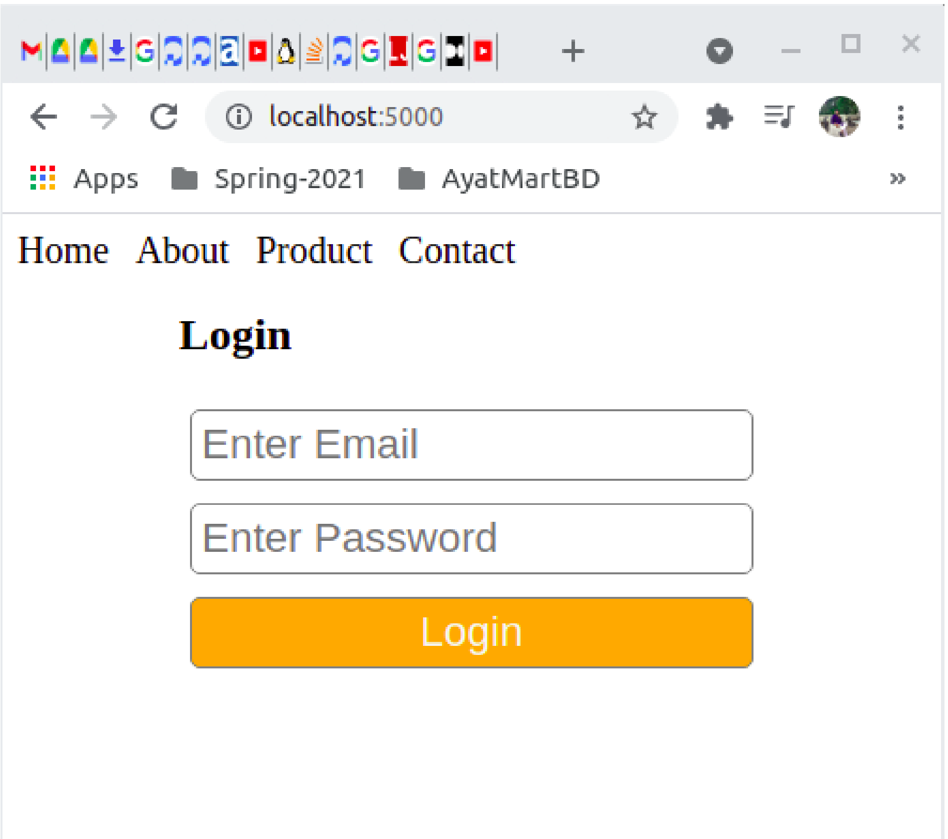
```
Removing intermediate container 9ca93f13d27e
 ---> 00a9f7558330
Step 4/6 : COPY index.html /var/www/html/
 ---> 3ebe3d354987
Step 5/6 : EXPOSE 80
 ---> Running in 79aac64b8152
Removing intermediate container 79aac64b8152
 ---> 48e87bdd5565
Step 6/6 : CMD ["nginx","-g","daemon off;"]
 ---> Running in bea1f33731f0
Removing intermediate container bea1f33731f0
 ---> 7dab484a977d
Successfully built 7dab484a977d
Successfully tagged docker-website:latest
```

Now i need to run that docker-website with the assign port of 5000



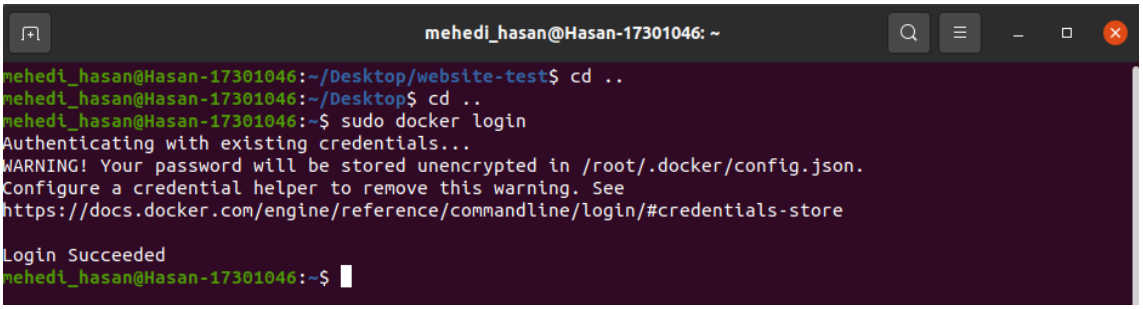
```
                    charming_turing
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ sudo docker run -d --name website -p 5000:80 docker
-website
6c1627e34de38e440dce1c75385d3c8fccc30eaeff06eeff09f3336c750a72c4
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ 
```

Now if i visit that address i can see my website is successfully hosted



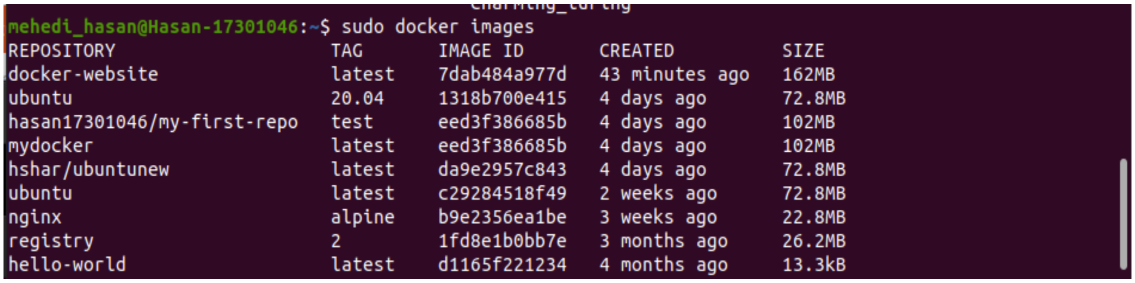
Part-9: Migrate the new container having the application into another machine. Again run the container and browse the URL. It should work.

To migrate my webster and docker container in docker repo. First i need to login to my docker server.



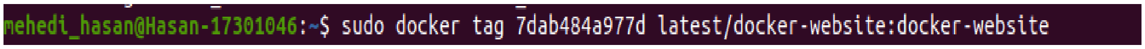
```
mehedi_hasan@Hasan-17301046:~/Desktop/website-test$ cd ..
mehedi_hasan@Hasan-17301046:~/Desktop$ cd ..
mehedi_hasan@Hasan-17301046:~$ sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
mehedi_hasan@Hasan-17301046:~$
```

After that i need to check my Docker-Images that i want to push it on my docker repository. Here i am going to push my **docker-website** in my repository.



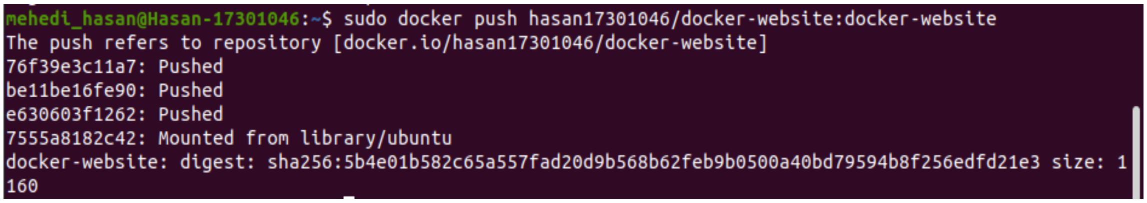
```
mehedi_hasan@Hasan-17301046:~$ sudo docker images
REPOSITORY                    TAG       IMAGE ID       CREATED          SIZE
docker-website                latest    7dab484a977d   43 minutes ago   162MB
ubuntu                        20.04     1318b700e415   4 days ago       72.8MB
hasan17301046/my-first-repo   test      eed3f386685b   4 days ago       102MB
mydocker                      latest    eed3f386685b   4 days ago       102MB
hshar/ubuntunew               latest    da9e2957c843   4 days ago       72.8MB
ubuntu                        latest    c29284518f49   2 weeks ago      72.8MB
nginx                         alpine    b9e2356ea1be   3 weeks ago      22.8MB
registry                      2         1fd8e1b0bb7e   3 months ago     26.2MB
hello-world                   latest    d1165f221234   4 months ago     13.3kB
```

Now i need to tag my **docker-images id** to my docker repository.



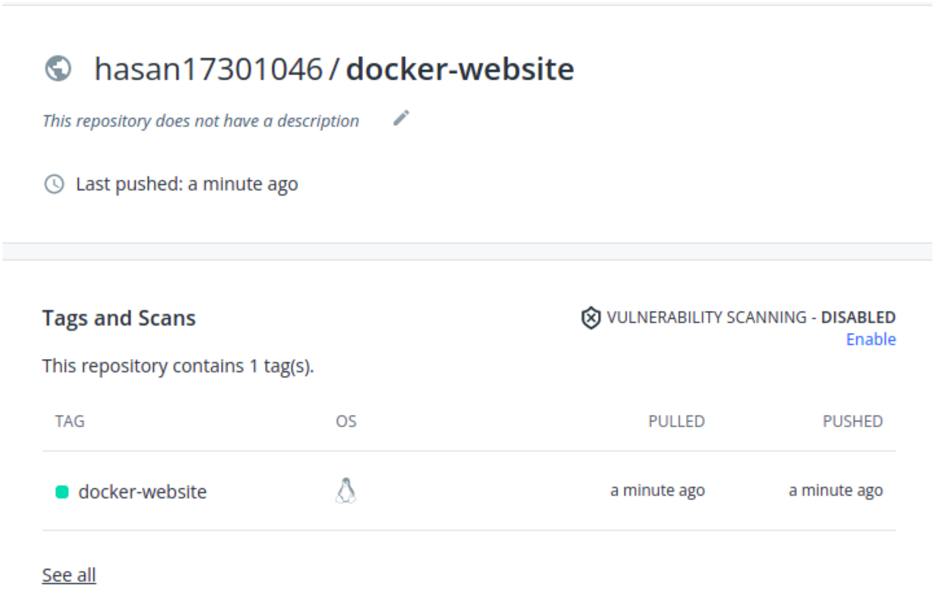
```
mehedi_hasan@Hasan-17301046:~$ sudo docker tag 7dab484a977d latest/docker-website:docker-website
```

After that i need to push my docker images in my docker repository.



```
mehedi_hasan@Hasan-17301046:~$ sudo docker push hasan17301046/docker-website:docker-website
The push refers to repository [docker.io/hasan17301046/docker-website]
76f39e3c11a7: Pushed
be11be16fe90: Pushed
e630603f1262: Pushed
7555a8182c42: Mounted from library/ubuntu
docker-website: digest: sha256:5b4e01b582c65a557fad20d9b568b62feb9b0500a40bd79594b8f256edfd21e3 size: 1
160
```

Now if i visit to my docker account i can see my web server docker container is successfully pushed in my docker repository



hasan17301046 / **docker-website**

This repository does not have a description

Last pushed: a minute ago

Tags and Scans

VULNERABILITY SCANNING - **DISABLED**
Enable

This repository contains 1 tag(s).

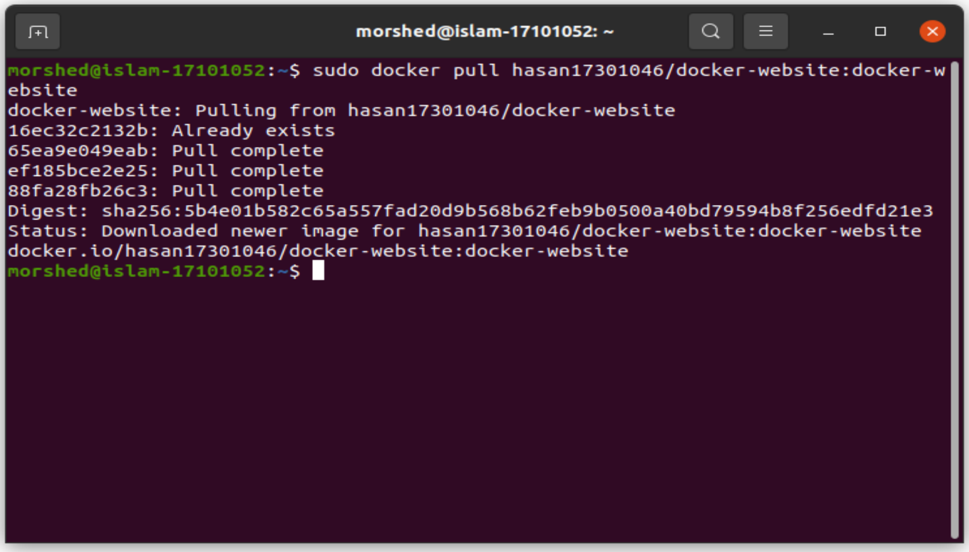
TAG	OS	PULLED	PUSHED
docker-website	🐧	a minute ago	a minute ago

See all

It's time to check my web server on my friend's pc.
My Friend Morshed help me to check my Migration of docker container web server

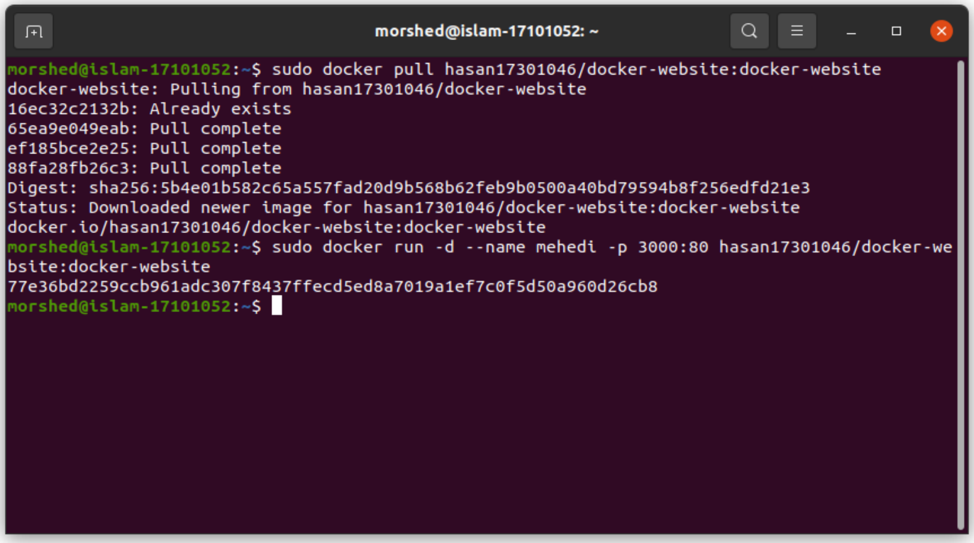
From My friend's Pc

First he pull the container in his pc



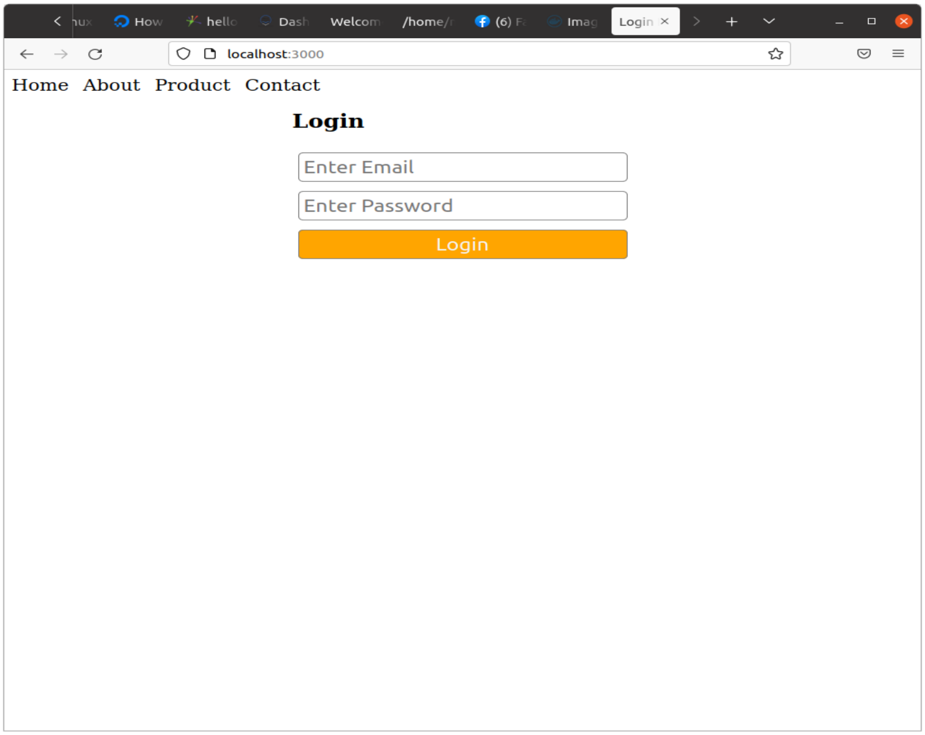
```
morshed@islam-17101052:~$ sudo docker pull hasan17301046/docker-website:docker-w
ebsite
docker-website: Pulling from hasan17301046/docker-website
16ec32c2132b: Already exists
65ea9e049eab: Pull complete
ef185bce2e25: Pull complete
88fa28fb26c3: Pull complete
Digest: sha256:5b4e01b582c65a557fad20d9b568b62feb9b0500a40bd79594b8f256edfd21e3
Status: Downloaded newer image for hasan17301046/docker-website:docker-website
docker.io/hasan17301046/docker-website:docker-website
morshed@islam-17101052:~$ 
```

After that run my docker container in their port



```
morshed@islam-17101052:~$ sudo docker pull hasan17301046/docker-website:docker-website
docker-website: Pulling from hasan17301046/docker-website
16ec32c2132b: Already exists
65ea9e049eab: Pull complete
ef185bce2e25: Pull complete
88fa28fb26c3: Pull complete
Digest: sha256:5b4e01b582c65a557fad20d9b568b62feb9b0500a40bd79594b8f256edfd21e3
Status: Downloaded newer image for hasan17301046/docker-website:docker-website
docker.io/hasan17301046/docker-website:docker-website
morshed@islam-17101052:~$ sudo docker run -d --name mehedi -p 3000:80 hasan17301046/docker-we
bsite:docker-website
77e36bd2259ccb961adc307f8437ffecd5ed8a7019a1ef7c0f5d50a960d26cb8
morshed@islam-17101052:~$ 
```

And Finally my Web server can access from my friend’s pc also



Enter Email

Enter Password

<-----The End----->