Light Container

by

Ishraq Ahmed Esha
19301261

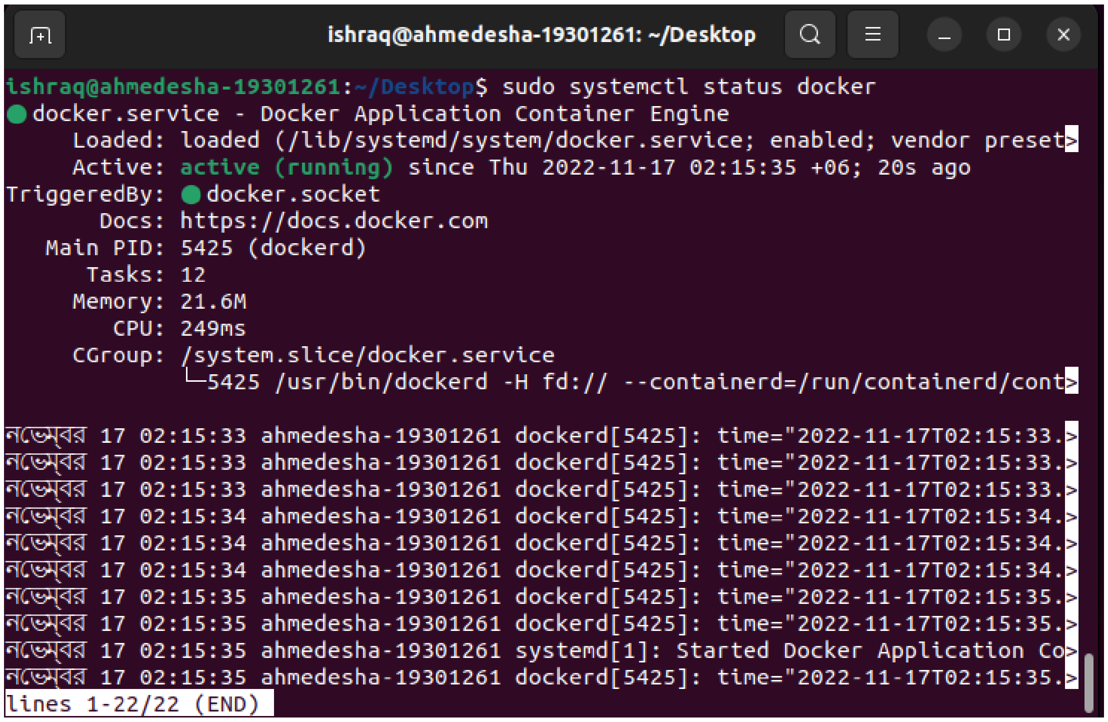
A Lab Assignment 3 submitted to the CSE484 Cloud Computing Course of Sec: 1

Brac University
November 17, 2022

Answer to the Question No 1

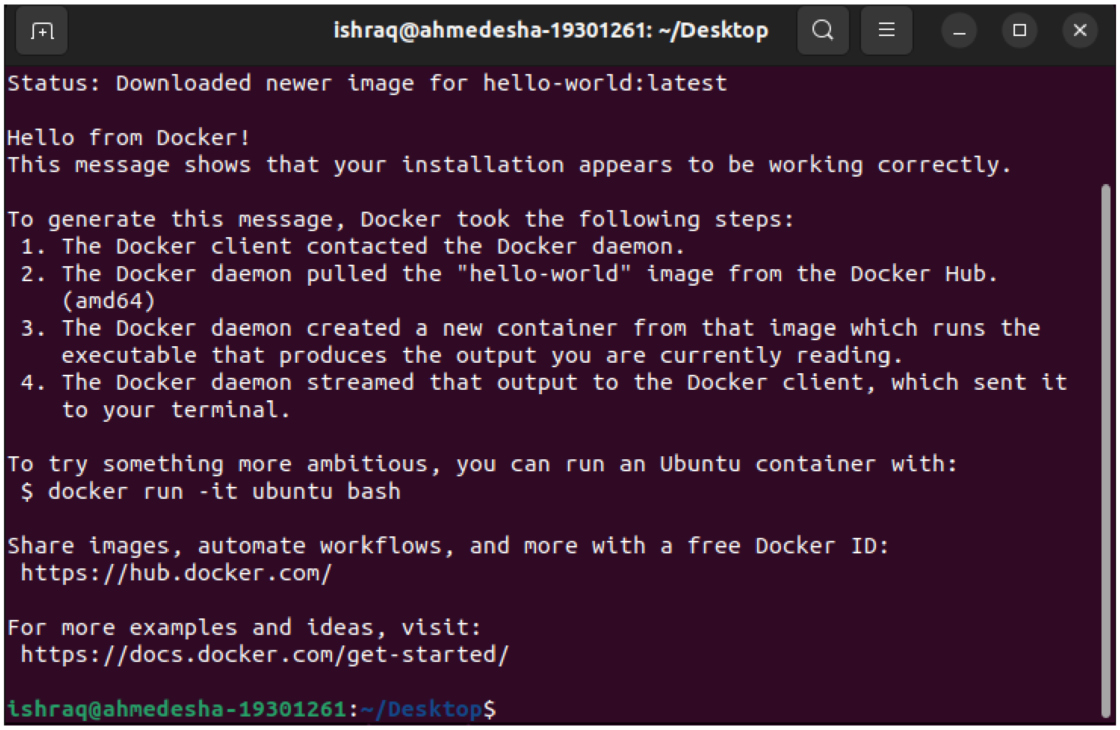
To get the latest version we will install Docker from the official Docker repository and will add a new package source, add the GPG key from Docker to ensure the downloads are valid, and then install the package. Now we have to follow a few steps underneath.

1. Update existing list of packages
   a. "***sudo apt update***"
2. Install few prerequisite packages which let "apt" use packages over HTTPS
   a. "***sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common***"
3. Now add the GPG key for the official Docker repository to the system
   a. "***curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -***"
4. Add a key
   a. "***sudo apt-key fingerprint 0EBFCD88***"
5. Have to add the docker repository to APT source
   a. "***sudo add-apt-repository \***
      "Deb [arch=amd64] https://download.docker.com/linux/ubuntu \
      $(lsb_release -cs) \
      stable" "
6. Install from the docker repo instead of the default ubuntu repo
   a. "***apt-cache policy docker-ce***"
7. Again update the list of packages
   a. "***sudo apt-get update***"
8. Install the docker
   a. "***sudo apt-get install docker-ce docker-ce-cli containerd.io***"
9. Check the docker status
   a. "***sudo systemctl status docker***"



```
ishraq@ahmedesha-19301261: ~/Desktop
ishraq@ahmedesha-19301261:~/Desktop$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Thu 2022-11-17 02:15:35 +06; 20s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 5425 (dockerd)
      Tasks: 12
     Memory: 21.6M
        CPU: 249ms
     CGroup: /system.slice/docker.service
             └─5425 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

নভেম্বর 17 02:15:33 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:33.>
নভেম্বর 17 02:15:33 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:33.>
নভেম্বর 17 02:15:33 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:33.>
নভেম্বর 17 02:15:34 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:34.>
নভেম্বর 17 02:15:34 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:34.>
নভেম্বর 17 02:15:34 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:34.>
নভেম্বর 17 02:15:35 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:35.>
নভেম্বর 17 02:15:35 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:35.>
নভেম্বর 17 02:15:35 ahmedesha-19301261 systemd[1]: Started Docker Application Co>
নভেম্বর 17 02:15:35 ahmedesha-19301261 dockerd[5425]: time="2022-11-17T02:15:35.>
lines 1-22/22 (END)
```

10. Now just check how to work with docker images
    a. "*sudo docker run hello-world*"

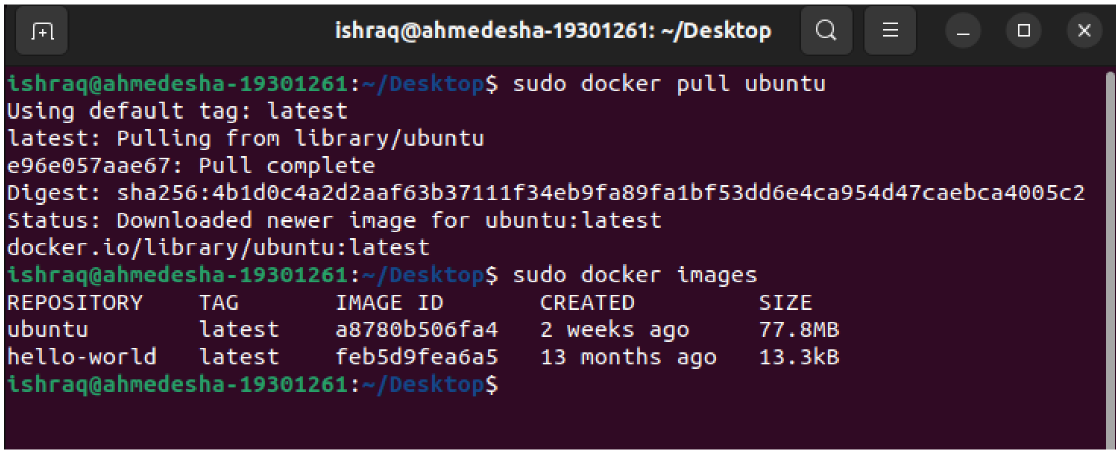


```
Status: Downloaded newer image for hello-world:latest

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.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

ishraq@ahmedesha-19301261:~/Desktop$
```

Answer to the Question No 2

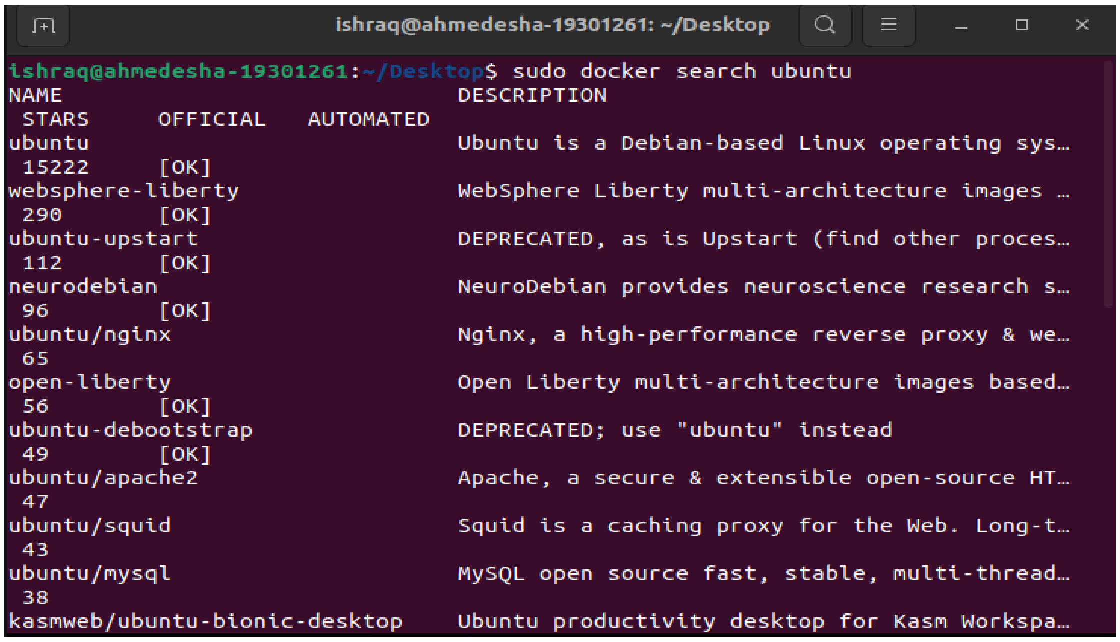
Few Basic docker command are:

1. **pull**: pull command is used to pull an image or a repository from a registry or docker hub "***sudo docker pull ubuntu***"
2. **image**: To see and manage the images this image command is used "***sudo docker images***"



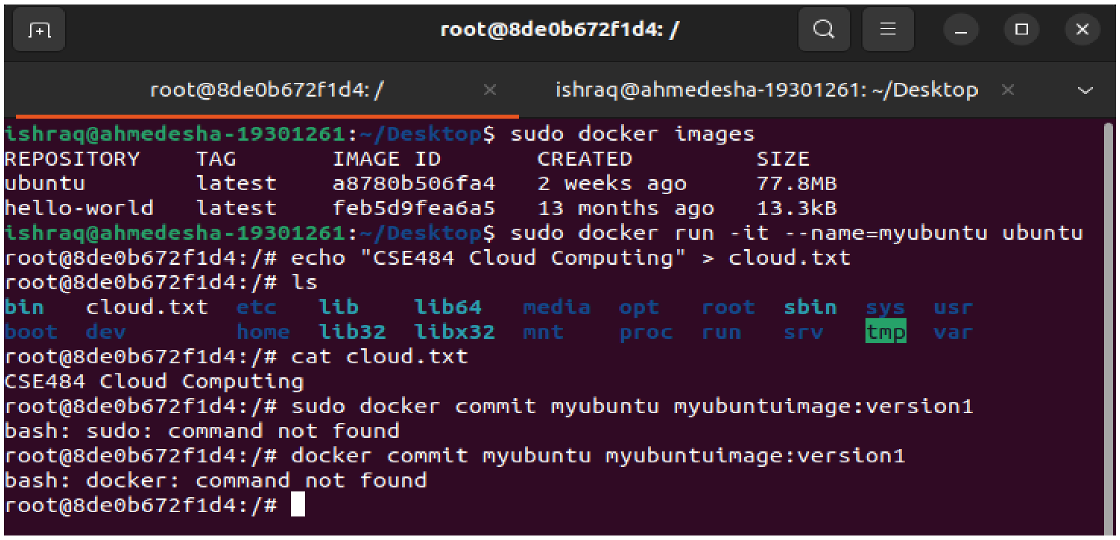
```
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
e96e057aae67: Pull complete
Digest: sha256:4b1d0c4a2d2aaf63b37111f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    a8780b506fa4   2 weeks ago     77.8MB
hello-world   latest    feb5d9fea6a5   13 months ago   13.3kB
ishraq@ahmedesha-19301261:~/Desktop$
```

3. **search**: In the Docker Hub search images ***"sudo docker search ubuntu***"
4. **build**: When we have to build an image from a Dockerfile we have to use the build command "***sudo docker build https://github.com/docker/rootfs.git#container:docker***"

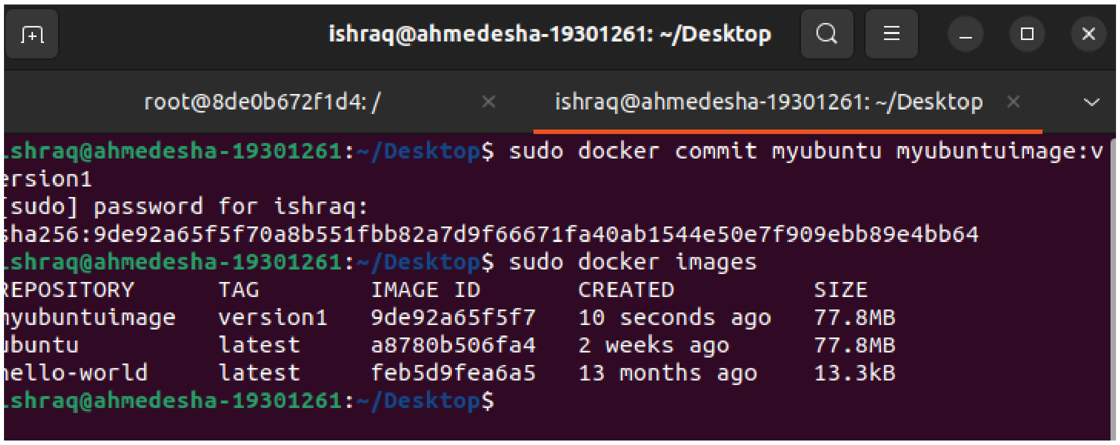


```
ishraq@ahmedesha-19301261: ~/Desktop
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker search ubuntu
NAME                               DESCRIPTION
 STARS     OFFICIAL   AUTOMATED
ubuntu                             Ubuntu is a Debian-based Linux operating sys…
 15222     [OK]
websphere-liberty                  WebSphere Liberty multi-architecture images …
 290       [OK]
ubuntu-upstart                     DEPRECATED, as is Upstart (find other proces…
 112       [OK]
neurodebian                        NeuroDebian provides neuroscience research s…
 96        [OK]
ubuntu/nginx                       Nginx, a high-performance reverse proxy & we…
 65
open-liberty                       Open Liberty multi-architecture images based…
 56        [OK]
ubuntu-debootstrap                 DEPRECATED; use "ubuntu" instead
 49        [OK]
ubuntu/apache2                     Apache, a secure & extensible open-source HT…
 47
ubuntu/squid                       Squid is a caching proxy for the Web. Long-t…
 43
ubuntu/mysql                       MySQL open source fast, stable, multi-thread…
 38
kasmweb/ubuntu-bionic-desktop      Ubuntu productivity desktop for Kasm Workspa…
```

5. **run**: To run a command in a new container as we have run the hello-world image above "***sudo docker run hello-world***" . Now if we want to create an ubuntu container and make some changes inside it. We will create a new file inside the container. "***sudo docker run -it --name=myubuntu ubuntu***"
6. **commit**: To save changes to a new image we use a commit command with some message. Now keeping the container running we have to open a new terminal and execute the following command for commit "***sudo docker commit myubuntu myubuntuimage:version1***" . The new file which we created earlier that's going to be committed with version1 message in new repository of myubuntuimage.

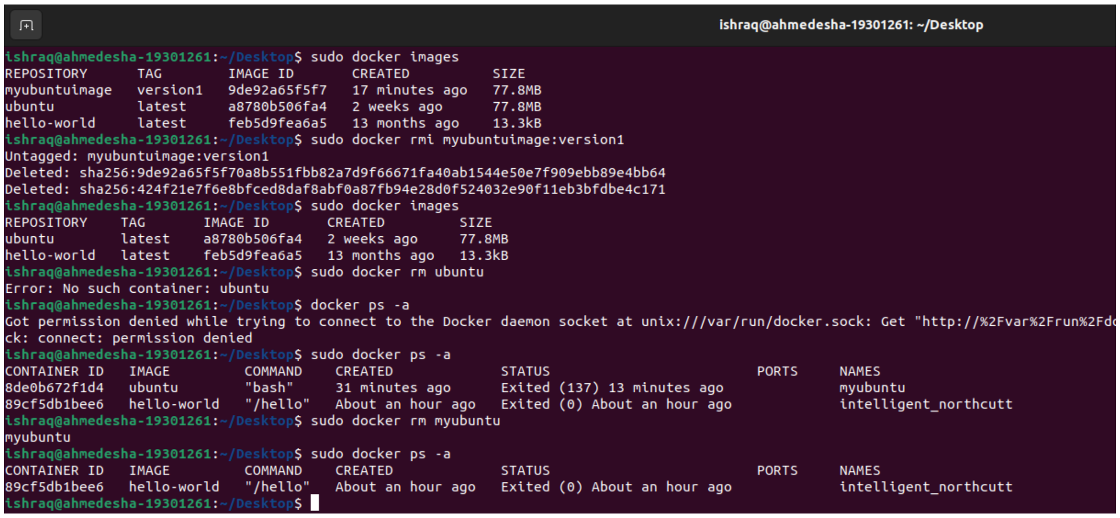


```
root@8de0b672f1d4: /
root@8de0b672f1d4: /          ishraq@ahmedesha-19301261: ~/Desktop
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
ubuntu        latest    a8780b506fa4   2 weeks ago      77.8MB
hello-world   latest    feb5d9fea6a5   13 months ago    13.3kB
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker run -it --name=myubuntu ubuntu
root@8de0b672f1d4:/# echo "CSE484 Cloud Computing" > cloud.txt
root@8de0b672f1d4:/# ls
bin   cloud.txt  etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  dev        home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@8de0b672f1d4:/# cat cloud.txt
CSE484 Cloud Computing
root@8de0b672f1d4:/# sudo docker commit myubuntu myubuntuimage:version1
bash: sudo: command not found
root@8de0b672f1d4:/# docker commit myubuntu myubuntuimage:version1
bash: docker: command not found
root@8de0b672f1d4:/# 
```



```
ishraq@ahmedesha-19301261: ~/Desktop
root@8de0b672f1d4: /        ishraq@ahmedesha-19301261: ~/Desktop
.shraq@ahmedesha-19301261:~/Desktop$ sudo docker commit myubuntu myubuntuimage:v
ersion1
[sudo] password for ishraq:
sha256:9de92a65f5f70a8b551fbb82a7d9f66671fa40ab1544e50e7f909ebb89e4bb64
.shraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY      TAG        IMAGE ID       CREATED          SIZE
nyubuntuimage   version1   9de92a65f5f7   10 seconds ago   77.8MB
ubuntu          latest     a8780b506fa4   2 weeks ago      77.8MB
hello-world     latest     feb5d9fea6a5   13 months ago    13.3kB
.shraq@ahmedesha-19301261:~/Desktop$
```

7. **stop**: To stop a running container we have to use the following command "***sudo docker stop myubuntu***" by executing this command running container myubuntu will stop.
8. **rmi**: To remove one or more images we use the following command "***sudo docker rmi myubuntuimage myubuntu:version1***" Here we will remove the newly created image which was myubuntuimage and Tag was version1
9. **rm**: To remove one or more containers we use the following command "***sudo docker rm myubuntu***" . Earlier we stopped the myubuntu container. Now if we execute the command then the myubuntu container will be removed.

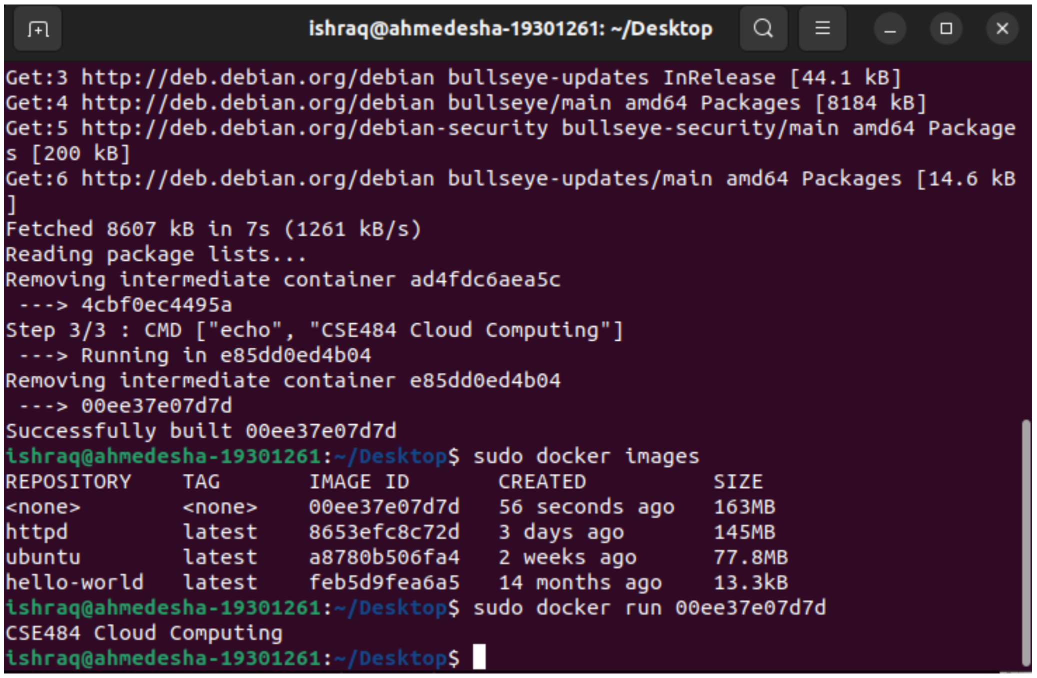


```
ishraq@ahmedesha-19301261: ~/Desktop
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY      TAG        IMAGE ID       CREATED          SIZE
myubuntuimage   version1   9de92a65f5f7   17 minutes ago   77.8MB
ubuntu          latest     a8780b506fa4   2 weeks ago      77.8MB
hello-world     latest     feb5d9fea6a5   13 months ago    13.3kB
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker rmi myubuntuimage:version1
Untagged: myubuntuimage:version1
Deleted: sha256:9de92a65f5f70a8b551fbb82a7d9f66671fa40ab1544e50e7f909ebb89e4bb64
Deleted: sha256:424f21e7f6e8bfced8daf8abf0a87fb94e28d0f524032e90f11eb3bfdbe4c171
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    a8780b506fa4   2 weeks ago     77.8MB
hello-world   latest    feb5d9fea6a5   13 months ago   13.3kB
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker rm ubuntu
Error: No such container: ubuntu
ishraq@ahmedesha-19301261:~/Desktop$ docker ps -a
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdo
ck: connect: permission denied
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED             STATUS                        PORTS     NAMES
8de0b672f1d4   ubuntu        "bash"     31 minutes ago      Exited (137) 13 minutes ago             myubuntu
89cf5db1bee6   hello-world   "/hello"   About an hour ago   Exited (0) About an hour ago            intelligent_northcutt
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker rm myubuntu
myubuntu
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED             STATUS                        PORTS     NAMES
89cf5db1bee6   hello-world   "/hello"   About an hour ago   Exited (0) About an hour ago            intelligent_northcutt
ishraq@ahmedesha-19301261:~/Desktop$
```

Answer to the Question No 3

To create a docker image using dockerfile we have to follow the following steps:

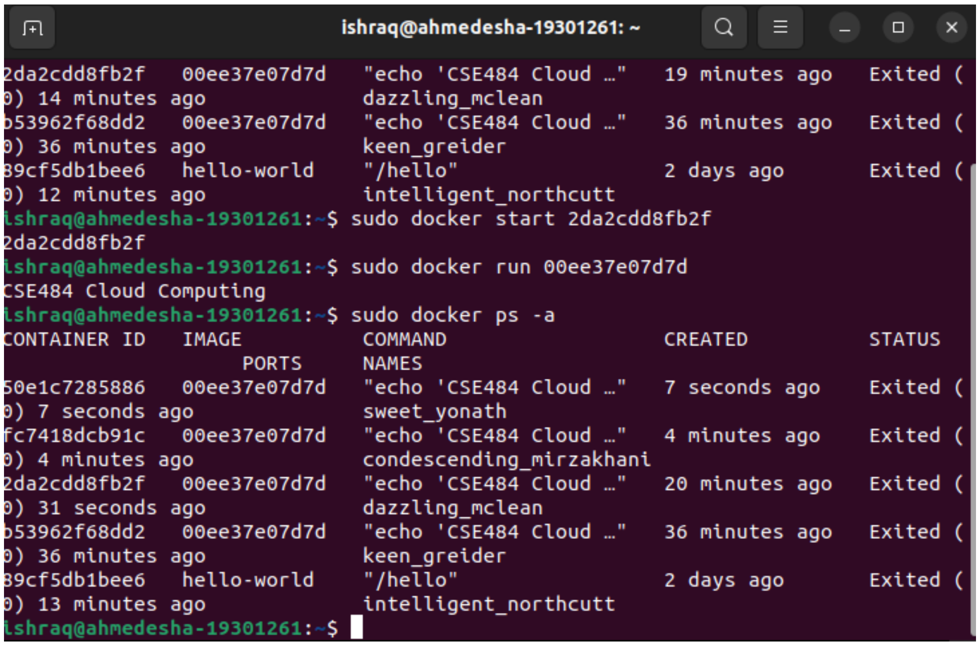
1. First of all, we have to create a directory to store the dockerfile. Using the following command "***mkdir Dockerfiles***"
2. In the dockerfile directory create a file named dockerfile and edit the file with the commands that want to execute and save the file. Use the following command "***vim Dockerfiles/dockerfile***"
3. Here we will build an apache web server image as an example. In the dockerfile we have to write the following commands to execute the server and show output.
   "***#Getting the image from docker hub***
   FROM httpd
   RUN apt-get update
   CMd ["echo", "CSE484 Cloud Computing"]"
4. Now to build the dockerfile run the following command from the terminal "***sudo docker build dockerfiles/***" . After completing the command "*successfully built*" message will be shown.
5. Now check the images with the following command "***sudo docker images***" . Here we can see the new image ID after successful build of our image from the dockerfile.
6. We have to run the image to see our output with the following command: "***sudo docker run 00ee37e07d7d***" . After that we can see our output "*CSE484 Cloud Computing*". So we have successfully created a docker image from the dockerfile and run it.



```
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Package
s [200 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB
]
Fetched 8607 kB in 7s (1261 kB/s)
Reading package lists...
Removing intermediate container ad4fdc6aea5c
 ---> 4cbf0ec4495a
Step 3/3 : CMD ["echo", "CSE484 Cloud Computing"]
 ---> Running in e85dd0ed4b04
Removing intermediate container e85dd0ed4b04
 ---> 00ee37e07d7d
Successfully built 00ee37e07d7d
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
<none>        <none>    00ee37e07d7d   56 seconds ago   163MB
httpd         latest    8653efc8c72d   3 days ago       145MB
ubuntu        latest    a8780b506fa4   2 weeks ago      77.8MB
hello-world   latest    feb5d9fea6a5   14 months ago    13.3kB
ishraq@ahmedesha-19301261:~/Desktop$ sudo docker run 00ee37e07d7d
CSE484 Cloud Computing
ishraq@ahmedesha-19301261:~/Desktop$ 
```

Answer to the Question No 4

To run a single container we have to start the container with the container ID, with following command "***sudo docker run 2da2ccdd8fb2f***" then to show the output of the docker image we have to run the docker image file with the docker image id, with following command "***sudo docker run 00ee37e07d7d***" . After that we can see the output as "*CSE484 Cloud Computing*". Then to show the status of all containers we will execute the following command "***sudo docker ps -a***" . After executing this command we can see the container ID, image ID, command, created time, status, ports and names. Underneath the image which is shown clearly.

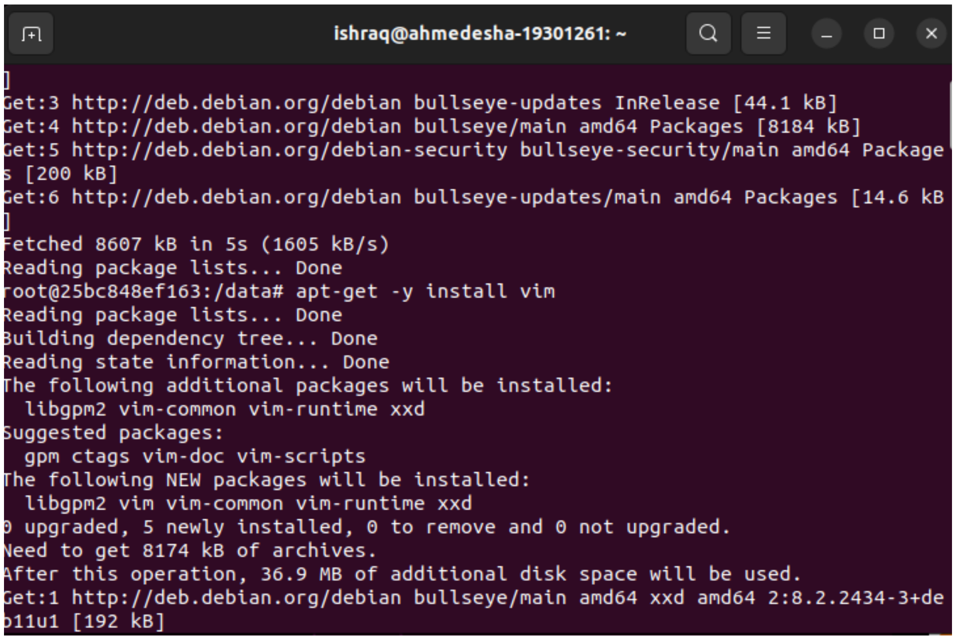


```
2da2cdd8fb2f   00ee37e07d7d   "echo 'CSE484 Cloud …"   19 minutes ago   Exited (
0) 14 minutes ago             dazzling_mclean
b53962f68dd2   00ee37e07d7d   "echo 'CSE484 Cloud …"   36 minutes ago   Exited (
0) 36 minutes ago             keen_greider
39cf5db1bee6   hello-world    "/hello"                 2 days ago       Exited (
0) 12 minutes ago             intelligent_northcutt
ishraq@ahmedesha-19301261:~$ sudo docker start 2da2cdd8fb2f
2da2cdd8fb2f
ishraq@ahmedesha-19301261:~$ sudo docker run 00ee37e07d7d
CSE484 Cloud Computing
ishraq@ahmedesha-19301261:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
                    PORTS     NAMES
50e1c7285886   00ee37e07d7d   "echo 'CSE484 Cloud …"   7 seconds ago    Exited (
0) 7 seconds ago              sweet_yonath
fc7418dcb91c   00ee37e07d7d   "echo 'CSE484 Cloud …"   4 minutes ago    Exited (
0) 4 minutes ago              condescending_mirzakhani
2da2cdd8fb2f   00ee37e07d7d   "echo 'CSE484 Cloud …"   20 minutes ago   Exited (
0) 31 seconds ago             dazzling_mclean
b53962f68dd2   00ee37e07d7d   "echo 'CSE484 Cloud …"   36 minutes ago   Exited (
0) 36 minutes ago             keen_greider
39cf5db1bee6   hello-world    "/hello"                 2 days ago       Exited (
0) 13 minutes ago             intelligent_northcutt
ishraq@ahmedesha-19301261:~$
```

Answer to the Question No 5

I will run the docker container in interactive mode. The advantage of docker interactive mode is that it allows us to execute commands at the time of running the container. Here I will use the Redis container. I can first start a Redis docker container in the background using the following command "***sudo docker run -d redis***" . Now to check the ID of the running containers we will use "***sudo docker ps -a***" . Lastly, using the ID of the container, I can use the following command to issue a different command to the running container in interactive mode " ***sudo docker exe -it redis-cli***" . Here I am issuing the command "redis-cli" on the container & "-it" for interactive mode. Or I can use bash to execute commands. This will open a redis-cli command prompt where I can execute commands on the Redis server. After that I execute the following command to install the vim in the running container. First of all, to generate a cache in the image following command "***apt-get update***" then install the vim package with the following command "***apt-get -y***

install vim" . By following these steps I can run a container in an interactive mode and install the packages in the running container.



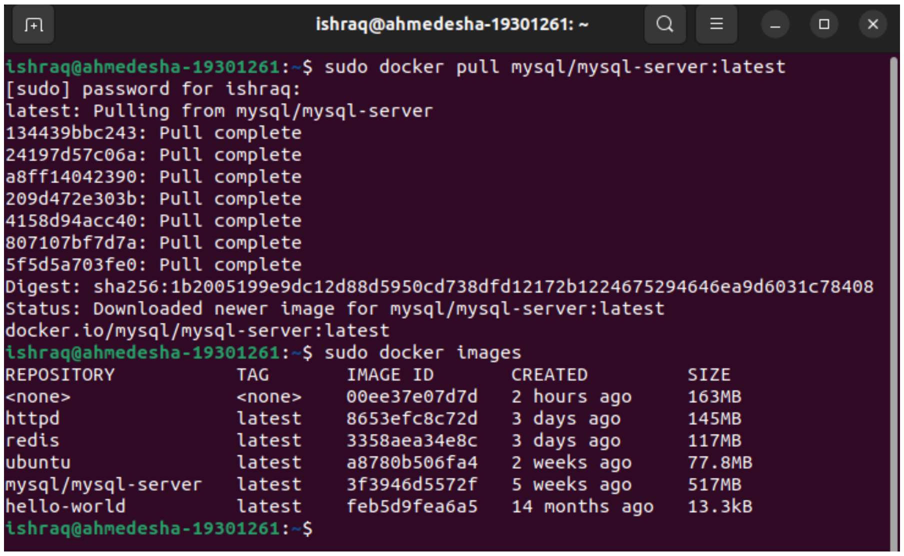
ishraq@ahmedesha-19301261: ~

```
]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Package
s [200 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB
]
Fetched 8607 kB in 5s (1605 kB/s)
Reading package lists... Done
root@25bc848ef163:/data# apt-get -y install vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 vim-common vim-runtime xxd
Suggested packages:
  gpm ctags vim-doc vim-scripts
The following NEW packages will be installed:
  libgpm2 vim vim-common vim-runtime xxd
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 8174 kB of archives.
After this operation, 36.9 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 xxd amd64 2:8.2.2434-3+de
b11u1 [192 kB]
```

Answer to the Question No 6

To run a database container in the background, showing logs, and some sql queries in interactive mode we have to follow below steps: Here I am using MySQL database.

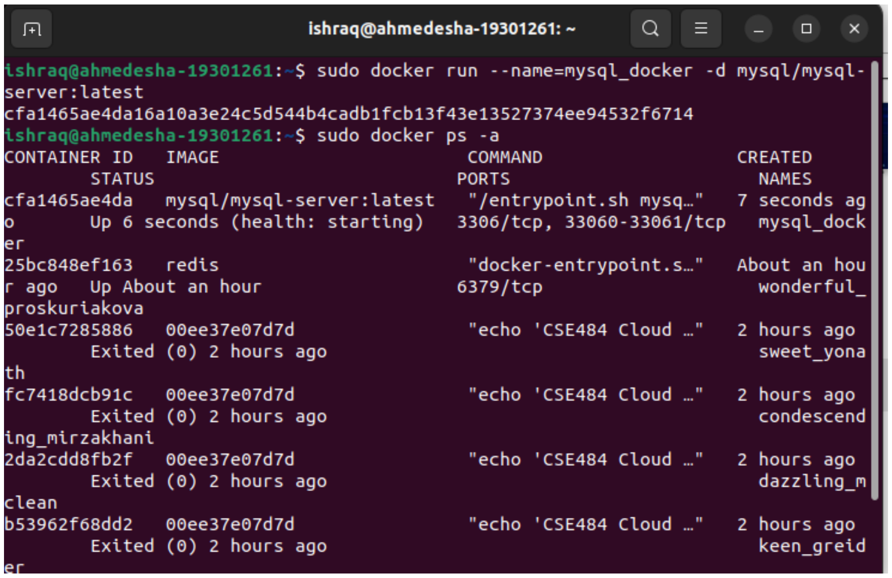
1. Pull the MySQL Docker Image: "***sudo docker pull mysql-mysql-server:latest***"
2. To verify the image I can use the "***sudo docker images***" command and will see the Image ID.



ishraq@ahmedesha-19301261: ~

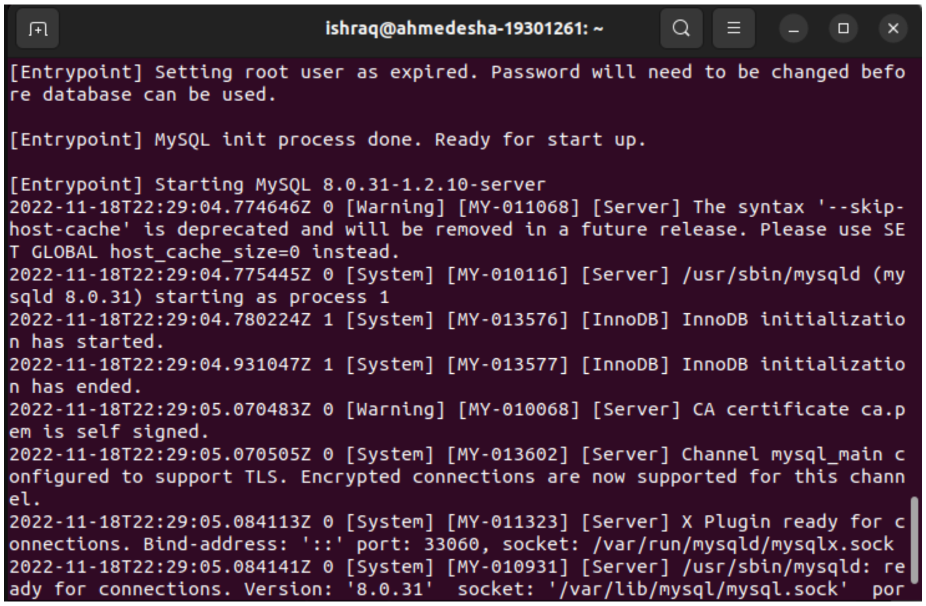
```
ishraq@ahmedesha-19301261:~$ sudo docker pull mysql/mysql-server:latest
[sudo] password for ishraq:
latest: Pulling from mysql/mysql-server
134439bbc243: Pull complete
24197d57c06a: Pull complete
a8ff14042390: Pull complete
209d472e303b: Pull complete
4158d94acc40: Pull complete
807107bf7d7a: Pull complete
5f5d5a703fe0: Pull complete
Digest: sha256:1b2005199e9dc12d88d5950cd738dfd12172b1224675294646ea9d6031c78408
Status: Downloaded newer image for mysql/mysql-server:latest
docker.io/mysql/mysql-server:latest
ishraq@ahmedesha-19301261:~$ sudo docker images
REPOSITORY           TAG       IMAGE ID       CREATED         SIZE
<none>               <none>    00ee37e07d7d   2 hours ago     163MB
httpd                latest    8653efc8c72d   3 days ago      145MB
redis                latest    3358aea34e8c   3 days ago      117MB
ubuntu               latest    a8780b506fa4   2 weeks ago     77.8MB
mysql/mysql-server   latest    3f3946d5572f   5 weeks ago     517MB
hello-world          latest    feb5d9fea6a5   14 months ago   13.3kB
ishraq@ahmedesha-19301261:~$
```

3. Deploy the MySQL Container: "***sudo docker run --name=mysql_docker -d mysql-server:latest***" . Here I have provided a container name as mysql_docker and "-d" for running in the background.
4. I can check the MySQL container running status with the following command "***sudo docker ps -a***"



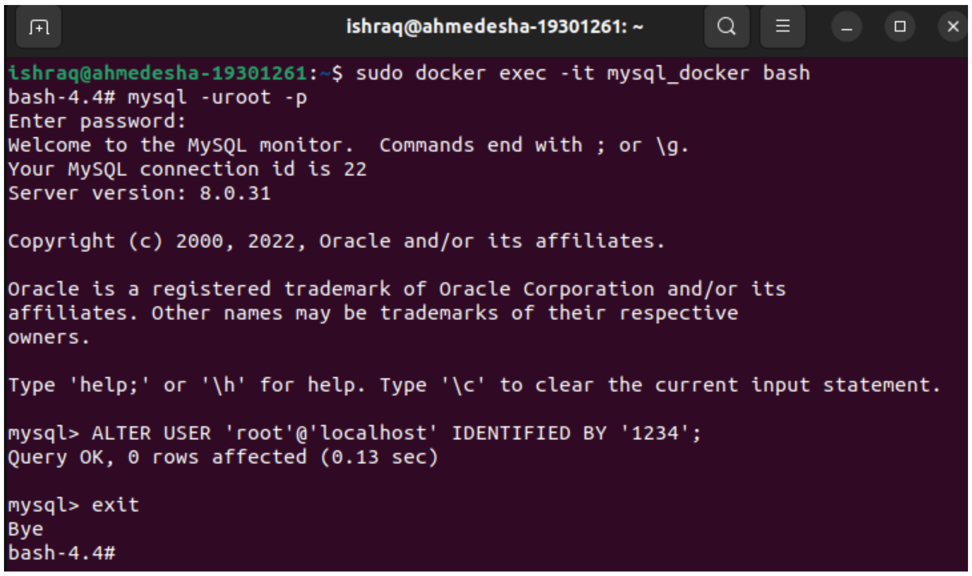
```
ishraq@ahmedesha-19301261:~$ sudo docker run --name=mysql_docker -d mysql/mysql-
server:latest
cfa1465ae4da16a10a3e24c5d544b4cadb1fcb13f43e13527374ee94532f6714
ishraq@ahmedesha-19301261:~$ sudo docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED
        STATUS                         PORTS                              NAMES
cfa1465ae4da   mysql/mysql-server:latest     "/entrypoint.sh mysq…"   7 seconds ag
o       Up 6 seconds (health: starting)   3306/tcp, 33060-33061/tcp   mysql_dock
er
25bc848ef163   redis                          "docker-entrypoint.s…"   About an hou
r ago   Up About an hour                  6379/tcp                    wonderful_
proskuriakova
50e1c7285886   00ee37e07d7d                   "echo 'CSE484 Cloud …"   2 hours ago
        Exited (0) 2 hours ago                                        sweet_yona
th
fc7418dcb91c   00ee37e07d7d                   "echo 'CSE484 Cloud …"   2 hours ago
        Exited (0) 2 hours ago                                        condescend
ing_mirzakhani
2da2cdd8fb2f   00ee37e07d7d                   "echo 'CSE484 Cloud …"   2 hours ago
        Exited (0) 2 hours ago                                        dazzling_m
clean
b53962f68dd2   00ee37e07d7d                   "echo 'CSE484 Cloud …"   2 hours ago
        Exited (0) 2 hours ago                                        keen_greid
er
```

5. Connect to the MySQL Docker Container: Before connecting with the MySQL server container with the host, I need to make sure the MySQL client package is installed with the following command "***apt-get install mysql-client***"
6. Then have to open the logs file for MySQL container to find the generated root password with the following command "***sudo docker logs mysql_docker***" . From here I have to copy the generated password for the next step.



```
[Entrypoint] Setting root user as expired. Password will need to be changed befo
re database can be used.

[Entrypoint] MySQL init process done. Ready for start up.

[Entrypoint] Starting MySQL 8.0.31-1.2.10-server
2022-11-18T22:29:04.774646Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-
host-cache' is deprecated and will be removed in a future release. Please use SE
T GLOBAL host_cache_size=0 instead.
2022-11-18T22:29:04.775445Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (my
sqld 8.0.31) starting as process 1
2022-11-18T22:29:04.780224Z 1 [System] [MY-013576] [InnoDB] InnoDB initializatio
n has started.
2022-11-18T22:29:04.931047Z 1 [System] [MY-013577] [InnoDB] InnoDB initializatio
n has ended.
2022-11-18T22:29:05.070483Z 0 [Warning] [MY-010068] [Server] CA certificate ca.p
em is self signed.
2022-11-18T22:29:05.070505Z 0 [System] [MY-013602] [Server] Channel mysql_main c
onfigured to support TLS. Encrypted connections are now supported for this chann
el.
2022-11-18T22:29:05.084113Z 0 [System] [MY-011323] [Server] X Plugin ready for c
onnections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2022-11-18T22:29:05.084141Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: re
ady for connections. Version: '8.0.31'  socket: '/var/lib/mysql/mysql.sock'  por
```

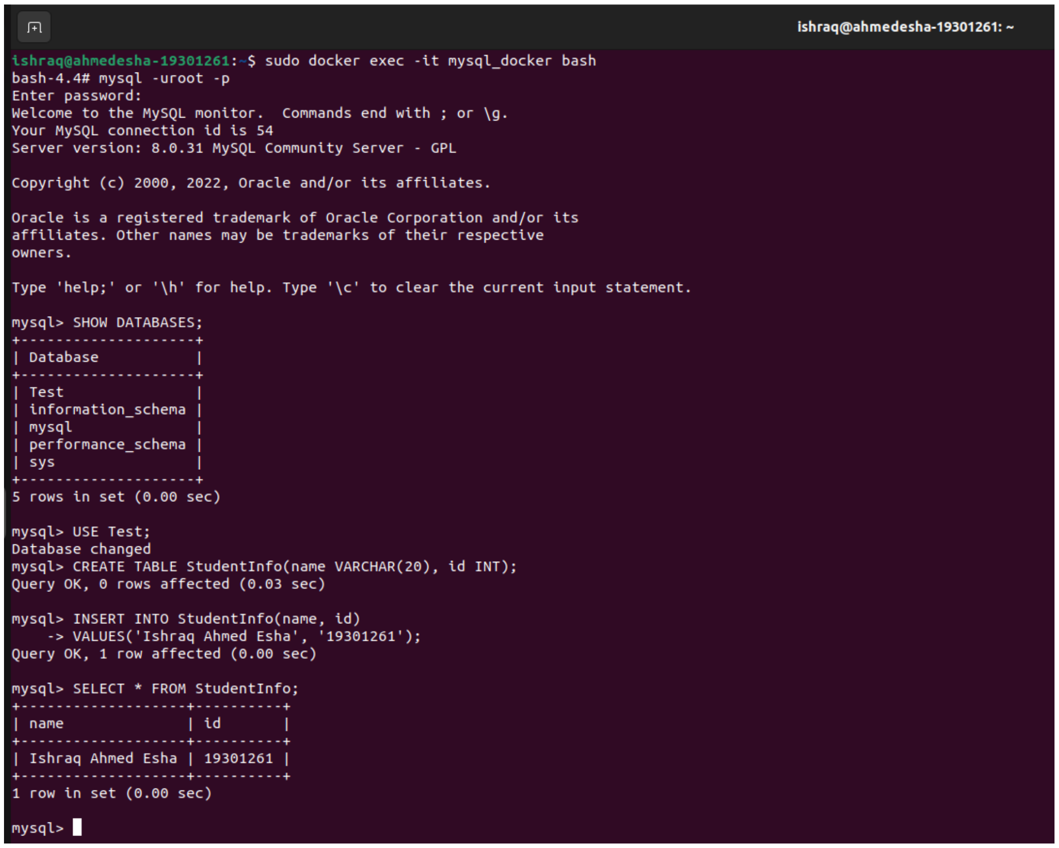
7. Now will go to the bash shell of the MySQL container with the following command "***sudo docker exec -it mysql_docker bash***"
8. Now I will write a sql query to change the root password from the container bash. Executing the following sql query I can change the root password for the database "***ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';***"



```
ishraq@ahmedesha-19301261:~$ sudo docker exec -it mysql_docker bash
bash-4.4# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.31

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.13 sec)

mysql> exit
Bye
bash-4.4#
```

9. Few sql queries to create a database and create a table then insert some data and show that.

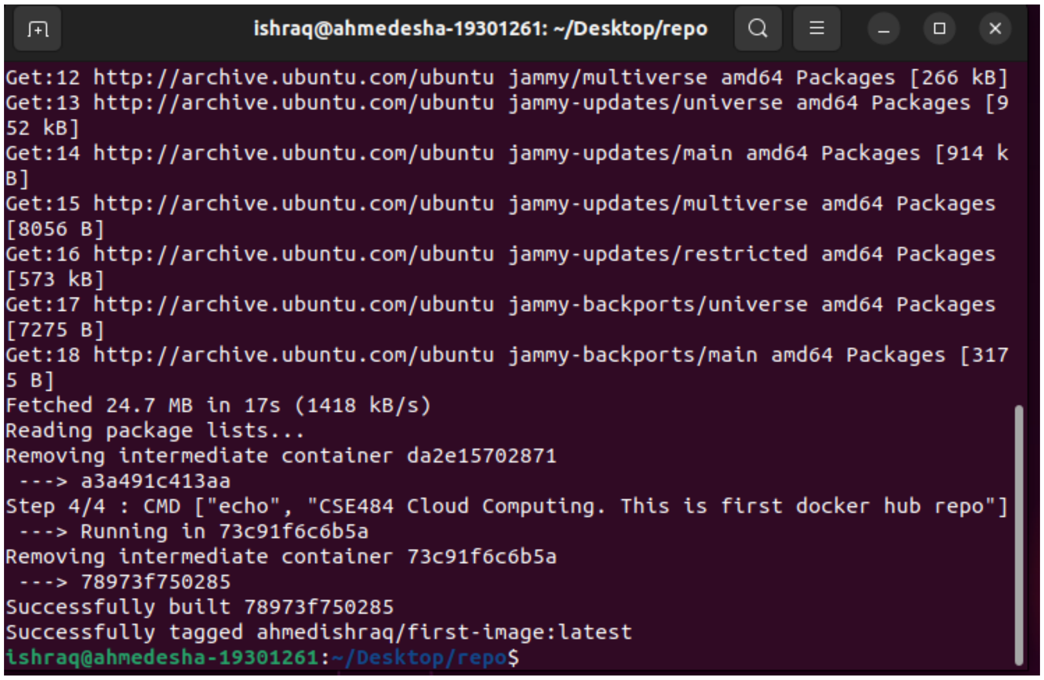
   "***CREATE DATABASE Test;***"
   "***CREATE TABLE StudentInfo(***
   name VARCHAR(20),
   id INT);"
   "***INSERT INTO StudentInfo (name, id)***
   VALUES
   ('Ishraq Ahmed Esha', '19301261');"



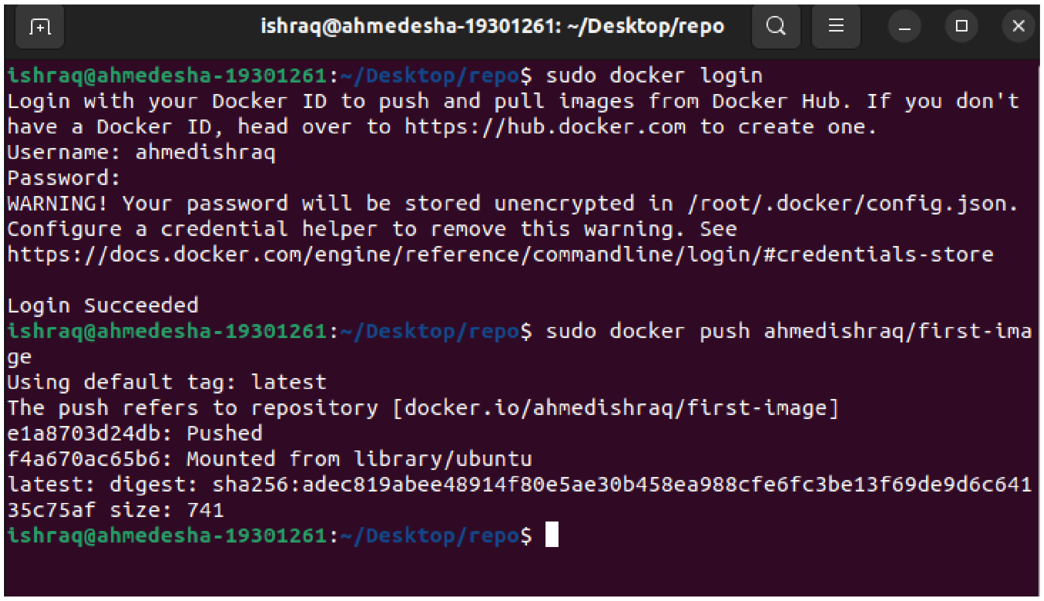
```
ishraq@ahmedesha-19301261:~$ sudo docker exec -it mysql_docker bash
bash-4.4# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Test               |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> USE Test;
Database changed
mysql> CREATE TABLE StudentInfo(name VARCHAR(20), id INT);
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO StudentInfo(name, id)
    -> VALUES('Ishraq Ahmed Esha', '19301261');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM StudentInfo;
+-------------------+----------+
| name              | id       |
+-------------------+----------+
| Ishraq Ahmed Esha | 19301261 |
+-------------------+----------+
1 row in set (0.00 sec)

mysql> 
```

Answer to the Question No 7

I will create a new dockerfile named dockerfile then will edit that with desired commands and save that. After that I will build the docker image with my docker hub username and repository name with the following command “***sudo docker build -t ahmedishraq/first-image .***”. Then I have to push the image to docker hub, to do so I need to login my docker hub account with the following command “***sudo docker login***” . Now push the docker image with the following command “***sudo docker push ahmedishraq/first-image***”



```
Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [9
52 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [914 k
B]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages
[8056 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages
[573 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages
[7275 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [317
5 B]
Fetched 24.7 MB in 17s (1418 kB/s)
Reading package lists...
Removing intermediate container da2e15702871
 ---> a3a491c413aa
Step 4/4 : CMD ["echo", "CSE484 Cloud Computing. This is first docker hub repo"]
 ---> Running in 73c91f6c6b5a
Removing intermediate container 73c91f6c6b5a
 ---> 78973f750285
Successfully built 78973f750285
Successfully tagged ahmedishraq/first-image:latest
ishraq@ahmedesha-19301261:~/Desktop/repo$
```



```
ishraq@ahmedesha-19301261:~/Desktop/repo$ 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: ahmedishraq
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
ishraq@ahmedesha-19301261:~/Desktop/repo$ sudo docker push ahmedishraq/first-ima
ge
Using default tag: latest
The push refers to repository [docker.io/ahmedishraq/first-image]
e1a8703d24db: Pushed
f4a670ac65b6: Mounted from library/ubuntu
latest: digest: sha256:adec819abee48914f80e5ae30b458ea988cfe6fc3be13f69de9d6c641
35c75af size: 741
ishraq@ahmedesha-19301261:~/Desktop/repo$
```

Answer to the Question No 8

To make my own private registry have to follow the steps:

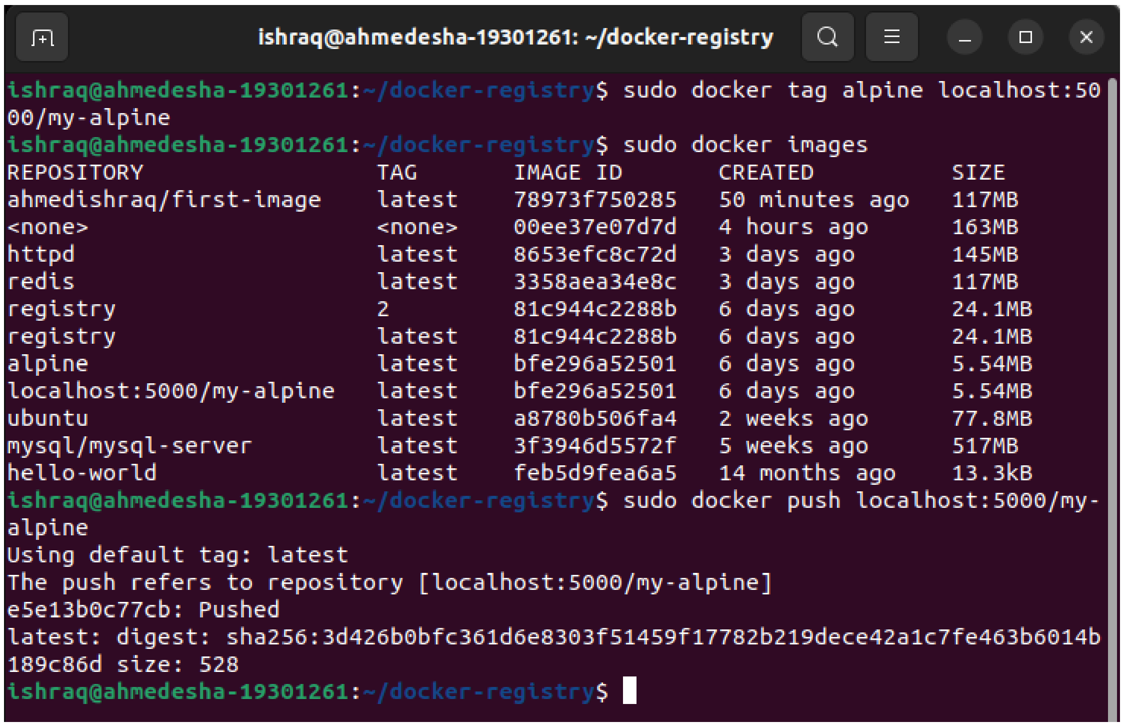
1. First of all, have to create own registry, to do so first of all install the docker-compose using following command "***sudo apt install docker-compose***"

2. Then edit the docker-compose.yml file with vim with the following commands

   ```
   "version: '3'
   services:
    registry:
     image: registry:2
     ports:
       -    "5000:5000"
   ```

The configuration uses the official registry image and forwards the port 5000 of the container to the host machine. This allows it to send requests to port 5000 on the server that runs the registry.

3. Now run the container with the following command "***sudo docker-compose up -d***"
4. Now I am ready to push an image to the registry, here I am going to use the alpine Linux image because it is small and downloads fast. Pull the image with the following command "***sudo docker pull alpine***" . Then add the tag "***sudo docker tag alpine localhost:5000/my-alpine***" . By using "***sudo docker images***" we can see the newly added images.
5. Now I can push the image to my private registry with the following command "***sudo docker push localhost:5000/my-alpine***" . This only works if my host registry on my local machine. If I want to host it on a server, I will need a secure SLL connection.

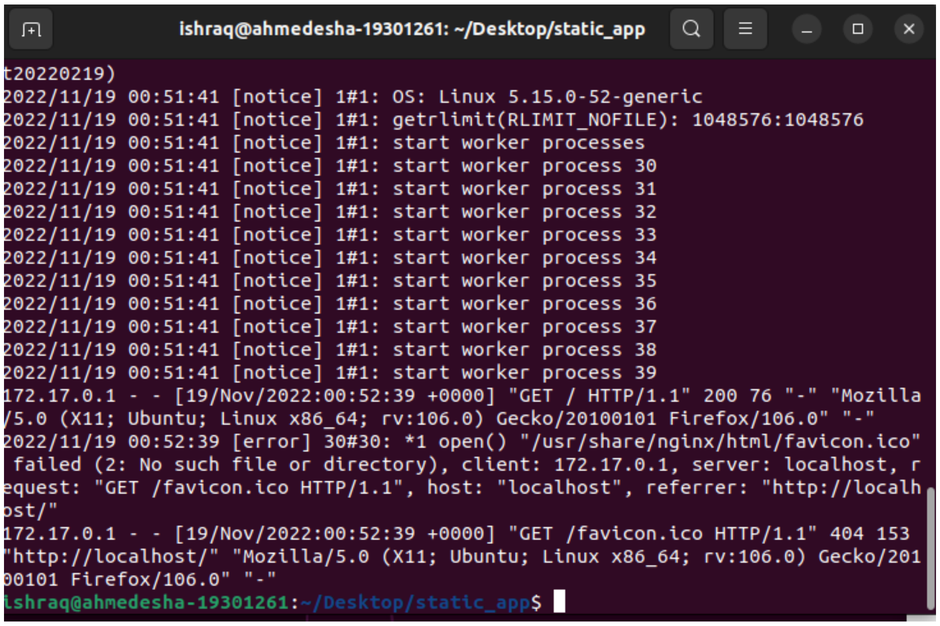


```
ishraq@ahmedesha-19301261:~/docker-registry$ sudo docker tag alpine localhost:50
00/my-alpine
ishraq@ahmedesha-19301261:~/docker-registry$ sudo docker images
REPOSITORY                 TAG       IMAGE ID       CREATED          SIZE
ahmedishraq/first-image    latest    78973f750285   50 minutes ago   117MB
<none>                     <none>    00ee37e07d7d   4 hours ago      163MB
httpd                      latest    8653efc8c72d   3 days ago       145MB
redis                      latest    3358aea34e8c   3 days ago       117MB
registry                   2         81c944c2288b   6 days ago       24.1MB
registry                   latest    81c944c2288b   6 days ago       24.1MB
alpine                     latest    bfe296a52501   6 days ago       5.54MB
localhost:5000/my-alpine   latest    bfe296a52501   6 days ago       5.54MB
ubuntu                     latest    a8780b506fa4   2 weeks ago      77.8MB
mysql/mysql-server         latest    3f3946d5572f   5 weeks ago      517MB
hello-world                latest    feb5d9fea6a5   14 months ago    13.3kB
ishraq@ahmedesha-19301261:~/docker-registry$ sudo docker push localhost:5000/my-
alpine
Using default tag: latest
The push refers to repository [localhost:5000/my-alpine]
e5e13b0c77cb: Pushed
latest: digest: sha256:3d426b0bfc361d6e8303f51459f17782b219dece42a1c7fe463b6014b
189c86d size: 528
ishraq@ahmedesha-19301261:~/docker-registry$ 
```

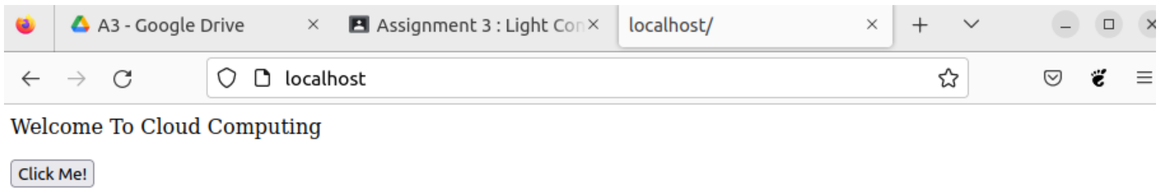
Answer to the Question No 9

I will create a small simple webapp that will run in the background of my host machine and I will browse that website from my local host machine. To do this follow the below steps:

1. Create a directory in home "**mkdir static_app**" there I will create a index.html file for the website instruction and edit that file with vim "***vim index.html***" "***<p>Welcome To Cloud Computing <button type="button">Click Me!</button>***"
2. Now I have to create the dockerfile for the image with the following command "***vim dockerfile***" edit this file with following instructions "***FROM nginx:alpine COPY . /usr/share/nginx/html***" save the dockerfile.
3. Now it's time to build the image with the following command "***sudo docker build -t static-app:v1 .***"
4. Image have build successfully now time to run the container and specify the port for the localhost with the following command "***sudo docker run -d --name=app-container -p 80:80 static-app:v1***"
5. Now if I type the localhost and port 80 to my host machine then I will see the website.



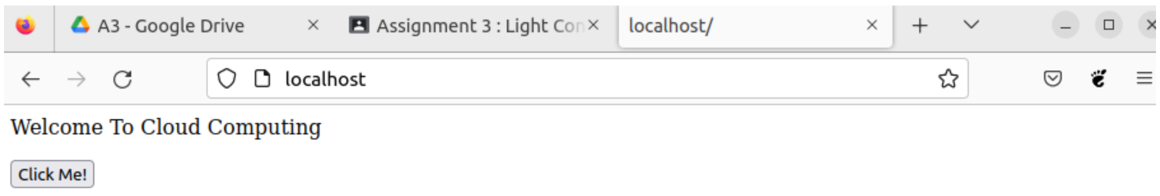
```
t20220219)
2022/11/19 00:51:41 [notice] 1#1: OS: Linux 5.15.0-52-generic
2022/11/19 00:51:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/11/19 00:51:41 [notice] 1#1: start worker processes
2022/11/19 00:51:41 [notice] 1#1: start worker process 30
2022/11/19 00:51:41 [notice] 1#1: start worker process 31
2022/11/19 00:51:41 [notice] 1#1: start worker process 32
2022/11/19 00:51:41 [notice] 1#1: start worker process 33
2022/11/19 00:51:41 [notice] 1#1: start worker process 34
2022/11/19 00:51:41 [notice] 1#1: start worker process 35
2022/11/19 00:51:41 [notice] 1#1: start worker process 36
2022/11/19 00:51:41 [notice] 1#1: start worker process 37
2022/11/19 00:51:41 [notice] 1#1: start worker process 38
2022/11/19 00:51:41 [notice] 1#1: start worker process 39
172.17.0.1 - - [19/Nov/2022:00:52:39 +0000] "GET / HTTP/1.1" 200 76 "-" "Mozilla
/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0) Gecko/20100101 Firefox/106.0" "-"
2022/11/19 00:52:39 [error] 30#30: *1 open() "/usr/share/nginx/html/favicon.ico"
 failed (2: No such file or directory), client: 172.17.0.1, server: localhost, r
equest: "GET /favicon.ico HTTP/1.1", host: "localhost", referrer: "http://localh
ost/"
172.17.0.1 - - [19/Nov/2022:00:52:39 +0000] "GET /favicon.ico HTTP/1.1" 404 153
"http://localhost/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0) Gecko/201
00101 Firefox/106.0" "-"
ishraq@ahmedesha-19301261:~/Desktop/static_app$
```



Welcome To Cloud Computing

Answer to the Question No 10

If I push the webapp image into a public registry like docker hub, then anyone can pull it from the hub and if they run the image into their local host machine they also can have the local access of the webapp.



Welcome To Cloud Computing