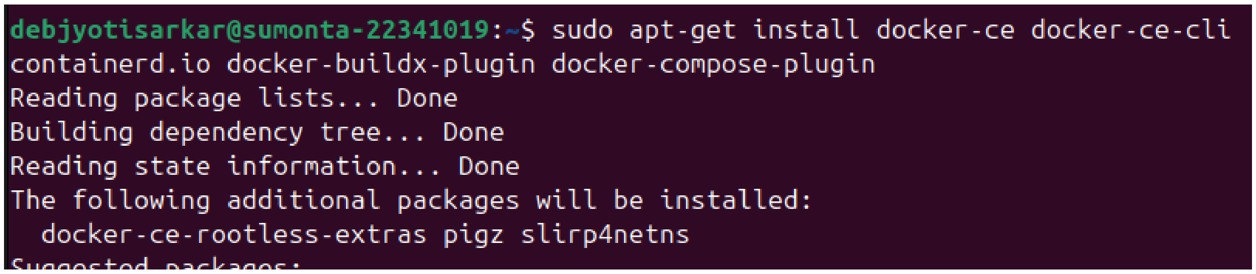
Task 1

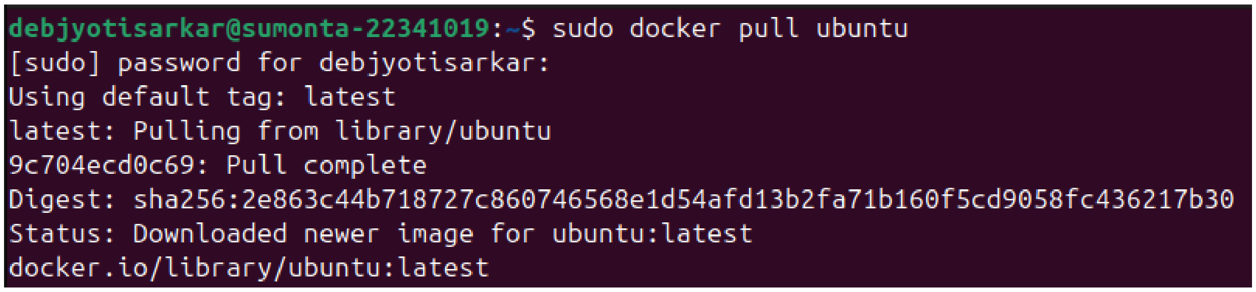
The following commands install docker and other necessary plugins.



```
debjyotisarkar@sumonta-22341019:~$ sudo apt-get install docker-ce docker-ce-cli
containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras pigz slirp4netns
```

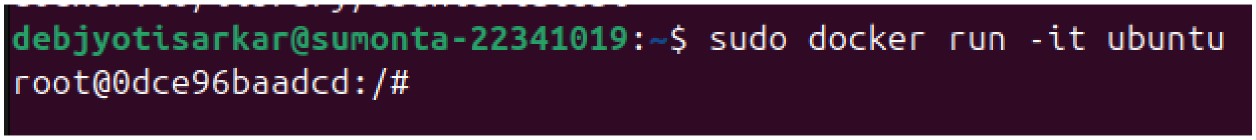
Task 2

This command pulls the image from the repository



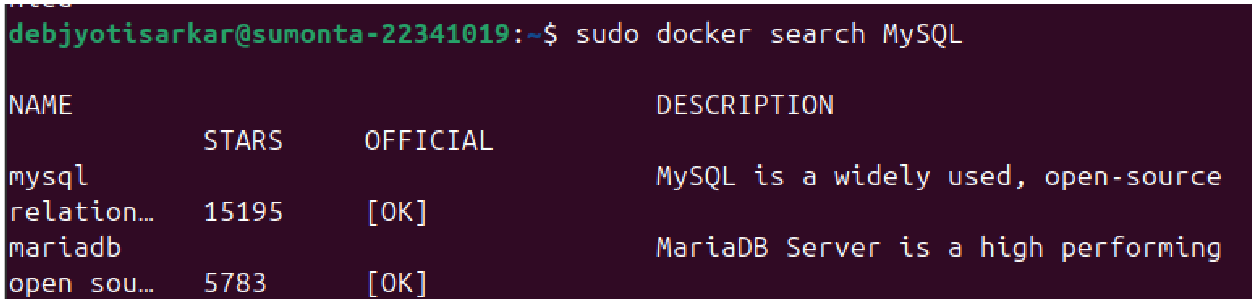
```
debjyotisarkar@sumonta-22341019:~$ sudo docker pull ubuntu
[sudo] password for debjyotisarkar:
Using default tag: latest
latest: Pulling from library/ubuntu
9c704ecd0c69: Pull complete
Digest: sha256:2e863c44b718727c860746568e1d54afd13b2fa71b160f5cd9058fc436217b30
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

This command runs the image iteratively



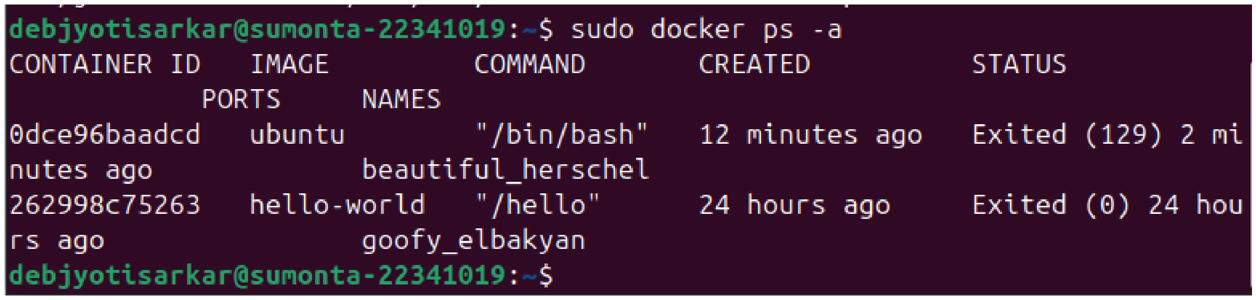
```
debjyotisarkar@sumonta-22341019:~$ sudo docker run -it ubuntu
root@0dce96baadcd:/#
```

This command searches images from the docker registry



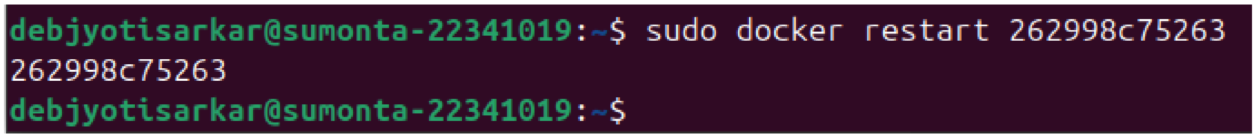
```
debjyotisarkar@sumonta-22341019:~$ sudo docker search MySQL

NAME                                       DESCRIPTION
            STARS     OFFICIAL
mysql                                      MySQL is a widely used, open-source
relation…   15195     [OK]
mariadb                                    MariaDB Server is a high performing
open sou…   5783      [OK]
```

This command shows a list of all the all the containers



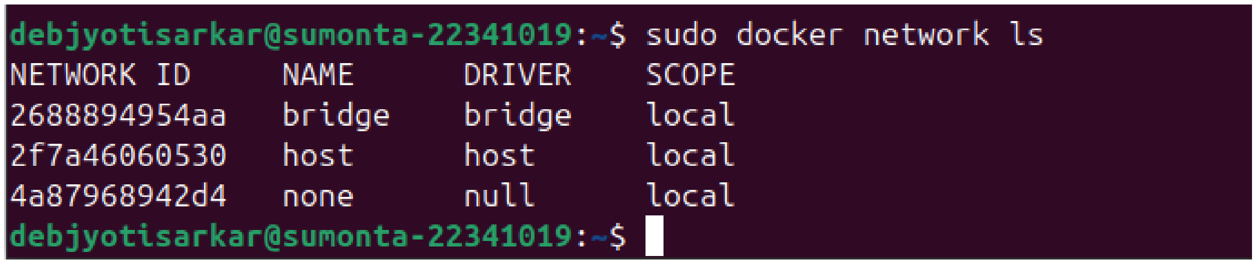
```
debjyotisarkar@sumonta-22341019:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED          STATUS
            PORTS     NAMES
0dce96baadcd   ubuntu        "/bin/bash"   12 minutes ago   Exited (129) 2 mi
nutes ago             beautiful_herschel
262998c75263   hello-world   "/hello"      24 hours ago     Exited (0) 24 hou
rs ago                goofy_elbakyan
debjyotisarkar@sumonta-22341019:~$
```

This command restarts the container using its container ID



```
debjyotisarkar@sumonta-22341019:~$ sudo docker restart 262998c75263
262998c75263
debjyotisarkar@sumonta-22341019:~$
```

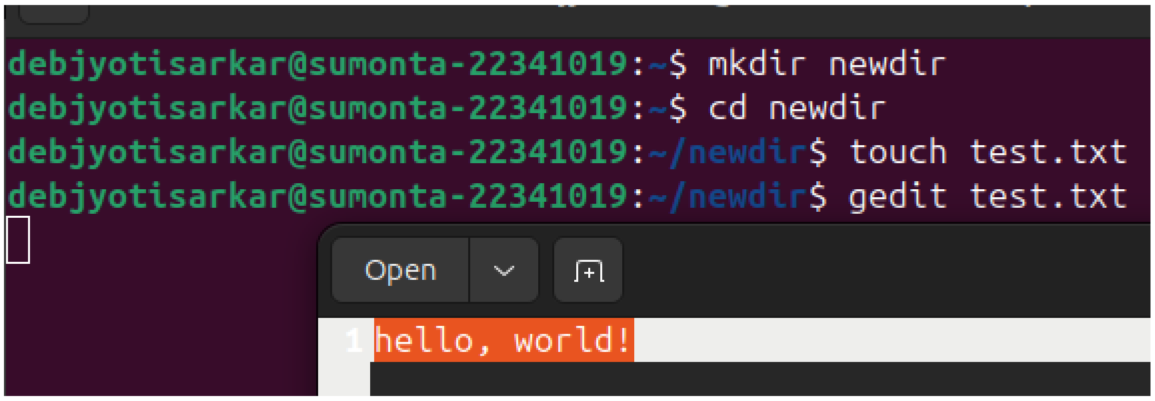
This command shows the list of all the networks available



```
debjyotisarkar@sumonta-22341019:~$ sudo docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
2688894954aa   bridge    bridge    local
2f7a46060530   host      host      local
4a87968942d4   none      null      local
debjyotisarkar@sumonta-22341019:~$
```

Task 3

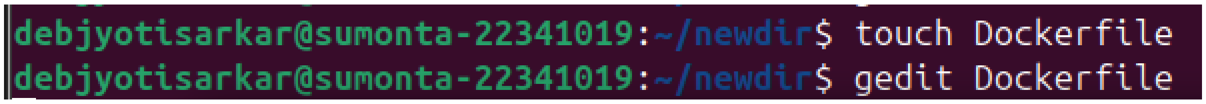
Firstly, let's create a new directory and inside that create a new txt file and write something in it



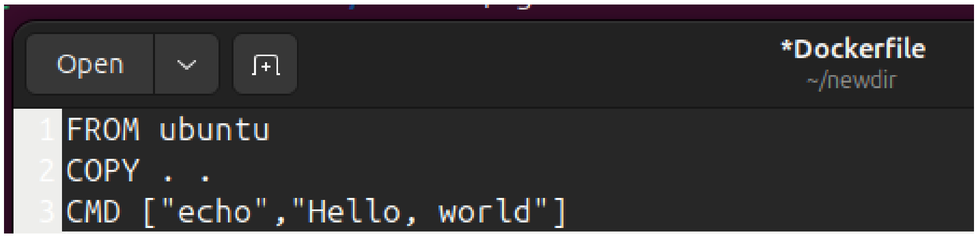
```
debjyotisarkar@sumonta-22341019:~$ mkdir newdir
debjyotisarkar@sumonta-22341019:~$ cd newdir
debjyotisarkar@sumonta-22341019:~/newdir$ touch test.txt
debjyotisarkar@sumonta-22341019:~/newdir$ gedit test.txt
```

```
hello, world!
```

Following that, create a new file named Dockerfile that contains the commands to run

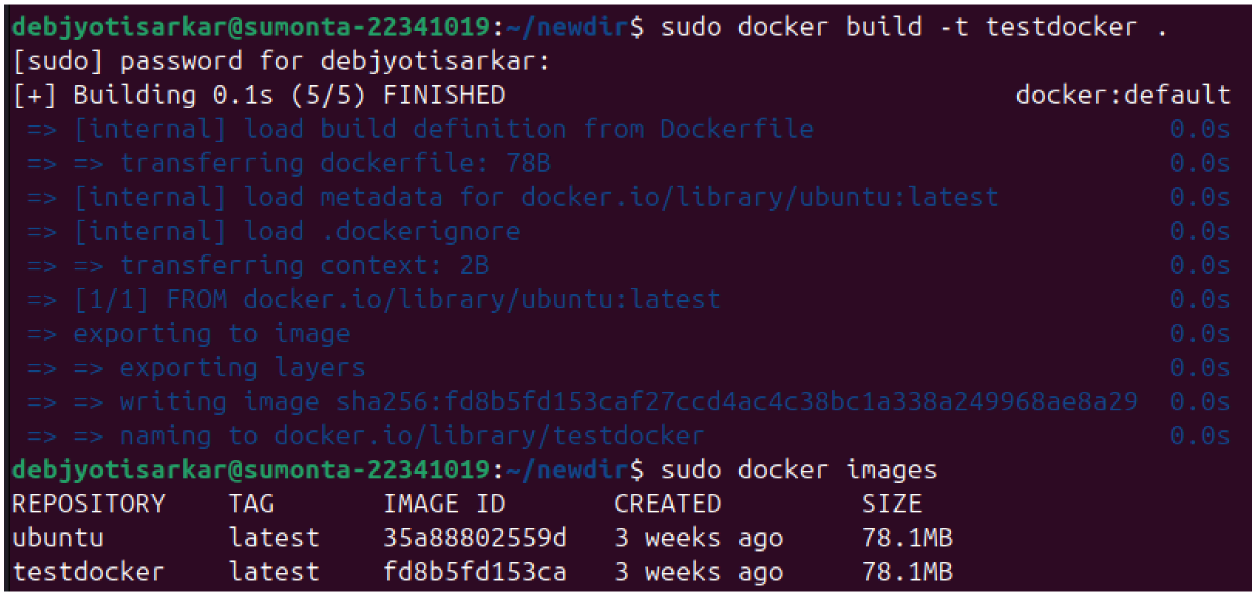


```
debjyotisarkar@sumonta-22341019:~/newdir$ touch Dockerfile
debjyotisarkar@sumonta-22341019:~/newdir$ gedit Dockerfile
```



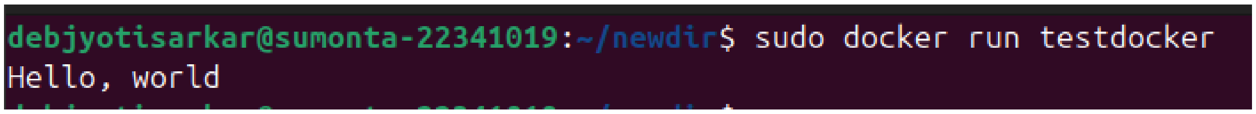
```
FROM ubuntu
COPY . .
CMD ["echo","Hello, world"]
```

This command builds the docker image and *docker images* shows all the images



```
debjyotisarkar@sumonta-22341019:~/newdir$ sudo docker build -t testdocker .
[sudo] password for debjyotisarkar:
[+] Building 0.1s (5/5) FINISHED                                  docker:default
 => [internal] load build definition from Dockerfile                       0.0s
 => => transferring dockerfile: 78B                                        0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest           0.0s
 => [internal] load .dockerignore                                          0.0s
 => => transferring context: 2B                                            0.0s
 => [1/1] FROM docker.io/library/ubuntu:latest                             0.0s
 => exporting to image                                                     0.0s
 => => exporting layers                                                    0.0s
 => => writing image sha256:fd8b5fd153caf27ccd4ac4c38bc1a338a249968ae8a29  0.0s
 => => naming to docker.io/library/testdocker                              0.0s
debjyotisarkar@sumonta-22341019:~/newdir$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    35a88802559d   3 weeks ago    78.1MB
testdocker    latest    fd8b5fd153ca   3 weeks ago    78.1MB
```

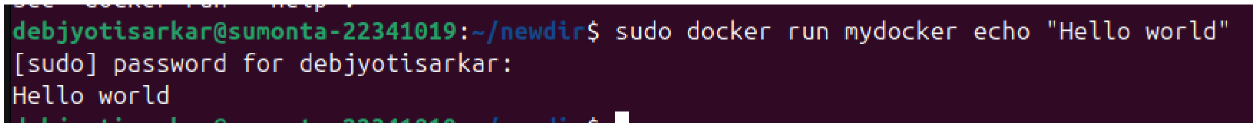
This command runs the docker image and in my case executes the command and outputs "Hello, world"



```
debjyotisarkar@sumonta-22341019:~/newdir$ sudo docker run testdocker
Hello, world
```

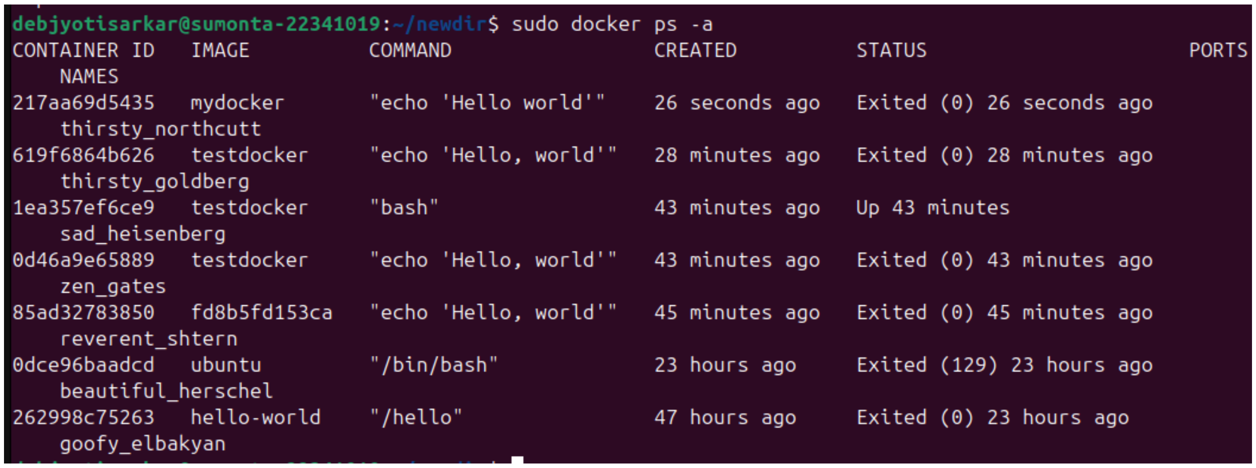
Task 4

This command runs the container directly in one line and show the output in the terminal



```
debjyotisarkar@sumonta-22341019:~/newdir$ sudo docker run mydocker echo "Hello world"
[sudo] password for debjyotisarkar:
Hello world
```

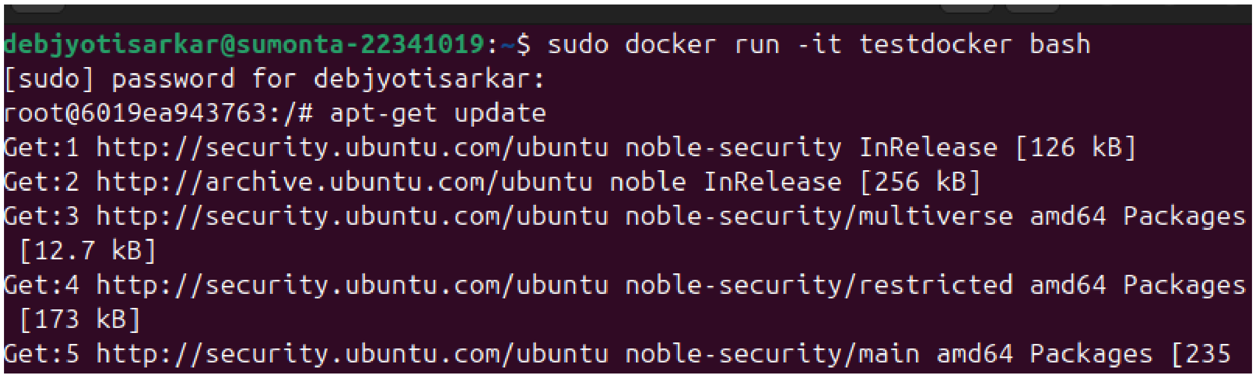
docker ps -a displays all the containers and their informations



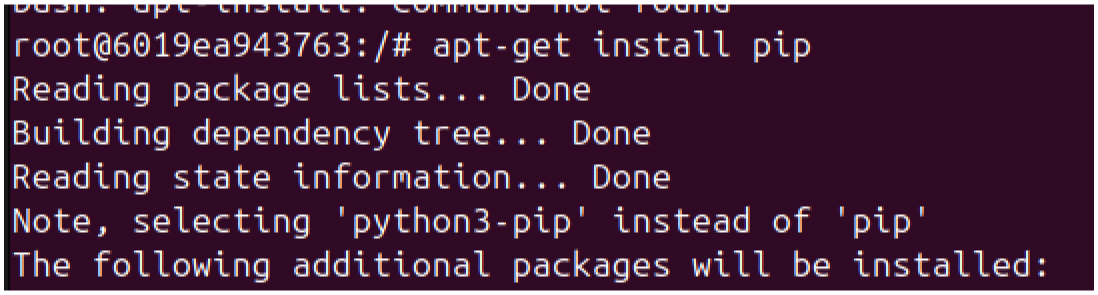
```
debjyotisarkar@sumonta-22341019:~/newdir$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                 CREATED          STATUS                        PORTS
    NAMES
217aa69d5435   mydocker       "echo 'Hello world'"    26 seconds ago   Exited (0) 26 seconds ago
    thirsty_northcutt
619f6864b626   testdocker     "echo 'Hello, world'"   28 minutes ago   Exited (0) 28 minutes ago
    thirsty_goldberg
1ea357ef6ce9   testdocker     "bash"                  43 minutes ago   Up 43 minutes
    sad_heisenberg
0d46a9e65889   testdocker     "echo 'Hello, world'"   43 minutes ago   Exited (0) 43 minutes ago
    zen_gates
85ad32783850   fd8b5fd153ca   "echo 'Hello, world'"   45 minutes ago   Exited (0) 45 minutes ago
    reverent_shtern
0dce96baadcd   ubuntu         "/bin/bash"             23 hours ago     Exited (129) 23 hours ago
    beautiful_herschel
262998c75263   hello-world    "/hello"                47 hours ago     Exited (0) 23 hours ago
    goofy_elbakyan
```

Task 5

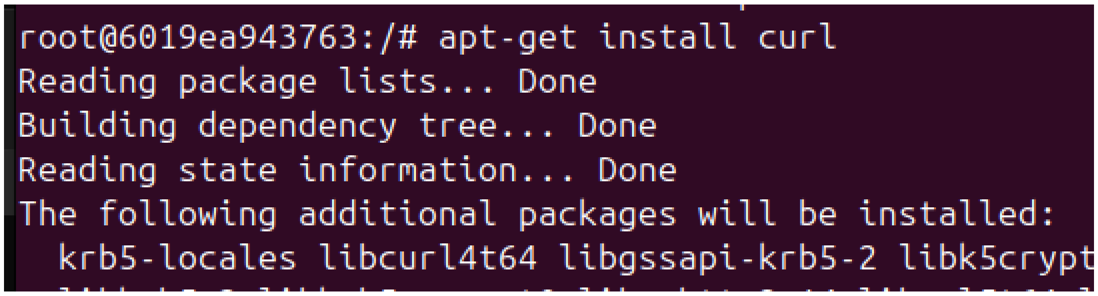
docker run -it runs the container in iterative mode, inside the container I first updated and then installed a few packages like pip, curl, git, net-tools



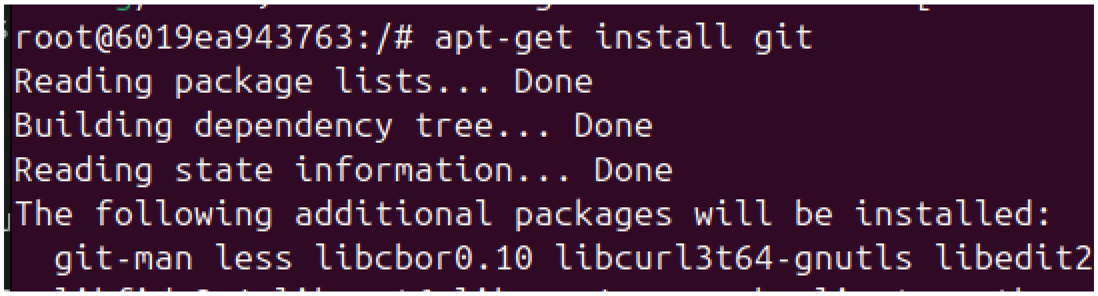
```
debjyotisarkar@sumonta-22341019:~$ sudo docker run -it testdocker bash
[sudo] password for debjyotisarkar:
root@6019ea943763:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages
 [12.7 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages
 [173 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [235
```



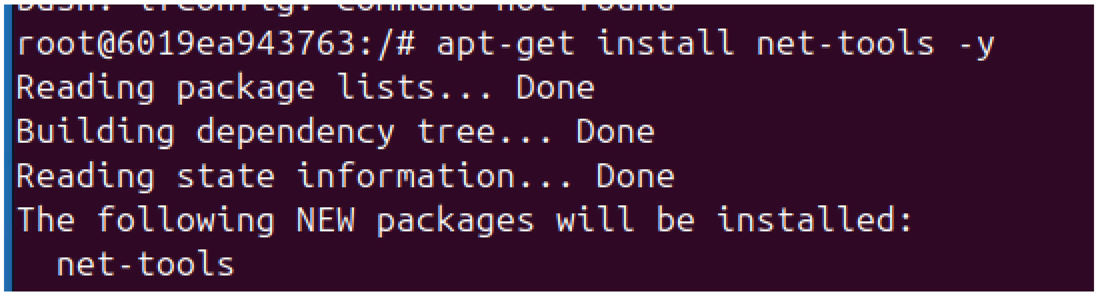
```
bash: apt-install: command not found
root@6019ea943763:/# apt-get install pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'python3-pip' instead of 'pip'
The following additional packages will be installed:
```



```
root@6019ea943763:/# apt-get install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  krb5-locales libcurl4t64 libgssapi-krb5-2 libk5crypt
```



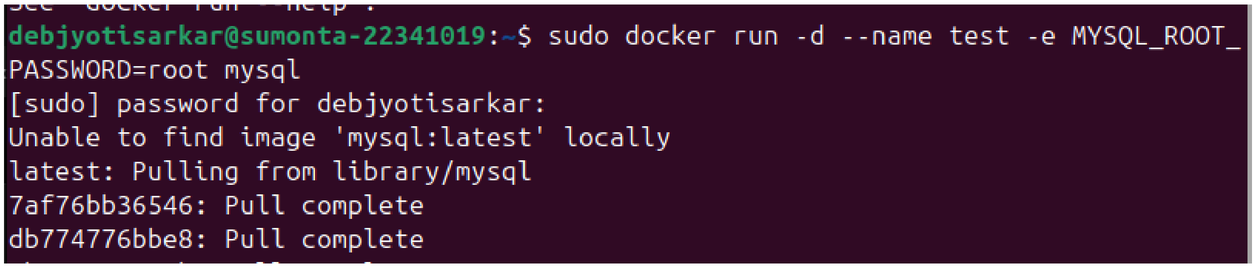
```
root@6019ea943763:/# apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man less libcbor0.10 libcurl3t64-gnutls libedit2
```



```
bash: ifconfig: command not found
root@6019ea943763:/# apt-get install net-tools -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
```

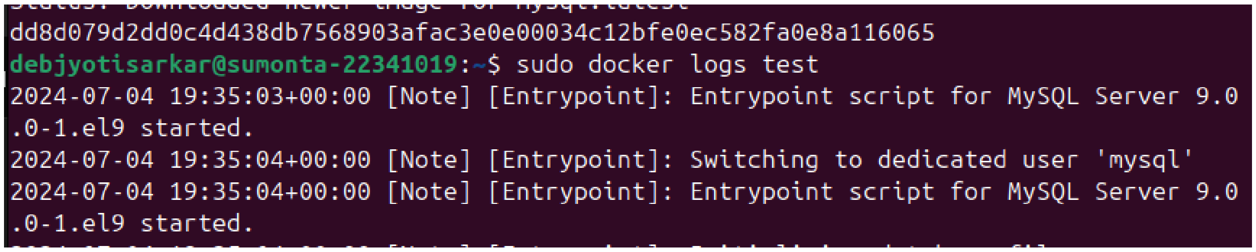
Task 6

This command runs the database in detached mode (in the background), assigns a name to the container and sets the password to root



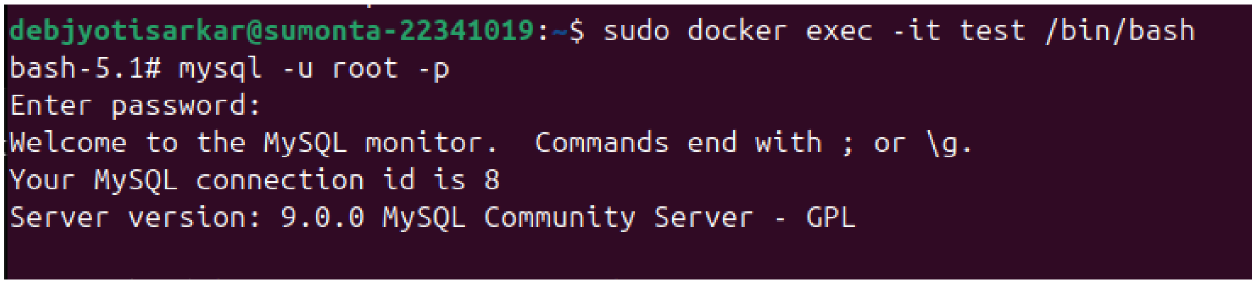
```
debjyotisarkar@sumonta-22341019:~$ sudo docker run -d --name test -e MYSQL_ROOT_
PASSWORD=root mysql
[sudo] password for debjyotisarkar:
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
7af76bb36546: Pull complete
db774776bbe8: Pull complete
```

This command shows the logs of the container



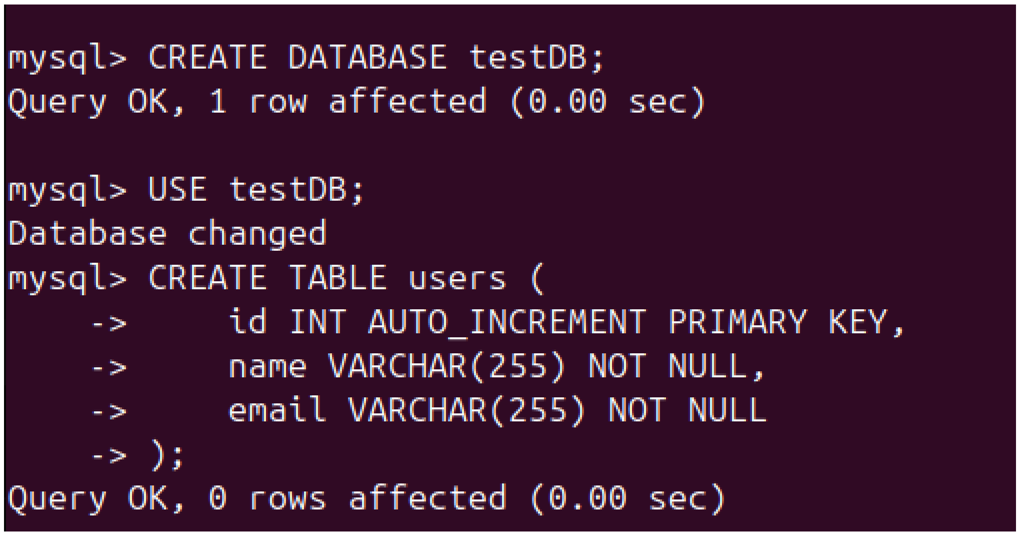
```
dd8d079d2dd0c4d438db7568903afac3e0e00034c12bfe0ec582fa0e8a116065
debjyotisarkar@sumonta-22341019:~$ sudo docker logs test
2024-07-04 19:35:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.0
.0-1.el9 started.
2024-07-04 19:35:04+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2024-07-04 19:35:04+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.0
.0-1.el9 started.
```

This command interactive session within the test container and the following command accesses the mySQL shell



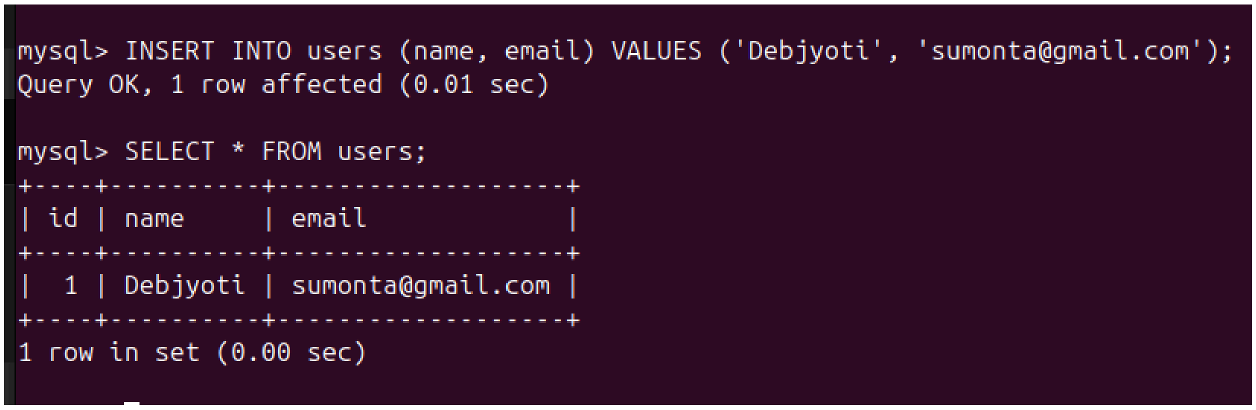
```
debjyotisarkar@sumonta-22341019:~$ sudo docker exec -it test /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.0 MySQL Community Server - GPL
```

This command is used to create and then use the database. Following command creates the table with attributes.



```
mysql> CREATE DATABASE testDB;
Query OK, 1 row affected (0.00 sec)

mysql> USE testDB;
Database changed
mysql> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     email VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.00 sec)
```

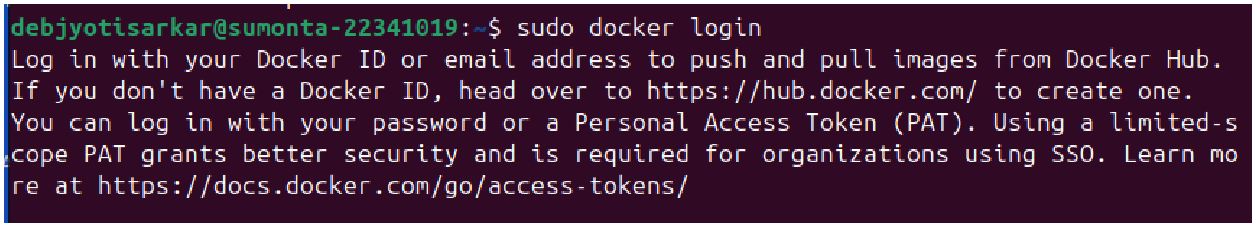
This command inserts the values into the table. The next command shows all the entries in the table.



```
mysql> INSERT INTO users (name, email) VALUES ('Debjyoti', 'sumonta@gmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM users;
+----+----------+-------------------+
| id | name     | email             |
+----+----------+-------------------+
|  1 | Debjyoti | sumonta@gmail.com |
+----+----------+-------------------+
1 row in set (0.00 sec)
```

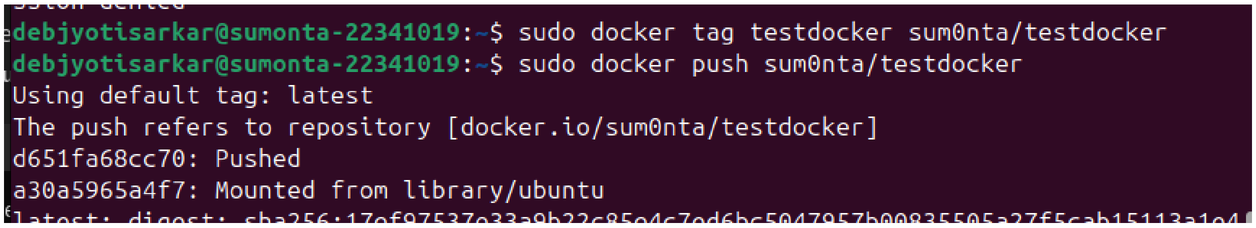
Task 7

In order to push my image into the docker public registry, first we need to login to docker using this command.



```
debjyotisarkar@sumonta-22341019:~$ sudo docker login
Log in with your Docker ID or email address 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.
You can log in with your password or a Personal Access Token (PAT). Using a limited-s
cope PAT grants better security and is required for organizations using SSO. Learn mo
re at https://docs.docker.com/go/access-tokens/
```

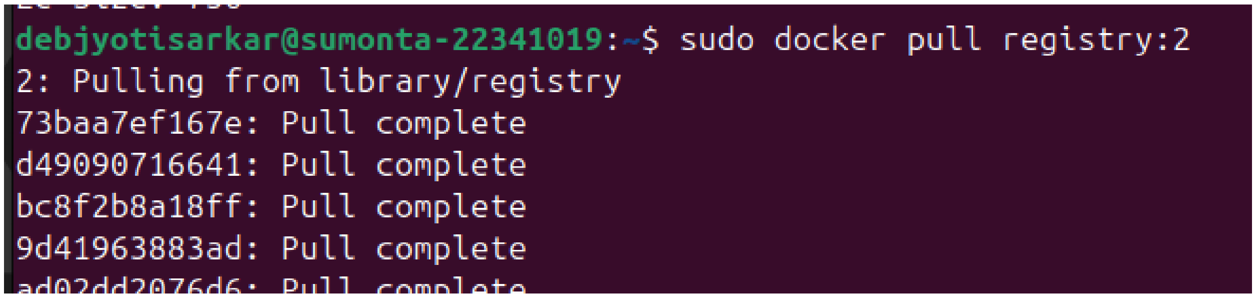
Following that, I need to tag the image I want to upload to my username/imagename and push the image in the directory and the image is uploaded to the docker registry.



```
debjyotisarkar@sumonta-22341019:~$ sudo docker tag testdocker sum0nta/testdocker
debjyotisarkar@sumonta-22341019:~$ sudo docker push sum0nta/testdocker
Using default tag: latest
The push refers to repository [docker.io/sum0nta/testdocker]
d651fa68cc70: Pushed
a30a5965a4f7: Mounted from library/ubuntu
latest: digest: sha256:17ef97537e33a9b22c85e4c7ed6bc5047957b00835505a27f5cab15113a1e4
```

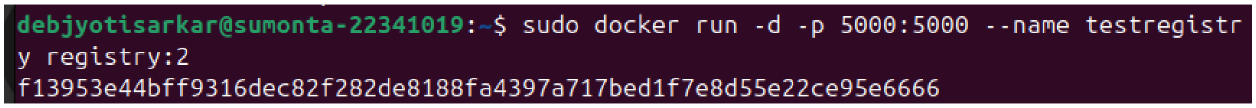
Task 8

Firstly, I need to pull the image from the docker hub.



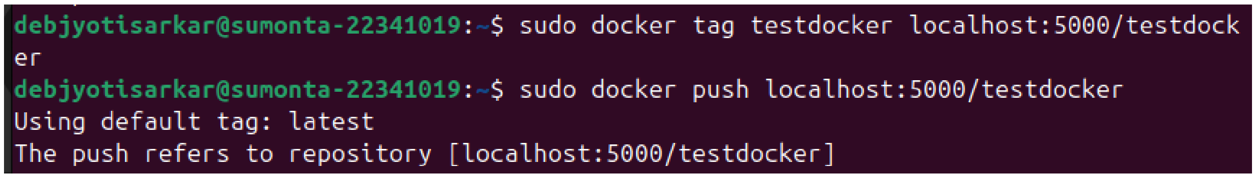
```
debjyotisarkar@sumonta-22341019:~$ sudo docker pull registry:2
2: Pulling from library/registry
73baa7ef167e: Pull complete
d49090716641: Pull complete
bc8f2b8a18ff: Pull complete
9d41963883ad: Pull complete
ad02dd2076d6: Pull complete
```

Following that, run the container in a detached mode with port 5000:5000 (host:guest) and name the container.



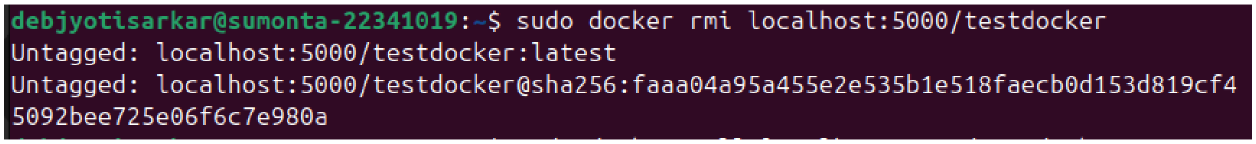
```
debjyotisarkar@sumonta-22341019:~$ sudo docker run -d -p 5000:5000 --name testregistr
y registry:2
f13953e44bff9316dec82f282de8188fa4397a717bed1f7e8d55e22ce95e6666
```

Next, the image needs to be tagged and pushed to the private local registry.

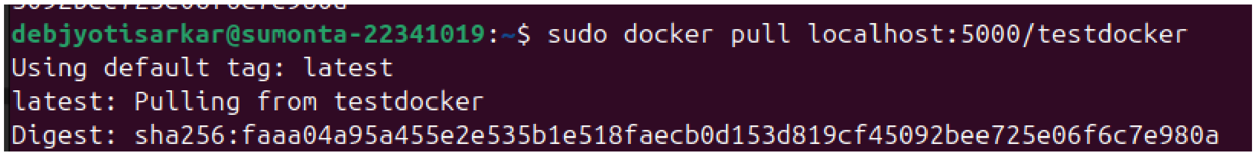


```
debjyotisarkar@sumonta-22341019:~$ sudo docker tag testdocker localhost:5000/testdock
er
debjyotisarkar@sumonta-22341019:~$ sudo docker push localhost:5000/testdocker
Using default tag: latest
The push refers to repository [localhost:5000/testdocker]
```

This image can also be removed or pulled from the local registry.



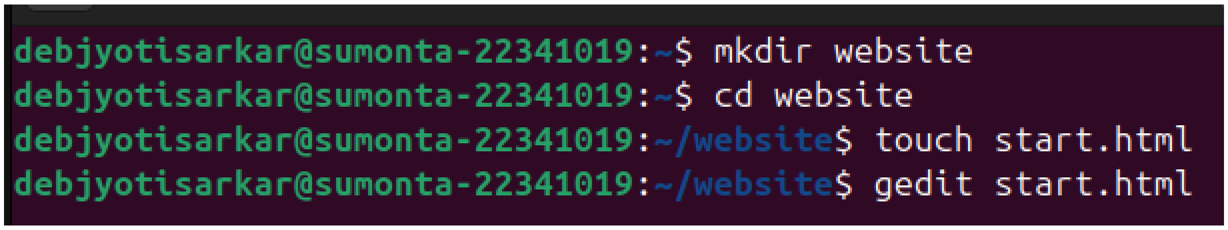
```
debjyotisarkar@sumonta-22341019:~$ sudo docker rmi localhost:5000/testdocker
Untagged: localhost:5000/testdocker:latest
Untagged: localhost:5000/testdocker@sha256:faaa04a95a455e2e535b1e518faecb0d153d819cf4
5092bee725e06f6c7e980a
```



```
debjyotisarkar@sumonta-22341019:~$ sudo docker pull localhost:5000/testdocker
Using default tag: latest
latest: Pulling from testdocker
Digest: sha256:faaa04a95a455e2e535b1e518faecb0d153d819cf45092bee725e06f6c7e980a
```

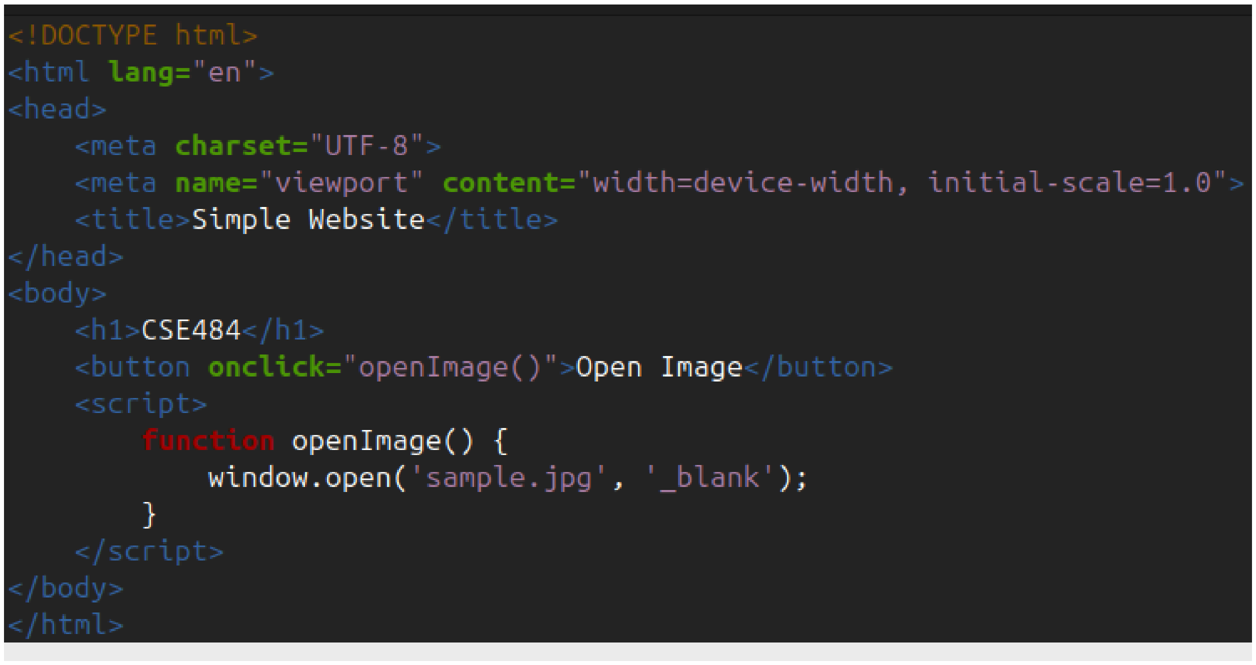
Task 9

First, we need to make a directory and create a html file inside that directory.



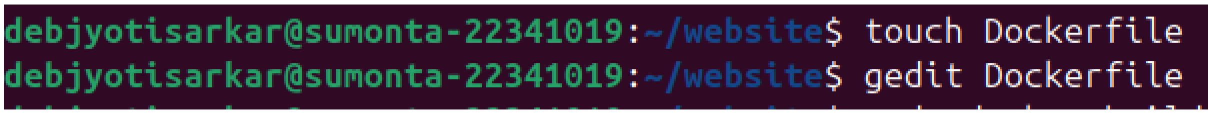
```
debjyotisarkar@sumonta-22341019:~$ mkdir website
debjyotisarkar@sumonta-22341019:~$ cd website
debjyotisarkar@sumonta-22341019:~/website$ touch start.html
debjyotisarkar@sumonta-22341019:~/website$ gedit start.html
```

Inside the HTML file, I made a simple button that opens up a picture.

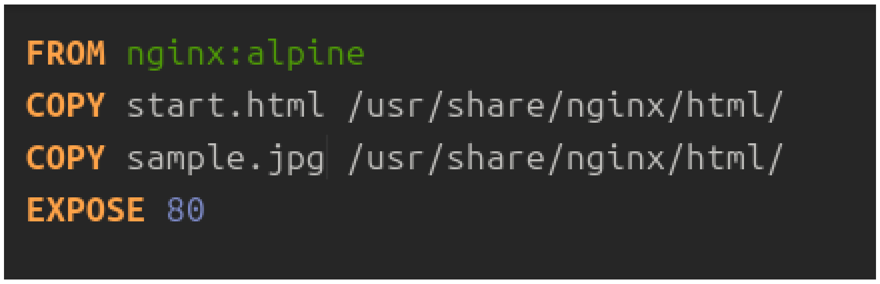


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Website</title>
</head>
<body>
    <h1>CSE484</h1>
    <button onclick="openImage()">Open Image</button>
    <script>
        function openImage() {
            window.open('sample.jpg', '_blank');
        }
    </script>
</body>
</html>
```

We also need to make a dockerfile and expose it to port 80.

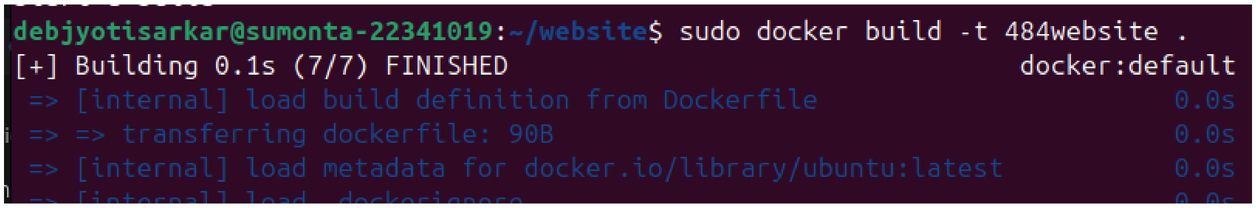


```
debjyotisarkar@sumonta-22341019:~/website$ touch Dockerfile
debjyotisarkar@sumonta-22341019:~/website$ gedit Dockerfile
```

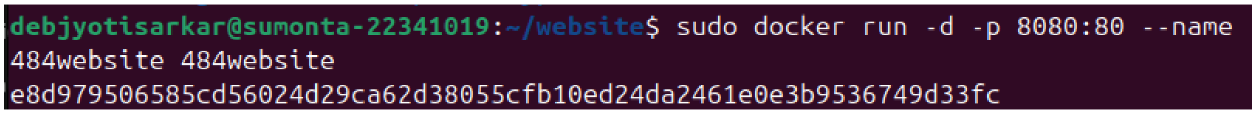


```
FROM nginx:alpine
COPY start.html /usr/share/nginx/html/
COPY sample.jpg /usr/share/nginx/html/
EXPOSE 80
```

Next, we need to build an image and run that image in the 8080:80 port.

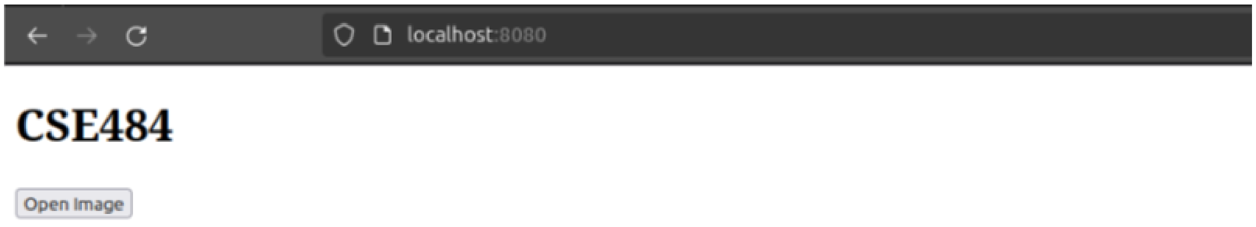


```
debjyotisarkar@sumonta-22341019:~/website$ sudo docker build -t 484website .
[+] Building 0.1s (7/7) FINISHED                                  docker:default
 => [internal] load build definition from Dockerfile                         0.0s
 => => transferring dockerfile: 90B                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest             0.0s
```



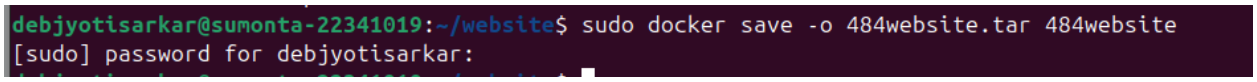
```
debjyotisarkar@sumonta-22341019:~/website$ sudo docker run -d -p 8080:80 --name
484website 484website
e8d979506585cd56024d29ca62d38055cfb10ed24da2461e0e3b9536749d33fc
```

Now, when we open the *localhost:8080* in the host machine, it shows the website.

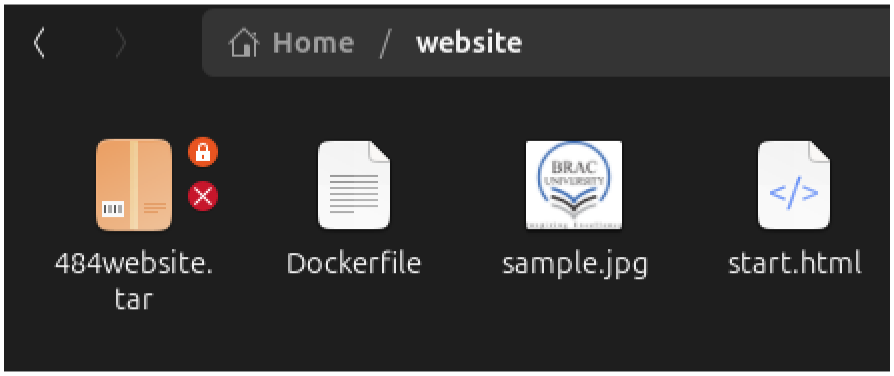


Task 10

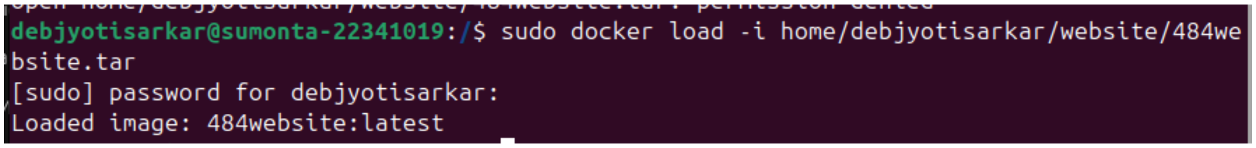
First, we need to save the .tar file for the image using docker save command.



```
debjyotisarkar@sumonta-22341019:~/website$ sudo docker save -o 484website.tar 484website
[sudo] password for debjyotisarkar:
```

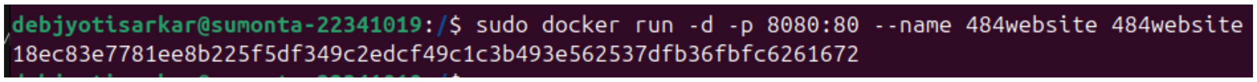


Next, the file needs to be transferred to another machine using a flash drive or through other means and loaded in the new machine.



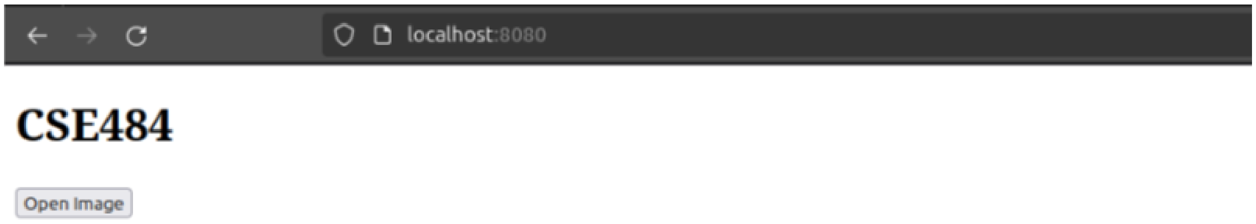
```
debjyotisarkar@sumonta-22341019:/$ sudo docker load -i home/debjyotisarkar/website/484we
bsite.tar
[sudo] password for debjyotisarkar:
Loaded image: 484website:latest
```

The loaded image has to be ran using docker run and the appropriate port number.



```
debjyotisarkar@sumonta-22341019:/$ sudo docker run -d -p 8080:80 --name 484website 484website
18ec83e7781ee8b225f5df349c2edcf49c1c3b493e562537dfb36fbfc6261672
```

Now if we open the *localhost:8080* page we get the webpage.



localhost:8080

CSE484