EXPERIMENT-08

NAME: SHAIKH MUBASHIRA TUFEL AHMED
ROLL NO: 612055 **COURSE:** ADVANCE DEVOPS(ITL504)
BRANCH: T.E. INFORMATION TECHNOLOGY (SEM 5)

1. **What is hub.docker.com?**

➔**Docker Hub** is a repository service and it is a cloud-based service where people push their Docker Container Images and also pull the Docker Container Images from the **Docker Hub** anytime or anywhere via the internet. It provides features such as you can push your images as private or public. Mainly DevOps team uses the Docker Hub. It is an open-source tool and freely available for all operating systems. It is like storage where we store the images and pull the images when it is required. When a person wants to push/pull images from the Docker Hub they must have a basic knowledge of Docker. Let us discuss the requirements of the Docker tool.

2. **What is docker hub used for?**

➢ Docker Hub Features:

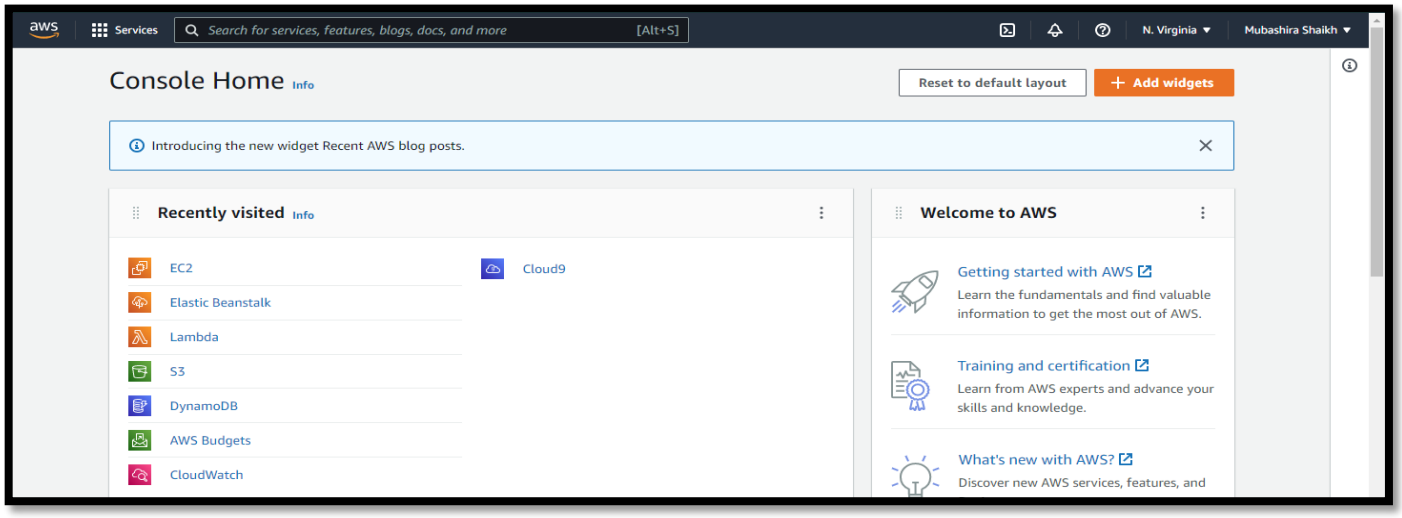
- **Repositories:** It contains the Push and Pull process for container images.
- **Teams and Organizations:** It allows access to developer/user to private repositories of container images.
- **Docker Official Images:** It Pulls and uses high-standard quality container images rendered by Docker.
- **Docker Verified Publisher Images:** It Pulls and uses high-standard quality container images rendered by outside vendors.
- **Builds:** It provides the mechanisms that automatically formulate container images from Bitbucket and GitHub and push them to Docker Hub.
- **Webhooks:** It triggers certain actions after a successful push to a container to combine Docker Hub with additional services.

3. **Install docker on AWS EC2 –Ubuntu by using curl**

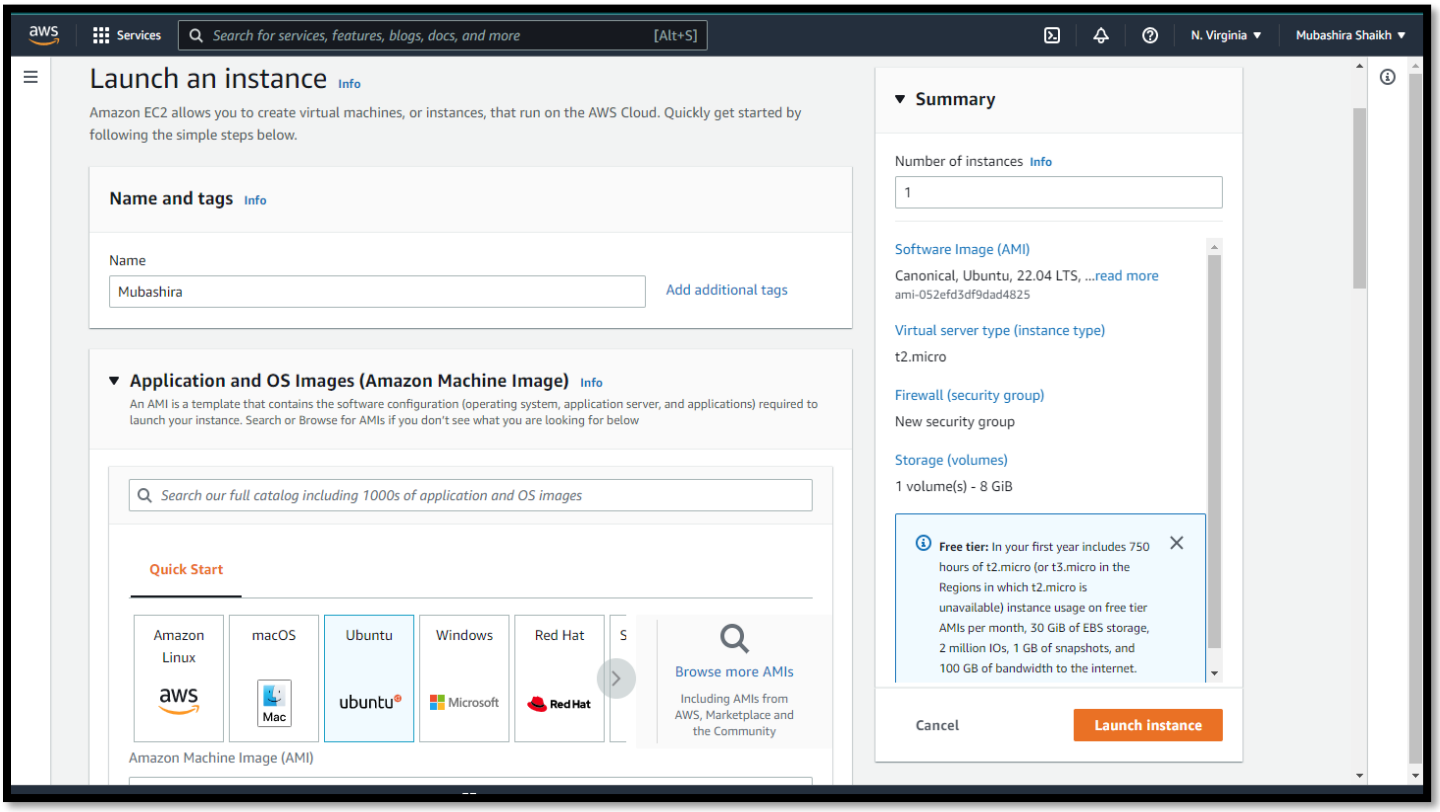
```
#curl -fsSL https://get.docker.com -o get-docker.sh
#sh get-docker.sh
```

4. **Run hello-world from docker hub and explain the steps**
5. **Pull 3 or 4 images, one of the python, run “Hello World “inside container.**
6. **Demonstrate any 15 docker command and explain its uses.**

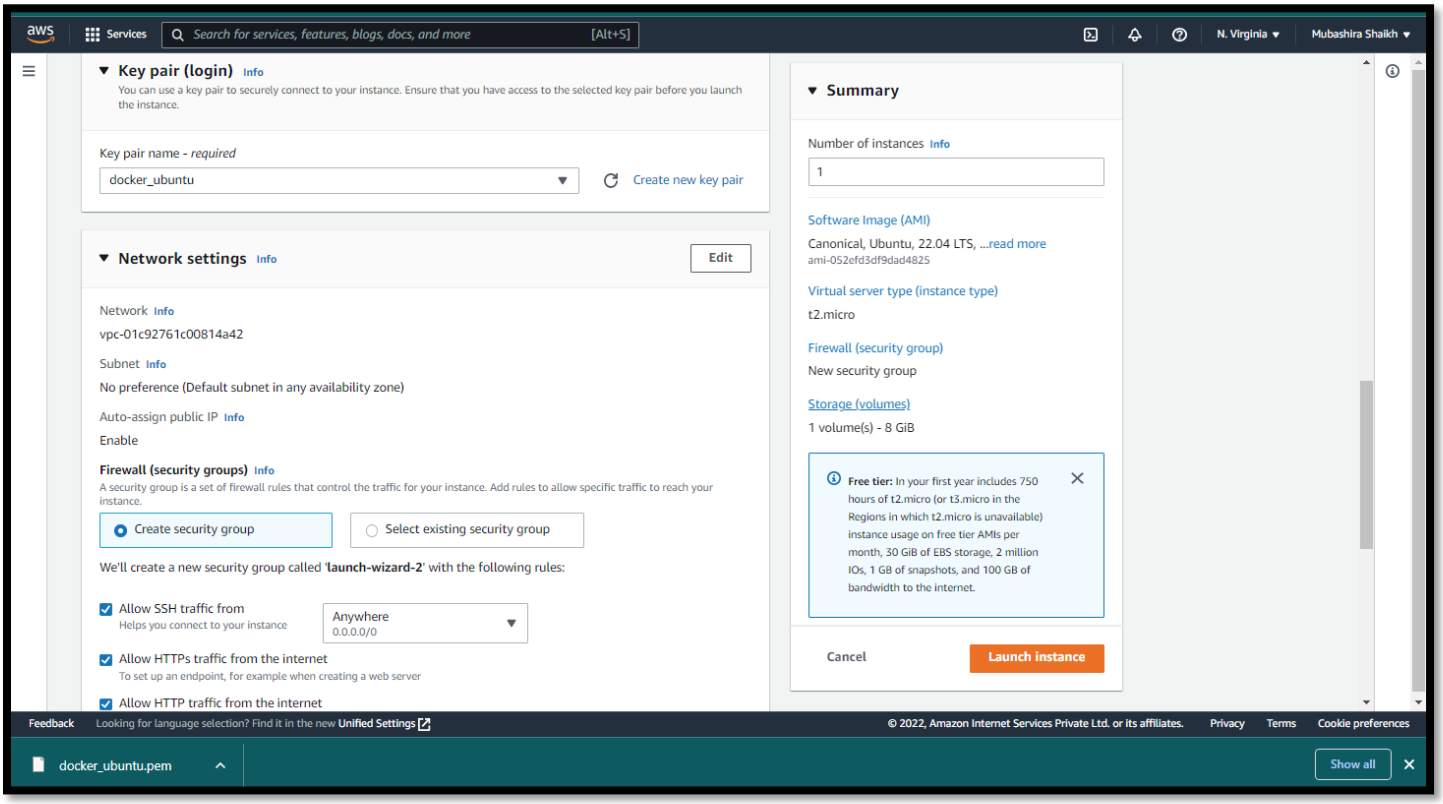
Step 1: AWS Management Console Dashboard.



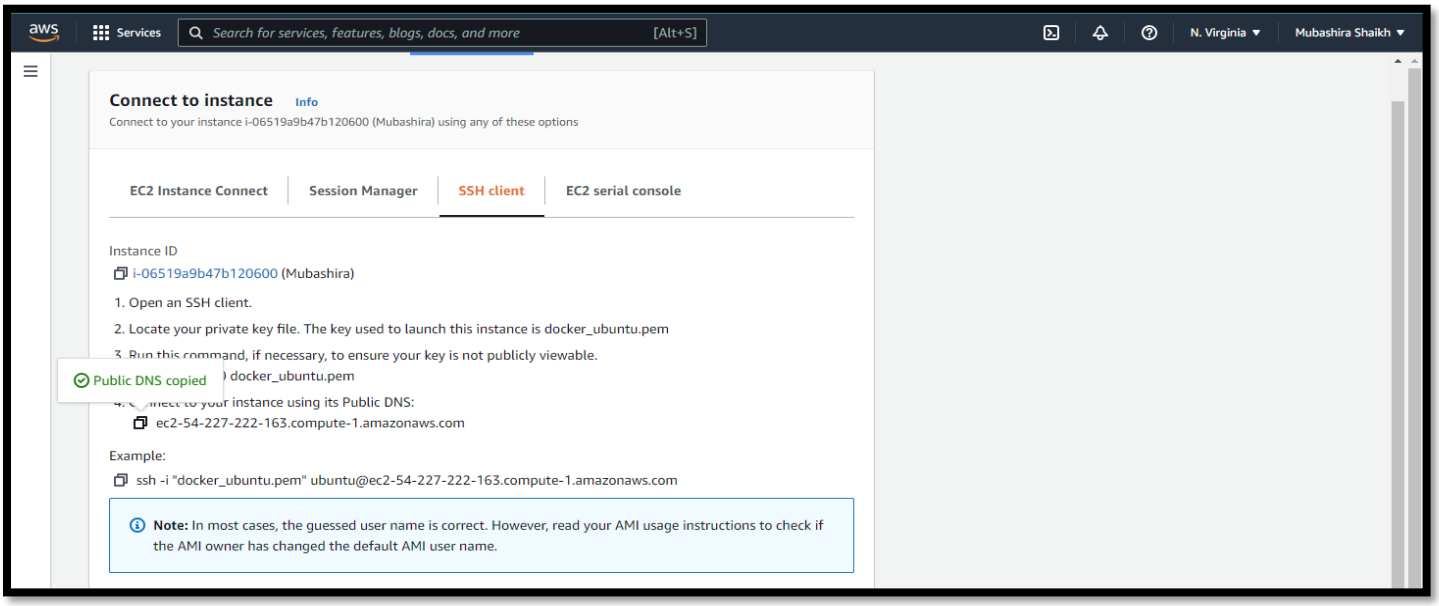
Step 2: Search for EC2→ Click on Launch instance→ Give a name to your instance and create an Ubuntu instance with 20.04 lts version.

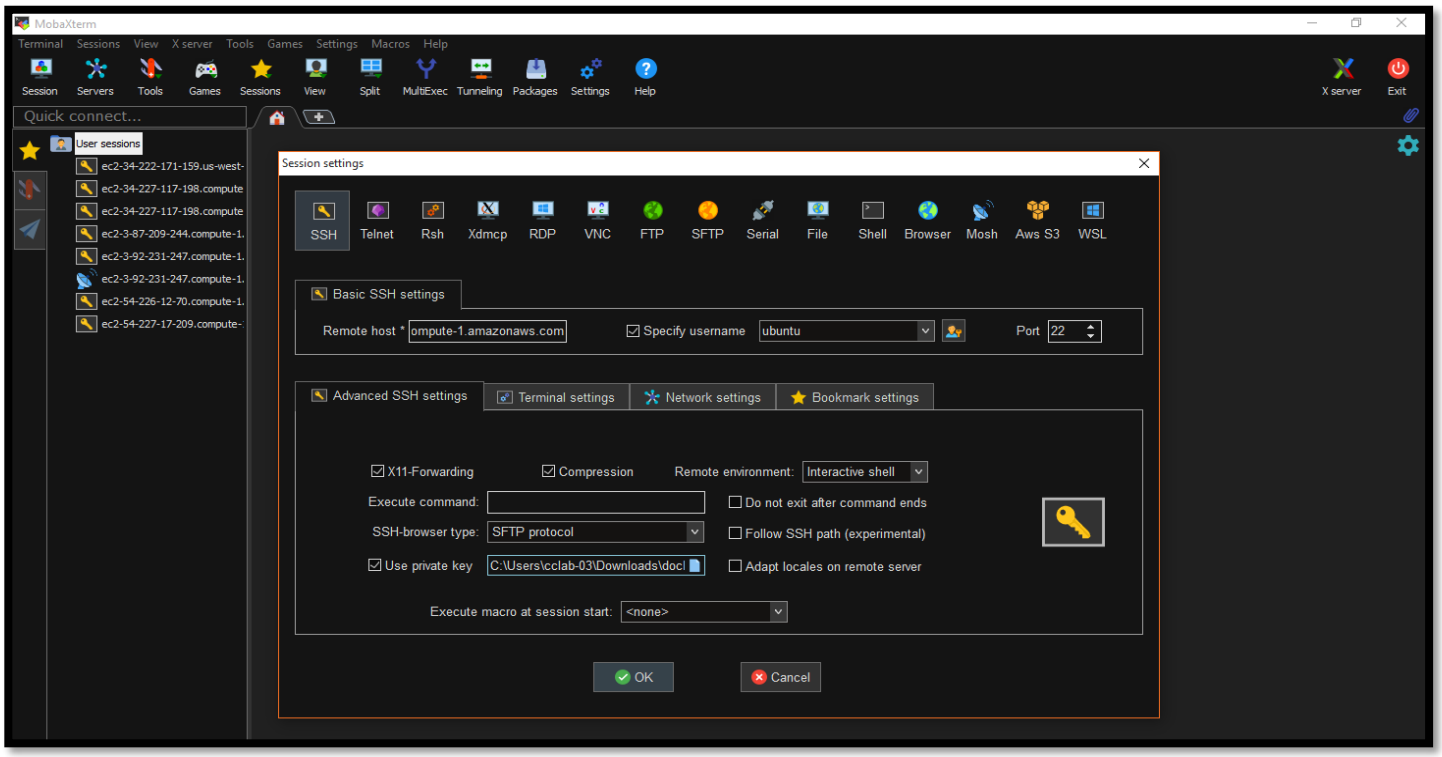


Step 3: Create new key pair for your instance→Network Settings→allow the HTTPS and HTTP traffic.



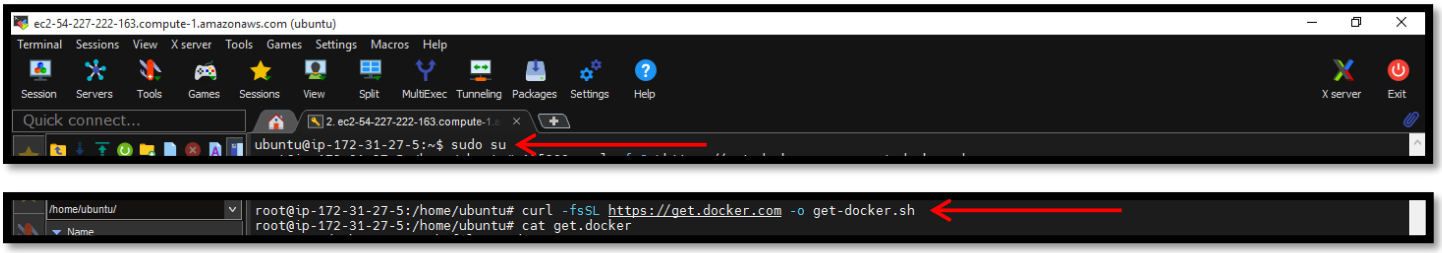
Step 4: Launch MobaXterm→Select SSH session→Copy the public DNS of your instance and paste it into the remote host. Use the downloaded key pair as the private key.

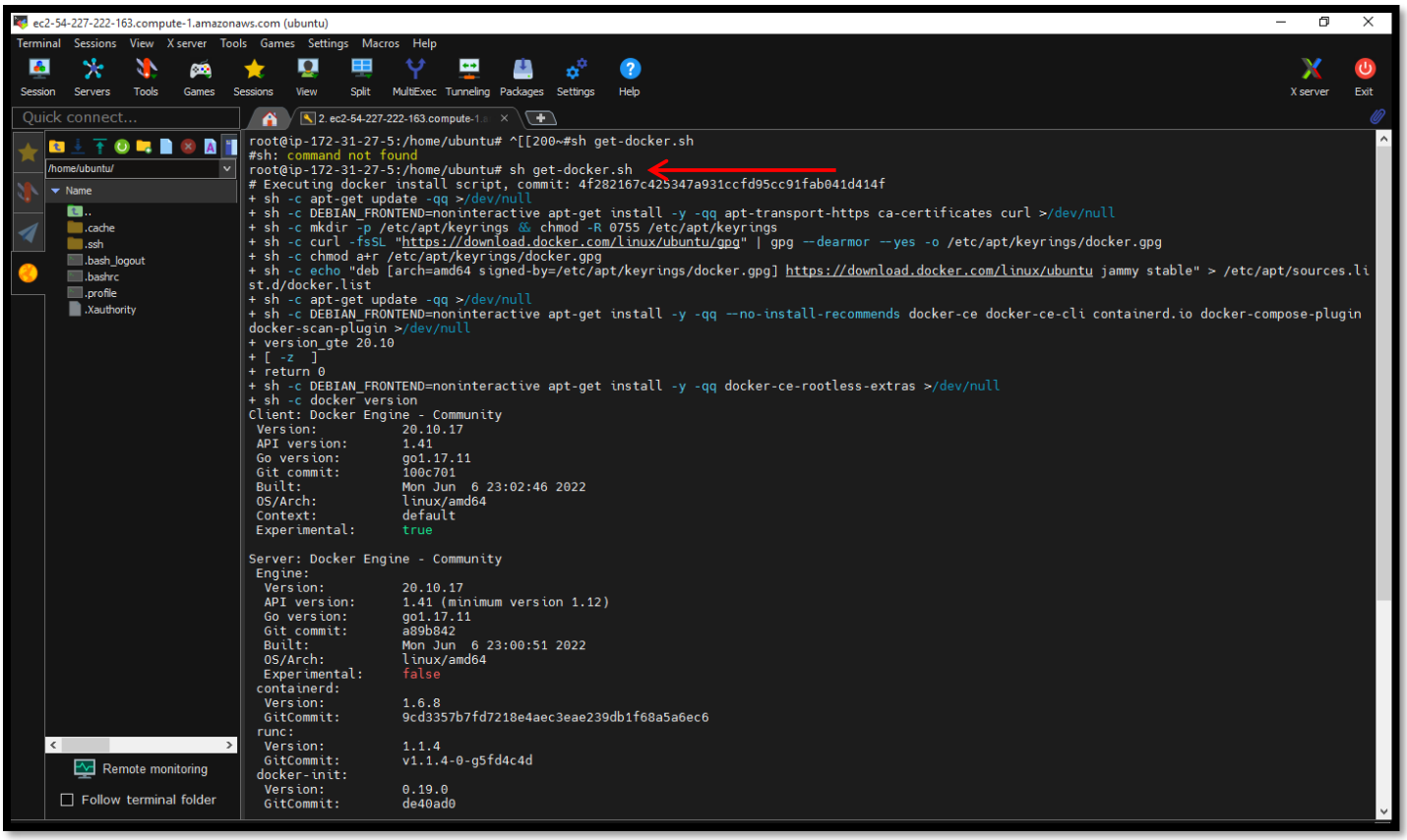




Step 5: Run the command 'sudo su' to gain root user access.
Then enter commands:

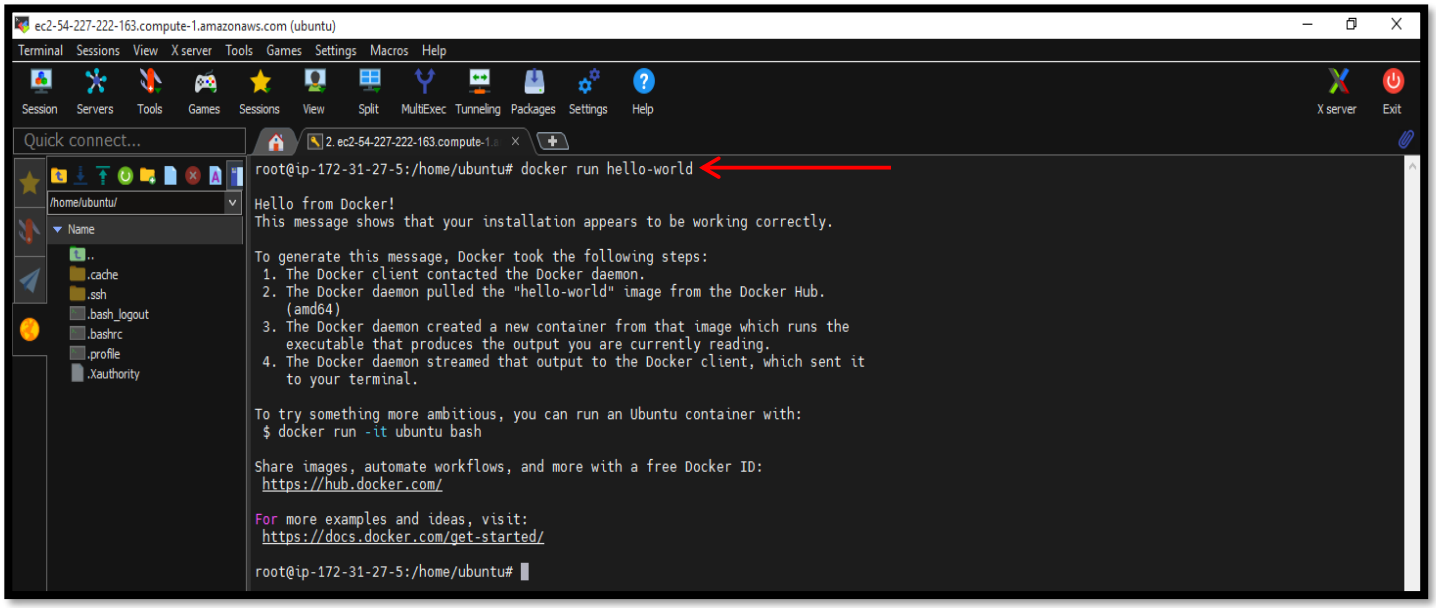
→curl -fsSL https://get.docker.com -o get-docker.sh and
→sh get-docker.sh





Step 6: Enter command 'docker –version' to see current docker version.

Step 7: Enter command 'docker images' to see installed images. At the beginning, there will be no images in the repository. Run command 'docker run hello-world' which will pull a hello-world image and run it.

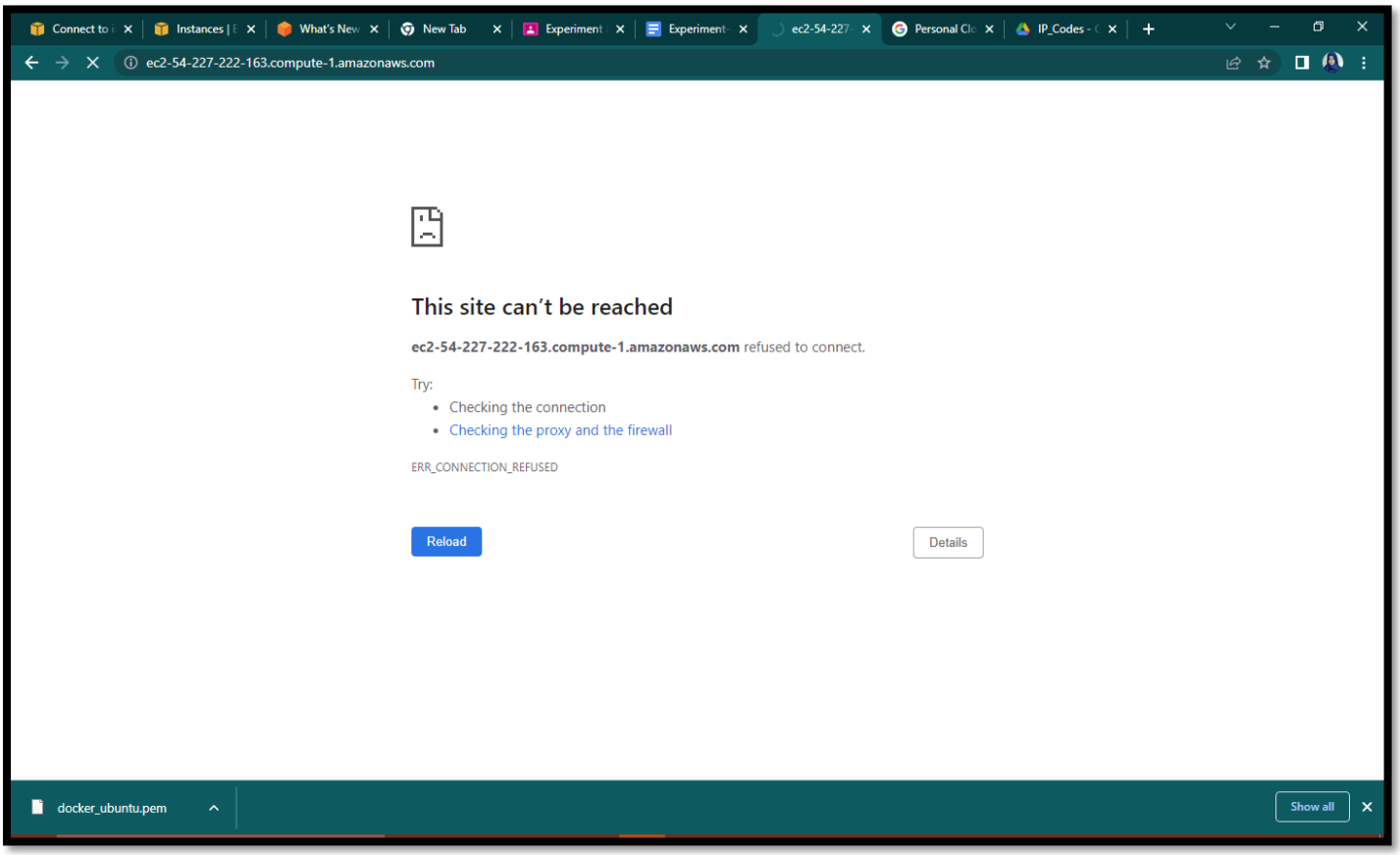


Step 8: Now, run ‘docker images’ again, the repository will have an image named ‘hello world’.

```
root@ip-172-31-27-5:/home/ubuntu# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   11 months ago   13.3kB
root@ip-172-31-27-5:/home/ubuntu#
```

Step 9: Run command ‘docker run -it -p 80:80 --name (name of webpage) nginx bash’. Now, to launch the nginx web server, copy the IPV4 address from the EC2 instance details and paste it into a web browser.

```
root@ip-172-31-27-5:/home/ubuntu# docker run -it -p 80:80 --name mhss-webpage nginx bash
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7a6db449b51b: Pull complete
ca1981974b58: Pull complete
d4019c921e20: Pull complete
7cb804d746d4: Pull complete
e7a561826262: Pull complete
7247f6e5c182: Pull complete
Digest: sha256:b95a99feebf7797479e0c5eb5ec0bdfa5d9f504bc94da550c2f58e839ea6914f
Status: Downloaded newer image for nginx:latest
root@2a15b51a6b78:/#
```

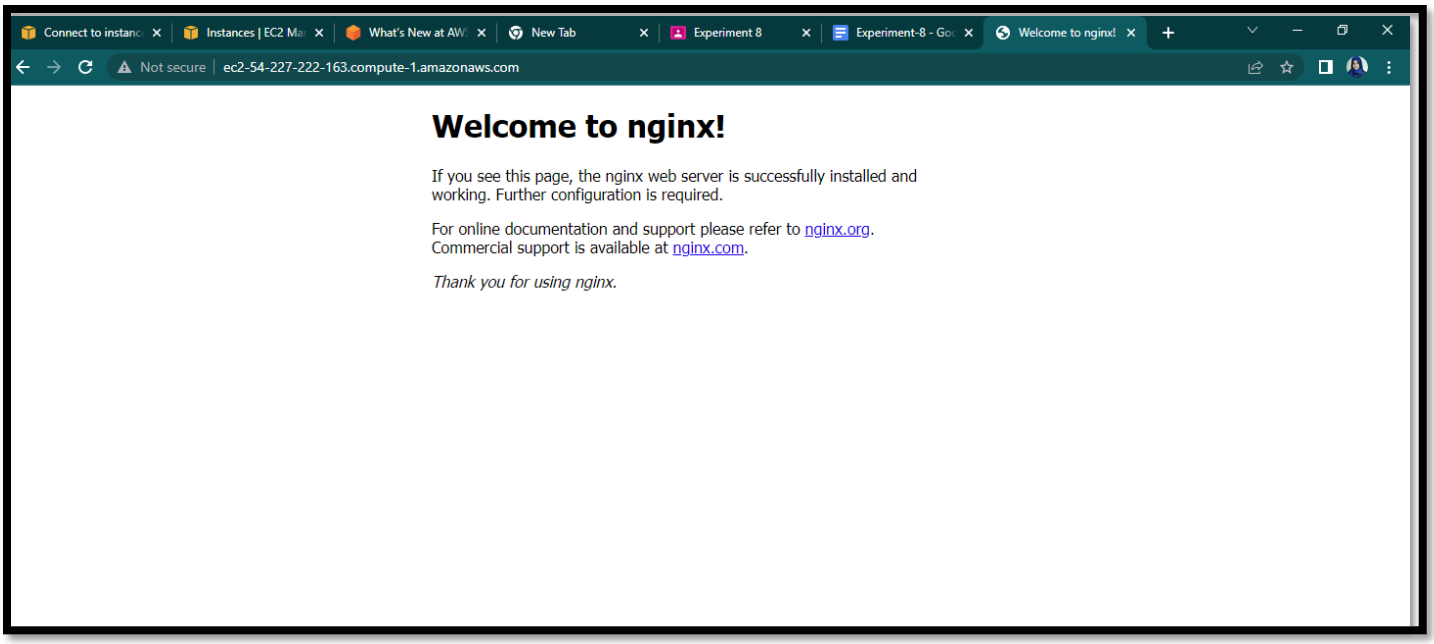


Step 10: Within the container use command ‘service nginx start’ to deploy the web server. After deploying the web server, the web page will be visible without any errors.

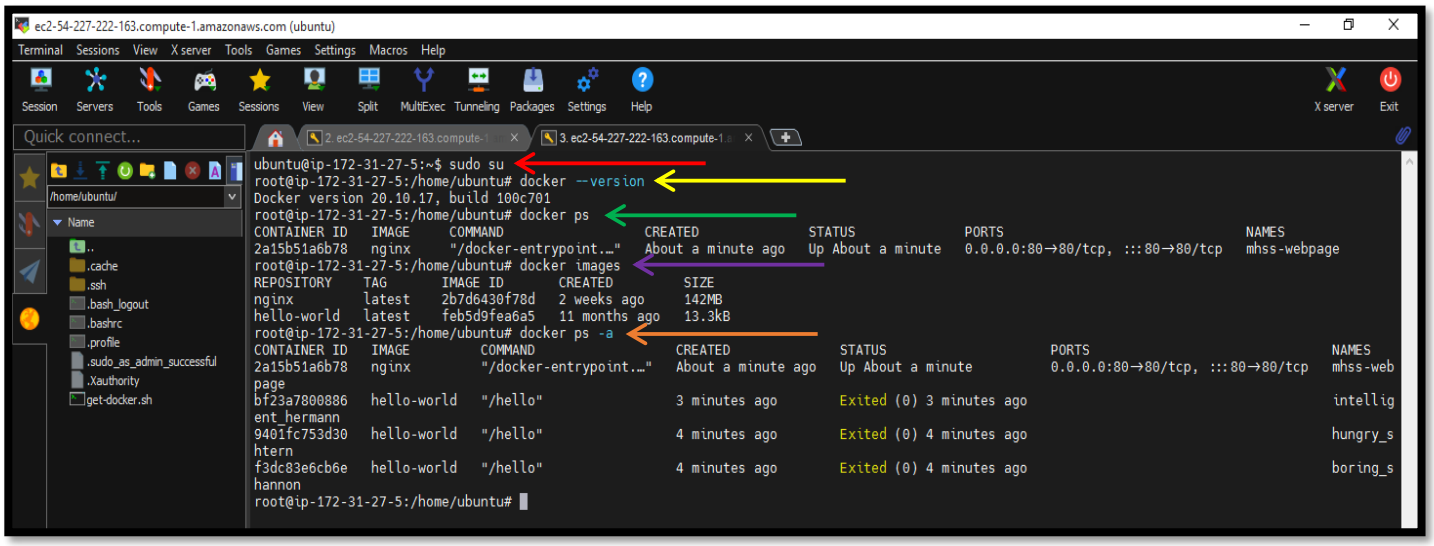
```
root@2a15b51a6b78:/# service nginx start
2022/09/08 06:52:09 [notice] 12#12: using the "epoll" event method
2022/09/08 06:52:09 [notice] 12#12: nginx/1.23.1
2022/09/08 06:52:09 [notice] 12#12: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/09/08 06:52:09 [notice] 12#12: OS: Linux 5.15.0-1011-aws
2022/09/08 06:52:09 [notice] 12#12: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/09/08 06:52:09 [notice] 13#13: start worker processes
2022/09/08 06:52:09 [notice] 13#13: start worker process 14
root@2a15b51a6b78:/#
```

Remote monitoring

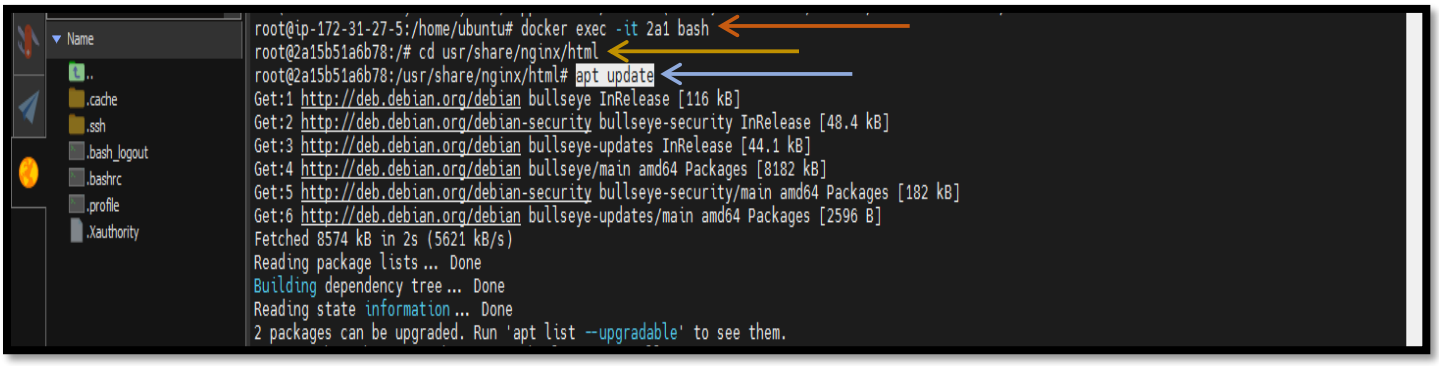
Follow terminal folder



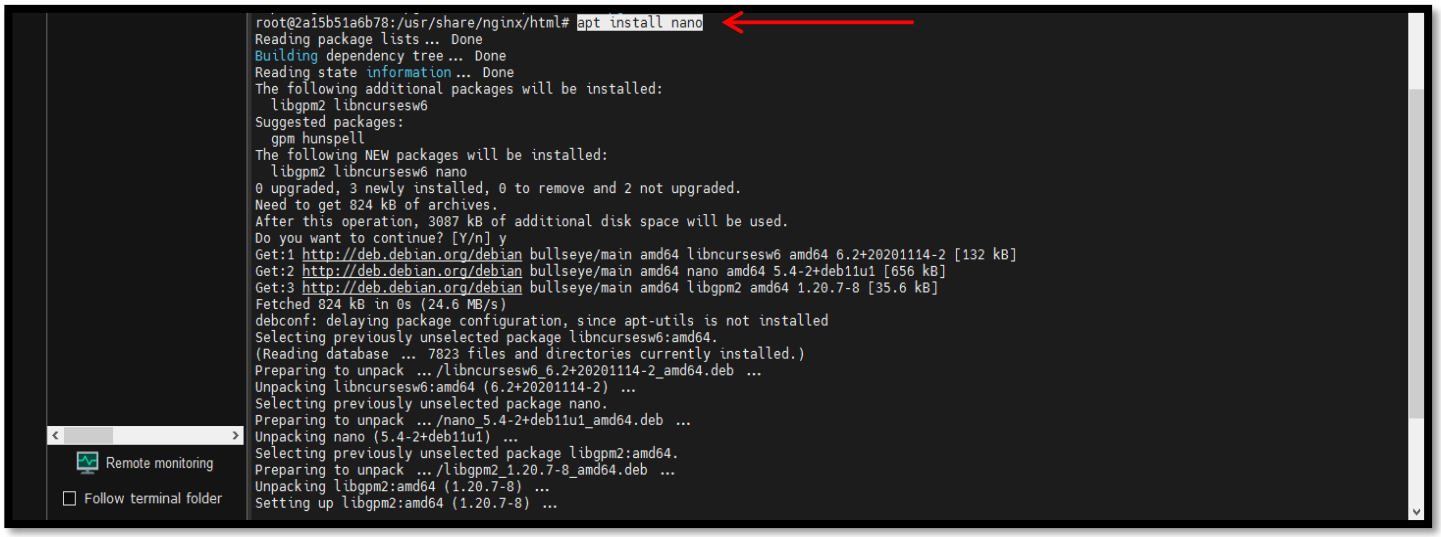
Step 11: To exit the container, use 'CTRL+P+Q'. Create a duplicate tab and take the root user rights. Run commands: 'docker ps', 'docker ps -a' Also run 'docker images' to check the images in the repository



Step 12: To make changes to a file within a container use command 'docker exec -it (container id) bash'.
Now within the container navigate to the html directory using command: 'cd usr/share/nginx/html'
Then update container using command 'apt update'

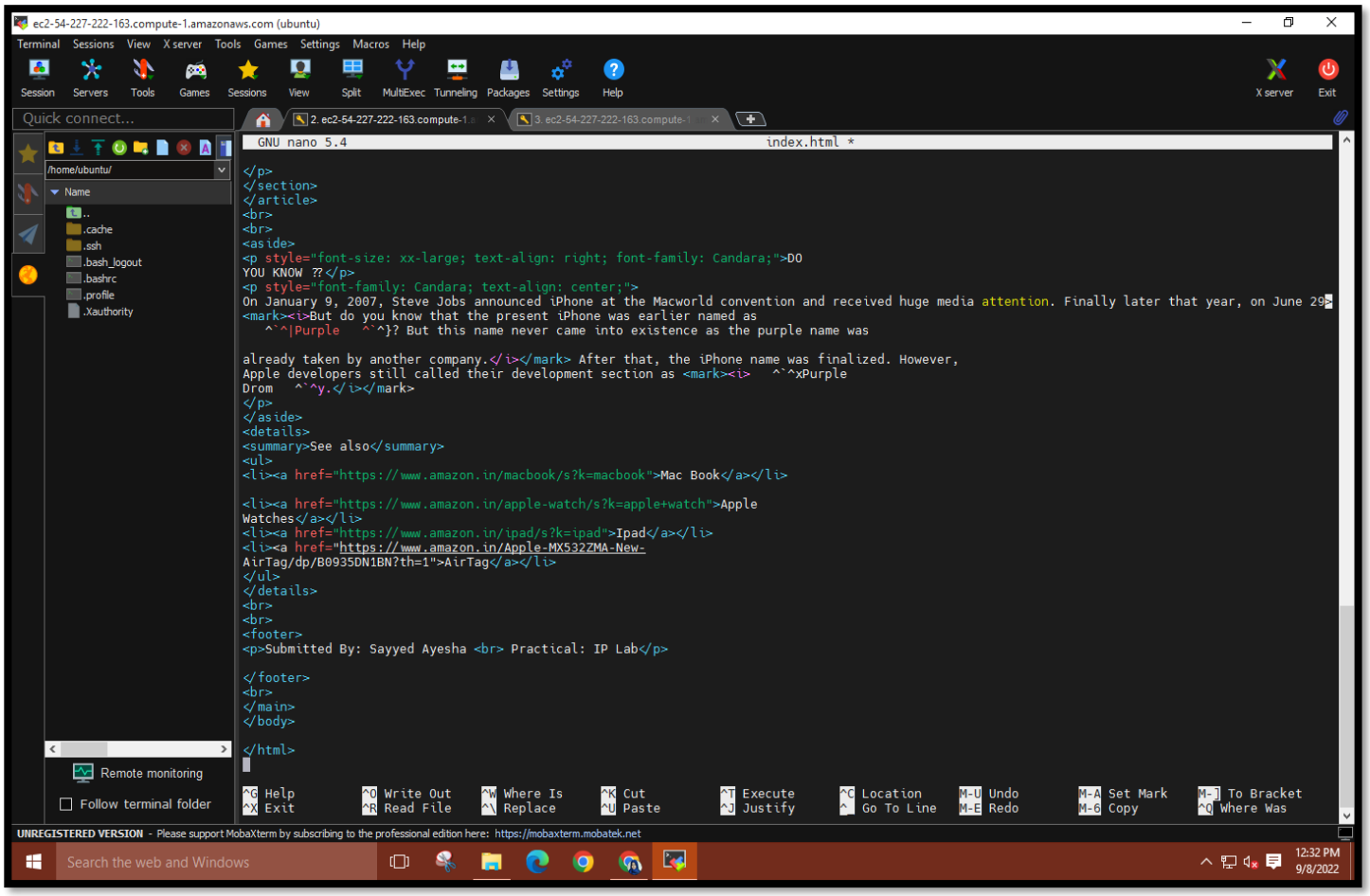


Step 13: Now use command: 'apt install nano' to install nano text editor.

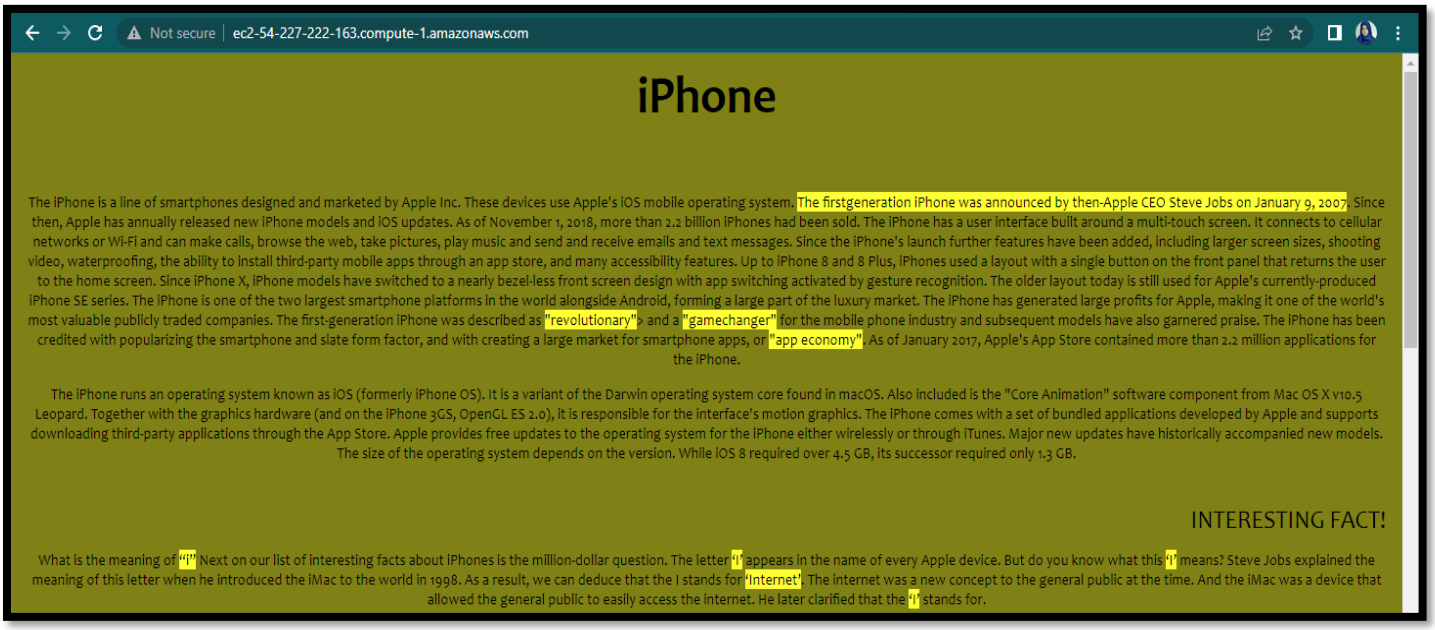


Step 14: Now move the original nginx index as a backup so you can create your own html index file using the command: 'mv index.html index.html.backup' Then, open nano text editor using command: 'nano index.html'

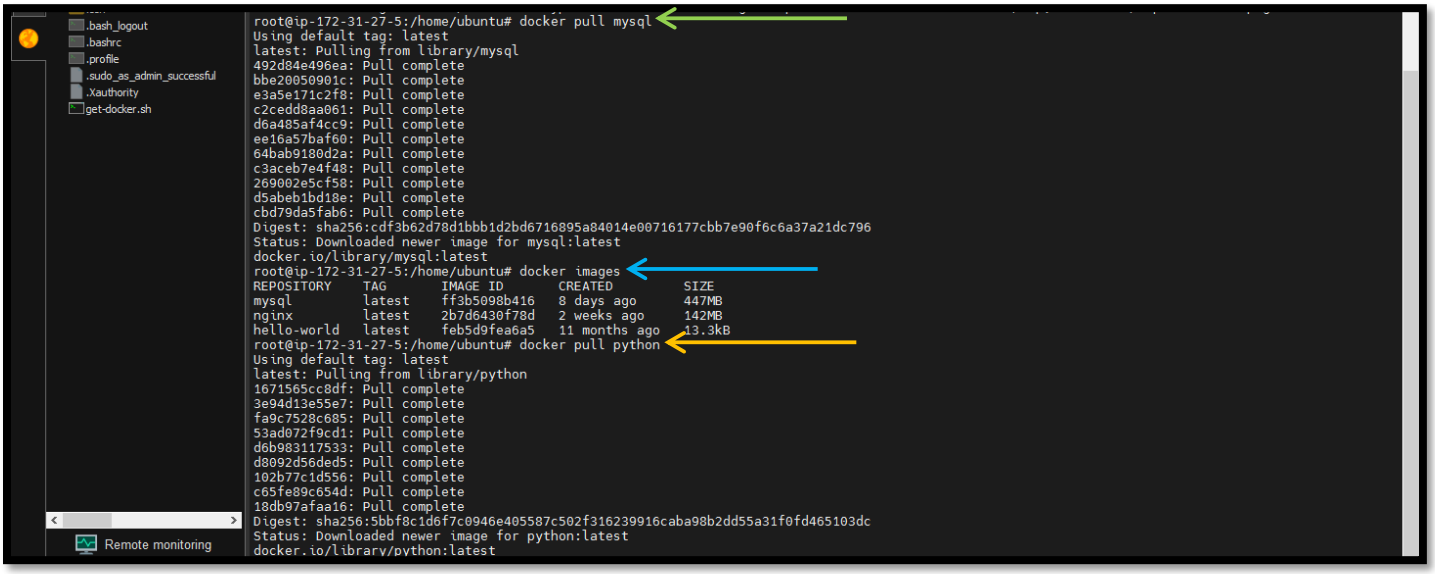
Step 15: Write an html code of your choice → CTRL+O → ENTER → CTRL X. This will save the code.



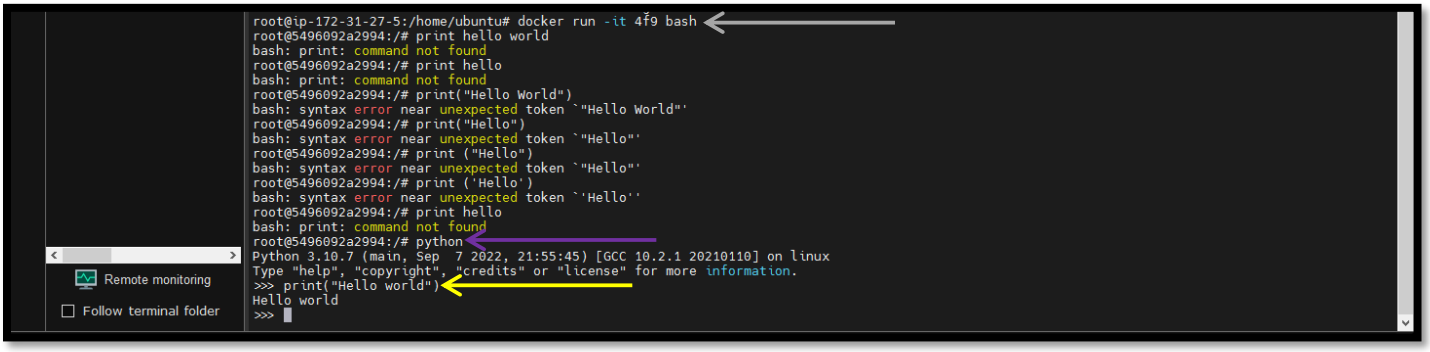
Step 16: Again, refresh the nginx web browser, it will be updated with the new changes.



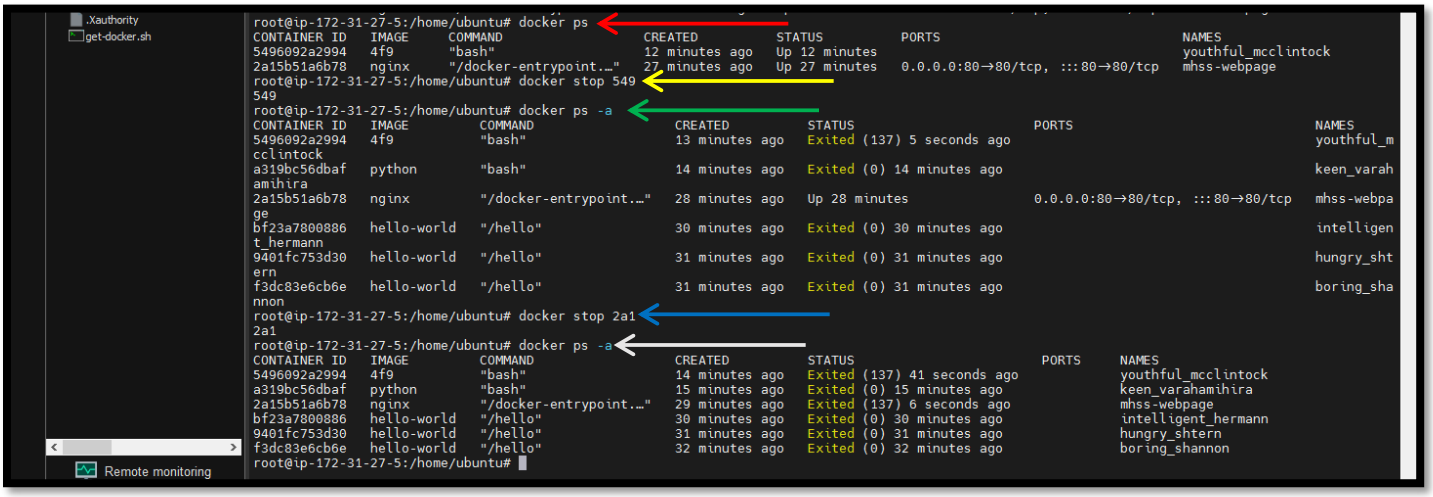
Step 17: Create another duplicate tab and get the root user access.
Run the command: 'docker pull mysql' to install a mysql image.
Then, run 'docker images' to check the image upload in the repository.
Subsequently, run commands such as 'docker pull python' and 'docker pull mongo' to install the respective images. Check the repository for the upload.

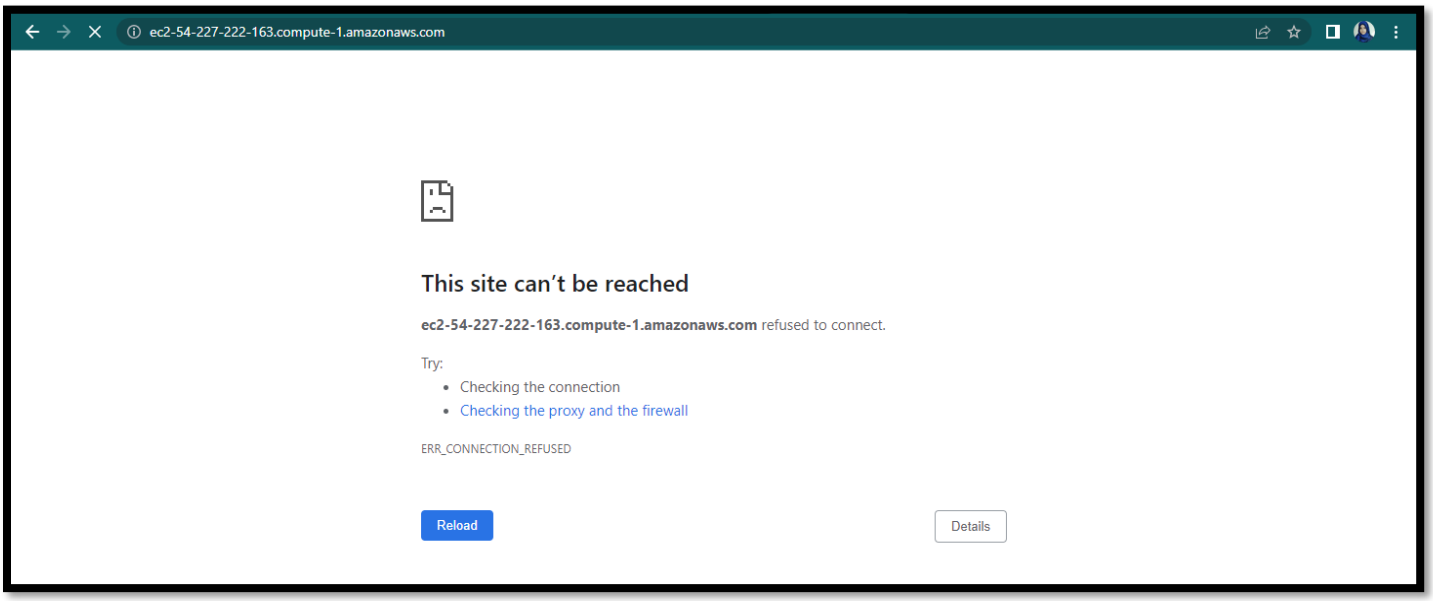


Step 18: Now, to enter the python container use command: 'docker run -it (image id) bash' and type 'python' to enter the shell Execute any command using python syntax such as print("Hello World")



Step 19: Run the command 'docker ps' to check the number of containers and their ID's.
To stop a container use command: 'docker stop (container id)'
You can use the command 'docker ps -a' to check the status of the container.





Step 20: Quit MobaXterm and then delete your EC2 instance.