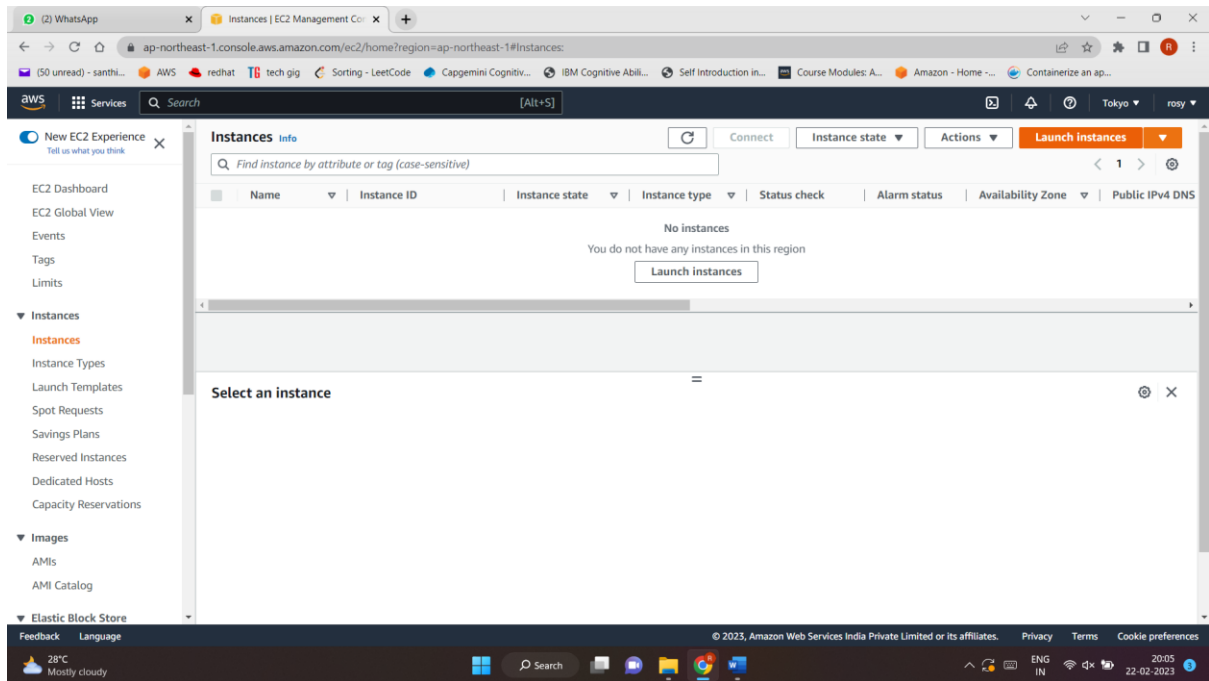


# ACTIVITY ON DOCKERS

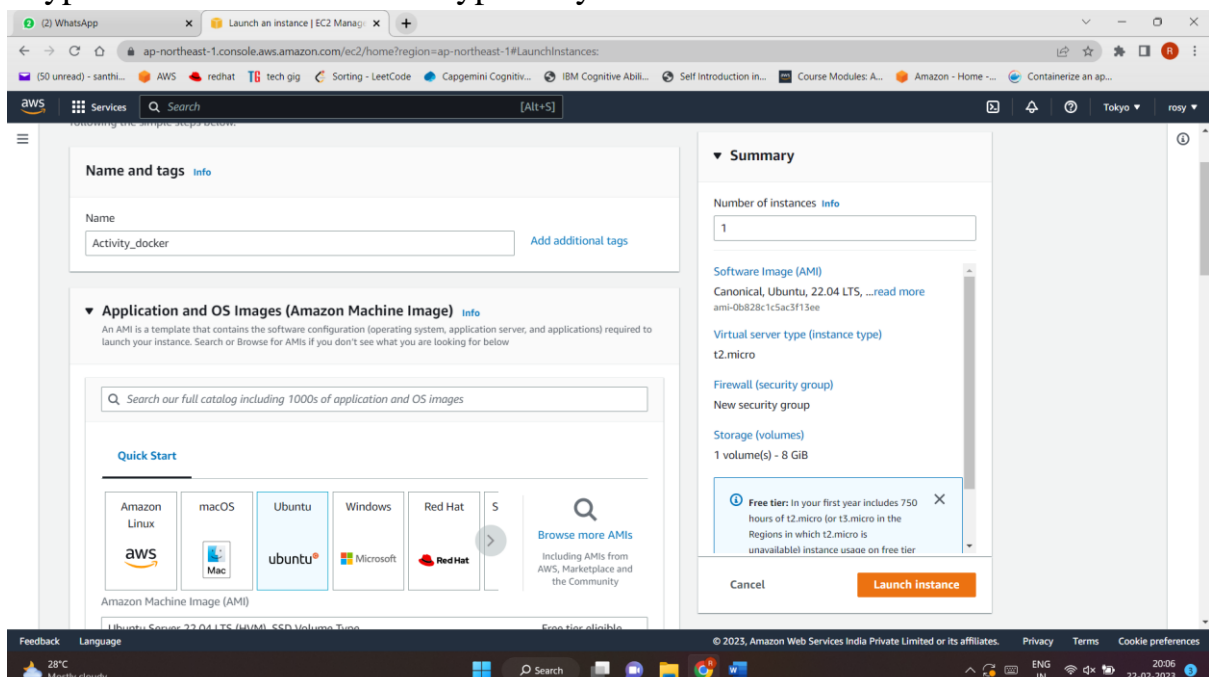
Name: T Rosy

Roll No: 20A91A0558

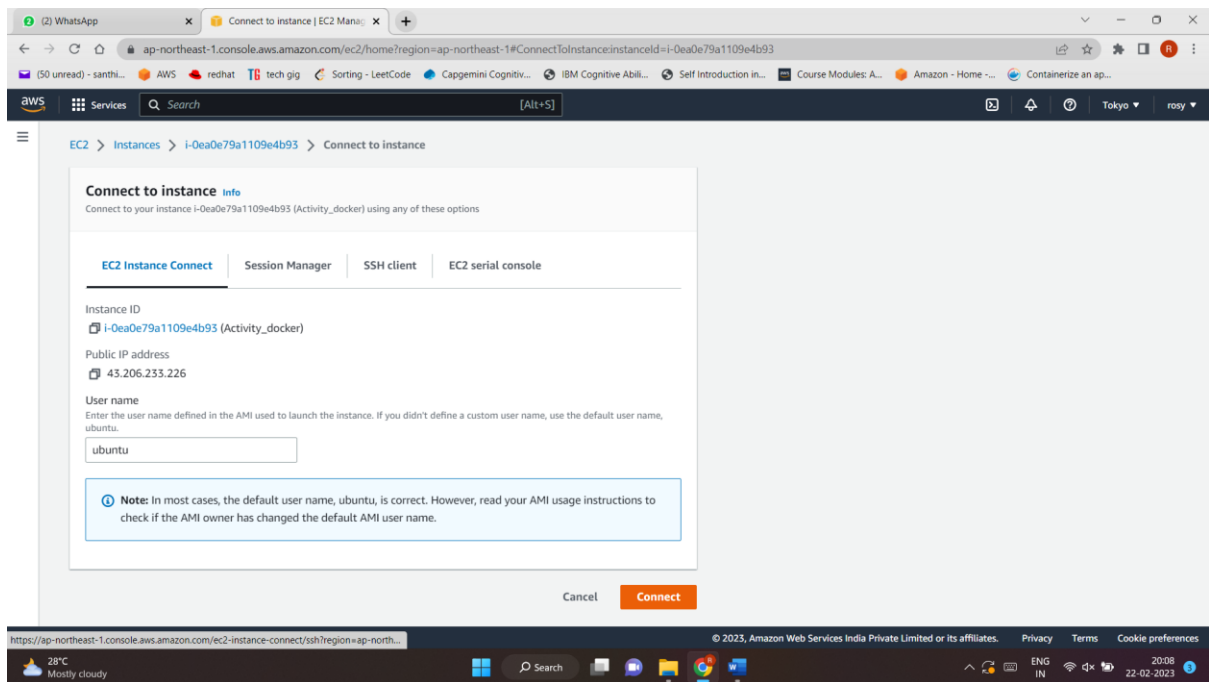
**Step-1:** We have to launch an EC2 instance in AWS console account.



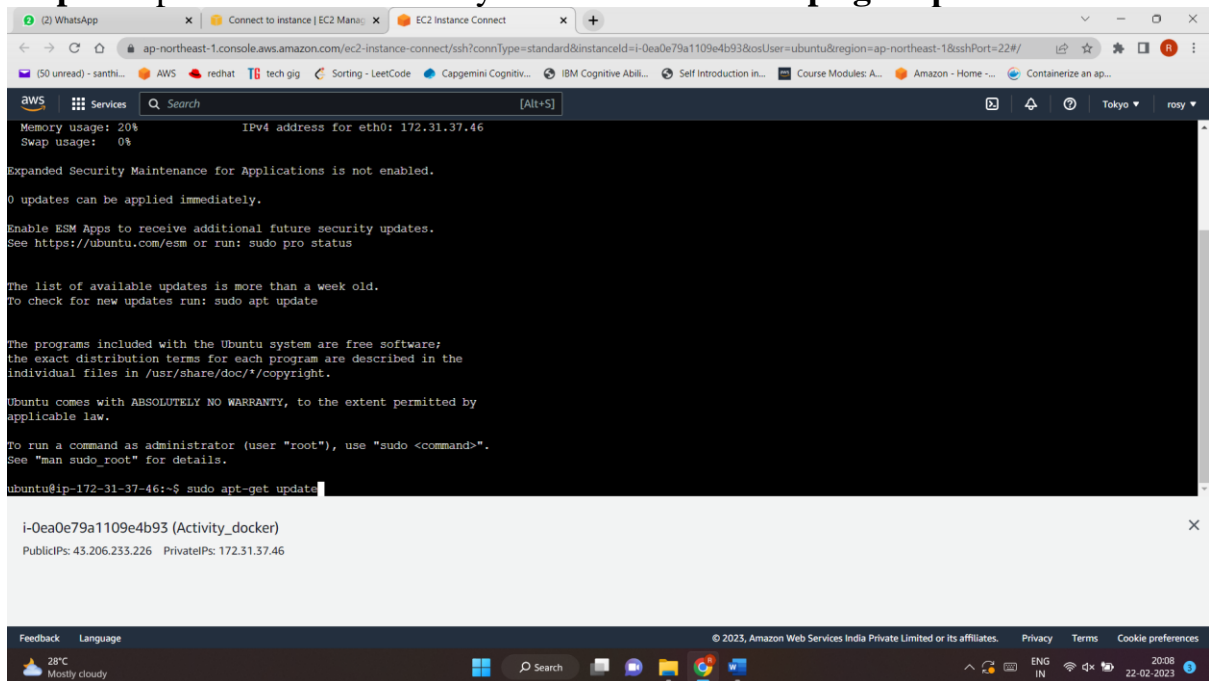
**Step-2:** we have to select Ubuntu from the AMI and edit the network settings as Type: All Traffic and sourcetype: Anywhere and click on “launch instance”.



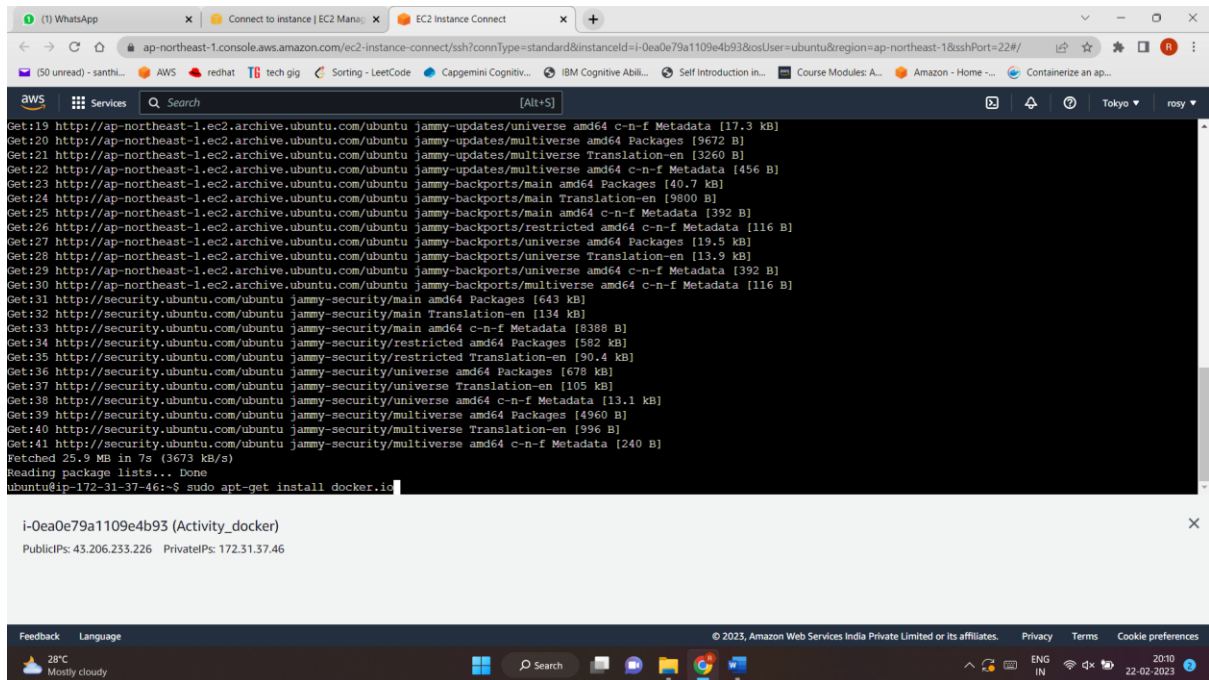
## Step-3: Connect the instance.



## Step-4: Update the ubuntu os by command – “sudo apt-get update.”

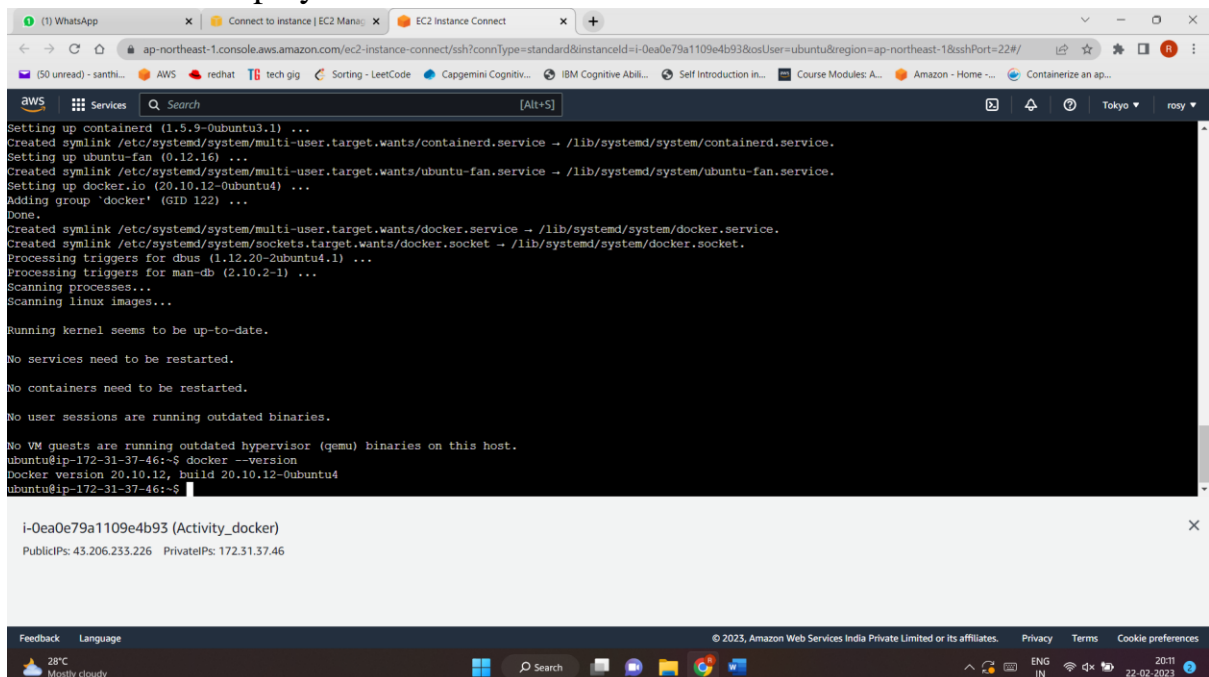


**Step-5:** Install the docker with a command – “**sudo apt-get install docker.io**”.



```
Get:19 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [17.3 kB]
Get:20 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [9672 B]
Get:21 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [3260 B]
Get:22 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [456 B]
Get:23 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.7 kB]
Get:24 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [9800 B]
Get:25 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [392 B]
Get:26 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [19.5 kB]
Get:28 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [13.9 kB]
Get:29 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [392 B]
Get:30 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [643 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [134 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [8388 B]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [582 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [90.4 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [678 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [105 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [13.1 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4960 B]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [996 B]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [240 B]
Fetched 25.9 MB in 7s (3673 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-37-46:~$ sudo apt-get install docker.io
```

**Step-6:** Check whether docker is installed or not by using command – “**docker --version**”. It display the current version of docker.



```
Setting up containerd (1.5.9-0ubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service -> /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service -> /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.12-0ubuntu4) ...
Adding group 'docker' (GID 122) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket -> /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

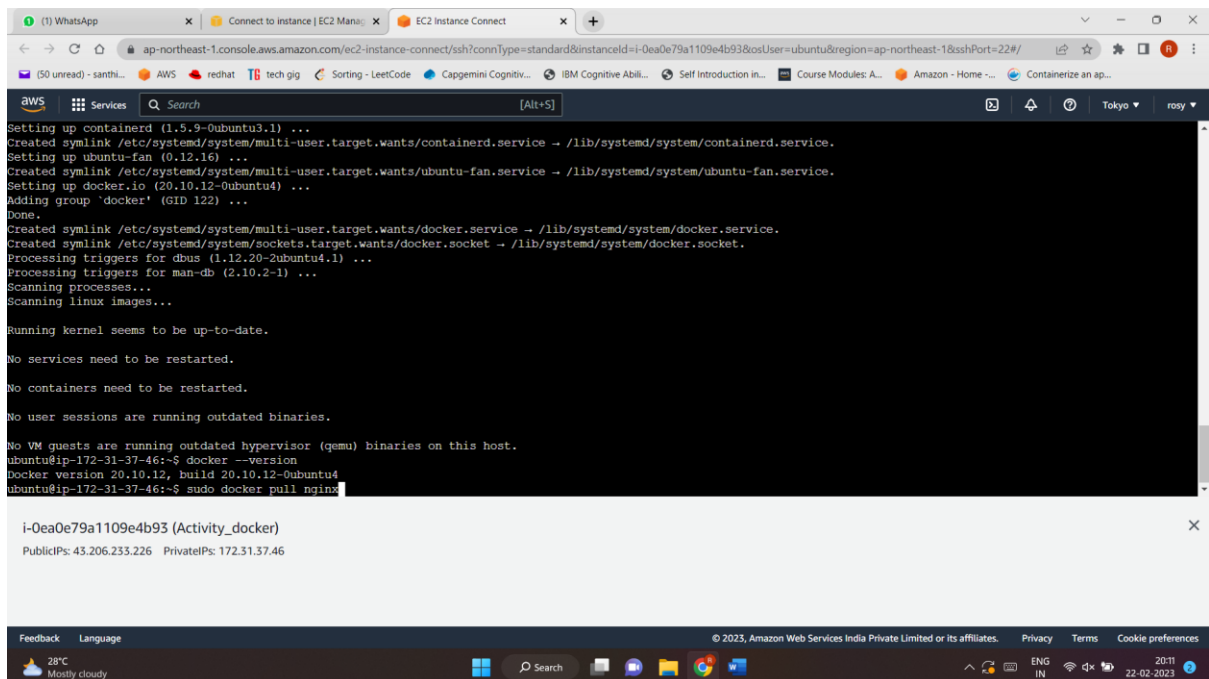
No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-37-46:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-37-46:~$
```

**Step-7:**For pulling the nginx latest image ,give the command “**sudo docker pull nginx**”.It will display nginx latest image.



```
Setting up containerd (1.5.9-0ubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.12-0ubuntu4) ...
Adding group 'docker' (GID 122) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

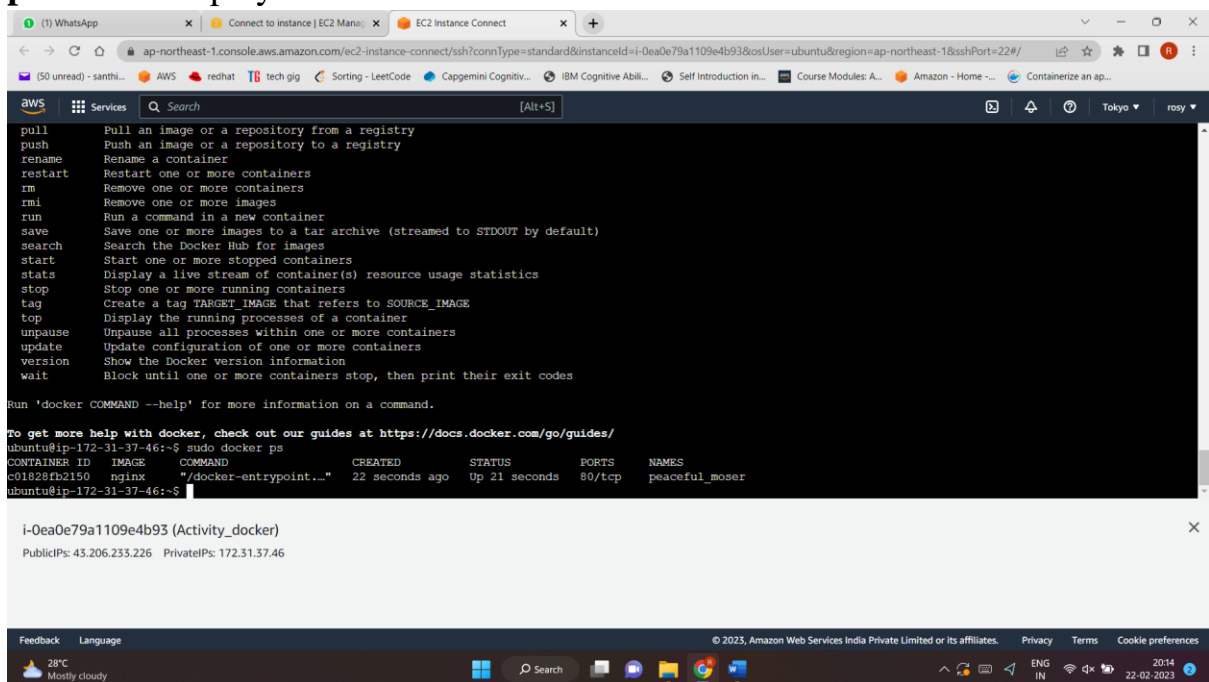
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-37-46:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-37-46:~$ sudo docker pull nginx
```

i-0ea0e79a1109e4b93 (Activity\_docker)

PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

**Step-8:** For running a container ,give command – “**sudo docker run -d nginx**” and check whether container is created or not by a command –“**sudo docker ps**”.It will display the created container id.



```
pull          Pull an image or a repository from a registry
push          Push an image or a repository to a registry
rename        Rename a container
restart       Restart one or more containers
rm            Remove one or more containers
rmi           Remove one or more images
run           Run a command in a new container
save          Save one or more images to a tar archive (streamed to STDOUT by default)
search        Search the Docker Hub for images
start         Start one or more stopped containers
stats         Display a live stream of container(s) resource usage statistics
stop          Stop one or more running containers
tag           Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top           Display the running processes of a container
unpause       Unpause all processes within one or more containers
update        Update configuration of one or more containers
version       Show the Docker version information
wait          Block until one or more containers stop, then print their exit codes

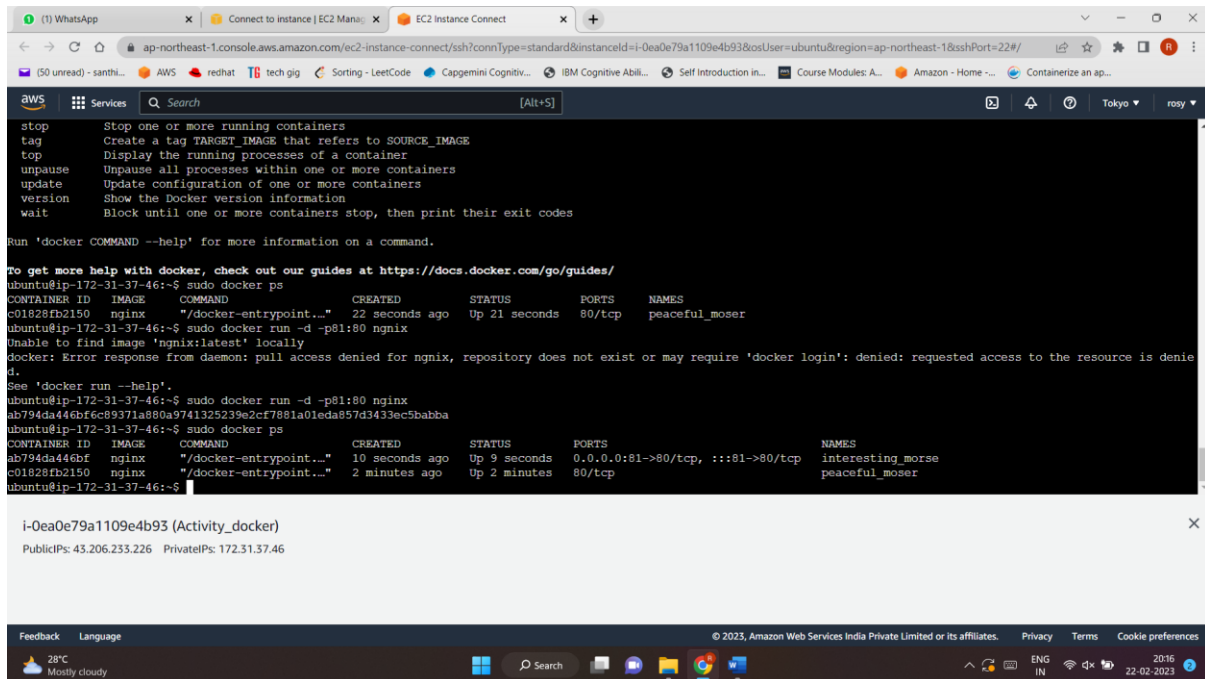
Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c91828fb2150   nginx    "/docker-entrypoint..." 22 seconds ago Up 21 seconds 80/tcp        peaceful_moser
ubuntu@ip-172-31-37-46:~$
```

i-0ea0e79a1109e4b93 (Activity\_docker)

PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

**Step-9:** Change the port number of container with 81 ( We are using 81 port,because AWS blocks 6000 port)with a command –“**sudo docker run -d -p81:80 nginx**”.Check whether port is changed or **not** by a command – “**sudo docker ps**”.



The screenshot shows a terminal window connected to an AWS EC2 instance. The user runs several Docker commands to manage containers. The output shows the status of containers and the ports they are using.

```
stop      Stop one or more running containers
tag       Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top       Display the running processes of a container
unpause   Unpause all processes within one or more containers
update    Update configuration of one or more containers
version   Show the Docker version information
wait      Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/

ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c01828fb2150   nginx    "/docker-entrypoint..." 22 seconds ago Up 21 seconds 80/tcp       peaceful_moser

ubuntu@ip-172-31-37-46:~$ sudo docker run -d -p81:80 nginx
Unable to find image 'nginx:latest' locally
docker: Error response from daemon: pull access denied for nginx, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

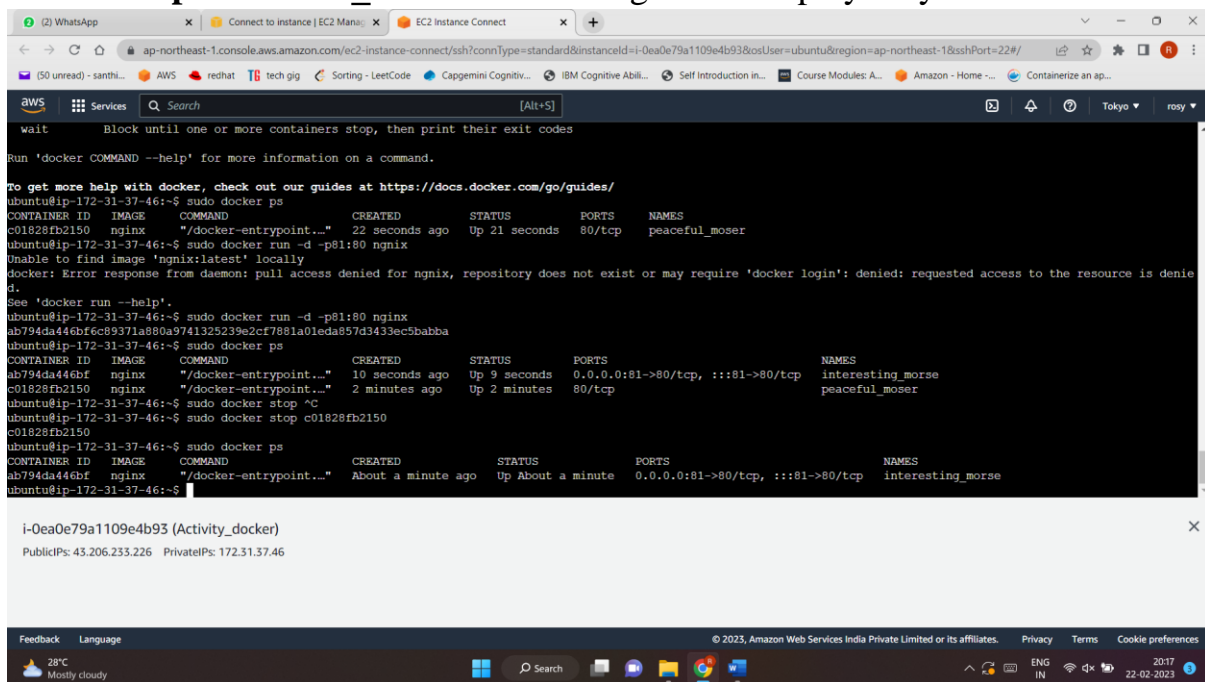
ubuntu@ip-172-31-37-46:~$ sudo docker run -d -p81:80 nginx
ab794da446bfc6e9371a880a9741325239e2cf7881a01eda857d3433ec5babba

ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                     NAMES
ab794da446bfc6e9371a880a9741325239e2cf7881a01eda857d3433ec5babba 10 seconds ago Up 9 seconds 0.0.0.0:81->80/tcp, :::81->80/tcp       interesting_morse
c01828fb2150   nginx    "/docker-entrypoint..." 2 minutes ago Up 2 minutes 80/tcp       peaceful_moser

ubuntu@ip-172-31-37-46:~$
```

i-OeaOe79a1109e4b93 (Activity\_docker)  
PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

**Step-10:** For deleting remaining containers,we are using command called “**sudo docker stop container\_id**”. After deleting it will display only one container.



The screenshot shows a terminal window connected to an AWS EC2 instance. The user runs several Docker commands to manage containers. The output shows the status of containers and the ports they are using.

```
wait      Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/

ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c01828fb2150   nginx    "/docker-entrypoint..." 22 seconds ago Up 21 seconds 80/tcp       peaceful_moser

ubuntu@ip-172-31-37-46:~$ sudo docker run -d -p81:80 nginx
Unable to find image 'nginx:latest' locally
docker: Error response from daemon: pull access denied for nginx, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

ubuntu@ip-172-31-37-46:~$ sudo docker run -d -p81:80 nginx
ab794da446bfc6e9371a880a9741325239e2cf7881a01eda857d3433ec5babba

ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                     NAMES
ab794da446bfc6e9371a880a9741325239e2cf7881a01eda857d3433ec5babba 10 seconds ago Up 9 seconds 0.0.0.0:81->80/tcp, :::81->80/tcp       interesting_morse
c01828fb2150   nginx    "/docker-entrypoint..." 2 minutes ago Up 2 minutes 80/tcp       peaceful_moser

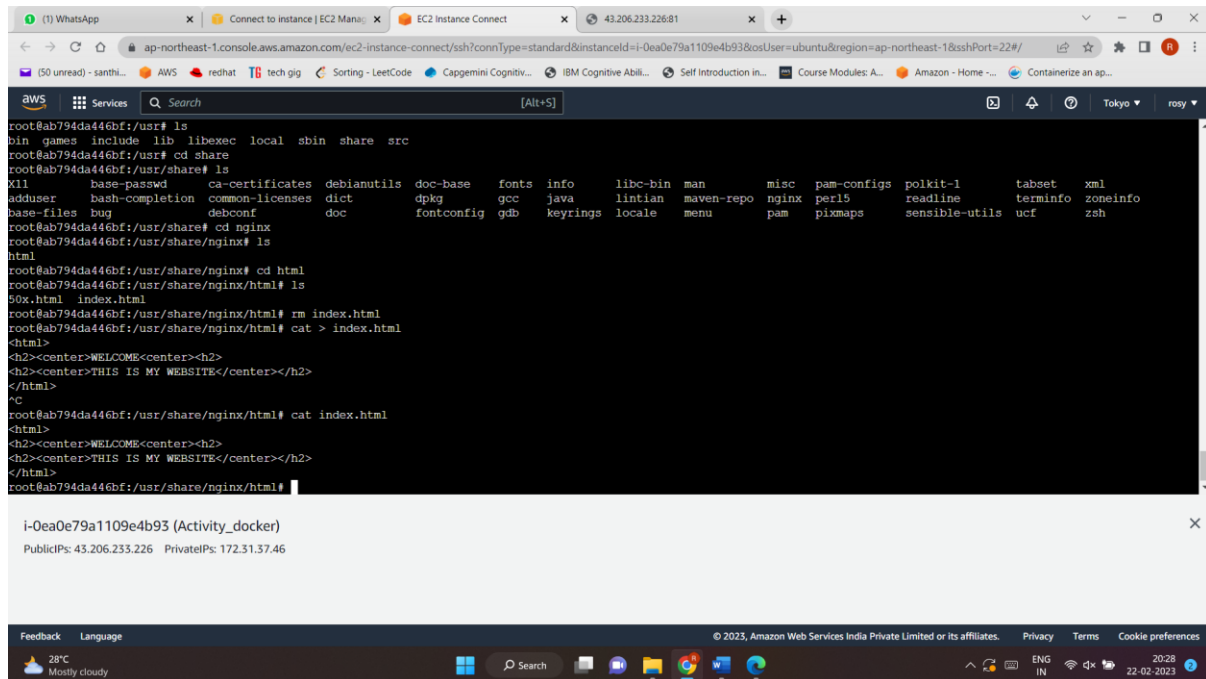
ubuntu@ip-172-31-37-46:~$ sudo docker stop ^C
ubuntu@ip-172-31-37-46:~$ sudo docker stop c01828fb2150
c01828fb2150

ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                                     NAMES
ab794da446bfc6e9371a880a9741325239e2cf7881a01eda857d3433ec5babba About a minute ago Up About a minute 0.0.0.0:81->80/tcp, :::81->80/tcp       interesting_morse

ubuntu@ip-172-31-37-46:~$
```

i-OeaOe79a1109e4b93 (Activity\_docker)  
PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

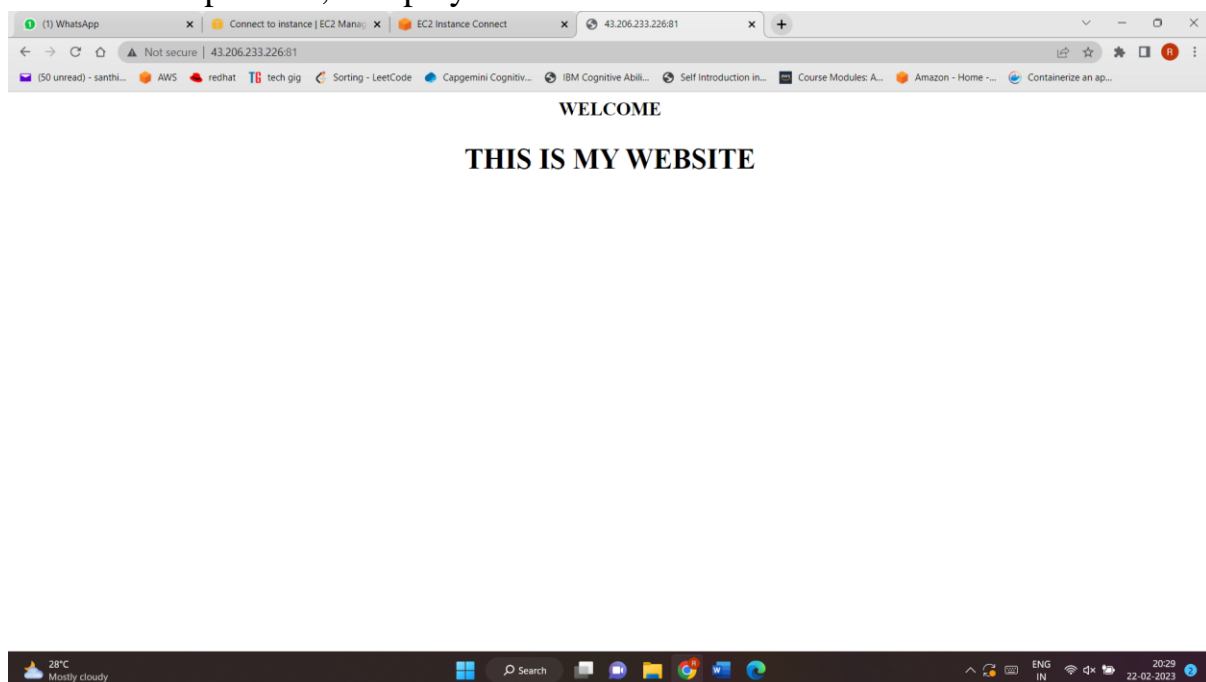
**Step-11:** Now, we are inserting the sample web template into an container by opening into the container using a command – “**sudo docker exec -it container\_id /bin/bash**” and we are going to the directory “**html**” which is presented in **/usr/share/nginx/html/** path and we are removing a index.html file and create that file with some html code.



```
root@ab794da446bf:/usr# ls
bin  games  include  lib  libexec  local /sbin  share  src
root@ab794da446bf:/usr# cd share
root@ab794da446bf:/usr/share# ls
X11      base-passwd  ca-certificates  debianutils  doc-base  fonts  info  libe-bin  man  misc  pam-configs  polkit-1  tabset  xml
adduser  bash-completion  common-licenses  dict  dpkg  fontconfig  gcc  java  lintian  maven-repo  nginx  perl5  readline  terminfo  zoneinfo
base-files  bug  debconf  doc  fontconfig  gdb  keyrings  locale  menu  pam  pixmaps  sensible-utils  ucf  zsh
root@ab794da446bf:/usr/share# cd nginx
root@ab794da446bf:/usr/share/nginx# ls
html
root@ab794da446bf:/usr/share/nginx# cd html
root@ab794da446bf:/usr/share/nginx/html# ls
50x.html  index.html
root@ab794da446bf:/usr/share/nginx/html# rm index.html
root@ab794da446bf:/usr/share/nginx/html# cat > index.html
<html>
<h2><center>WELCOME</center></h2>
<h2><center>THIS IS MY WEBSITE</center></h2>
</html>
^C
root@ab794da446bf:/usr/share/nginx/html# cat index.html
<html>
<h2><center>WELCOME</center></h2>
<h2><center>THIS IS MY WEBSITE</center></h2>
</html>
root@ab794da446bf:/usr/share/nginx/html#
```

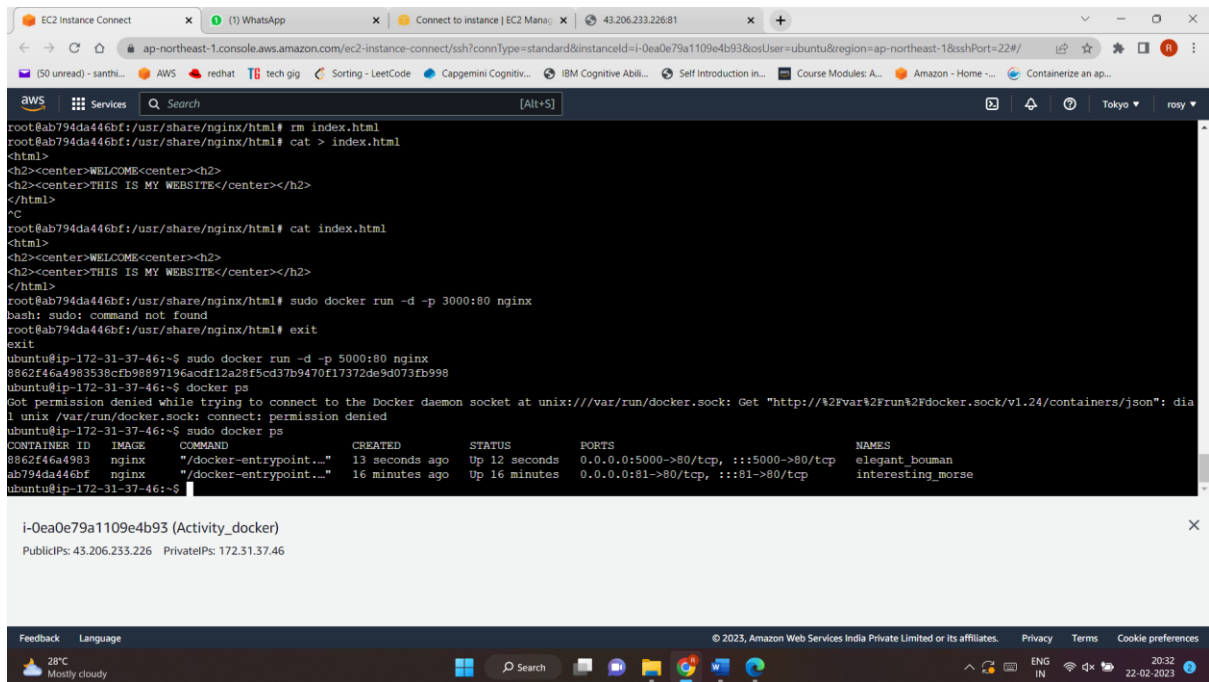
I-Oea0e79a1109e4b93 (Activity\_docker)  
PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

**Step-12:** When we are opening the created sample website with an public ip address with port 81 ,it displays a created website.





**Step-13:** Pull the other version of nginx image by using “**sudo docker pull nginx:1.23**” command and create another container with other version of nginx image.

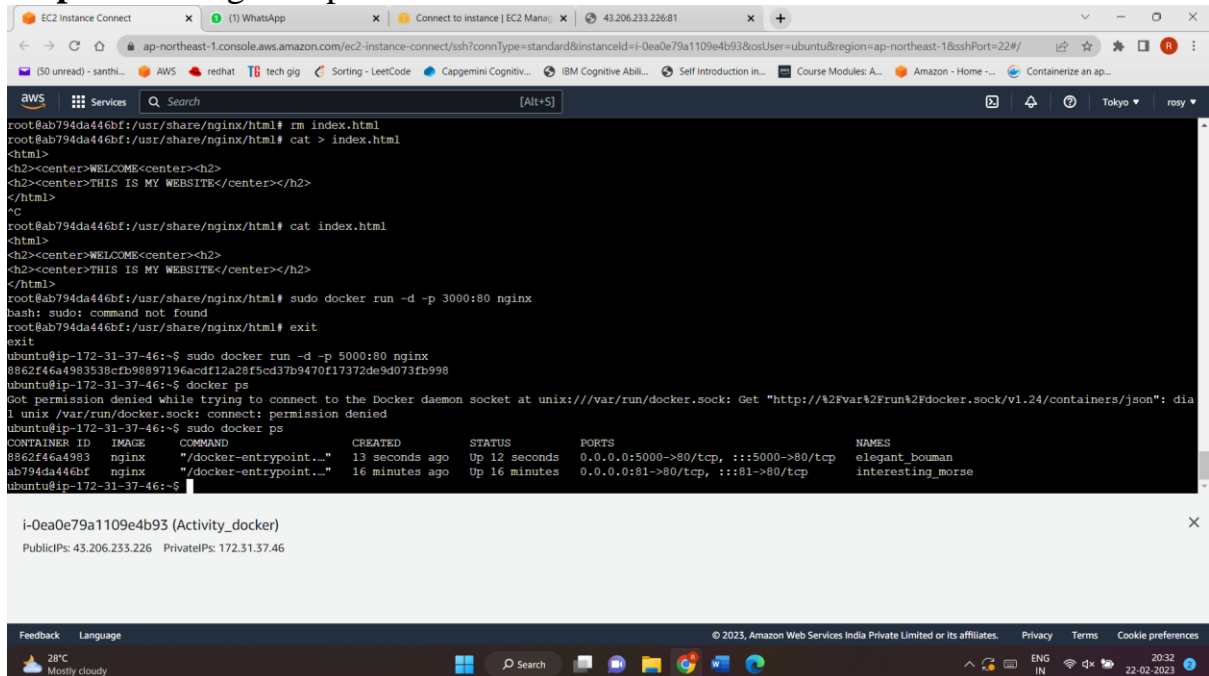


The screenshot shows the AWS Management Console terminal for an EC2 instance. The user is in the root shell of an Ubuntu instance. They create an index.html file with a simple HTML structure. Then, they attempt to run 'sudo docker run -d -p 3000:80 nginx', but receive a 'command not found' error. They then exit the root shell and log in as 'ubuntu' on IP 172.31.37.46. They run 'sudo docker run -d -p 5000:80 nginx' successfully. When they run 'docker ps', they see two containers: 'elegant\_bouman' (nginx:1.23) and 'interesting\_morse' (nginx:1.23.6-ubuntu). The console also shows a notification for the instance 'i-OeaOe79a1109e4b93' with public IP 43.206.233.226 and private IP 172.31.37.46.

```
root@ab794da446bf:/usr/share/nginx/html# rm index.html
root@ab794da446bf:/usr/share/nginx/html# cat > index.html
<html>
<h2><center>WELCOME</center><h2>
<h2><center>THIS IS MY WEBSITE</center></h2>
</html>
^C
root@ab794da446bf:/usr/share/nginx/html# cat index.html
<html>
<h2><center>WELCOME</center><h2>
<h2><center>THIS IS MY WEBSITE</center></h2>
</html>
root@ab794da446bf:/usr/share/nginx/html# sudo docker run -d -p 3000:80 nginx
bash: sudo: command not found
root@ab794da446bf:/usr/share/nginx/html# exit
exit
ubuntu@ip-172-31-37-46:~$ sudo docker run -d -p 5000:80 nginx
8862f46a4983538c9b98897196acdf12a28f5cd37b9470f17372de9d073fb998
ubuntu@ip-172-31-37-46:~$ docker ps
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json": dial
l unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-37-46:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
8862f46a4983   nginx    "/docker-entrypoint..." 13 seconds ago Up 12 seconds 0.0.0.0:5000->80/tcp, :::5000->80/tcp   elegant_bouman
ab794da446bf    nginx    "/docker-entrypoint..." 16 minutes ago Up 16 minutes 0.0.0.0:81->80/tcp, :::81->80/tcp       interesting_morse
ubuntu@ip-172-31-37-46:~$
```

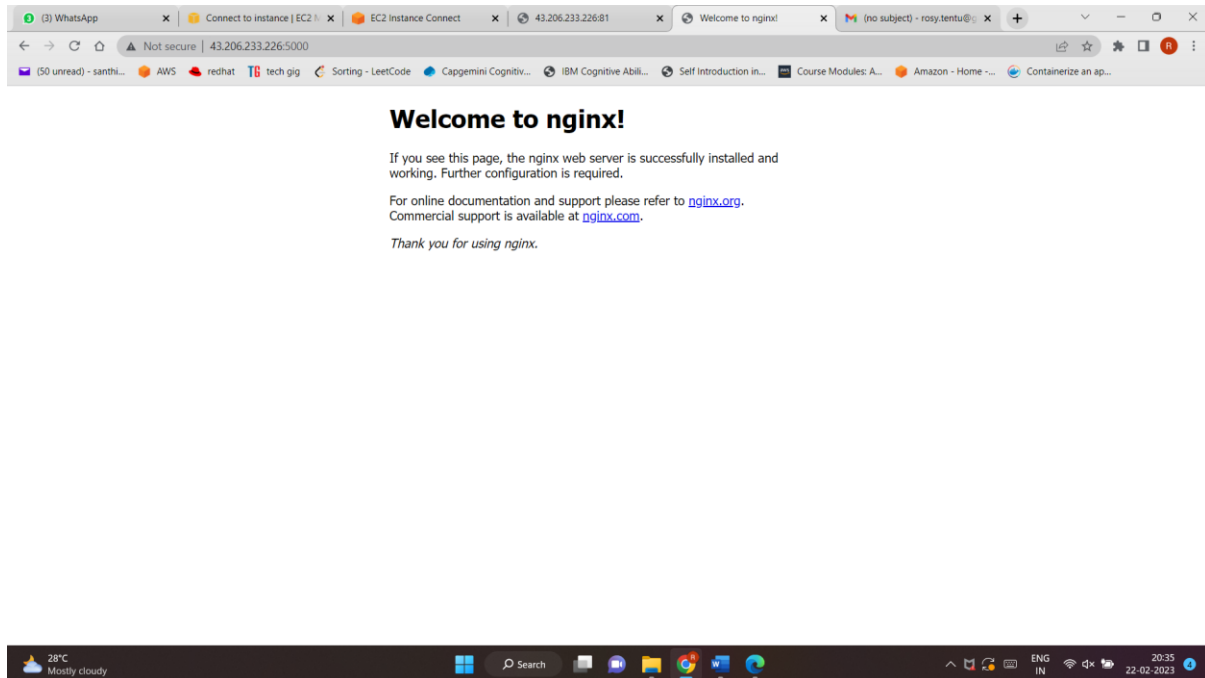
i-OeaOe79a1109e4b93 (Activity\_docker)  
PublicIPs: 43.206.233.226 PrivateIPs: 172.31.37.46

**Step-14:** Change the port number with 5000.

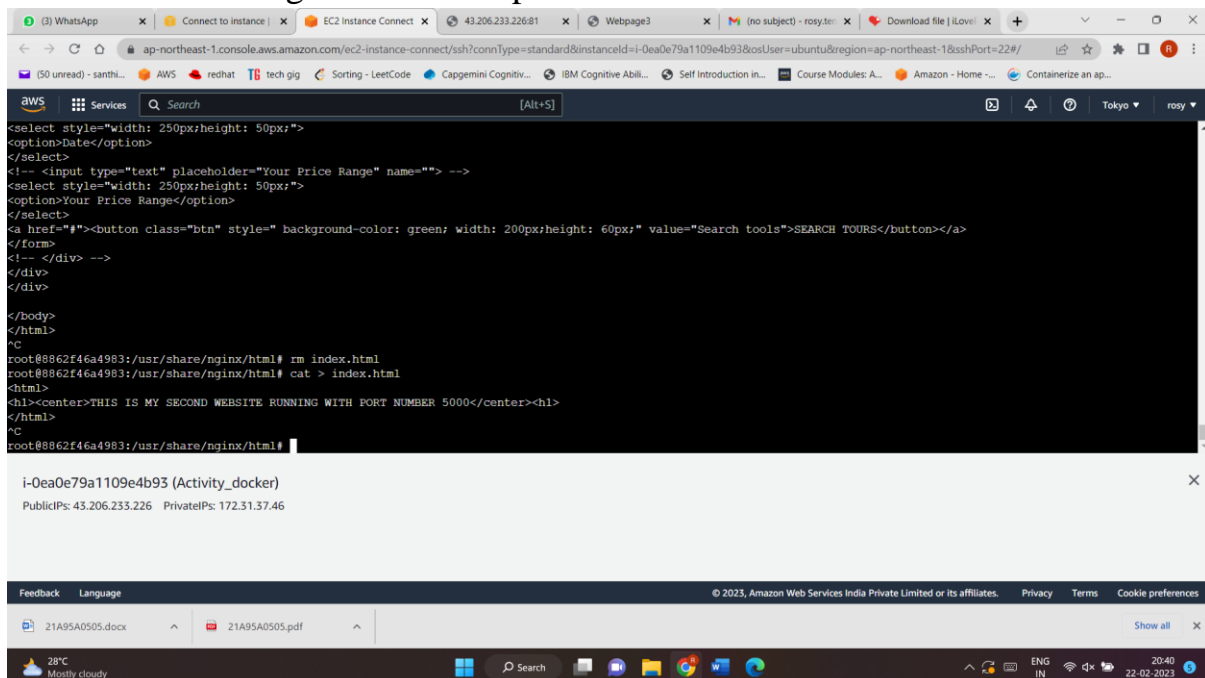


This screenshot is identical to the one for Step 13, showing the same terminal session and console notification. The key difference is in the Docker command used in Step 14: 'sudo docker run -d -p 5000:80 nginx'. The terminal output and the 'docker ps' result are the same as in Step 13, showing the 'elegant\_bouman' container mapped to port 5000.

## Step-15: Default website of nginx image version of port 5000.



## Step-16: Now we can change the content of html file with our html code in newer version of nginx:1.23 with port 5000.





**Step-17:** When we are opening the created sample website with an public ip address with port 5000 ,it displays a created website.



Both the websites should run parallely with same ip address on different ports 81 and 5000.