What is Docker?

Docker is a infrastructure provisioning tool.

Docker is a containerization tool.

What is Containerization?

IS the next level of virtualization (or) advanced level of virtualization.

Containerization eliminated the need for Guest OS. There is no point of allocating hardware system.

**Virtualization:** ex- VMware

**Virtualization --** Fixed hardware allocation.

**Containerization -** No Fixed Hardware

Process isolation ( Dependency in os is removed )

**+++++++++++++++++++++++++**

In comparison to the traditional virtualization functionalities of hypervisors,

Docker containers eliminate the need for a separate guest operating system for every new virtual machine.

Docker implements a high-level API to provide lightweight containers that run processes in isolation.

A Docker container enables rapid deployment with minimum run-time requirements. It also ensures better management and simplified portability.

This helps developers and operations team in rapid deployment of an application.

+++++++++++++++++++++++++

**Create Ubuntu Machine on AWS**

All Traffic - anywhere

Connect using git bash

https://get.docker.com/

Go to Root Account

$ sudo su -

# curl -fsSL https://get.docker.com -o get-docker.sh ( this will download shell script in the machine)

# sh get-docker.sh ( This will execute the shell script, which will install docker )

How to check the docker is installed or not

# docker --version

We should be comfortable with four terms

1) docker images

Combinations of binaries / libraries which are necessary for one software application.

2) Docker Containers

When image is executed comes into running condition, it is called container.

3) Docker Host

Machine on which docker is installed, is called as Docker host.

4) Docker Client

Terminal used to run docker run commands ( Git bash )

On linux machine, git bash will work like docker client.

+++++++++++

**Docker Commands**

**--------------------**

**Working on Images**

**-------------------------**

1 To download a docker image

docker pull image\_name

docker pull tomee

docker pull ubuntu

docker pull jenkins/jenkins

2 To see the list of docker images

docker image ls

(or)

docker images

3 To delete a docker image from docker host

docker rmi image\_name/image\_id

4) To upload a docker image into docker hub

docker push image\_name

5) To tag an image (tags refers to version)

docker tag image\_name ipaddress\_of\_local\_registry:5000/image\_name

6) To build an image from a customised container

docker commit container\_name/container\_id new\_image\_name

7) To create an image from docker file

docker build -t new\_image\_name

8) To search for a docker image

docker search image\_name

9) To delete all images that are not attached to containers

docker system prune -a

++++++++++++++++++++++++++++++++++++++++++++++

**Working on containers**

**-----------------------------**

10) To see the list of all running continers

docker container ls

11) To see the list of running and stopped containers

docker ps -a

12) To start a container

docker start container\_name/container\_id

13) To stop a running container

docker stop container\_name/container\_id

14) To restart a running container

docker restart container\_name/container\_id

To restart after 10 seconds

docker restart -t 10 container\_name/container\_id

15) To delete a stopped container

docker rm container\_name/container\_id

16) To delete a running container

docker rm -f container\_name/container id

17) To stop all running containers

docker stop $(docker ps -aq)

18) To restart all containers

docker restart $(docker ps -aq)

19) To remove all stopped containers

docker rm $(docker ps -aq)

20) To remove all contianers(running and stopped)

docker rm -f $(docker ps -aq)

21) To see the logs generated by a container

docker logs container\_name/container\_id

22) To see the ports used by a container

docker port container\_name/container\_id

23) To get detailed info about a container

docker inspect container\_name/container\_id

24) To go into the shell of a running container which is moved into background

docker attach container\_name/container id

25) To execute any command in a container

docker exec -it container\_name/container\_id command

Eg: To launch the bash shell in a contianer

docker exec -it container\_name/container\_id bash

26) To create a container from a docker image ( imp )

docker run image\_name

++++++++++++++++++++++++++++++++++++++++++++++++

**Run command options**

-it for opening an interactive terminal in a container

--name Used for giving a name to a container

-d Used for running the container in detached mode as a background process

-e Used for passing environment variables to the container

-p Used for port mapping between port of container with the dockerhost port.

-P Used for automatic port mapping ie, it will map the internal port of the container

with some port on host machine.

This host port will be some number greater than 30000

-v Used for attaching a volume to the container

--volume-from Used for sharing volume between containers

--network Used to run the contianer on a specific network

--link Used for linking the container for creating a multi container architecture

--memory Used to specify the maximum amount of ram that the container can use

++++++++++++++++++++++++++++++++++++++++++++++++++=

# docker images ( There are no images )

To download tomcat image

# docker pull tomee

# docker images

# docker pull ubuntu

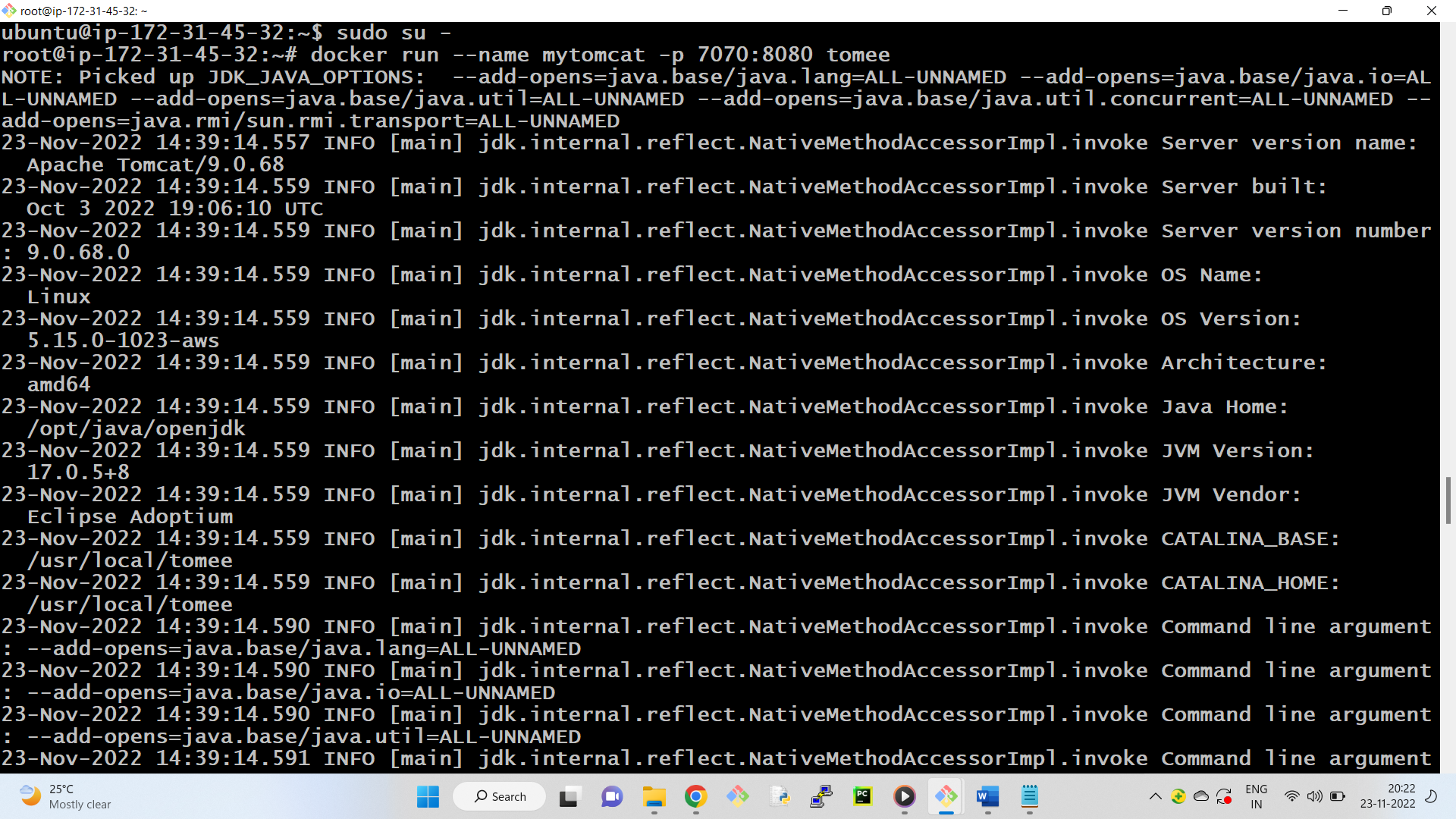
If you do not specify the version, by default, we get latest version

I want to download jenkins

# docker pull jenkins/jenkins

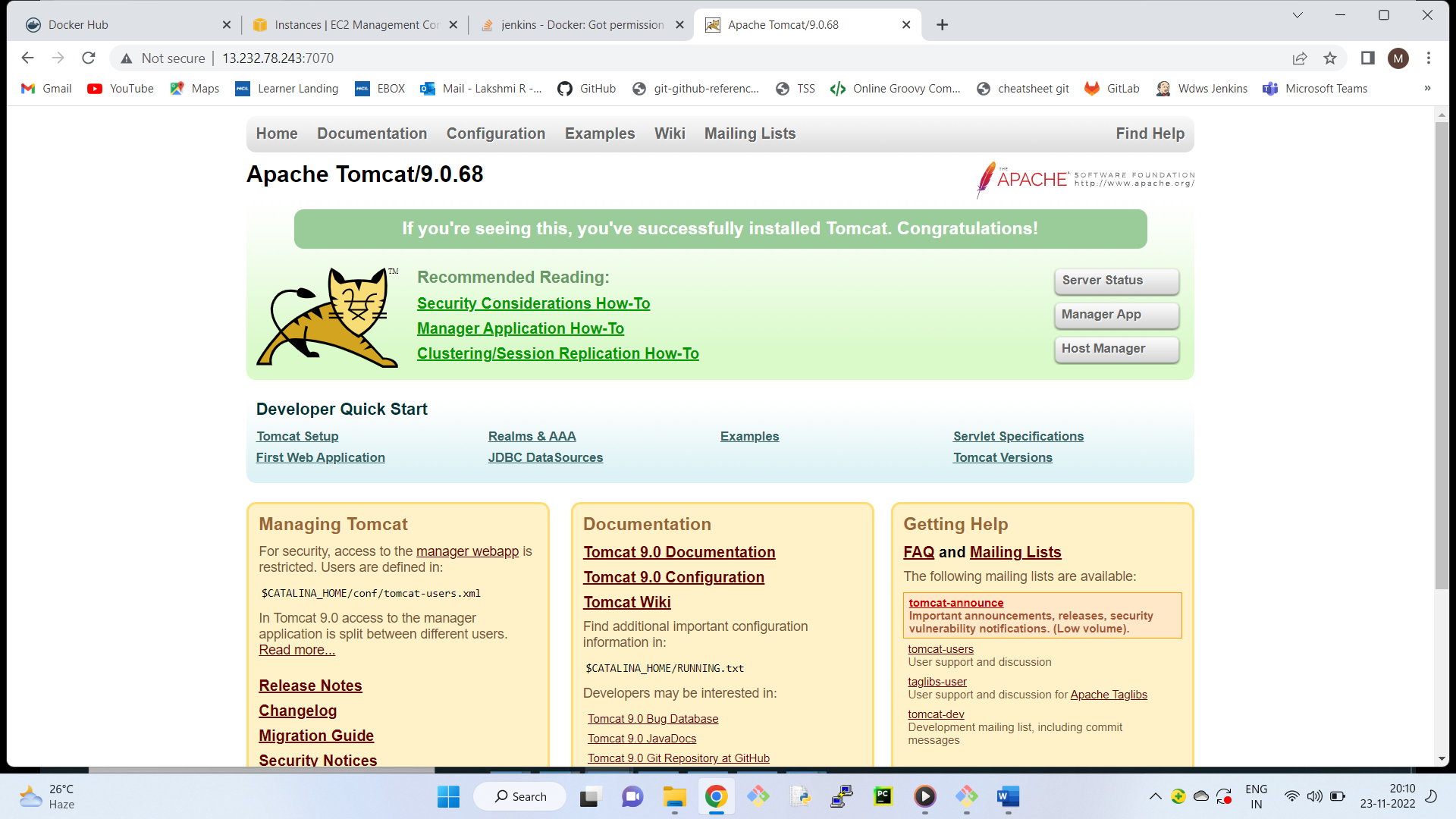
---------------------------------------------------------------------------------------------------

docker run --name mytomcat -p 7070:8080 tomee –runs with log messages ( small p refers to manual port mapping)



browse public\_ip\_of\_dockerhost:port\_number

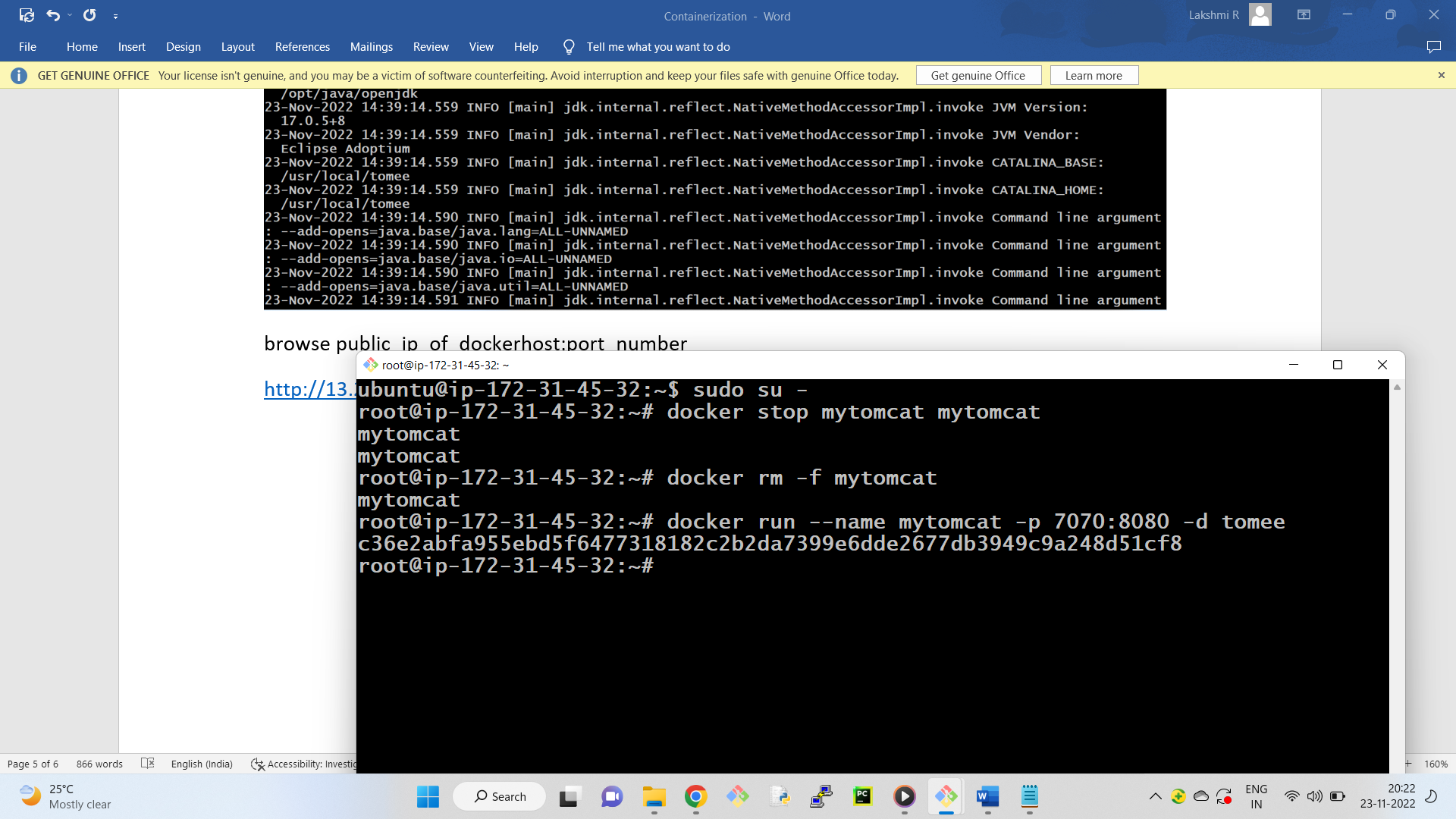
<http://13.232.78.243:7070/> -- to check whether the tomcat is running



docker stop mytomcat –-stop the container

docker rm -f mytomcat –removes the container

docker run --name mytomcat -p 7070:8080 -d tome –runs the container in detach mode and avoids log messages

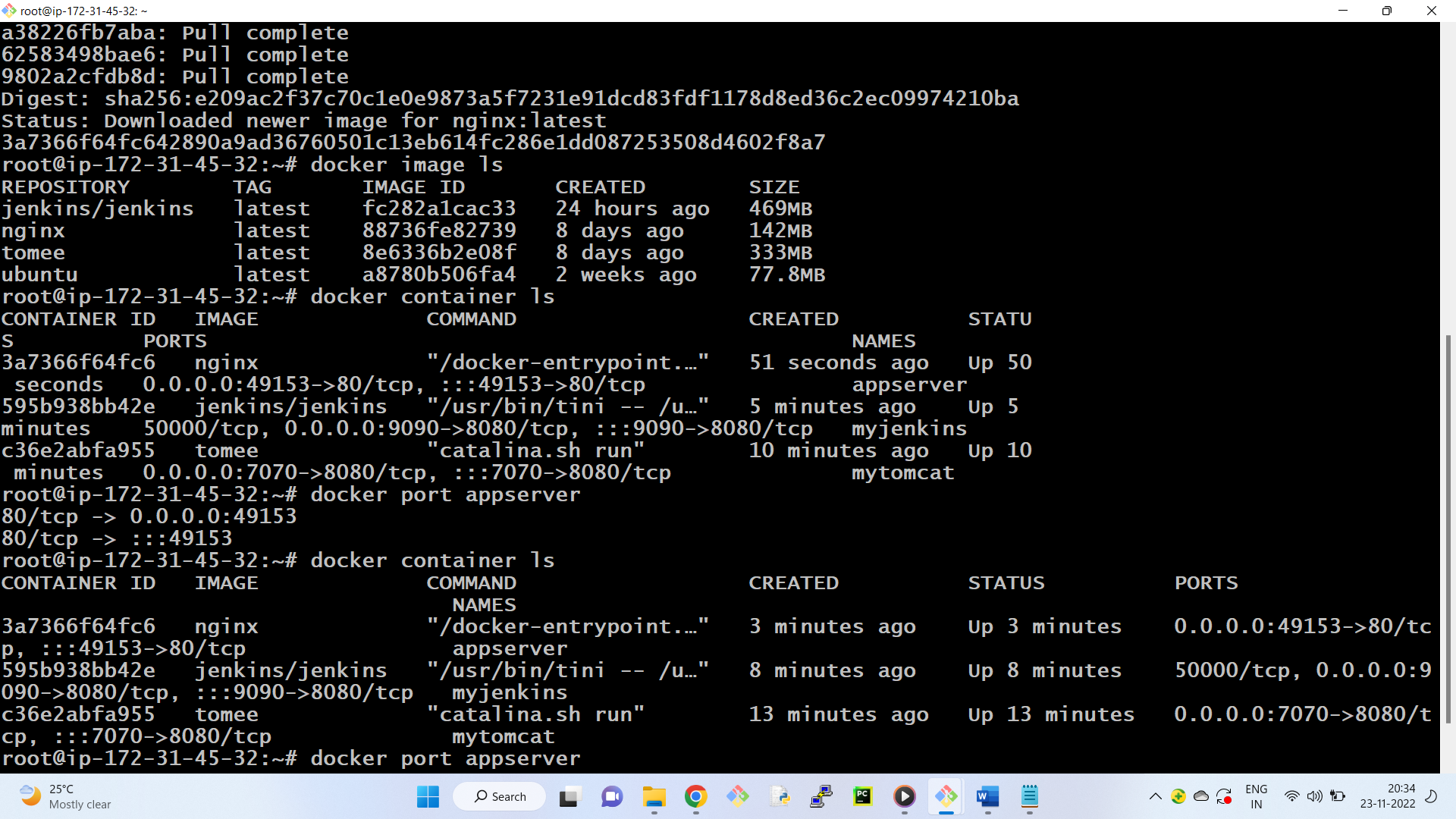


docker container ls –lists containers

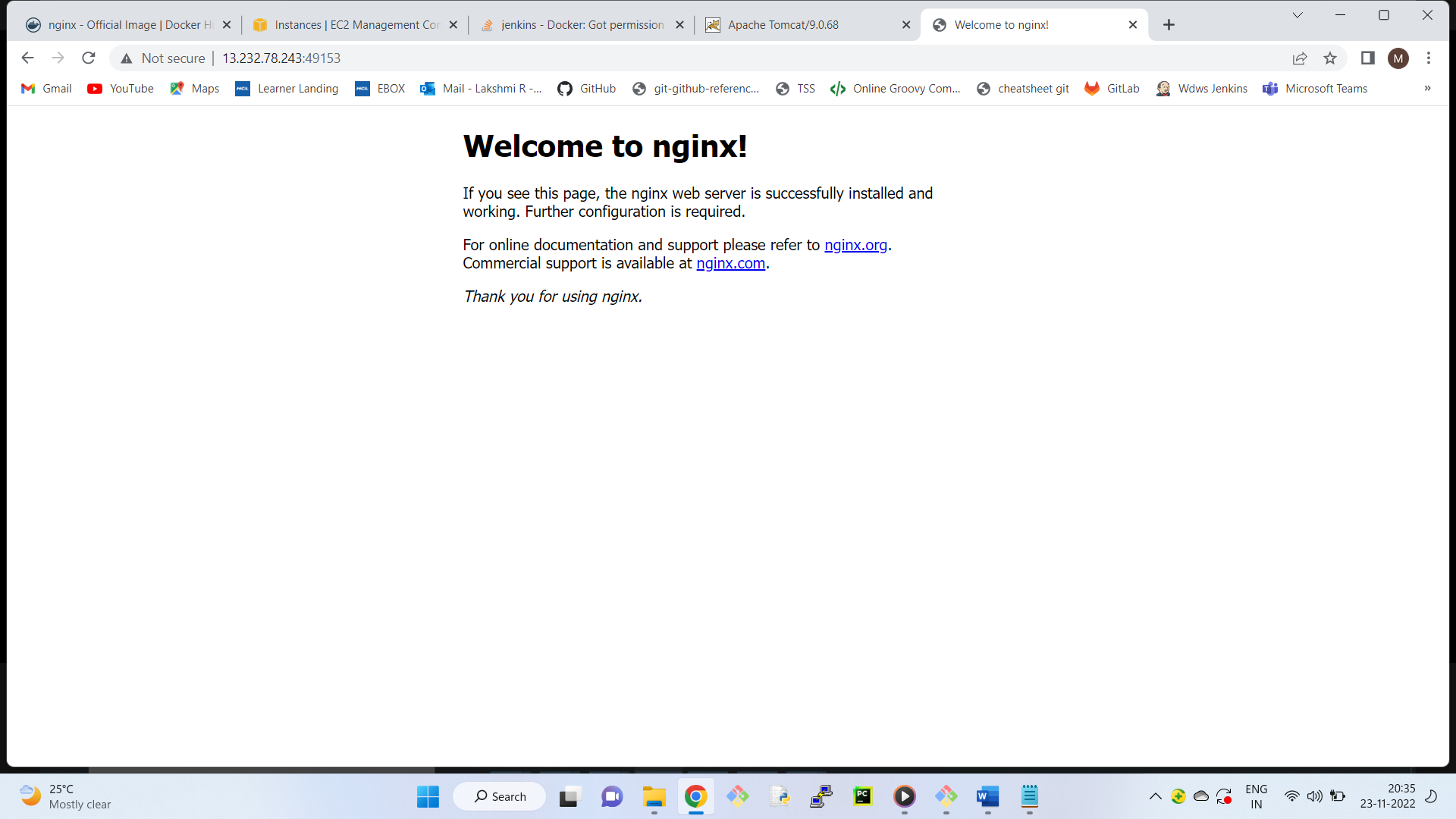
docker run --name myjenkins -p 9090:8080 -d jenkins/jenkins

docker run --name appserver -P -d nginx –directly pulls the image( capital P refers to default port mapping )

docker port appserver – lists port number



http://13.232.78.243:49153/



docker run --name mycentos -it centos (-it container is created and immediately jumping into container)

