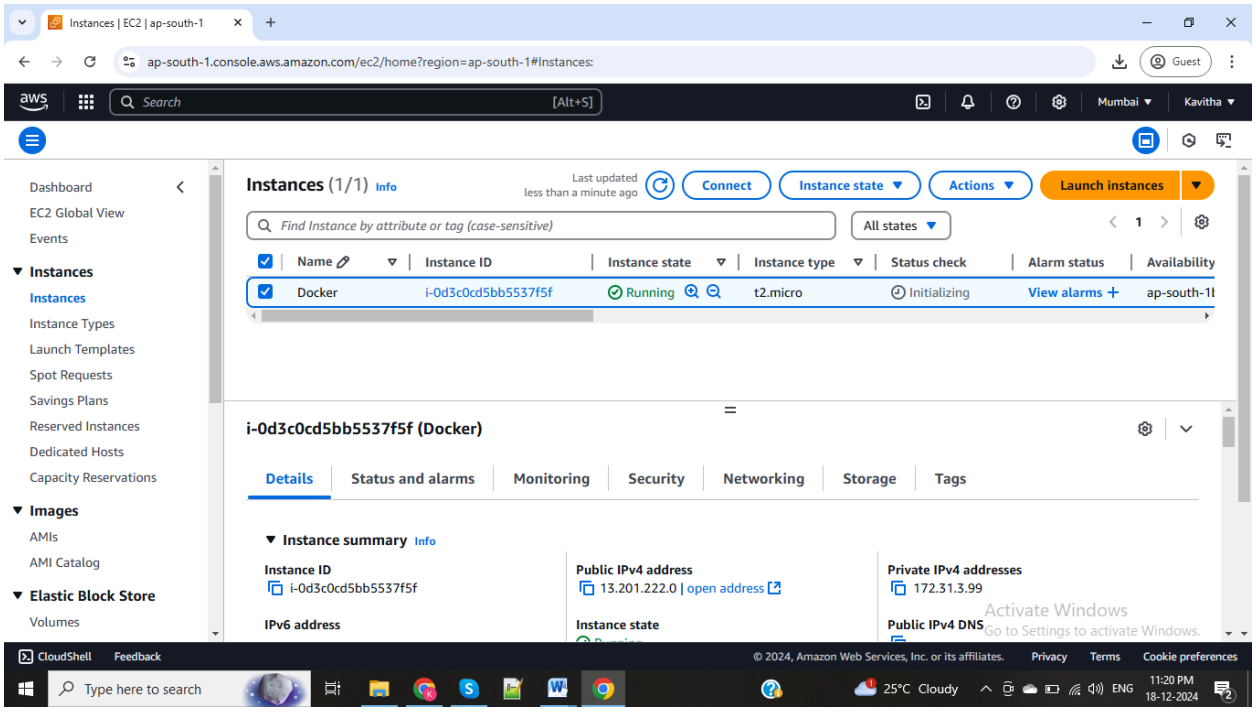
DOCKER

A Docker is a set of platform as a service (PAAS) products that use OS level virtualization to deliver software in packages called containers. Docker is an open-source platform that enables developers and system administrators to build, ship, and run applications in containers. Containers are lightweight, portable, and self-sufficient environments that include all the dependencies an application needs to run.

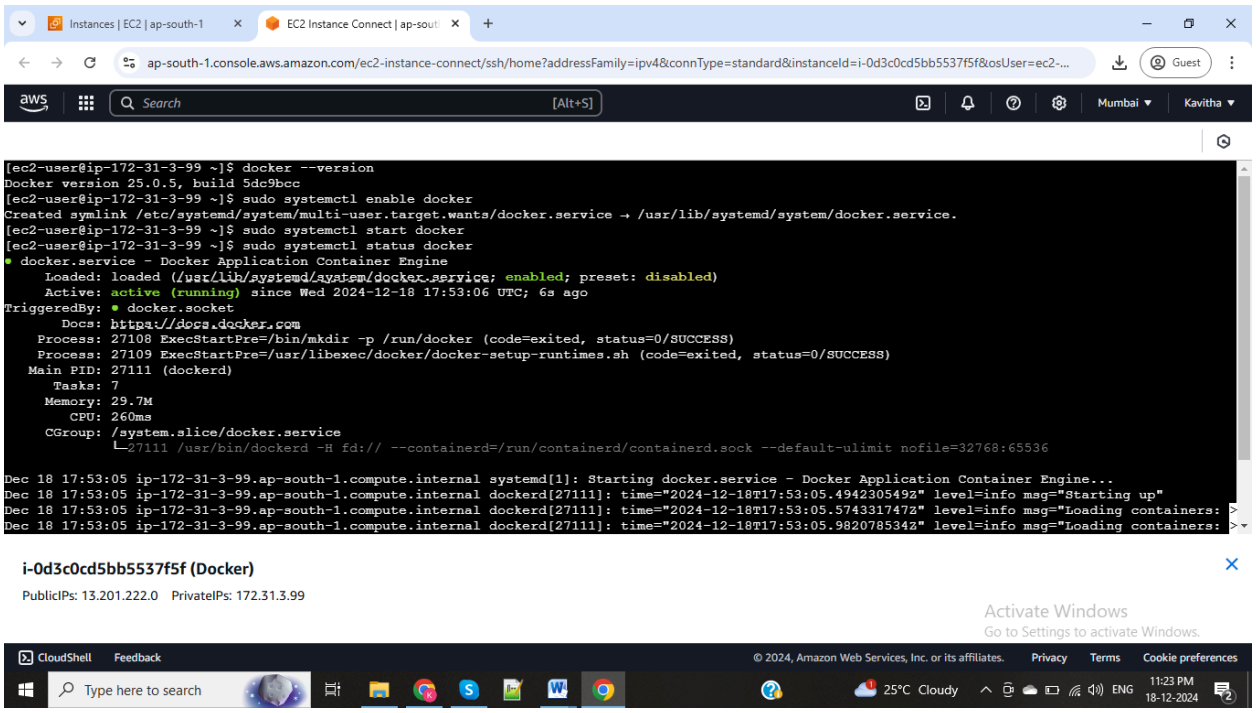
Key Features of Docker

1. **Containerization**: Packages applications and their dependencies into isolated containers.
2. **Portability**: Containers can run consistently across different environments (local, development, or production).
3. **Efficiency**: Containers share the host operating system kernel, making them more resource-efficient than virtual machines.
4. **Speed**: Containers are lightweight, start quickly, and require fewer resources.
5. **Version Control**: Docker enables versioning for container images, allowing easy updates and rollbacks.

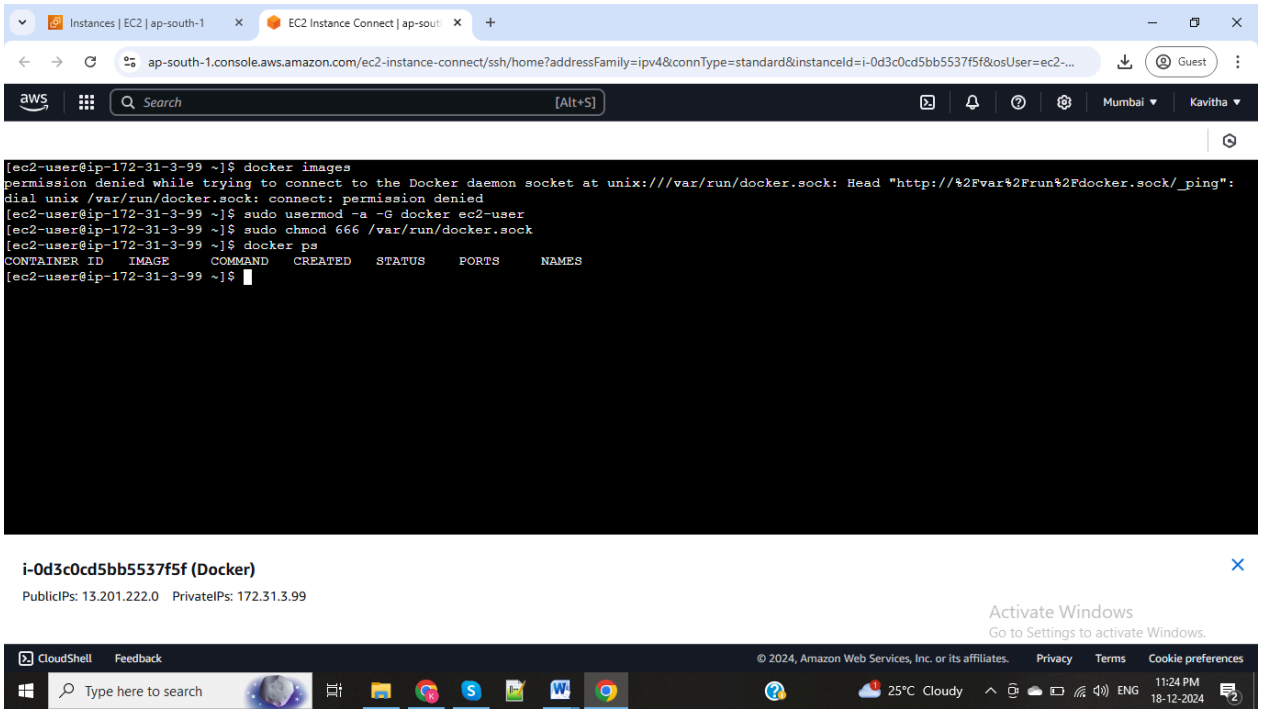
Created a EC2 instance and launch instance



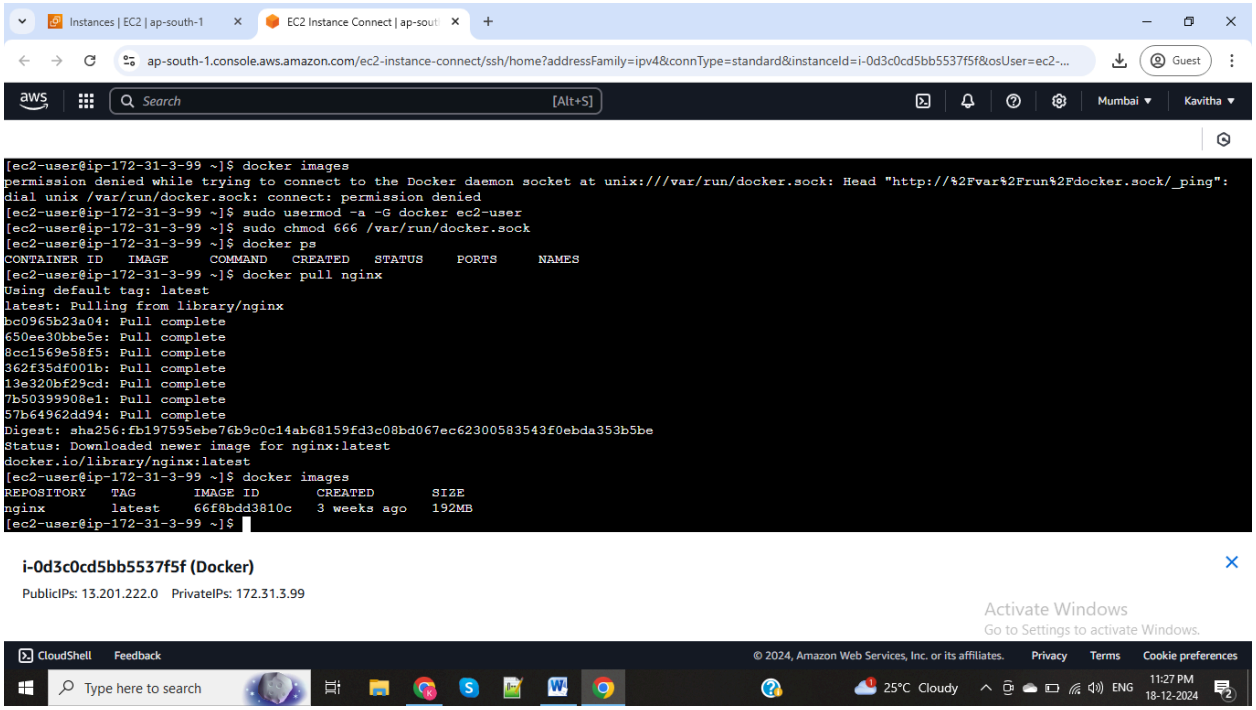
Install docker on EC2 instance – enable, start and check status



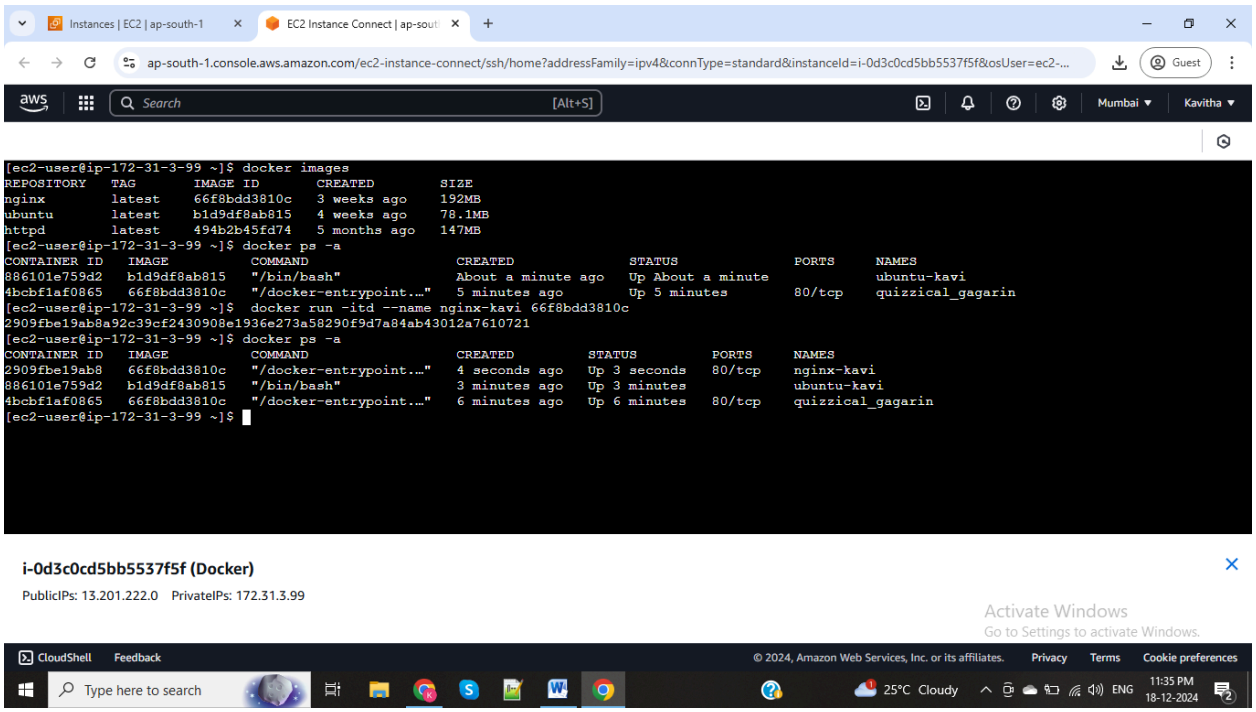
Add ec2-user with docker in groups --> sudo usermod -a -G docker ec2-user --> sudo chmod 666 /var/run/docker.sock. Now check docker command whether it works.



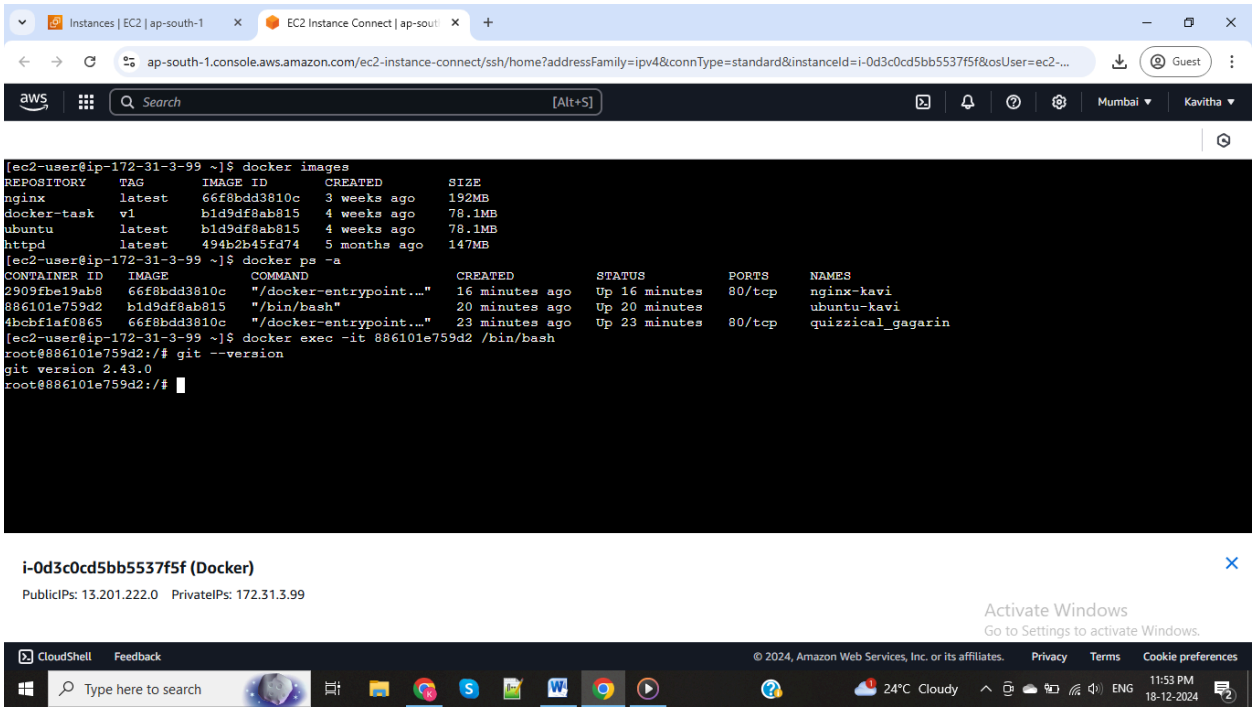
Login hub.docker.com and search images then nginx and ubuntu pulled from docker hub.



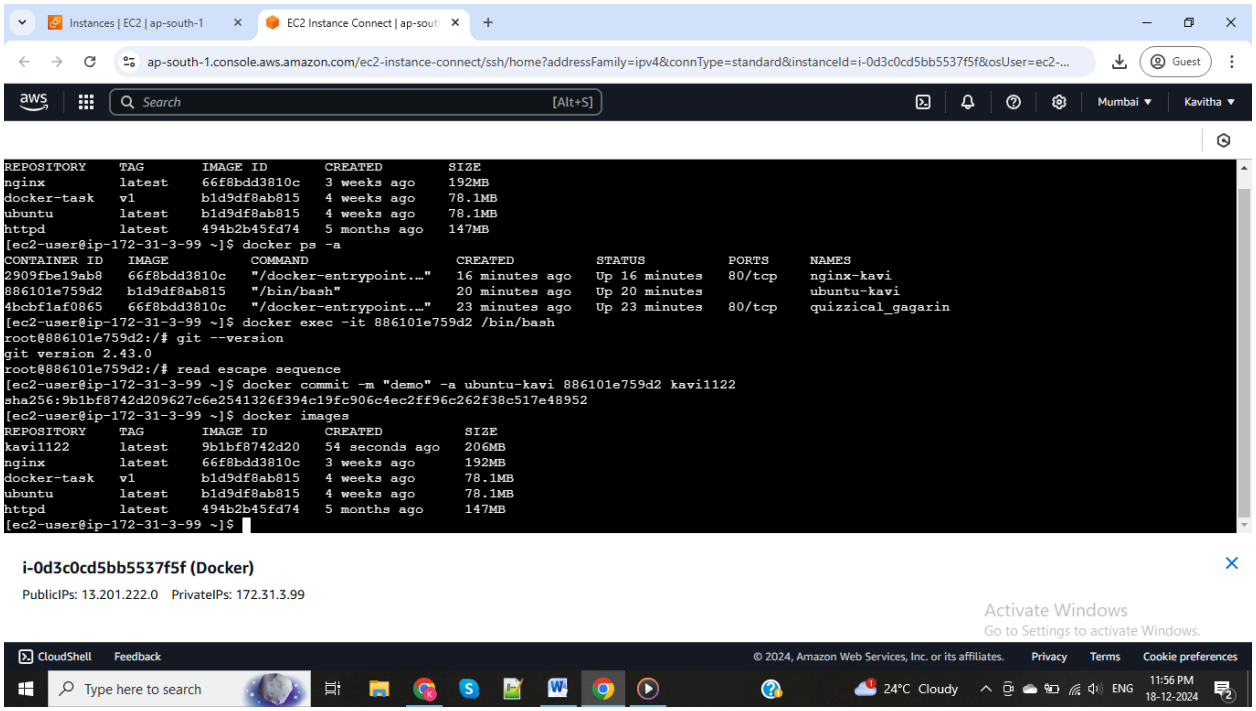
List the images with the help of docker images command and created containers from the docker image using --> docker run -itd --name nginx-kavi (image id)



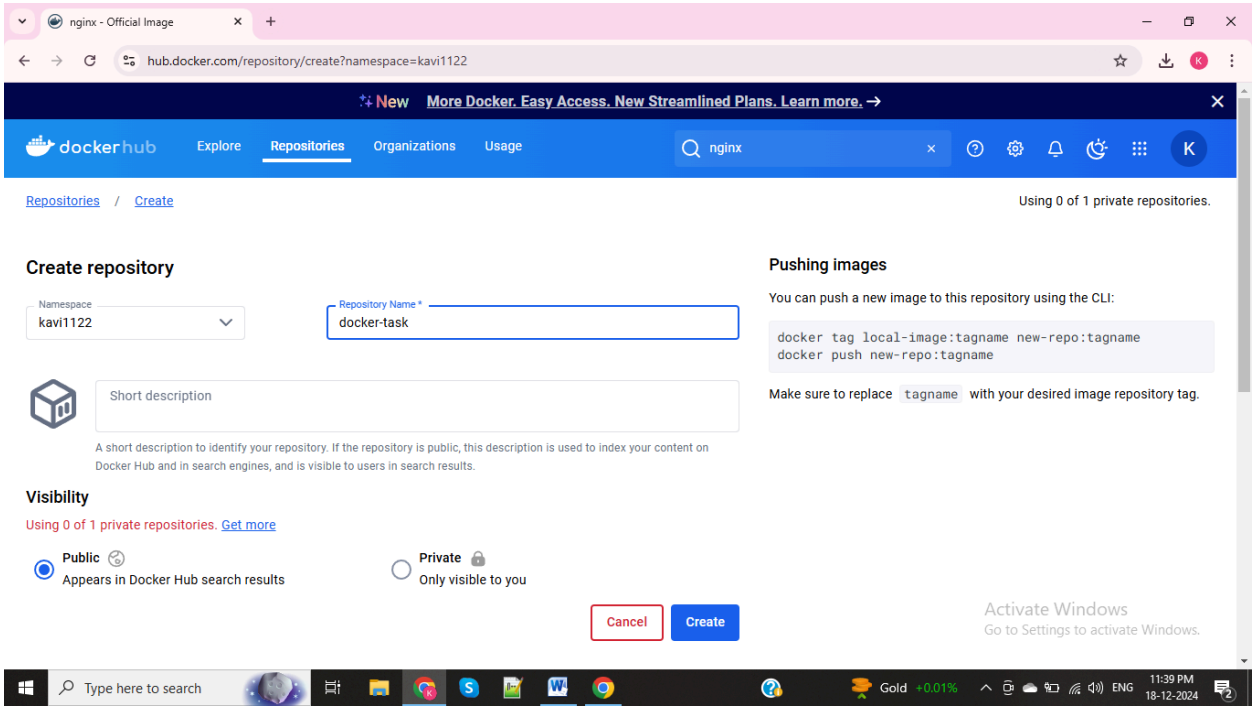
Go into the container --> docker exec -it (ubuntu container id) /bin/bash –> apt update -y –> apt install git -y and check git --version and for come outside the container use –> control p+q



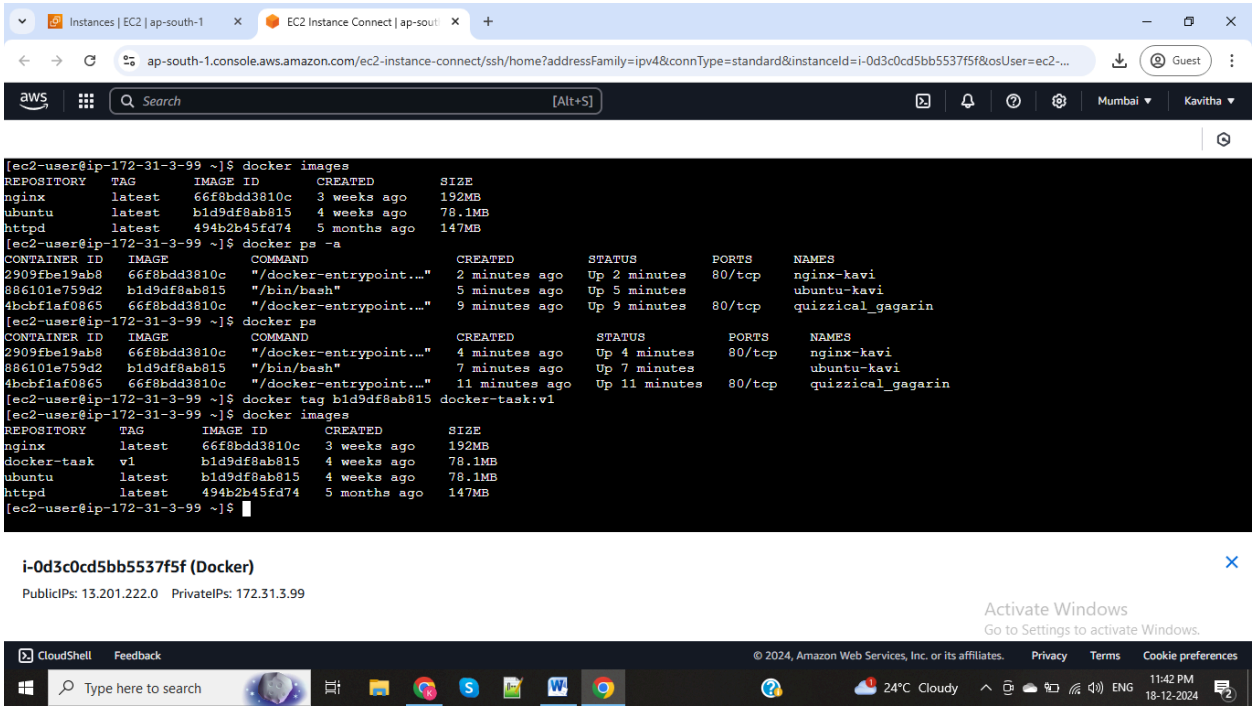
Created a Image from the container to push docker hub: Docker commit -m “demo” -a “name” (container id) docker hub username



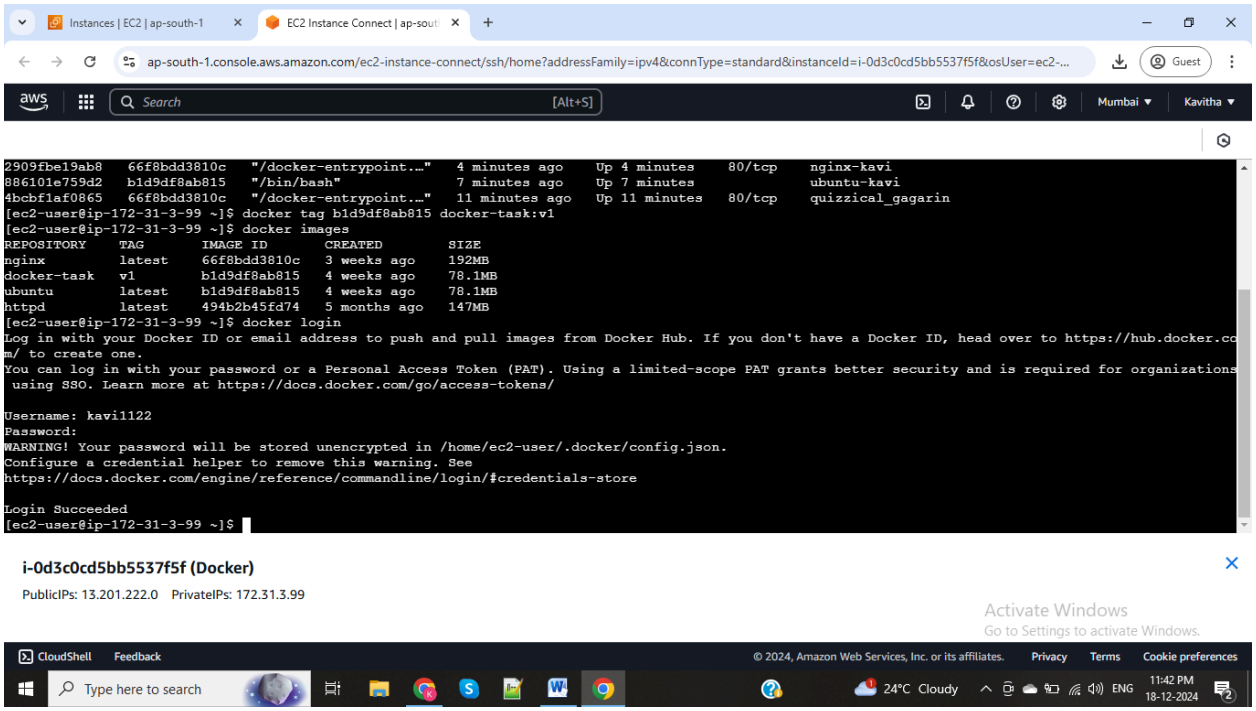
Login hub.docker.com –> Created a new repository



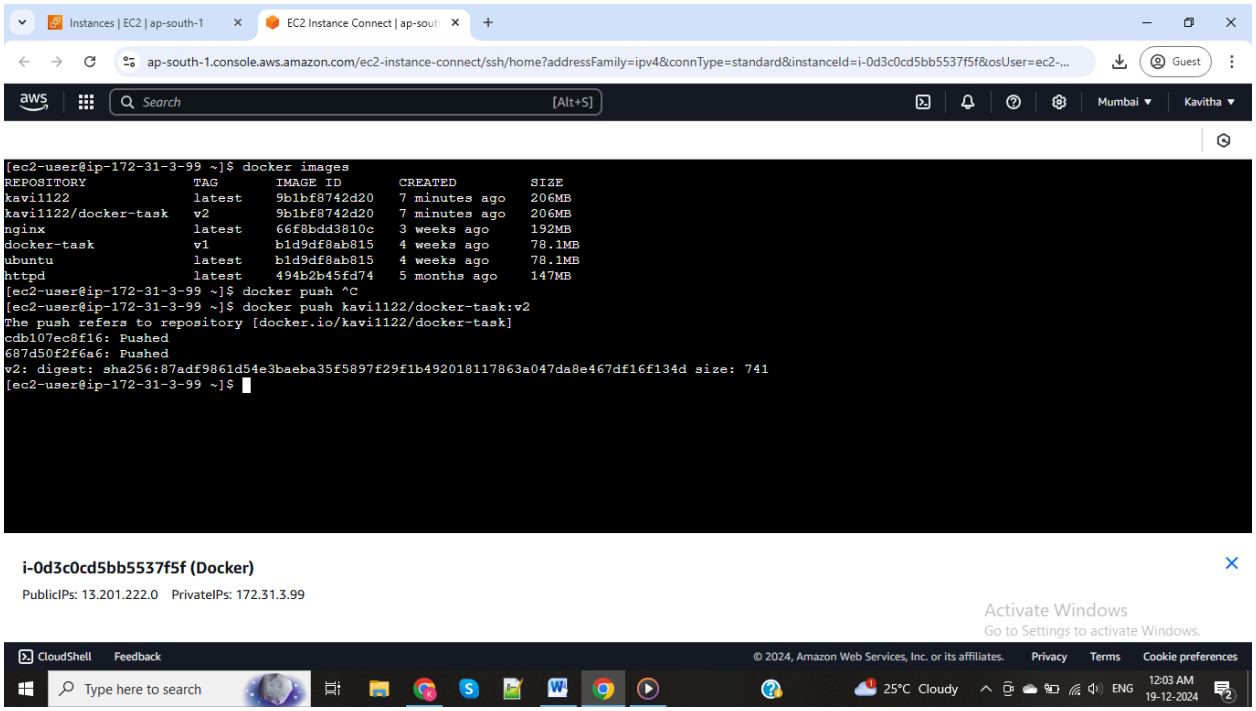
Check docker images –> docker tag (image id) (reponame):tag - created a new image



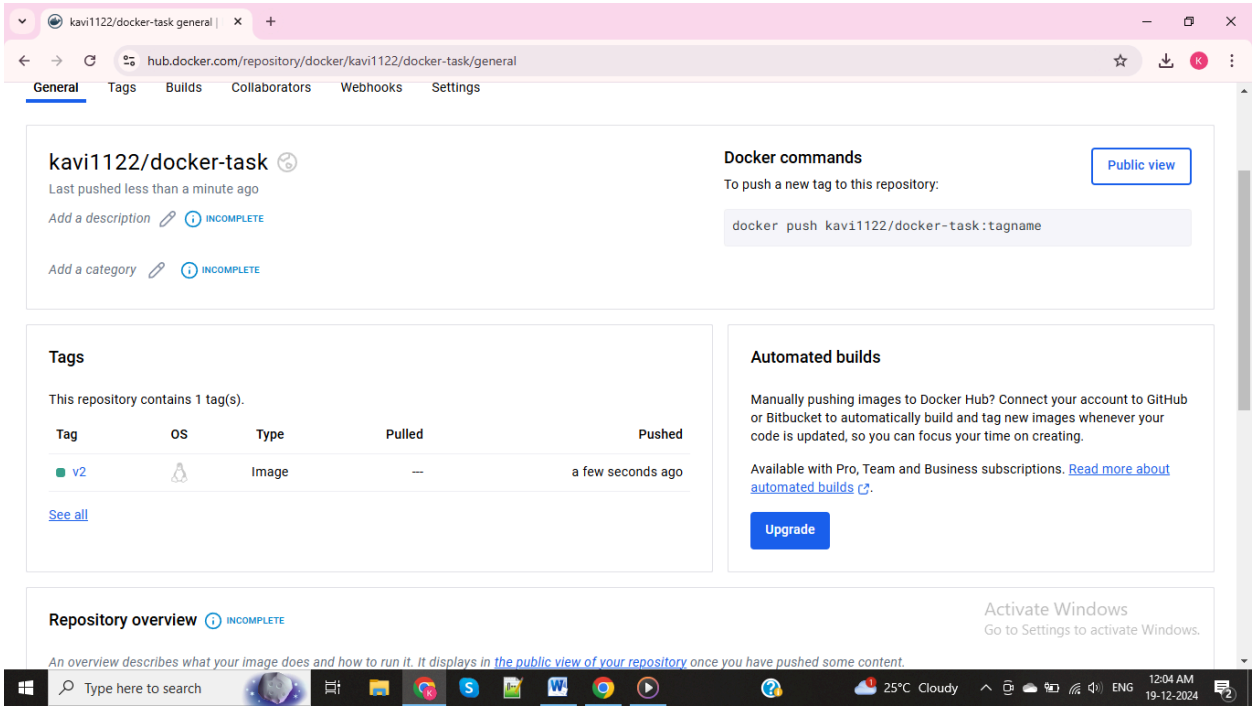
For pushing the image to docker hub for this give –> docker login and provide credentials and docker login succeeded



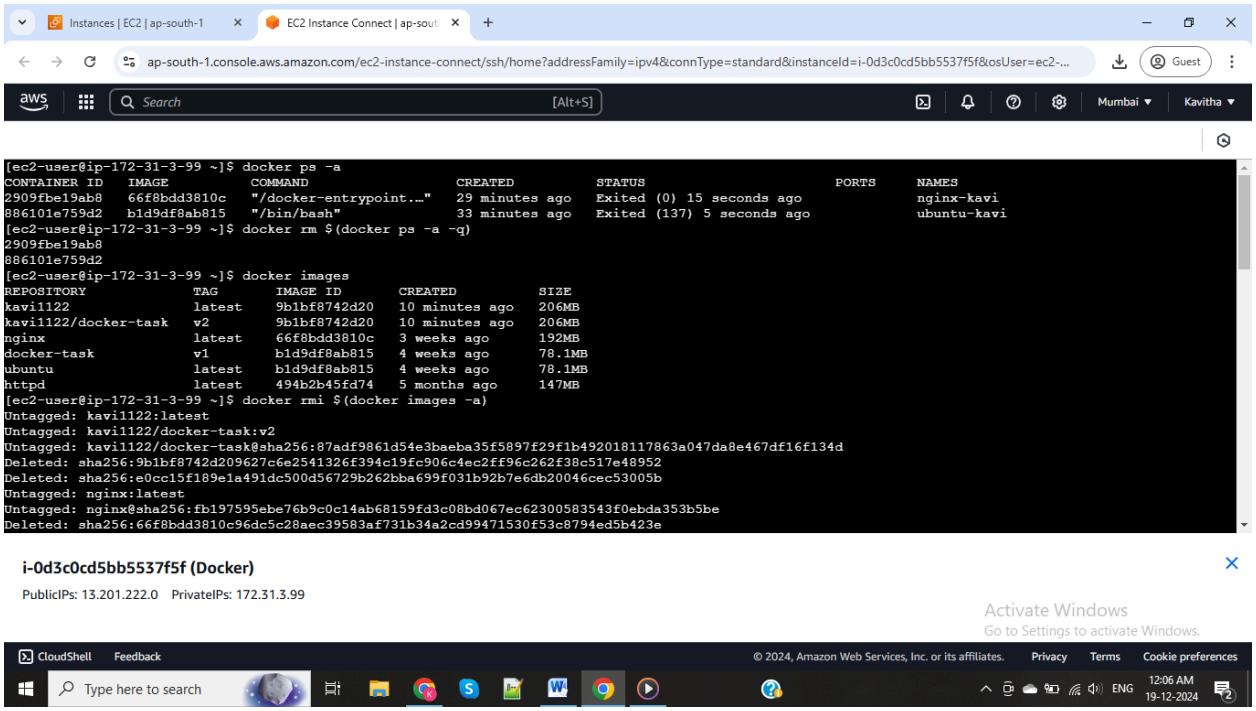
Now, give docker push command: docker push repository name:tag –> image successfully pushed to Docker Hub



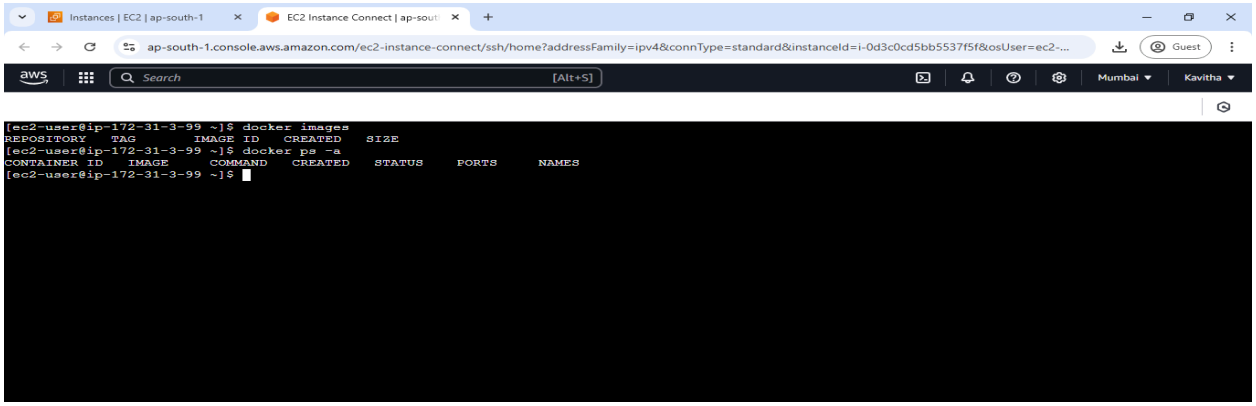
Successfully image pushed into Docker Hub and output



For Deleting all the containers and images –> docker rm $(docker ps -a -q) –> docker rmi $(docker images -a)



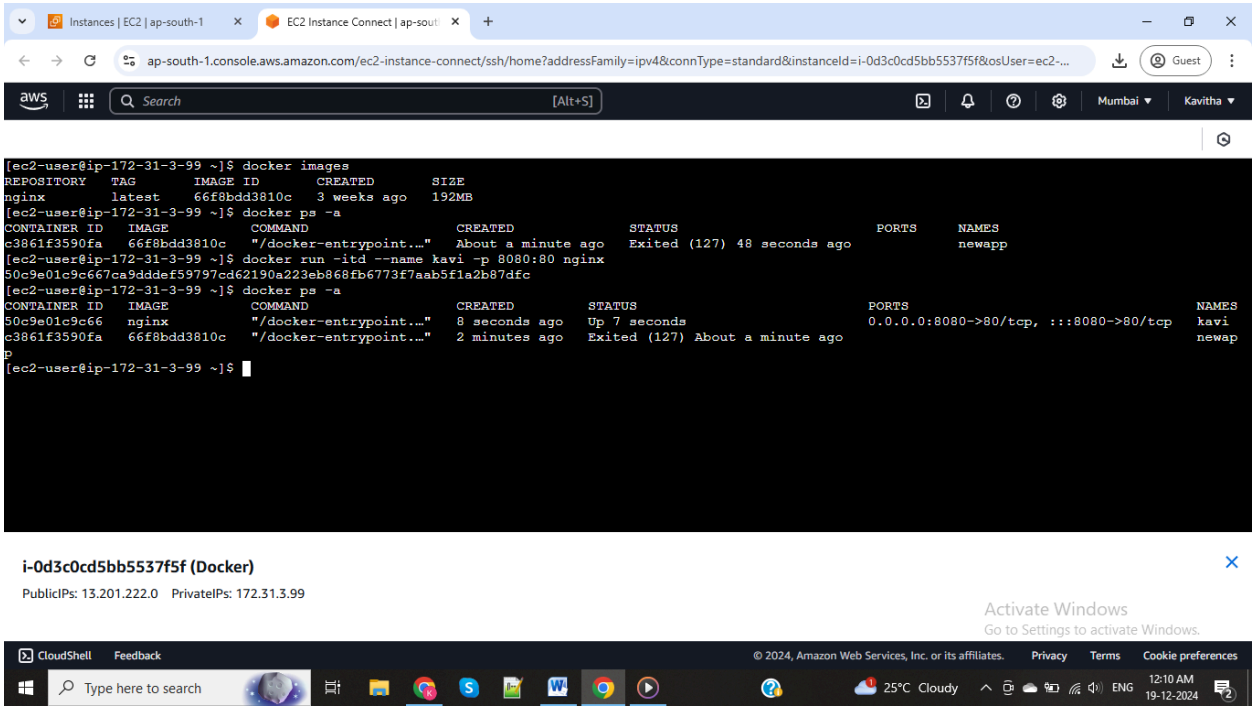
Deleted containers and images output



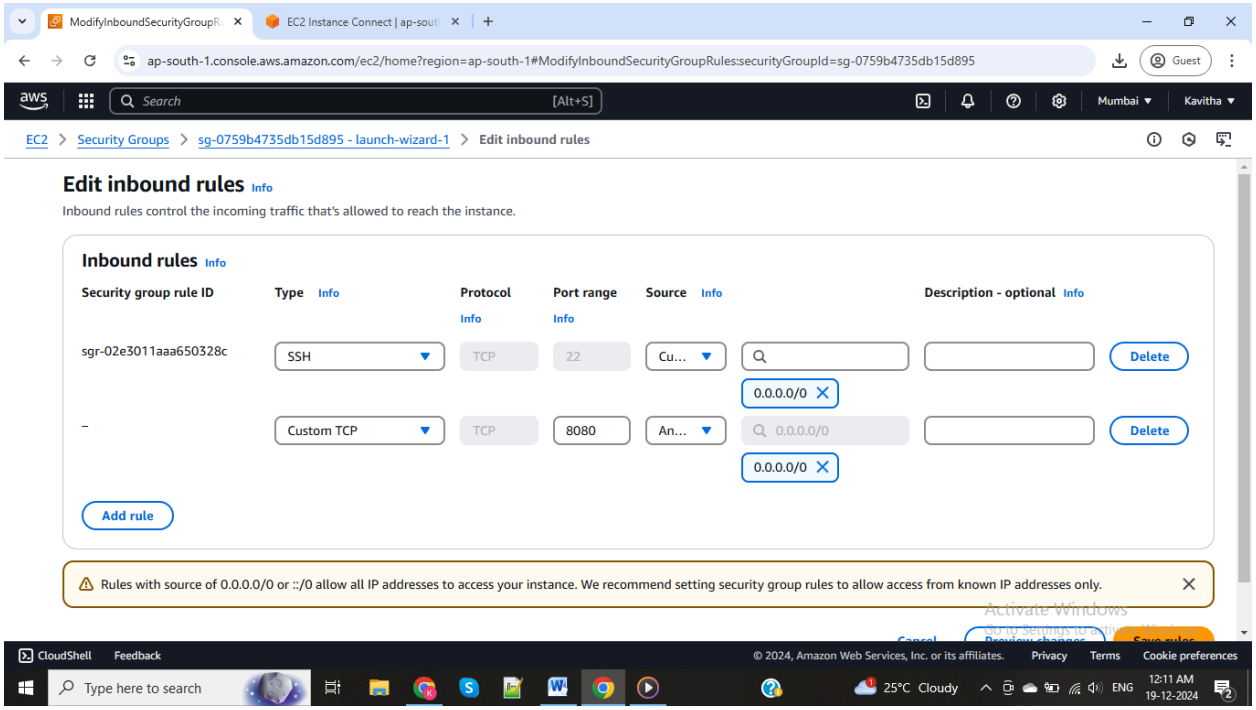
Inside the Container we have to access the application

Pull nginx image from dockerhub and create the container with port number for run the application in new web server or outside the server.

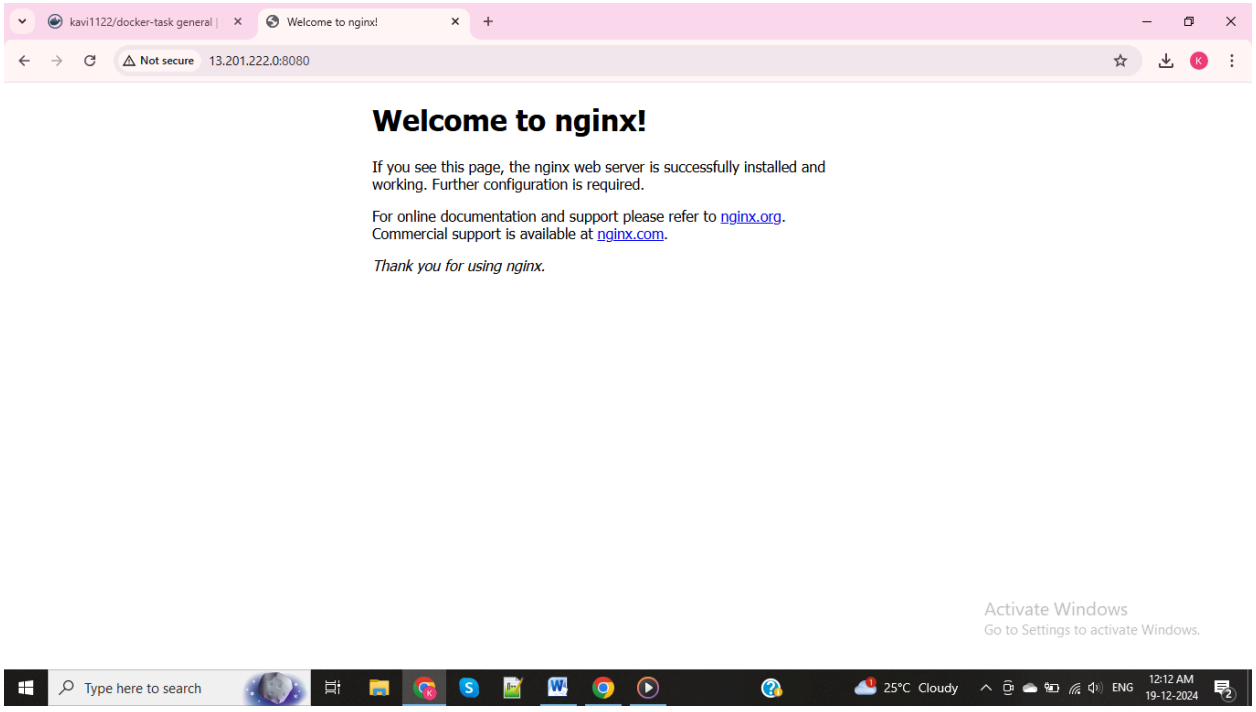
Docker run -it --name kavi -p 8080:80 (image name)



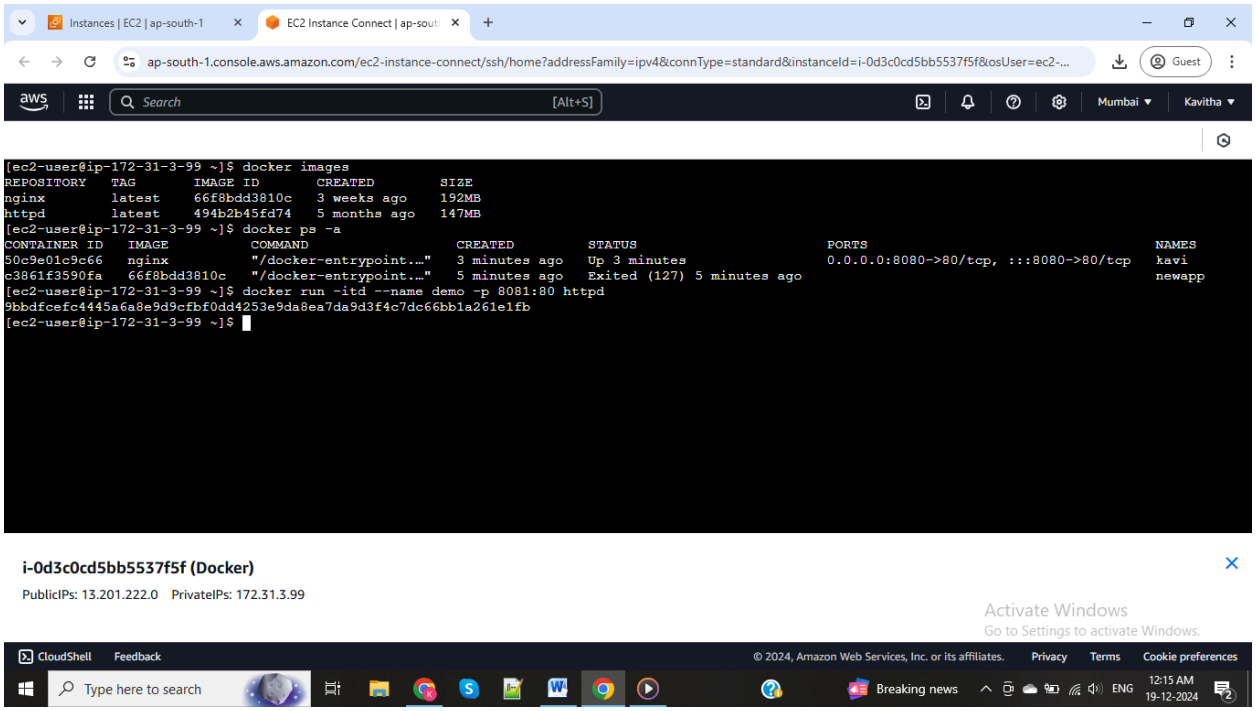
To access the application edit the inbound rules in security group and add 8080 port



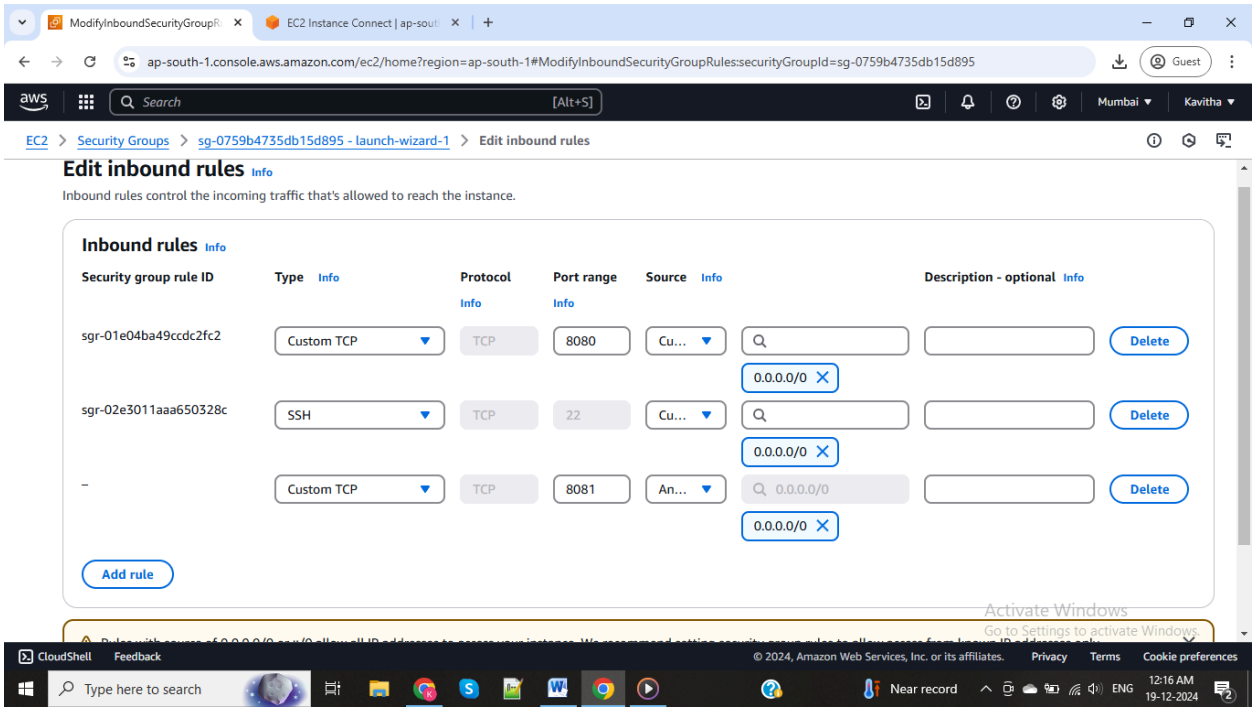
For checking whether it works use EC2 instance public ip address with port 8080



Pull httpd image from Docker Hub and create a new container with new port number for access the application in container with new port number



Edited the inbound rules to run the application in outside the web server



Access the application with public ip with port 8081