**DOCKER PULL IMAGE**

**DOCKER**

Docker is an open-source platform that enables developers to build, ship, and run applications inside lightweight, portable containers. A container bundles an application along with its dependencies, ensuring consistency across development, testing, and production environments.

**CONTAINER**

A **Docker container** is a lightweight, standalone, executable package that includes everything needed to run software, such as the code, runtime, libraries, and system tools. Containers provide an isolated environment for applications, ensuring they run consistently across various platforms and environments.

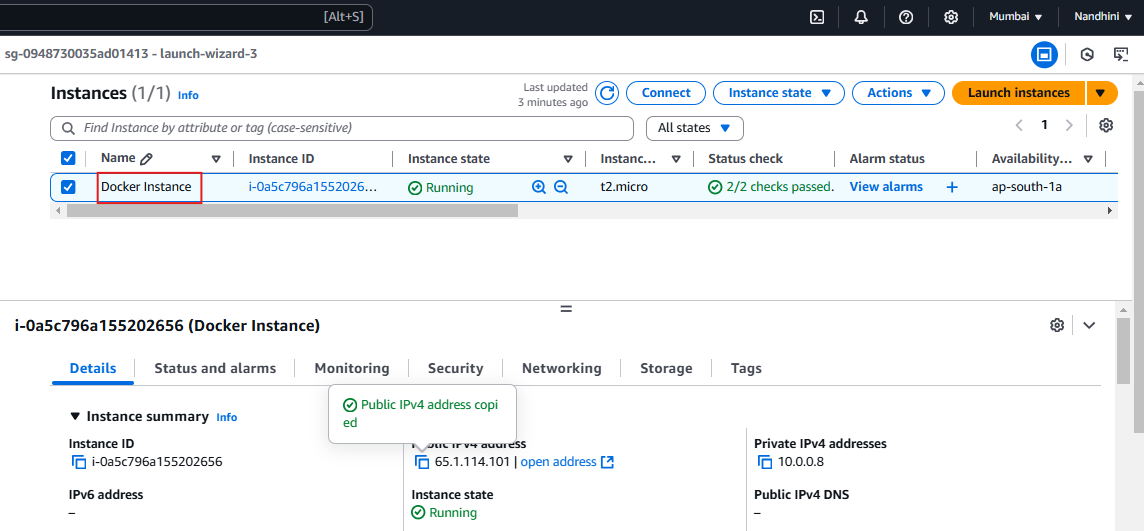
**IMAGES**

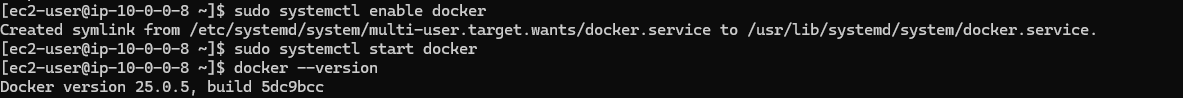
Docker images are the blueprint or template used to create Docker containers. An image contains everything needed to run an application: the code, libraries, configuration files, environment variables, and dependencies.

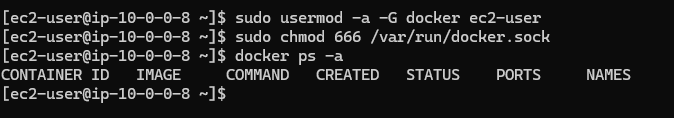
**PULL IMAGES**

Pulling a Docker image involves downloading it from a container registry to your local environment. This process ensures you have the required image to create and run containers.

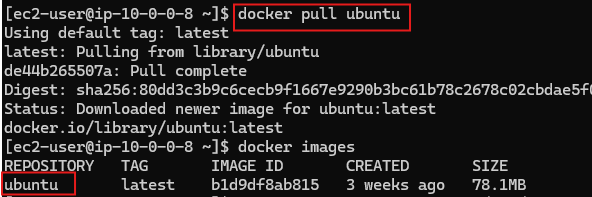
**Create an instance and deploy Docker.**





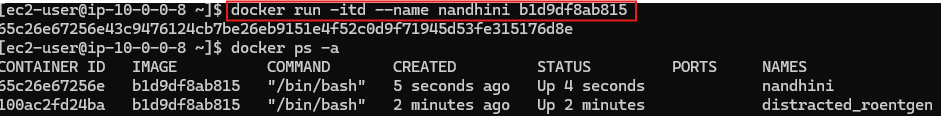
**Grouping ec2-user in Docker.**

**Pull image from Docker hub:**

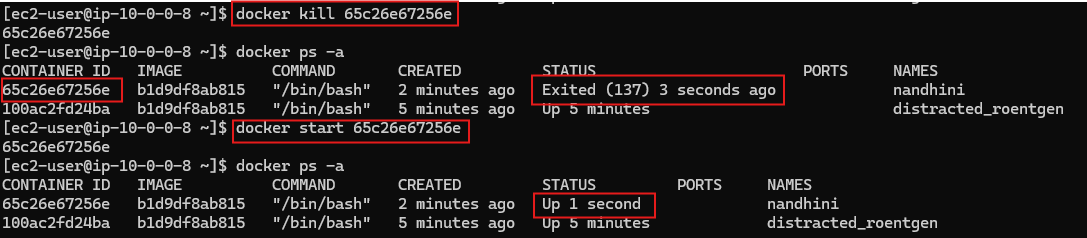


**Creating a container from a pulled image,**

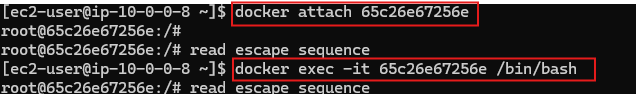
I have created two Containers. Container with default name and container with custom name



**To stop and start the container, below commands are used,**

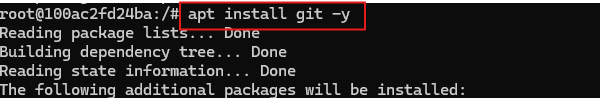
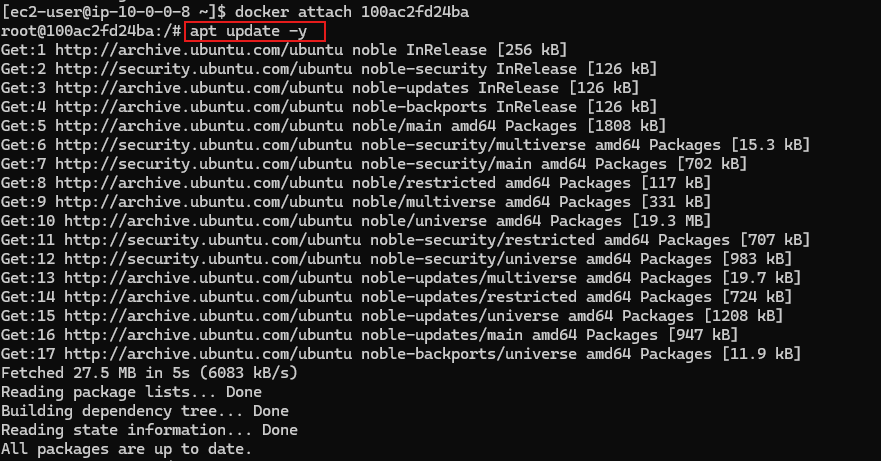


**To enter into the container instead of using start commands, we can use below commands,**



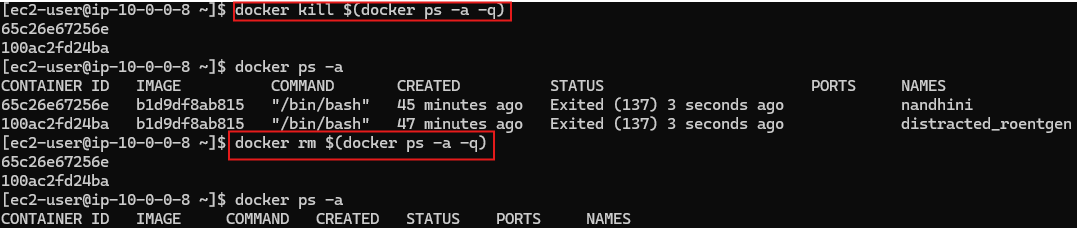
**Even though EC2 instance launched using Linux OS, we can update or install other OS into the container.**

I have updated the Ubuntu OS and installed Git in container.



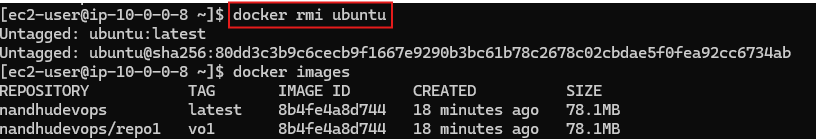
**How to delete multiple containers and images:**

Before deleting a container we have to stop the container, using Kill command.

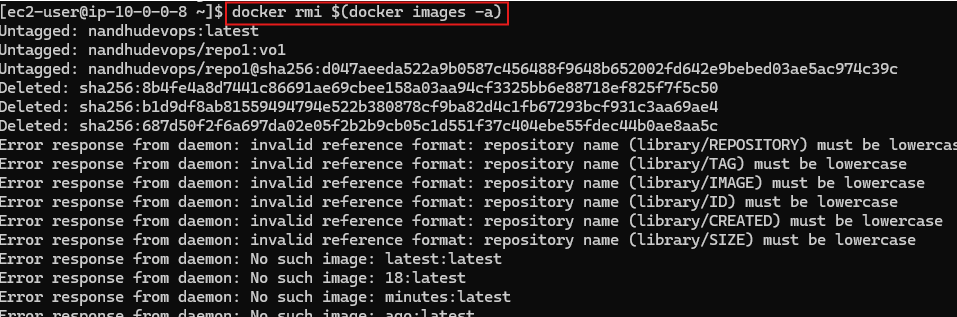


**Before deleting image we have to delete related container, “rmi” command used to delete image.**

Deleting single image,

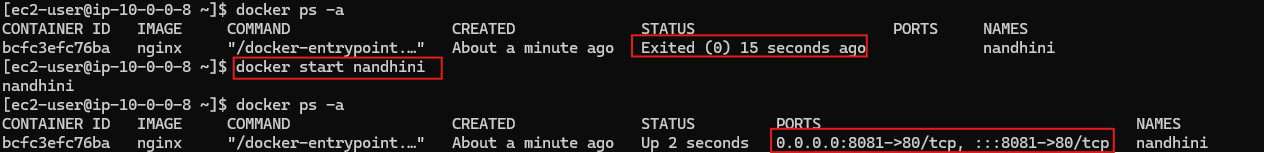
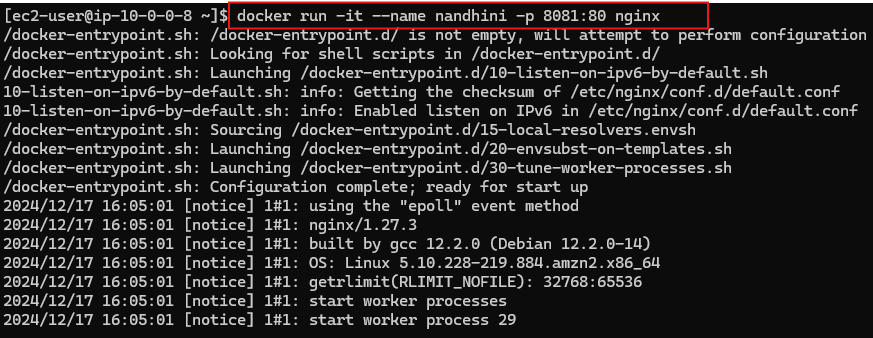


Deleting multiple image,

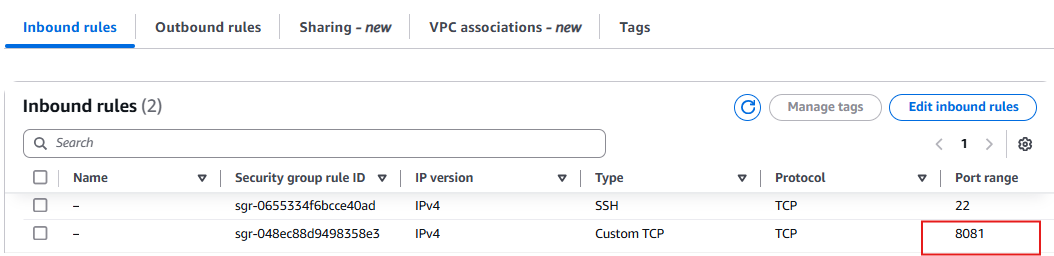


**Accessing our container application using Public IP and port number.**

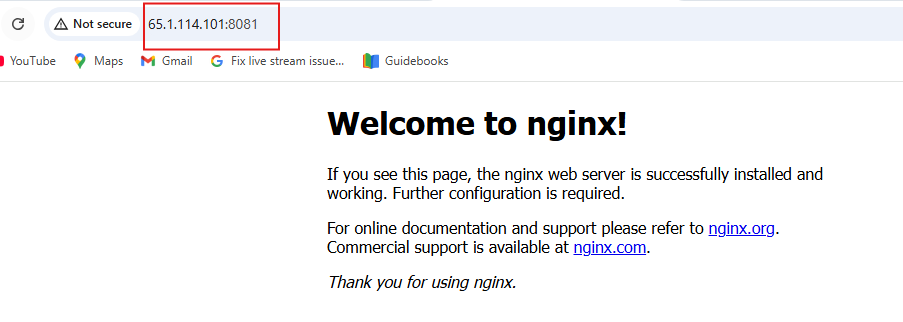
Pull required image form Docker hub, create container using port no, to access it outside the server. Here I have created container with port no 8081.



In security group edit the inbound rule, provide access to port 8081,



Now, I can able to access application from container outside using post no.



**Create image using “—rm” command, which is used to delete the container whenever it get stopped or killed.**

