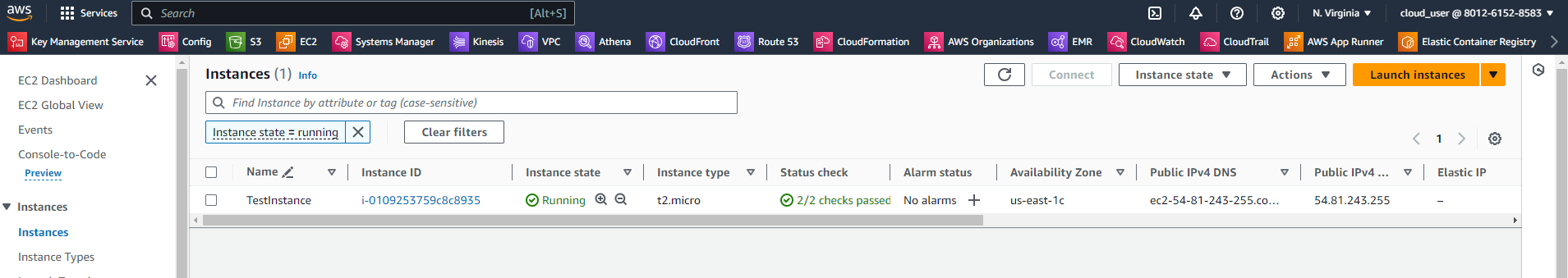
1. **Connect to your EC2 instance:** Herewe SSH to connect to our EC2 instance. Here I have created EC2 Instance named TestInstance with ubuntu OS and a security group to ssh from my IP



SSH to instance using following command

**ssh -i "ubuntutest.pem"** [**ubuntu@ec2-54-81-243-255.compute-1.amazonaws.com**](mailto:ubuntu@ec2-54-81-243-255.compute-1.amazonaws.com)

login to instance as root user using **sh -i**

1. **Update your package index:**

sudo apt update

1. **Install necessary packages to allow apt to use a repository over HTTPS:**

sudo apt install apt-transport-https ca-certificates curl software-properties-common

1. **Add Docker's official GPG key:**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

1. **Set up the stable Docker repository:**

echo "deb [signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

1. **Update the package index again:**

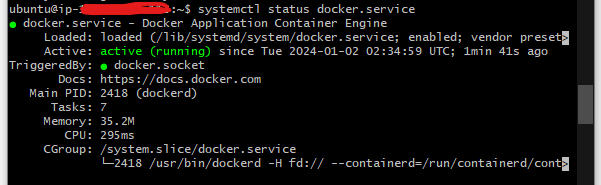
sudo apt update

1. **Install Docker:**

sudo apt install docker-ce docker-ce-cli containerd.io

1. **Start the Docker service:**

sudo systemctl start docker

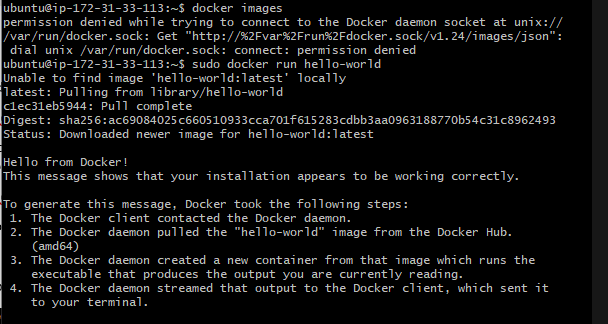


Additionally, if we want Docker to start on boot we use:

sudo systemctl enable docker

Now, Docker should be installed on your EC2 instance. we can test it by running a simple Docker command, like:

sudo docker run hello-world



Some of docker basic commands are:

* **List Docker images:**

docker images



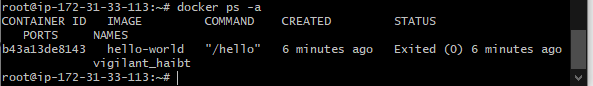
* **List running containers:**

docker ps



* **List all containers (including stopped ones):**

docker ps -a



* **Docker Search:**

Docker search is used to search for specified images through dockerhub

* **Docker Pull**

Docker pull is used to pull a specific image from docker hub

* **Docker stop**

Docker stop stops a container using container name or container ID

* **Docker Kill**

Stops container immediately by killing its execution while docker stop is used to shut down the container in its own time

* **Docker Restart**

Restarts the stopped container , it’s recommended to use this after rebooting the system

* **Docker exec**

Used to access the container that is running

* **Docker login**

To login to docker hub with login credentials

* **Docker Commit**

Used to create or save an image of the edited container

* **Docker push**

Helps to push or upload a docker image on the repo or docker hub

* **Docker Network**

Used to know the details of the list of networks in the cluster

* **Docker rmi**

Used to free up some disk space we can use image id along with above command to do it for a specific container

* **Docker copy**

Copied a file from docker to local system

* **Docker logs**

Used to check logs of all docker container with corresponding container id

* **Docker volume**

Creates a volume so that the docker container can use it to store data

* **Docker logout**

Will log out of the docker hub