Project-2 Deployment of Online Ticketing and Event Management Applications and Infrastructure.

Submitted By – Arya Chaurasia

Date of Submission – 14/12/2025

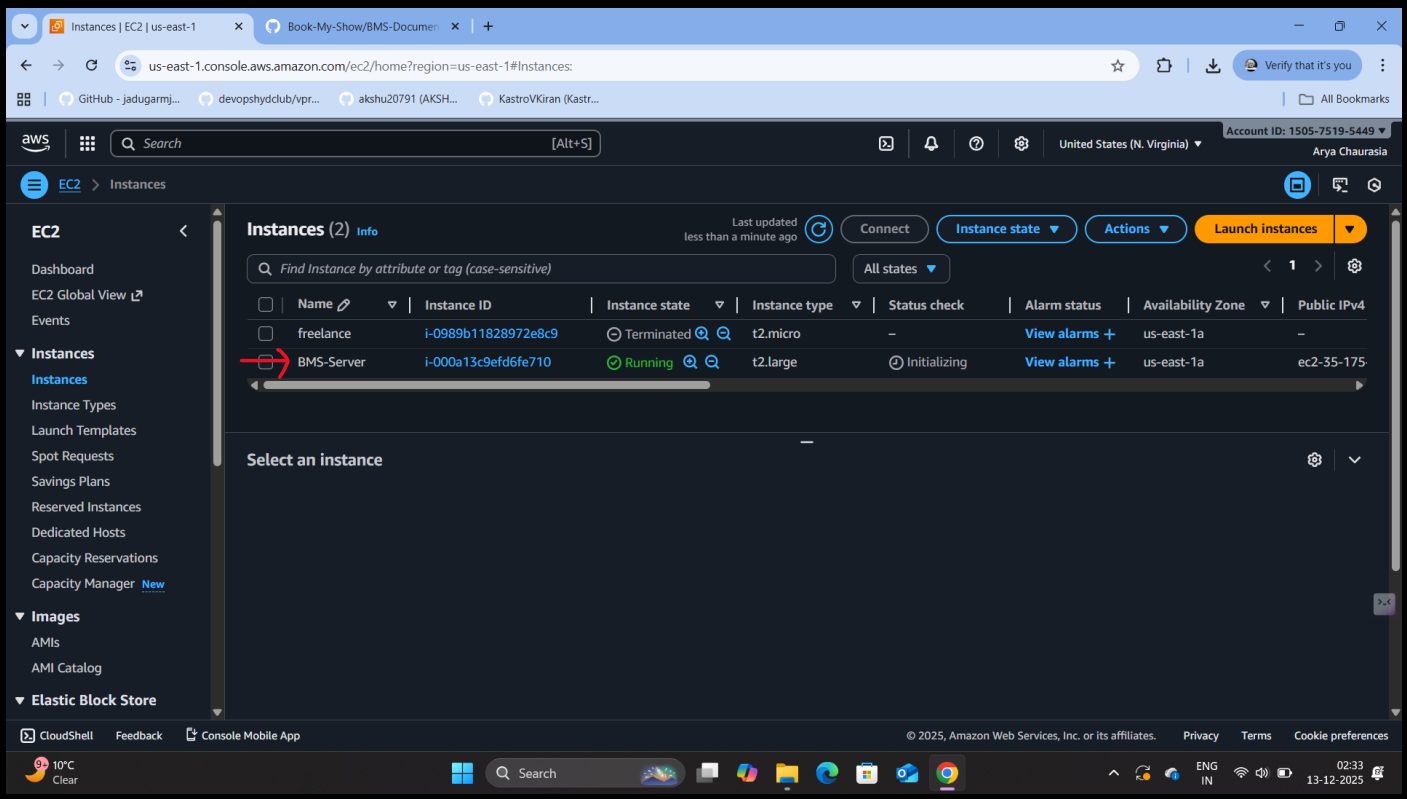
Project Overview

BookMyShow is a popular online movie ticketing platform that allows users browse movies, book tickets, and manage their bookings. The company is transitioning from a monolithic architecture to a microservices-based architecture to improve scalability, deployment efficiency, and infrastructure management.

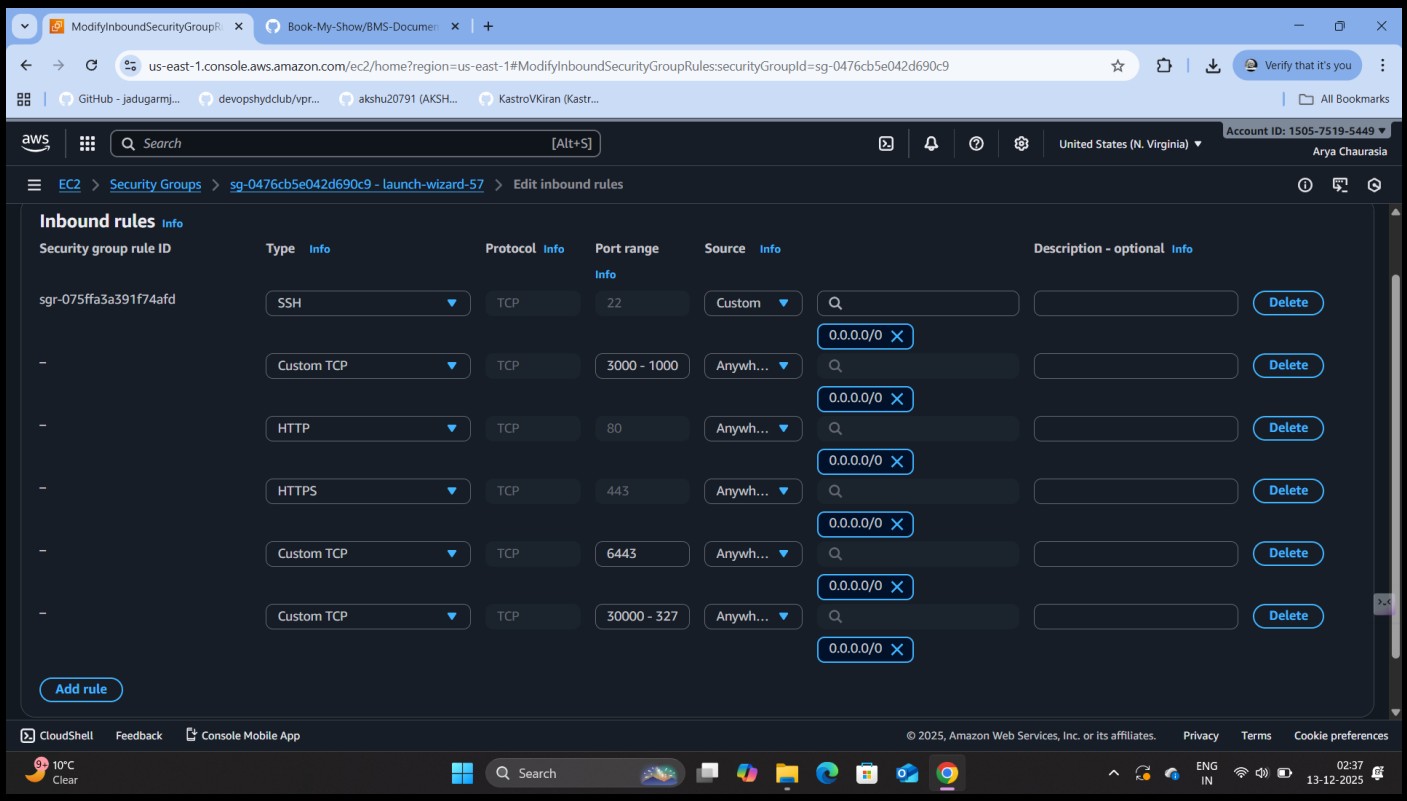
GitHub link - https://github.com/Aryachaurasia23/Book-My-Show

Steps to Deploy the Website successfully

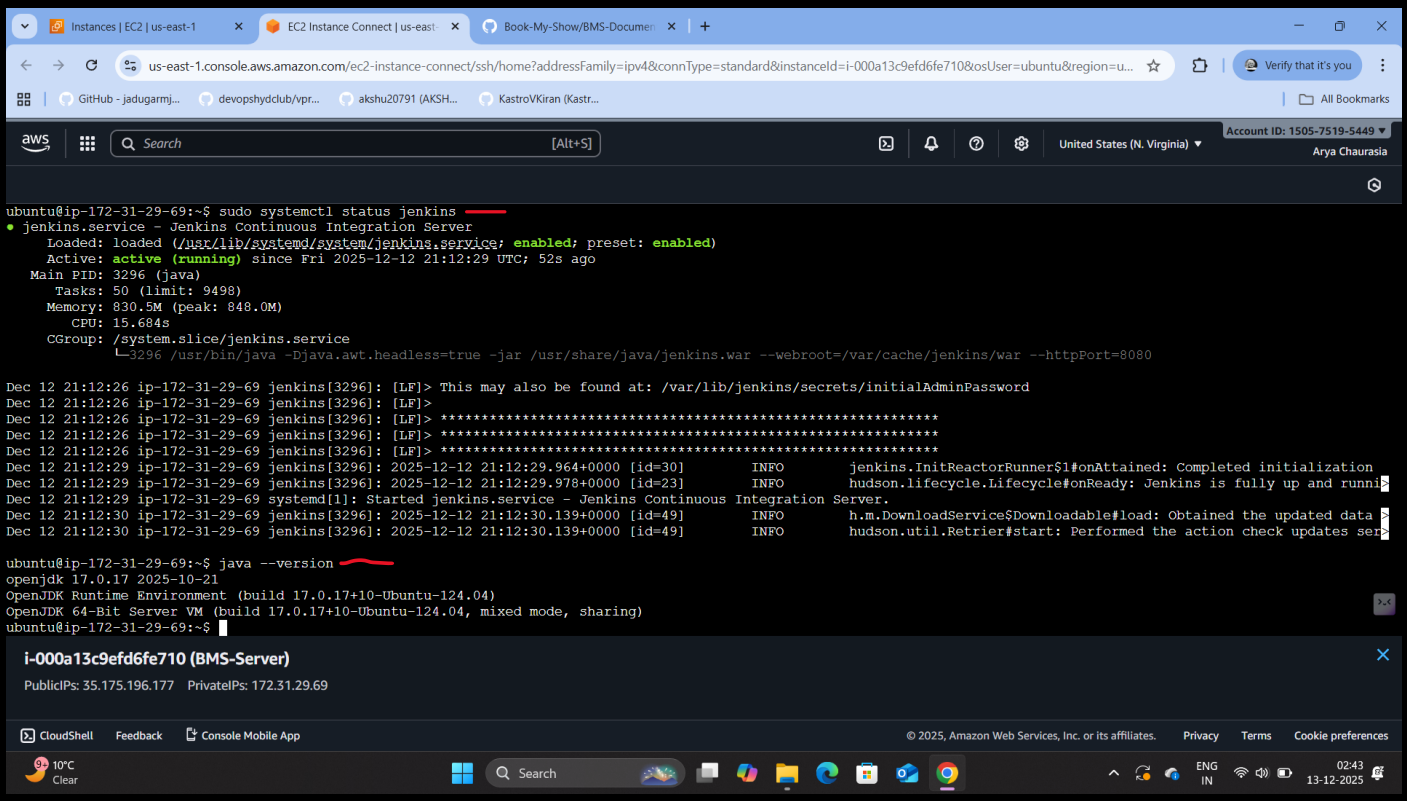
Step-1- We will create an EC2 Instance BMS-Server (Ubuntu 24.04,t2.large) with EBS storage 28 Gib.



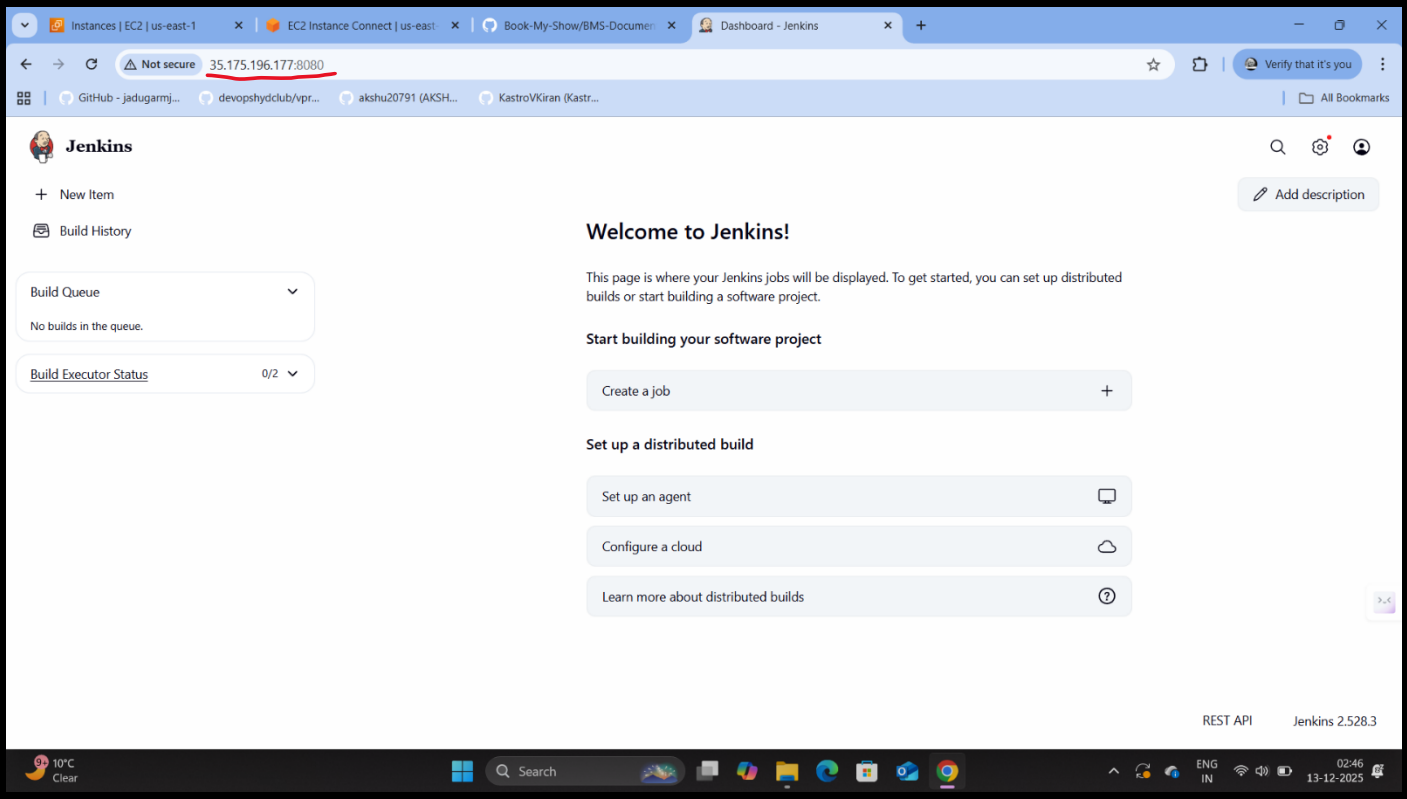
Step-2-Opened below ports for the security group of the Bms-Server – 22, 80, 3000-10000, 443, 6443, 30000-32767.



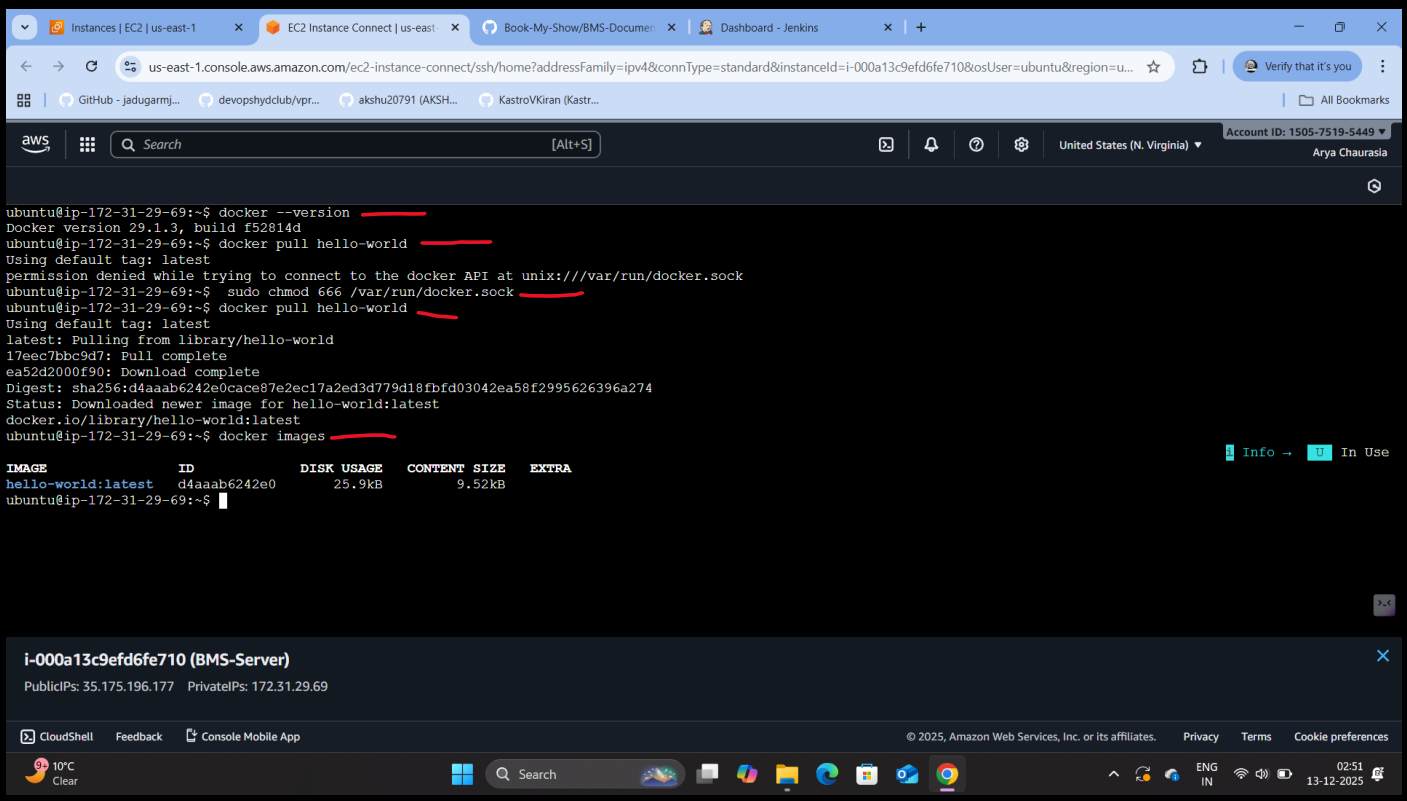
Step-3- We installed Jenkins and JAVA and validated if they are installed or not.



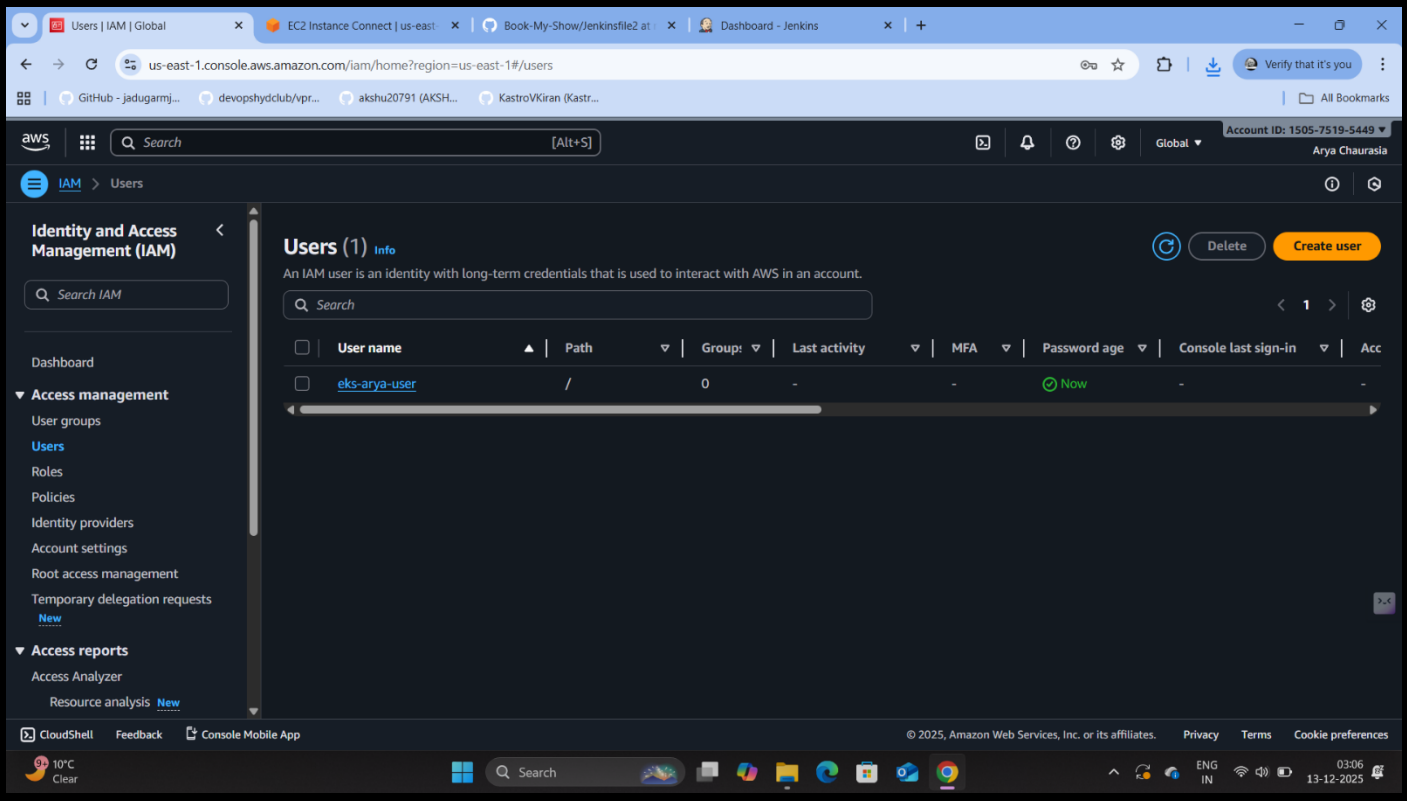
Step-4- we opened Jenkins with <Bms-server IP> : 8080



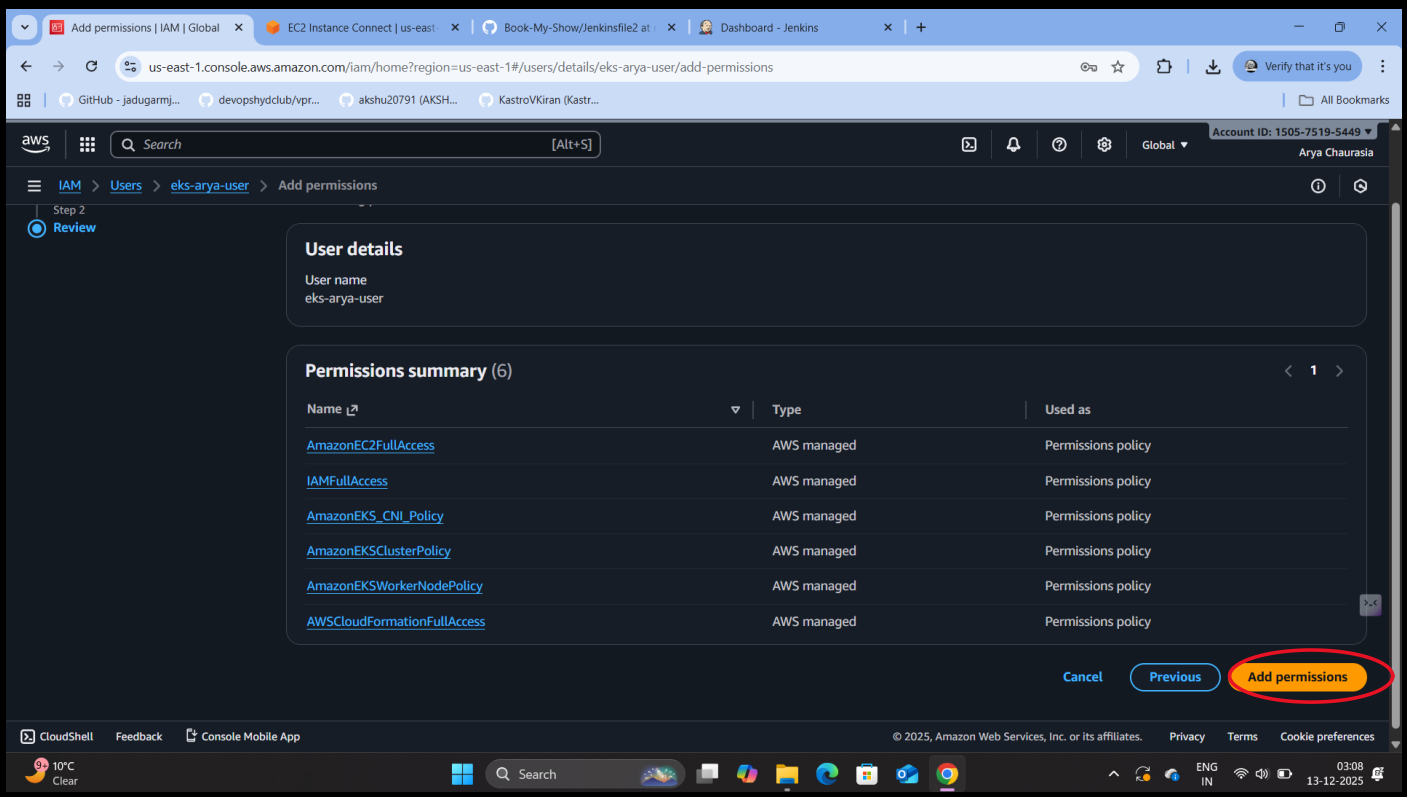
Step-5 Installed docker and giving permission to pull the images.



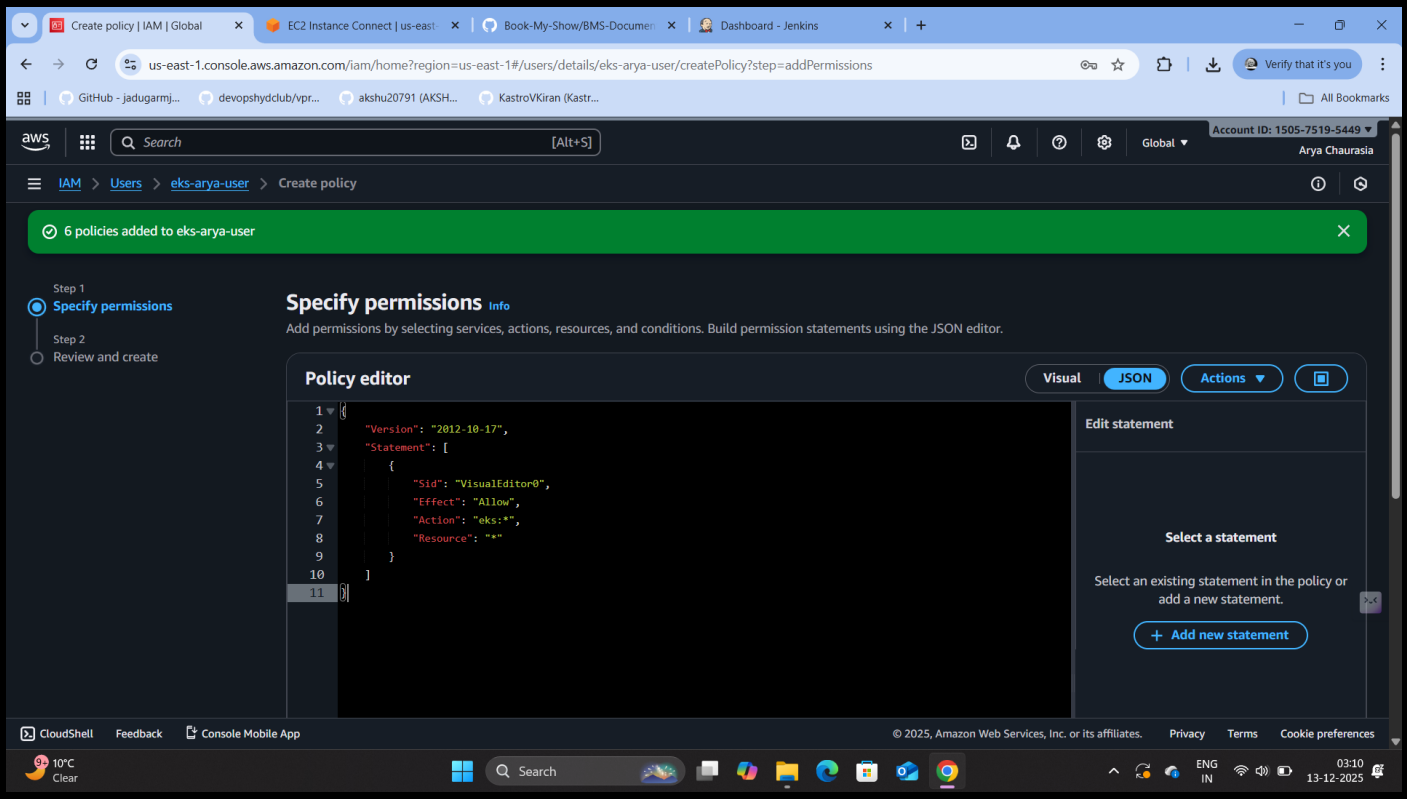
Step-6- To create EKS Cluster, its not recommended to create using Root account so we created an IAM user – eks-arya-user



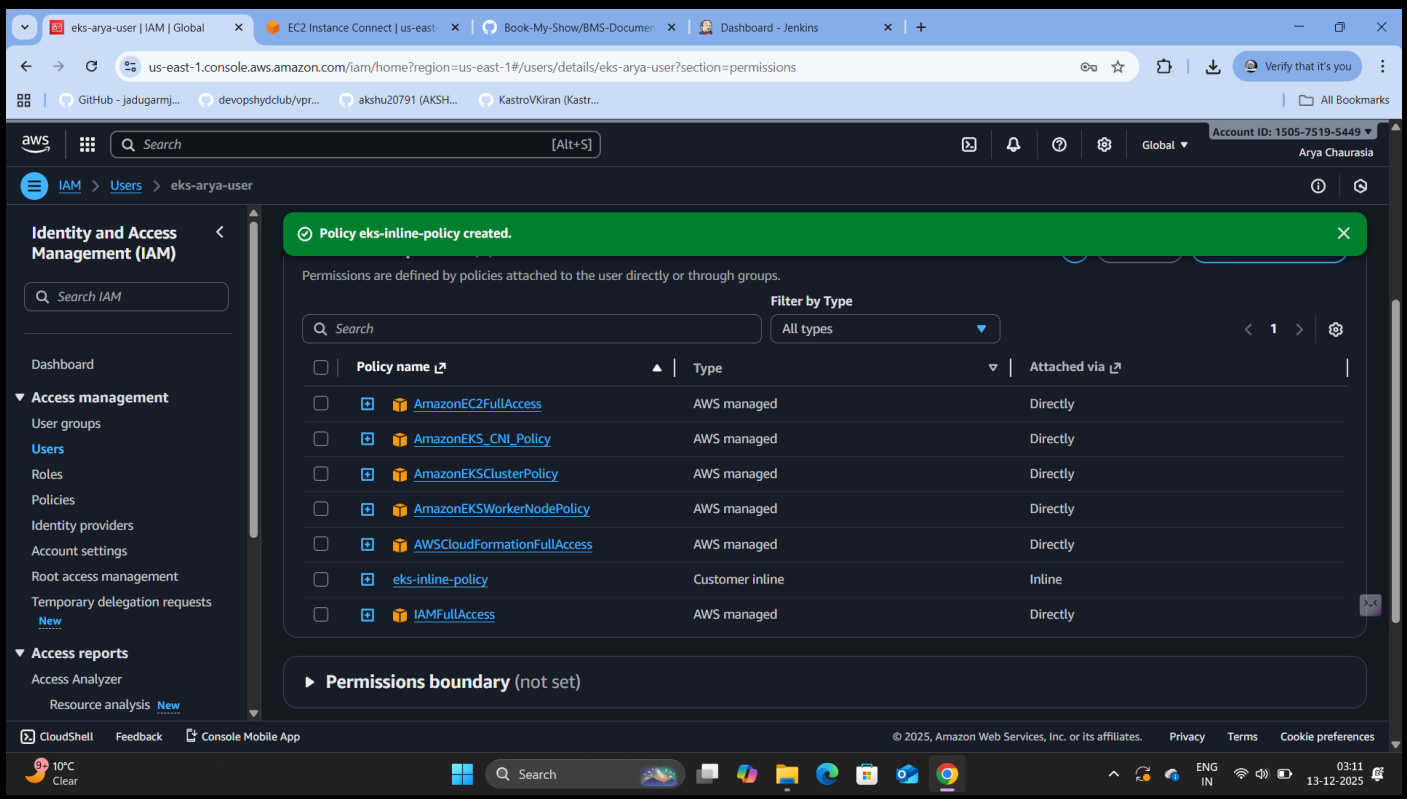
Step-7- We attached these policies to the User.



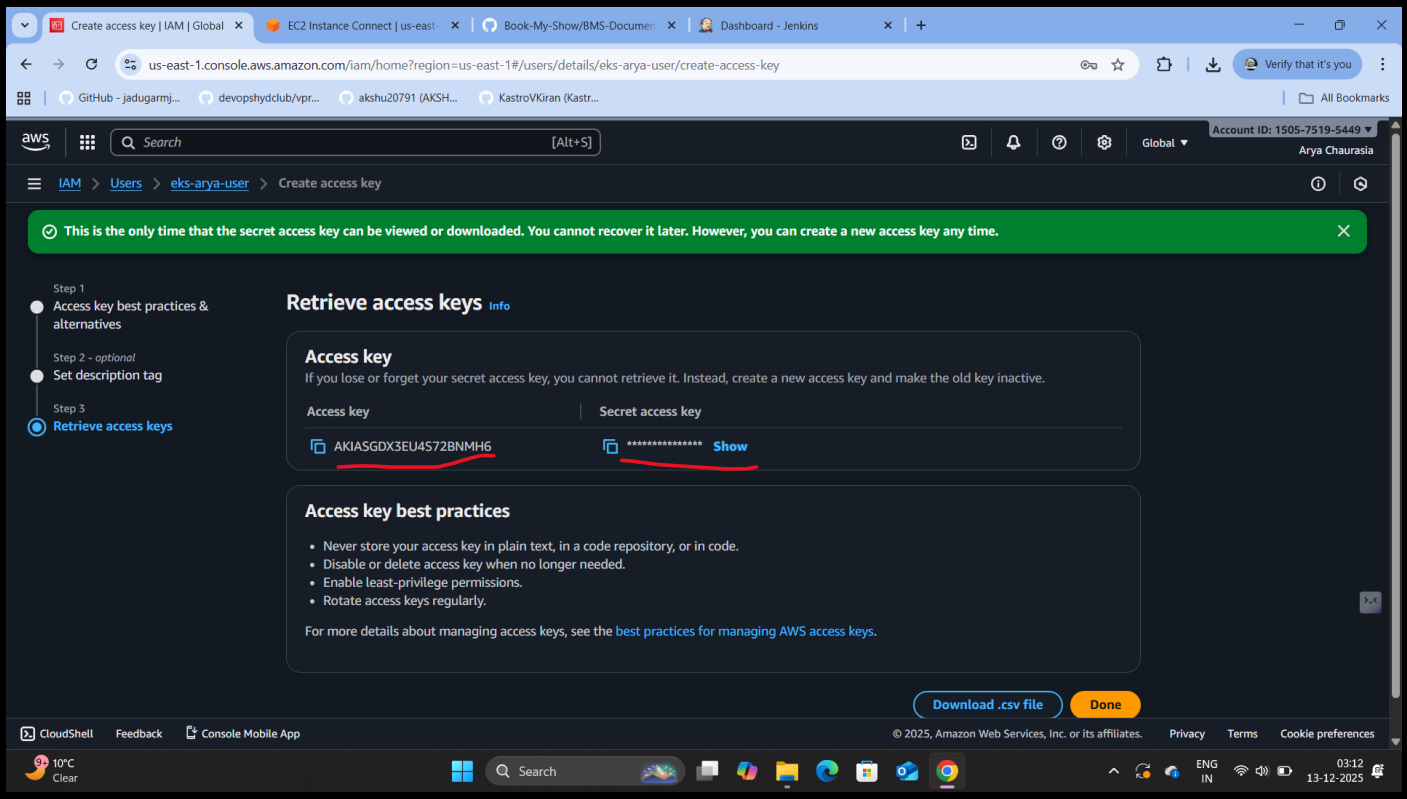
Step-8- Attached this inline policy for the same user.



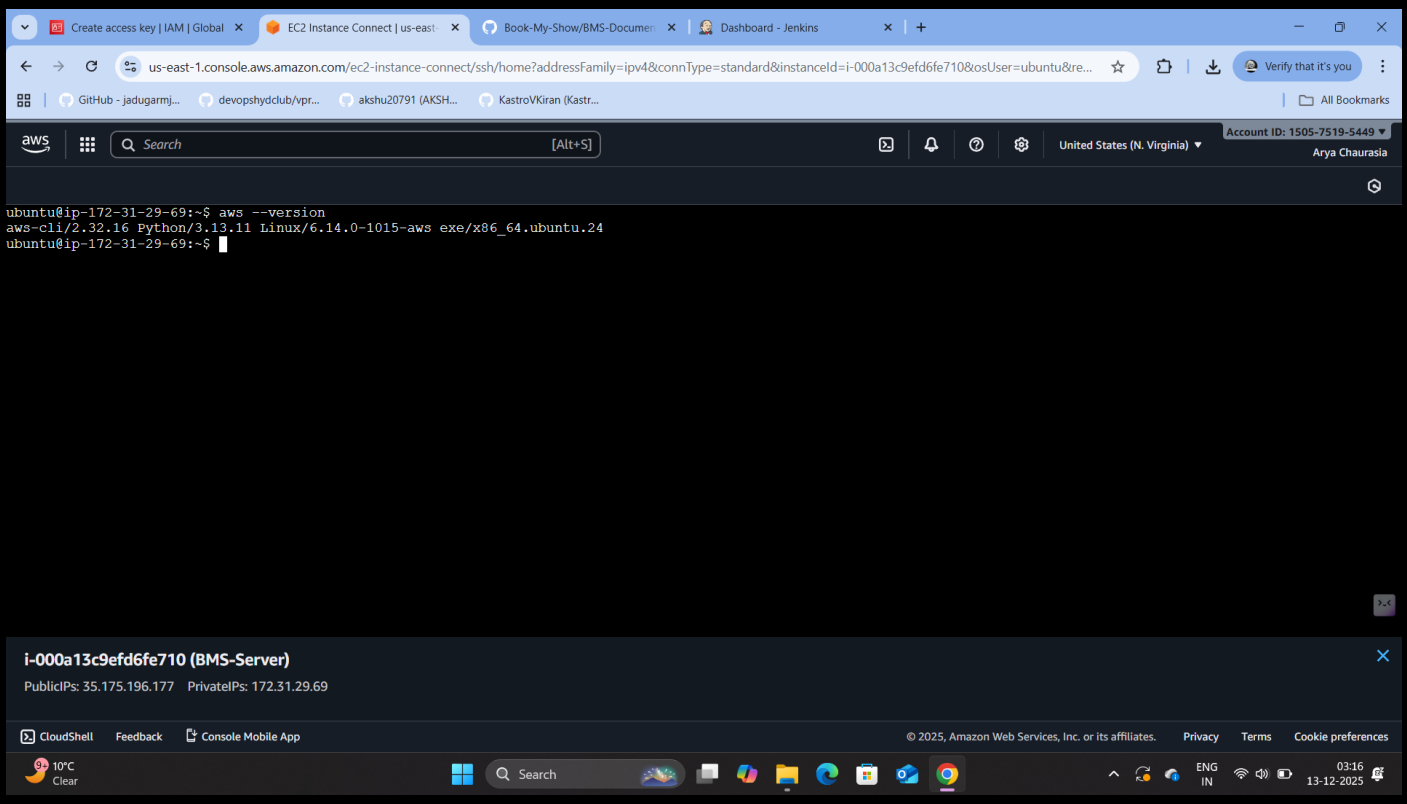
Step-9- We attached total 7 policies.



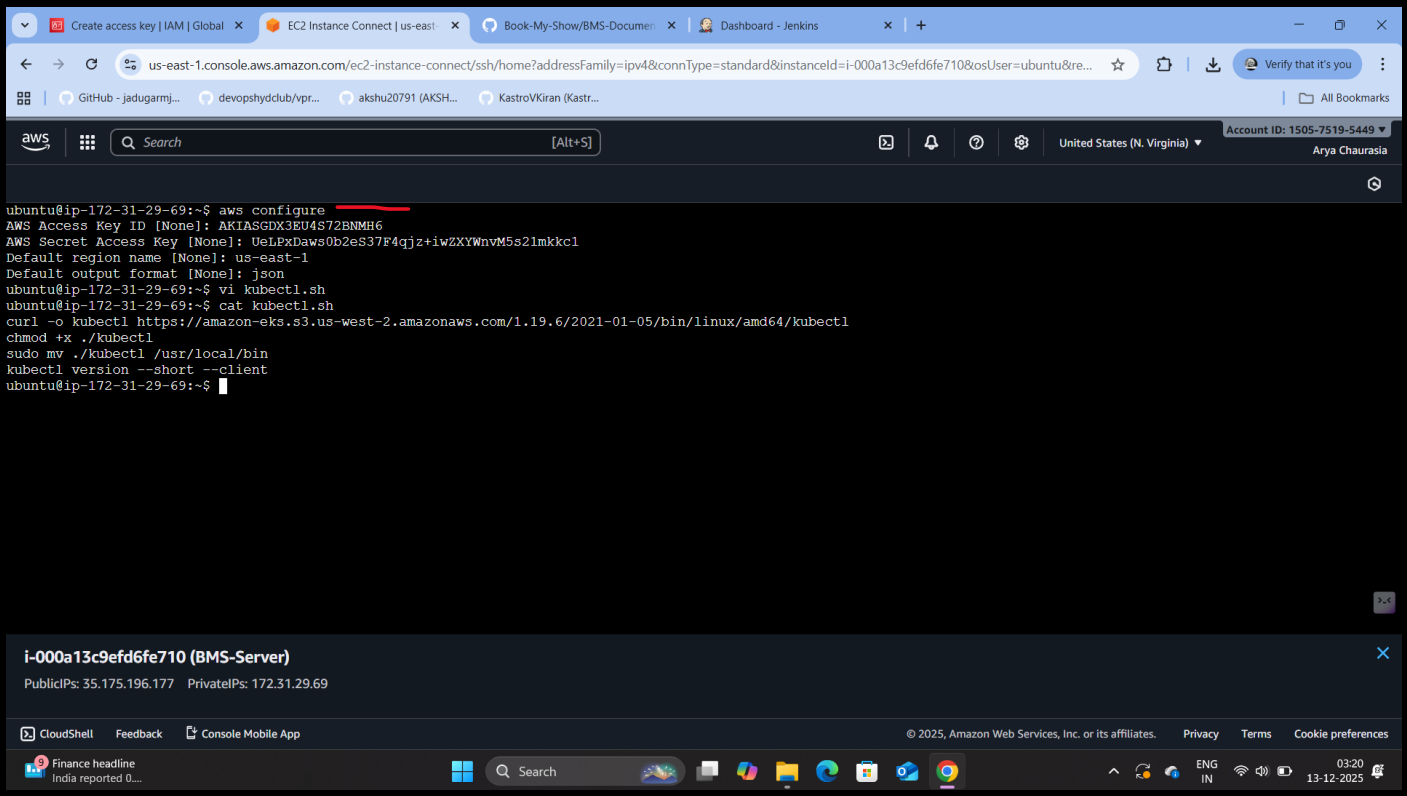
Step-10- Now we generated the access key for the User.



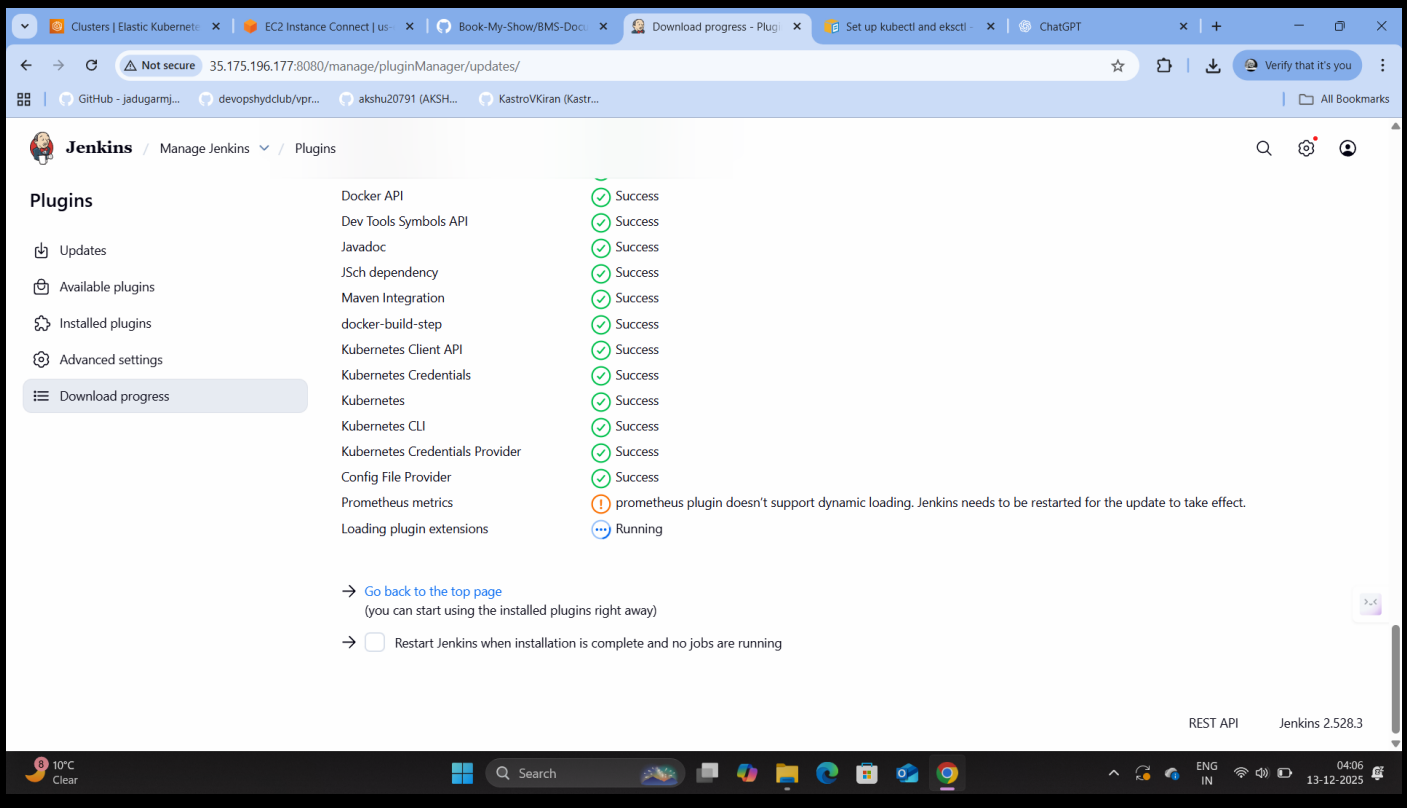
Step-11- Installed AWS CLI to interact with AWS account.



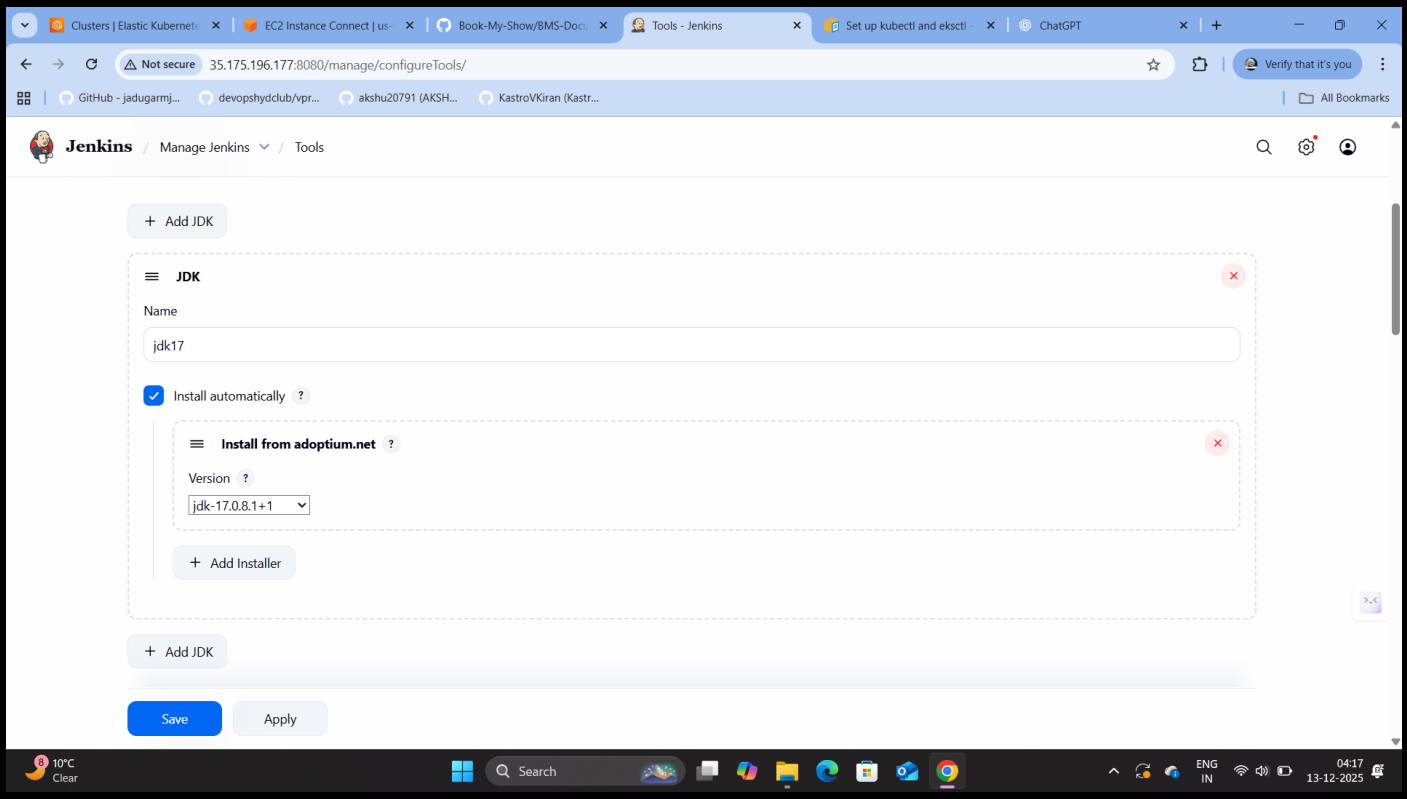
Step-12- Configuring AWS and installing KubeCTL to interact with K8S.



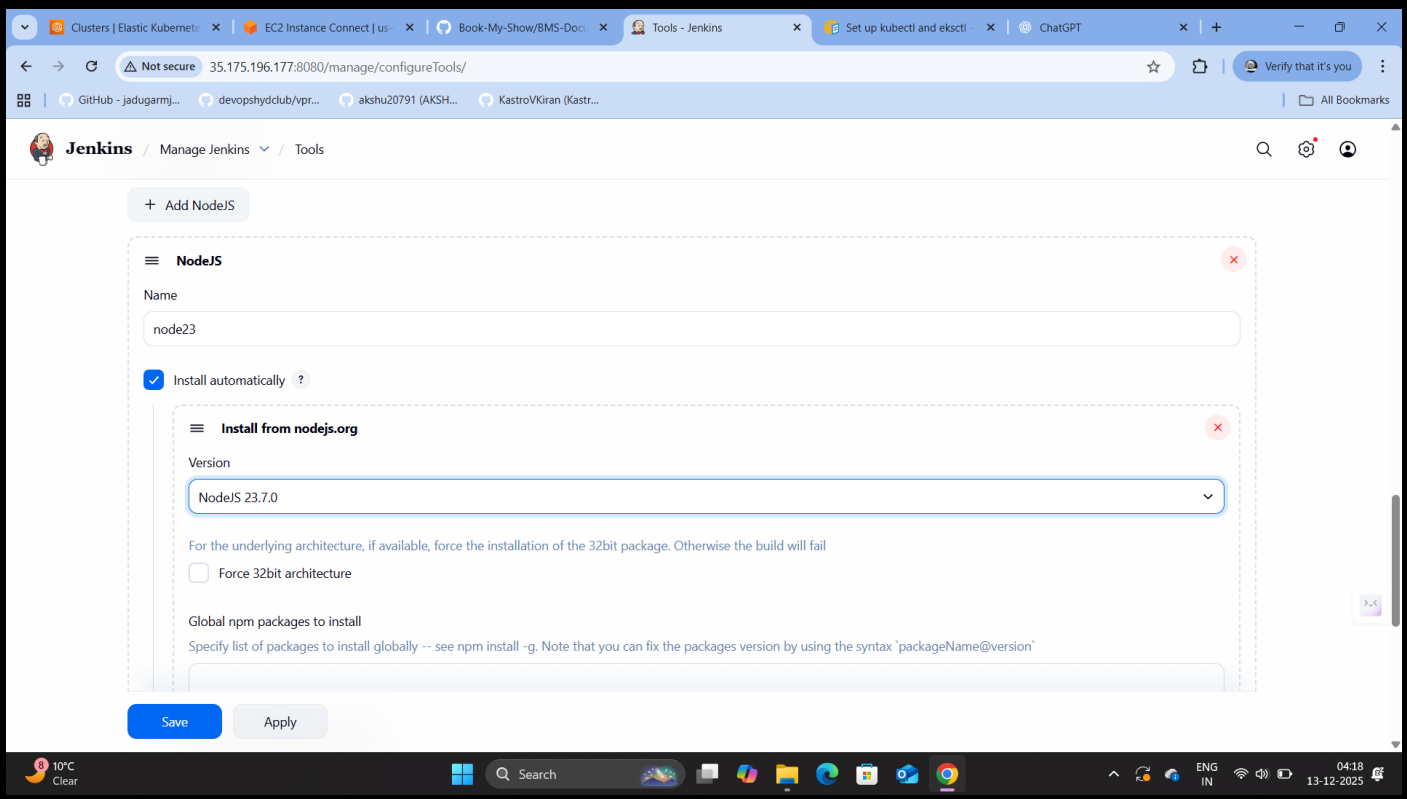
Step-13- Installed plugins in Jenkins – Eclipse Temurian Installer, NodeJS, Docker, Docker commons, Docker pipeline, Docker API, docker-build-step, Pipeline stage view, K8S, K8S CLI, K8S Client API, K8S Credentials, Config file provider, Prometheus metrics.



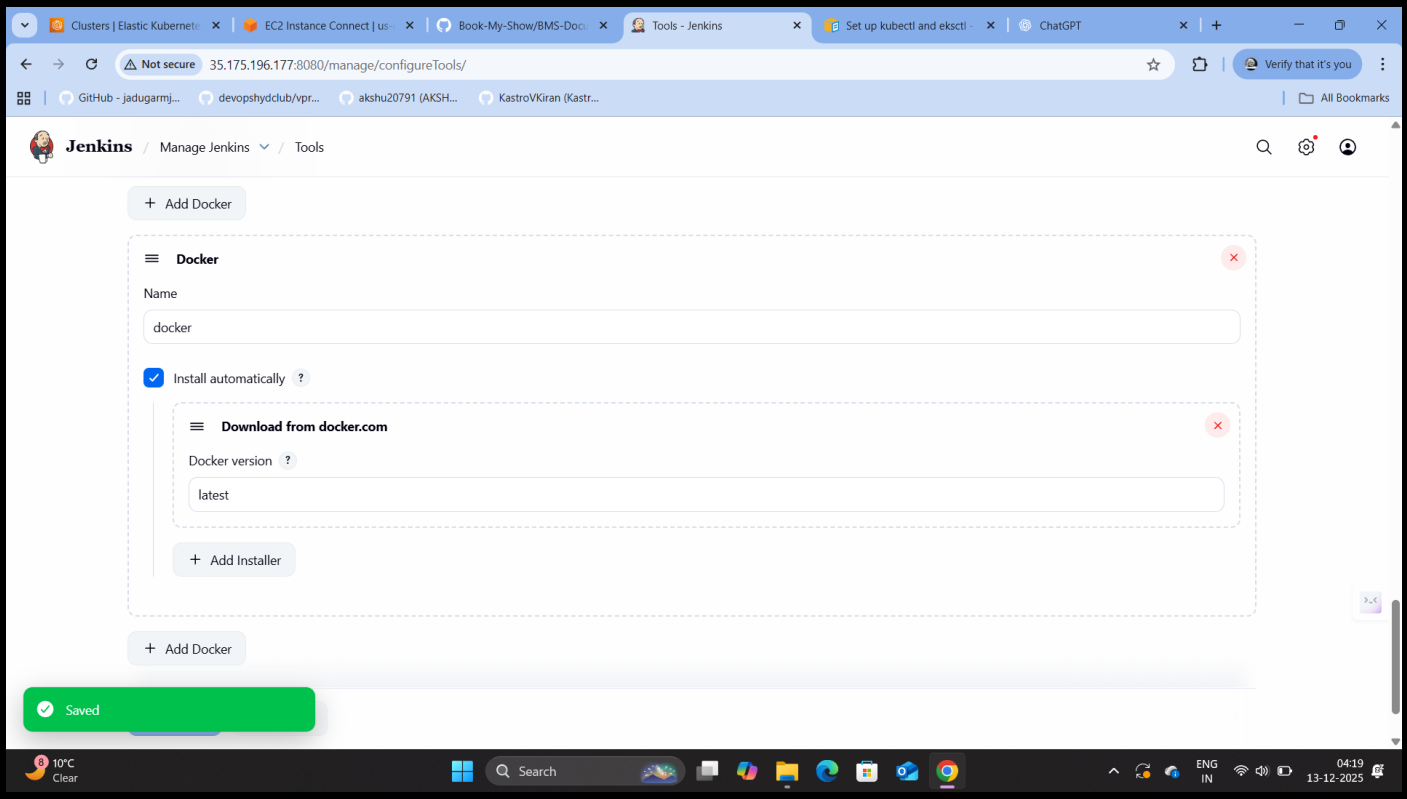
Step-14- Managed tools for jdk



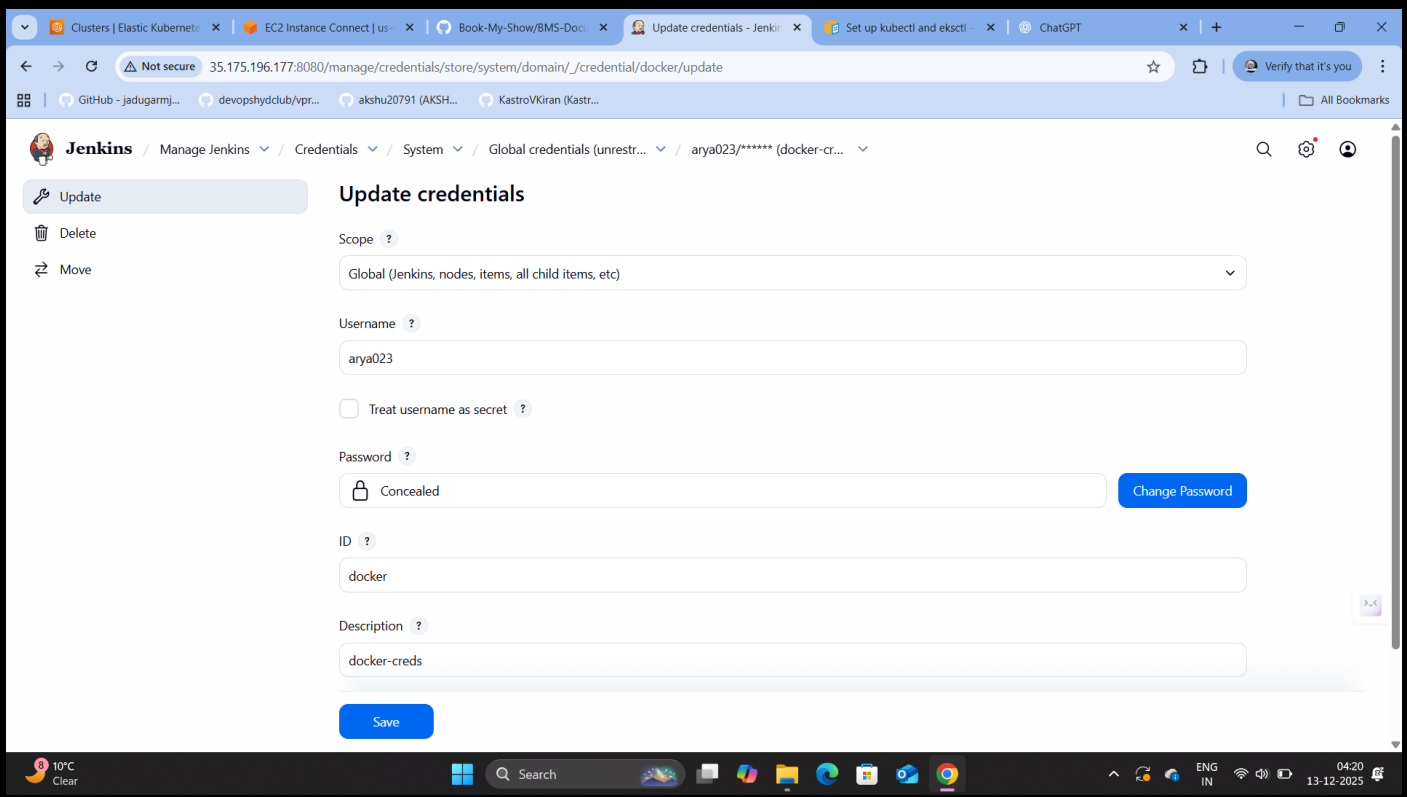
Step-15 Managed tool for NodeJS



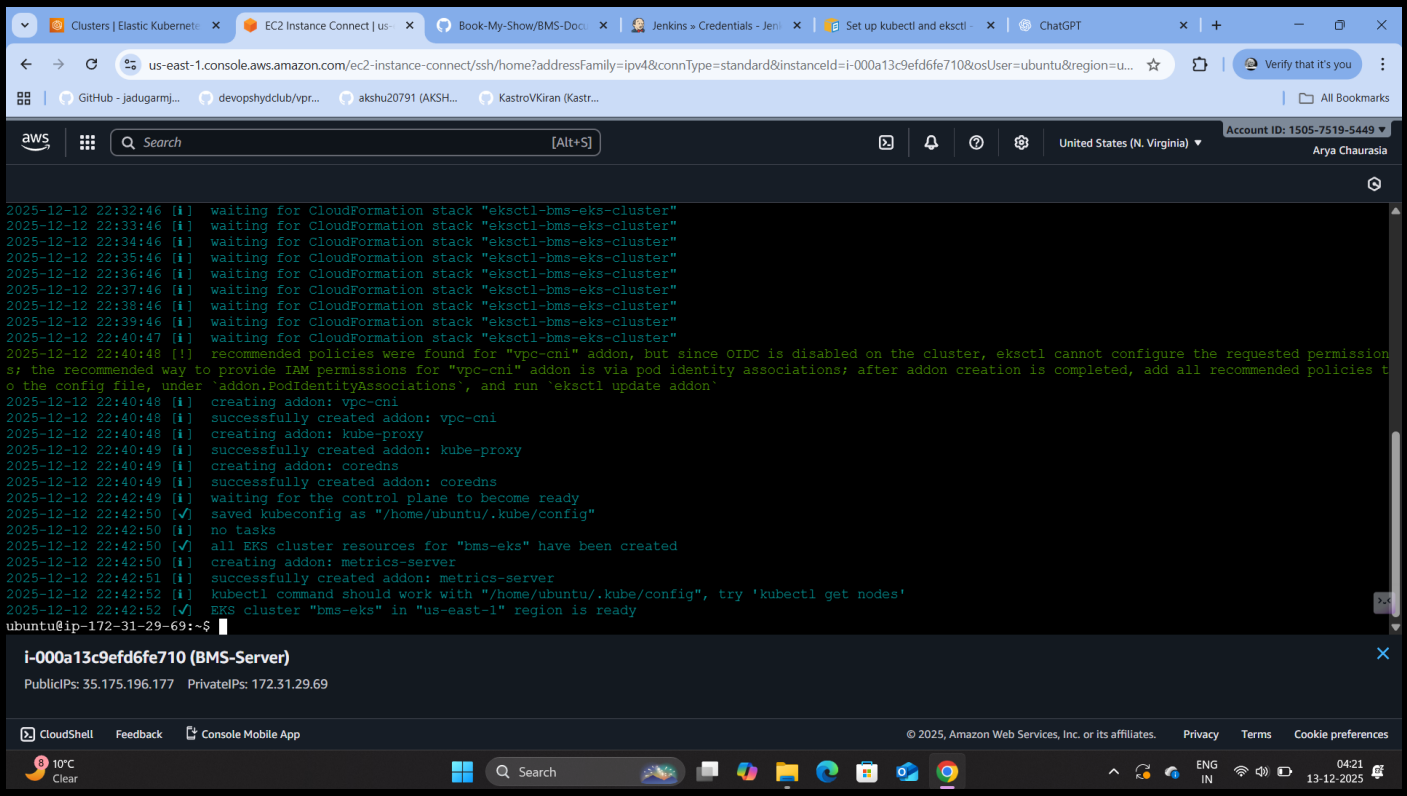
Step-16- Managed tool for docker and clicked Save.



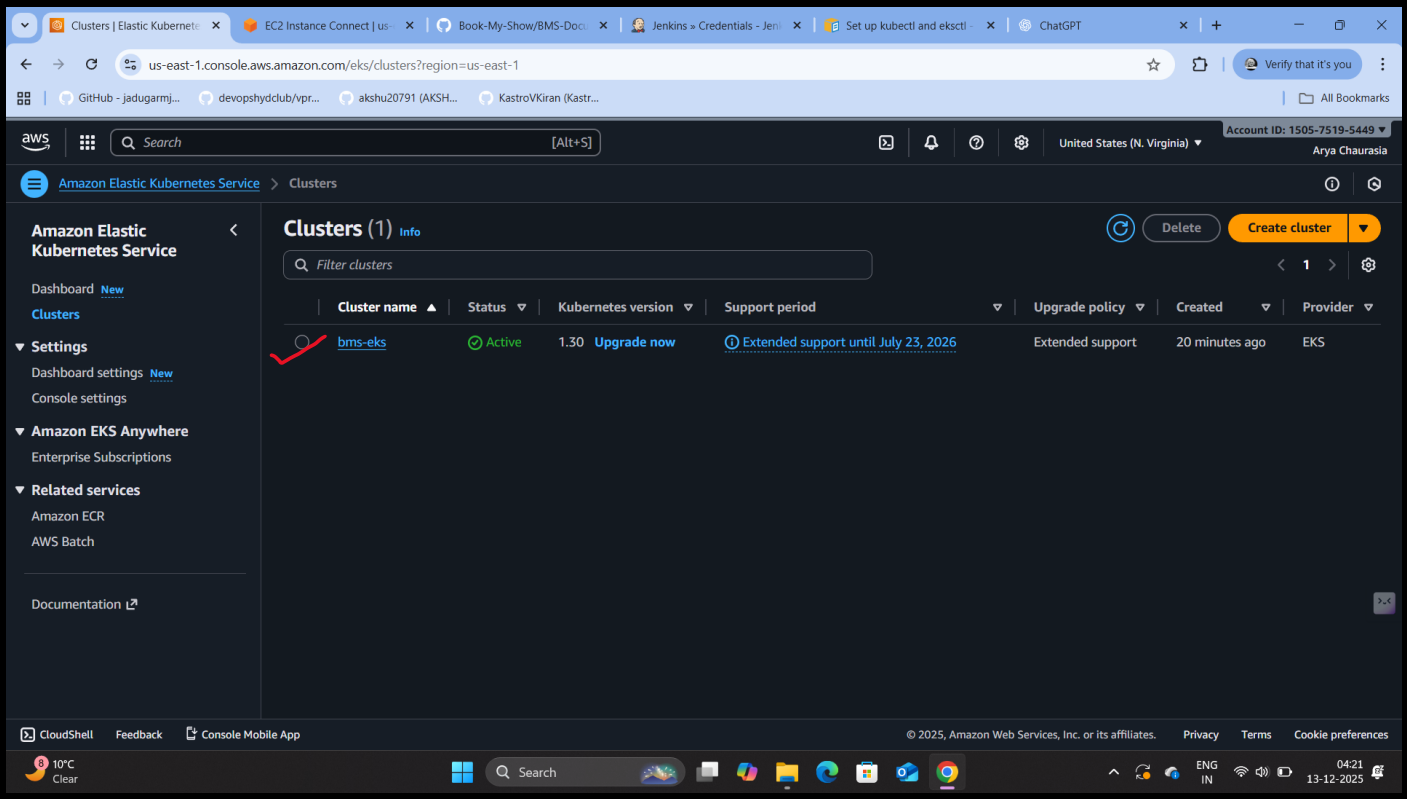
Step-17- We gave credentials for docker to Jenkins



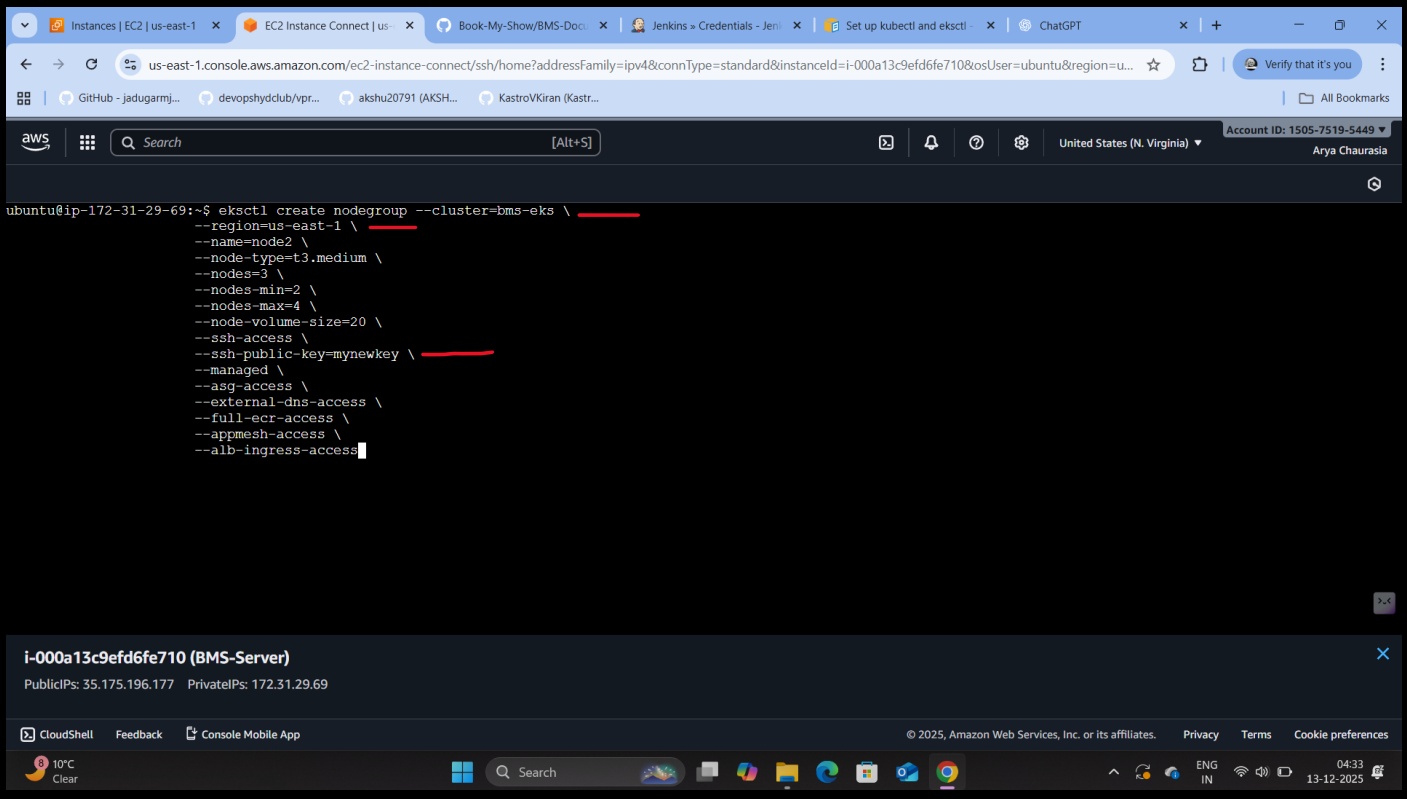
Step-18- Here we created the EKS Cluster and it had taken sometime.



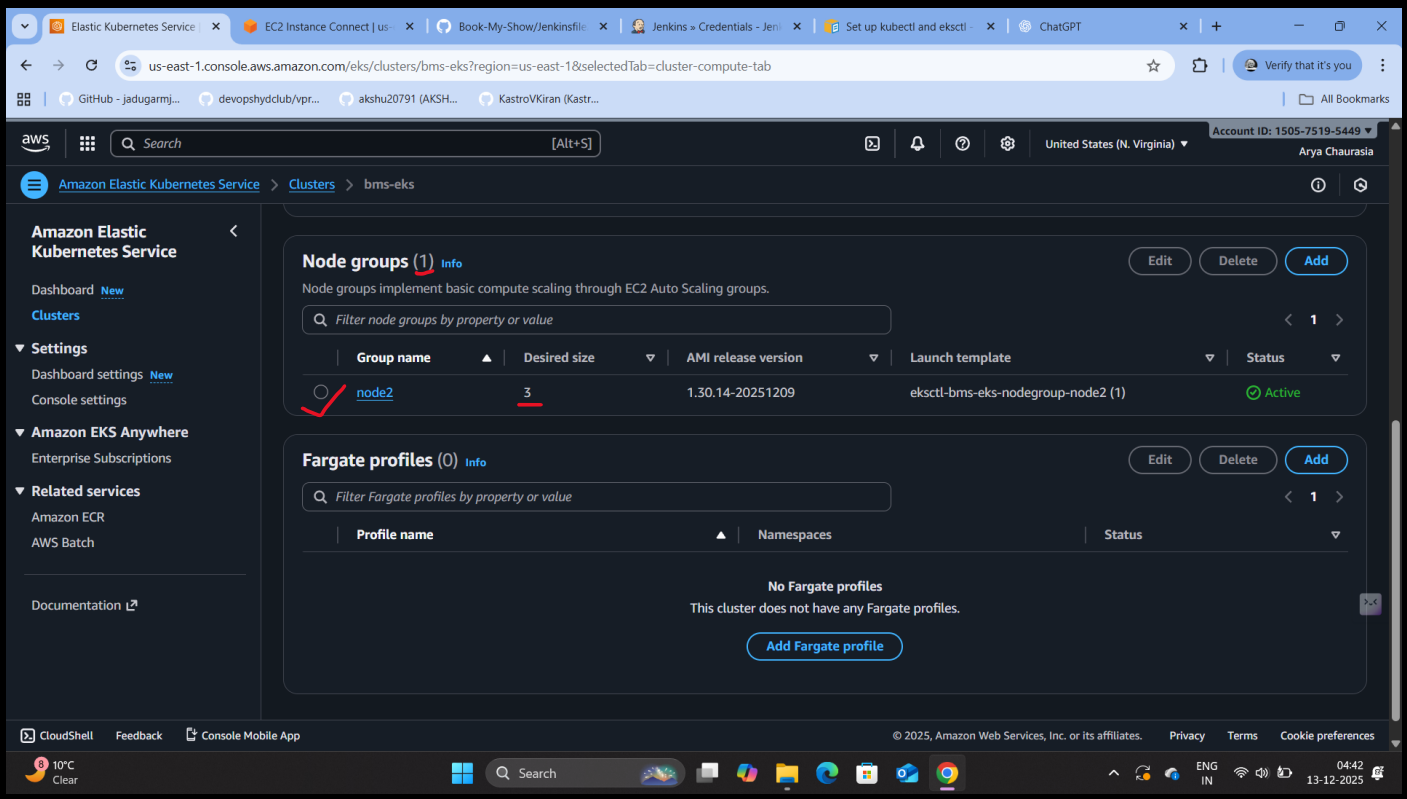
Step-19- After sometime, we verified that our cluster was made.



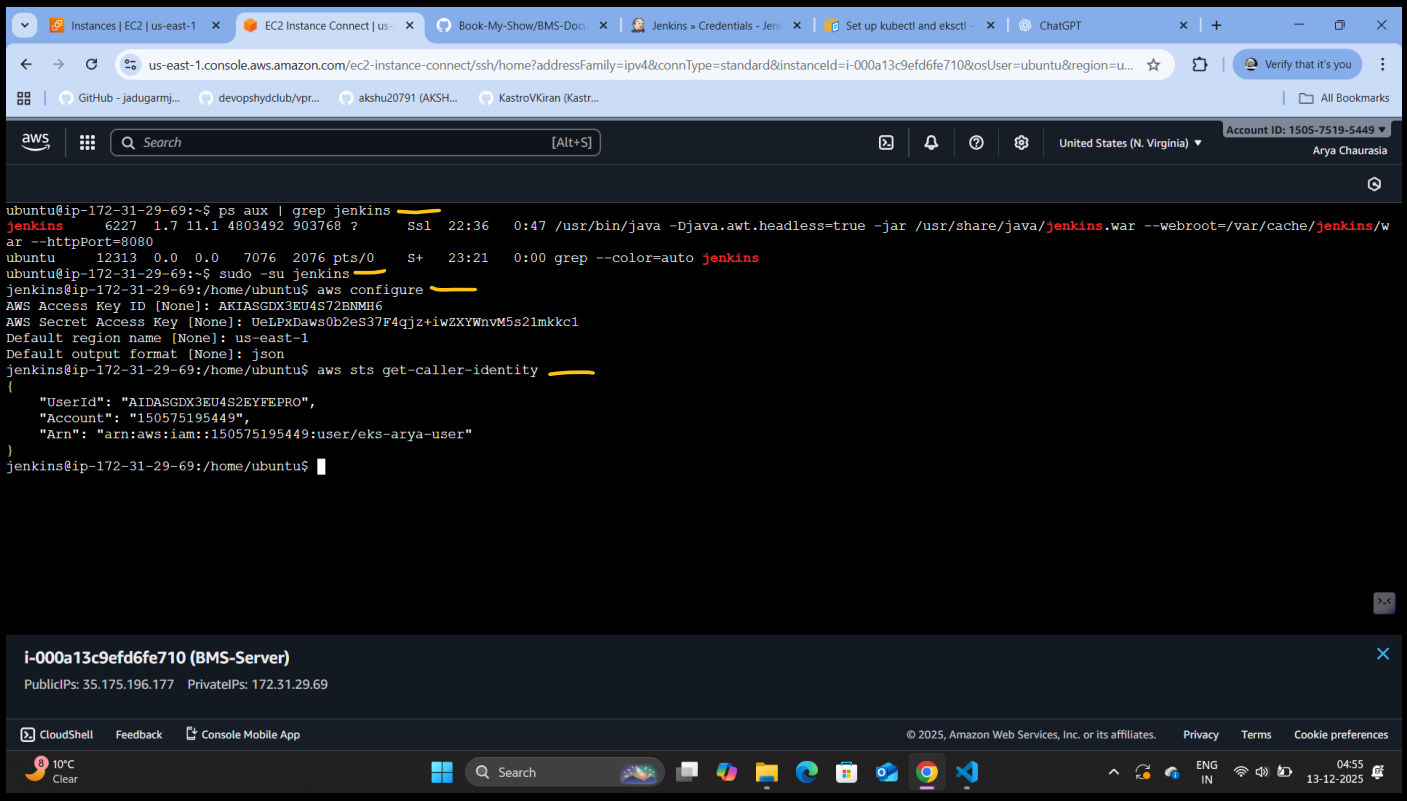
Step-20- Here we created nodes for our EKS Cluster. Note – in ssh-public-key, give only key name not .pem



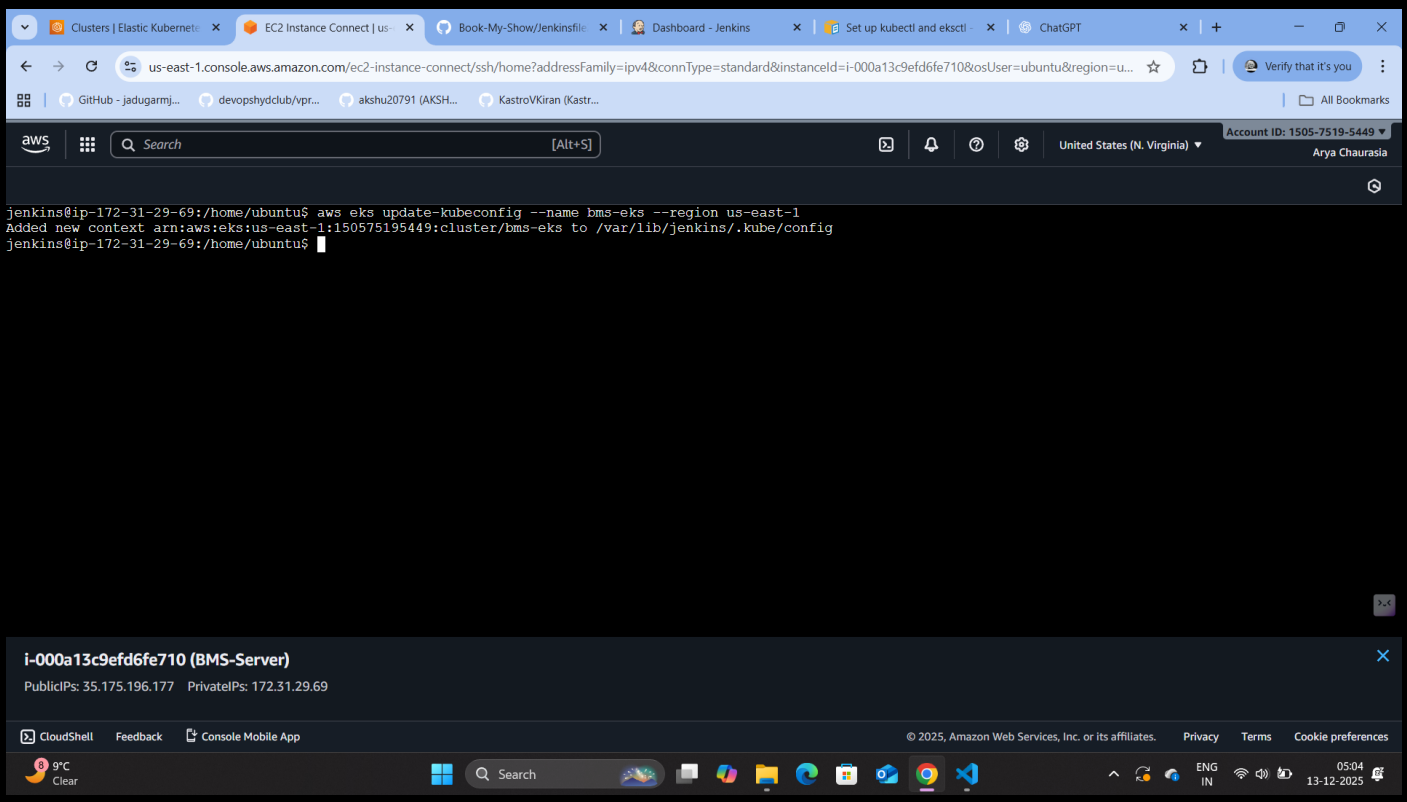
Step-21- It took 7-10 minutes and after that our nodes were created.



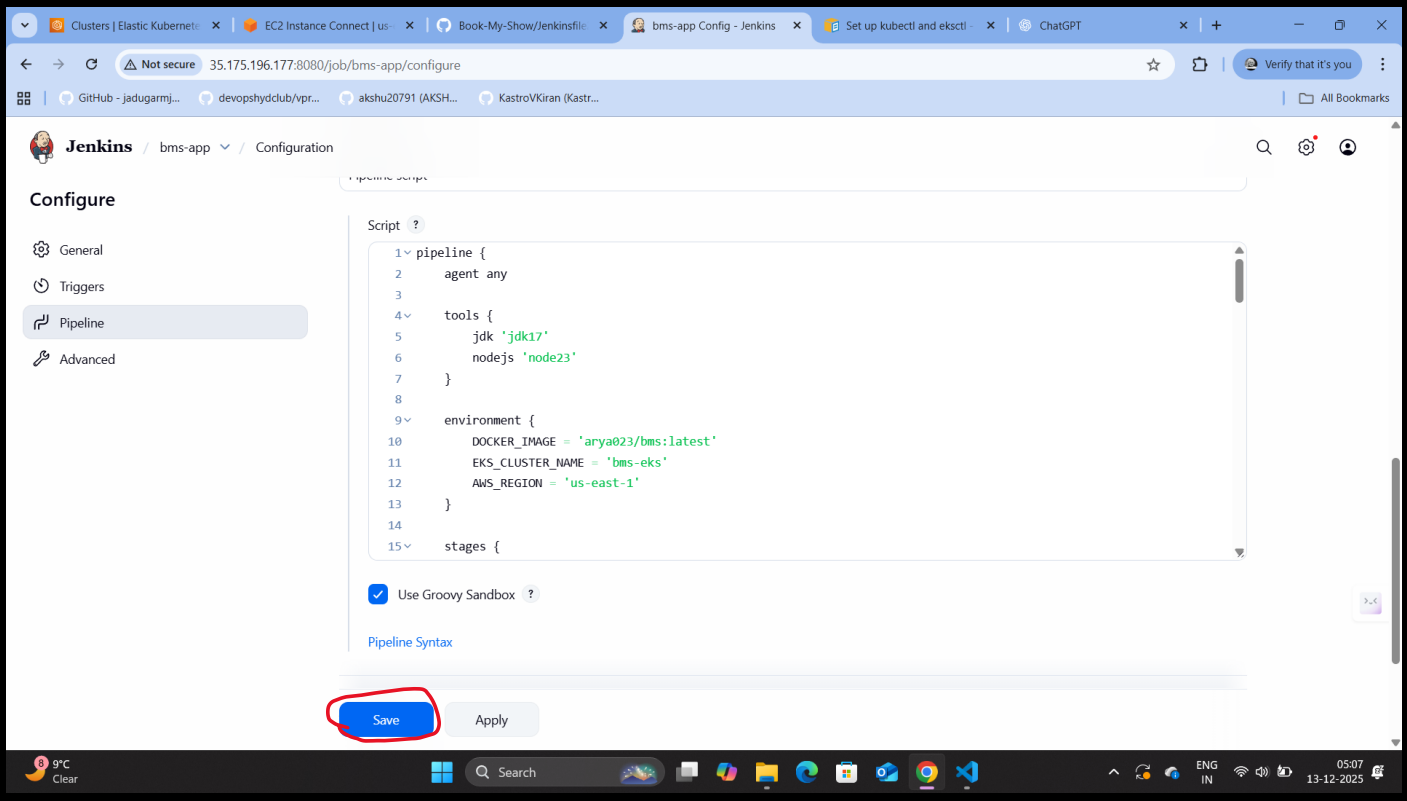
Step-22-To Know Jenkins is running on which user , if its Jenkins then switch to jenkins user ad verify the credentials.



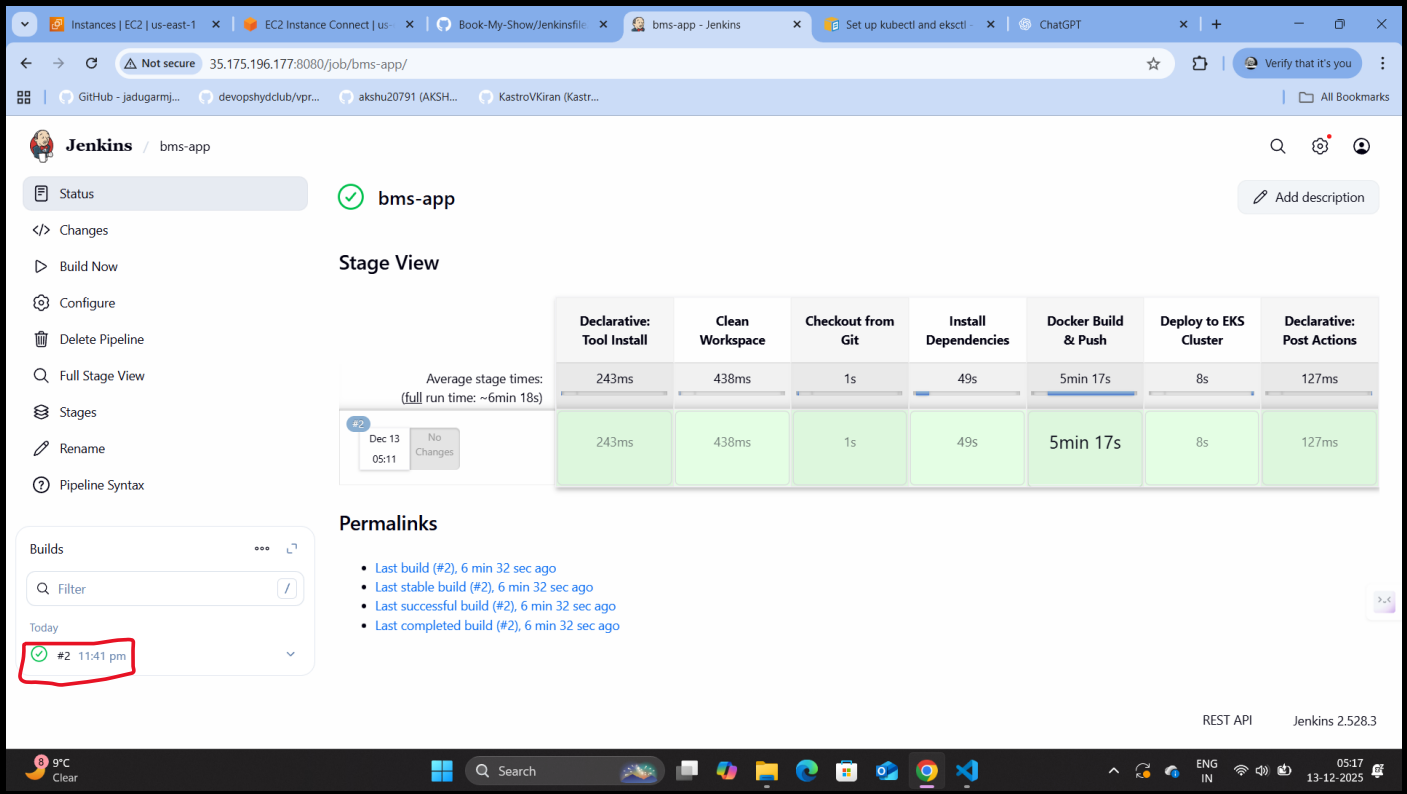
Step-23-Here we update the kubeconfig file



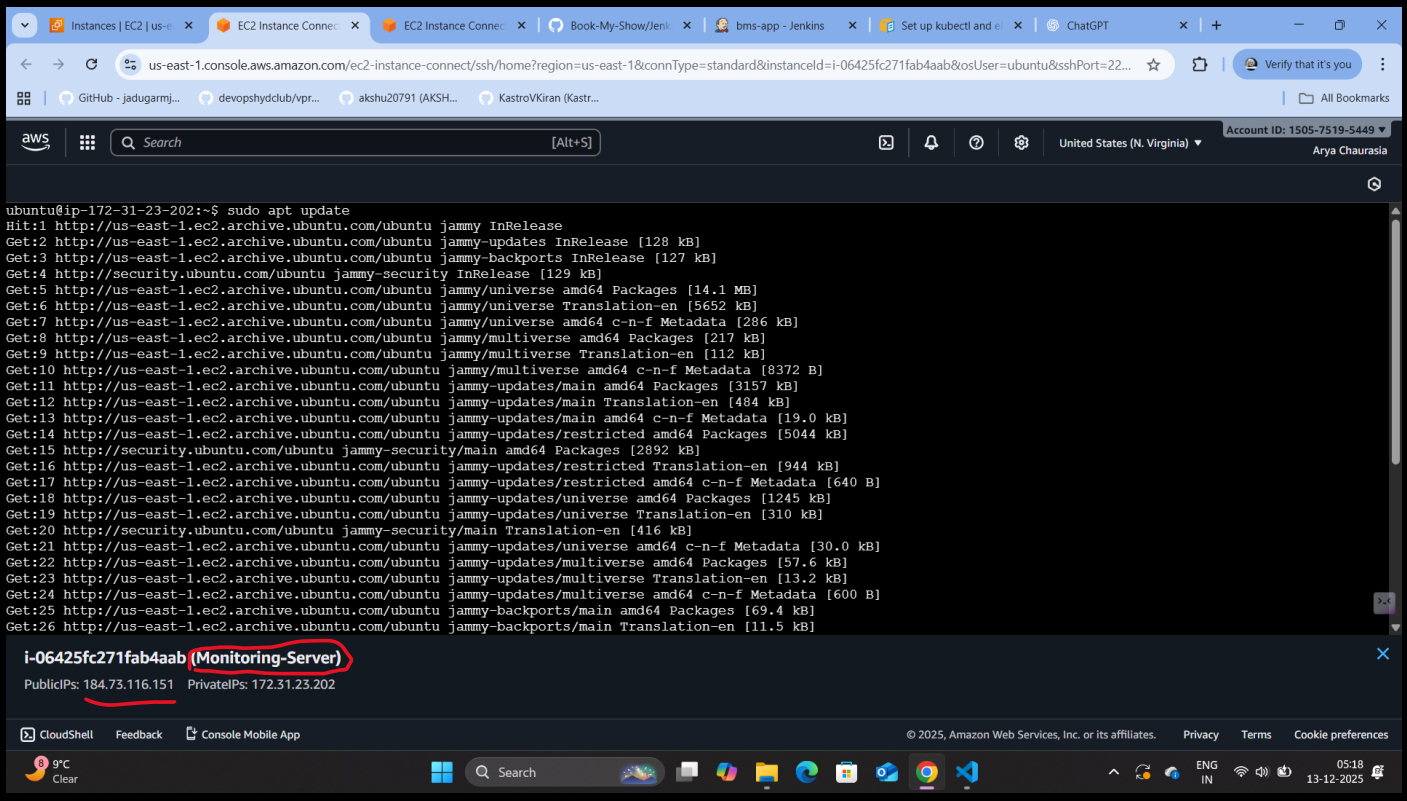
Step-24- Now we will create a job named – bms-app and in that we will run the pipeline.



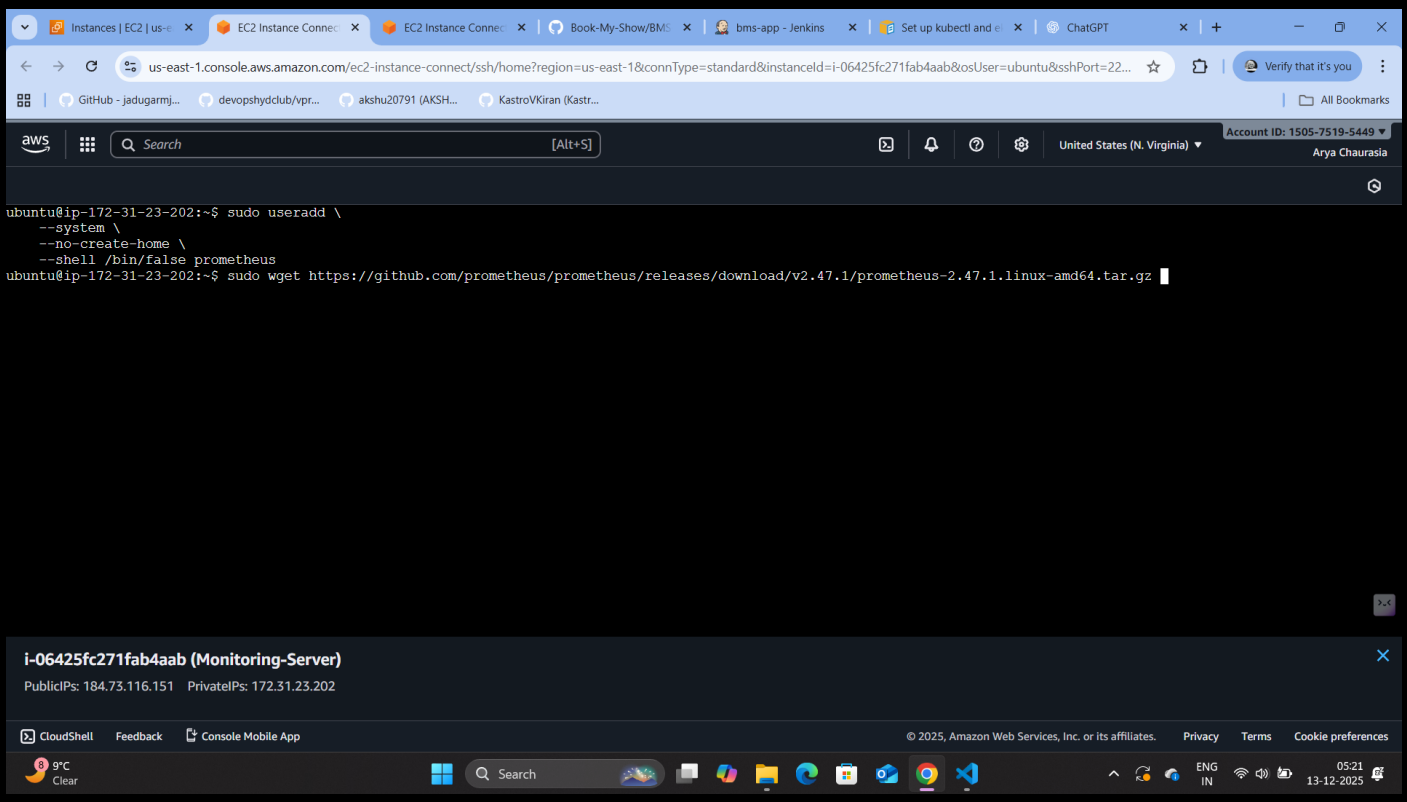
Step-25- The build is successful for the bms-app



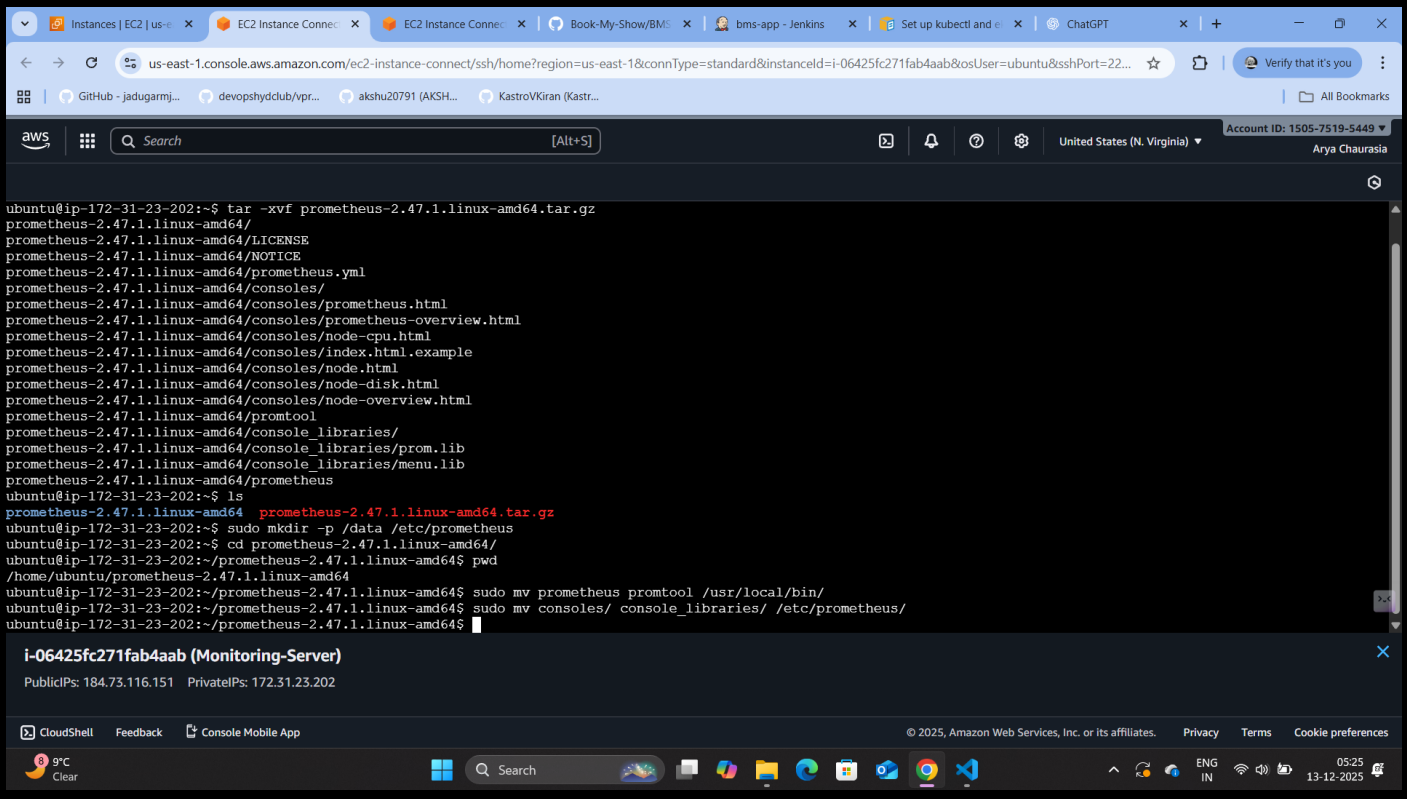
Step-26- Now we will make another instance for Prometheus which is Monitoring server.



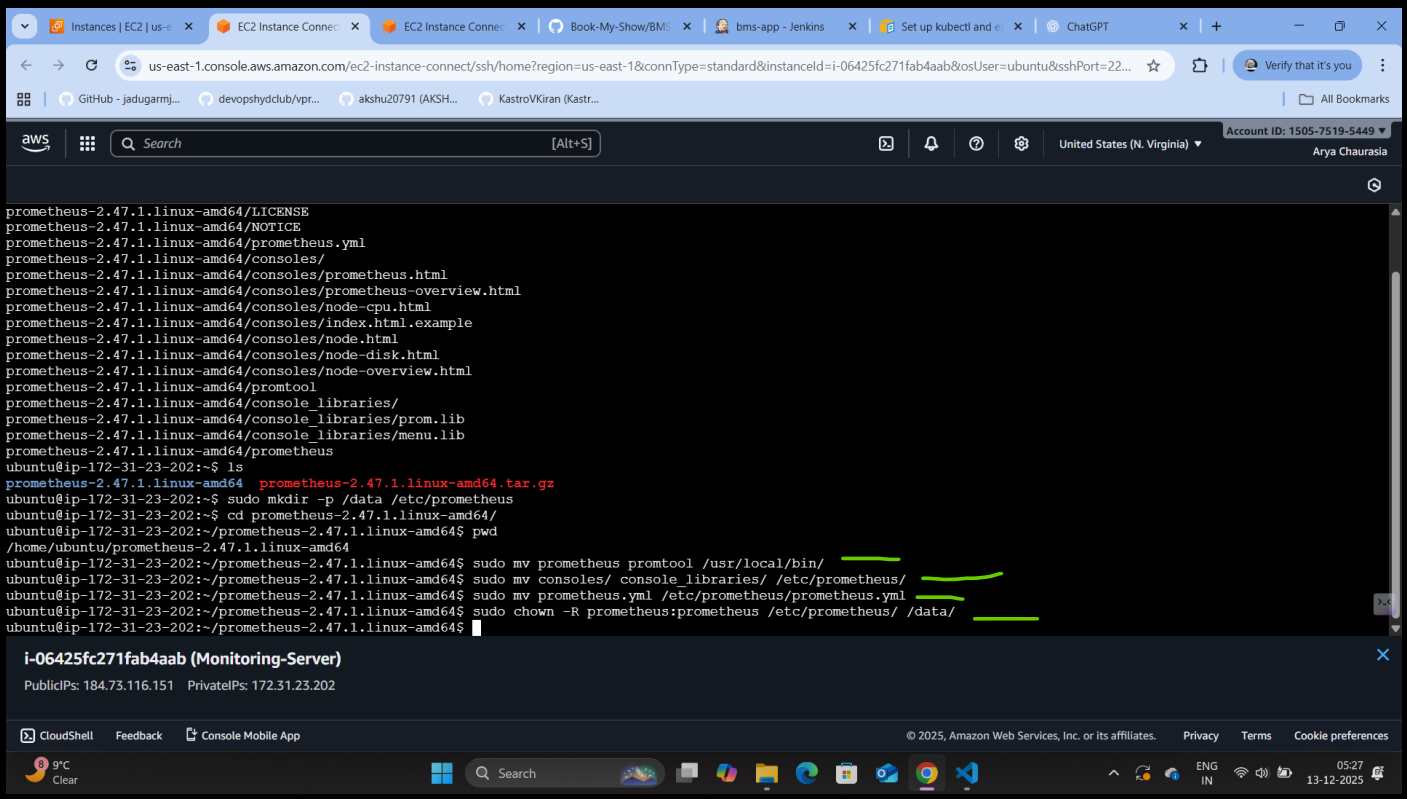
Step-27- With this command we created the Prometheus user.



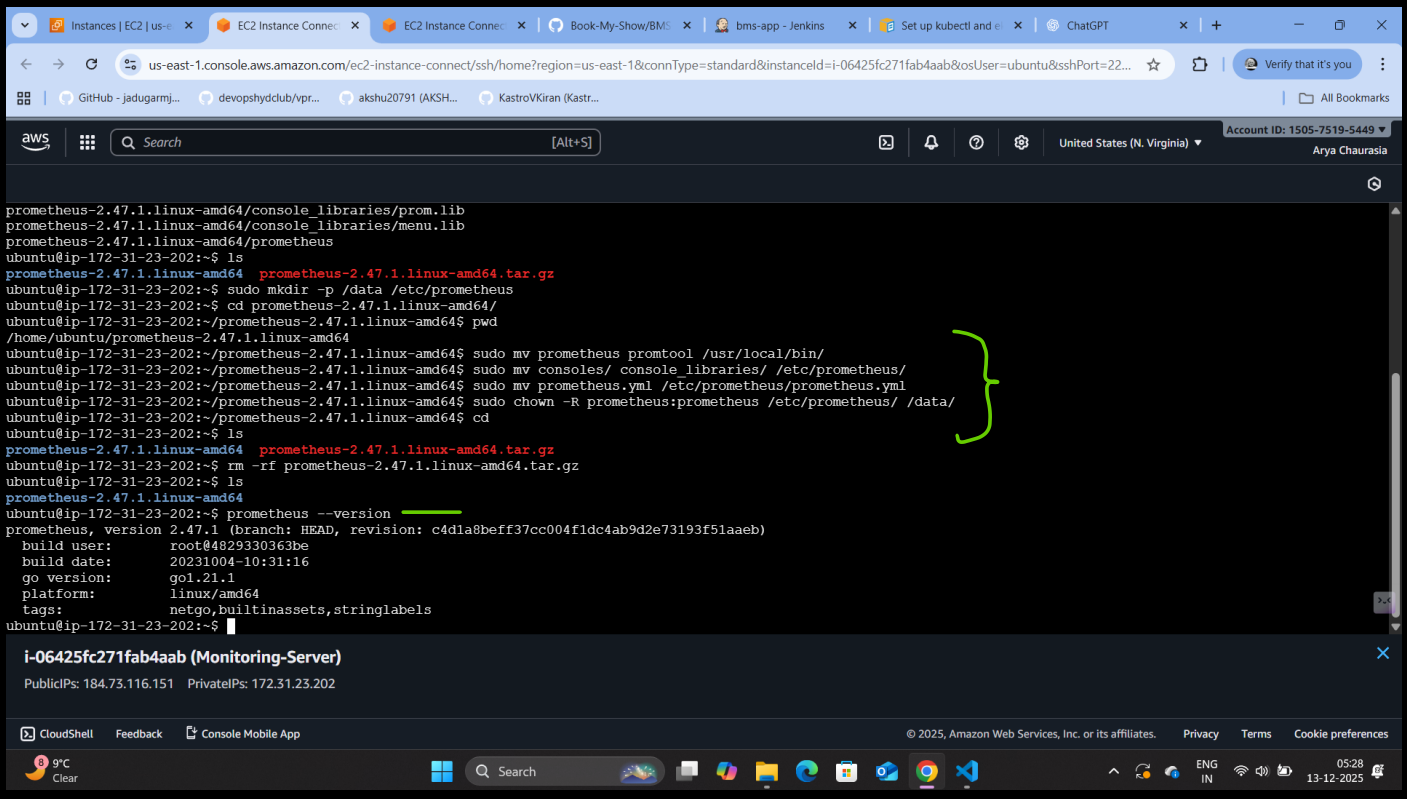
Step-28- Downloading the Prometheus.



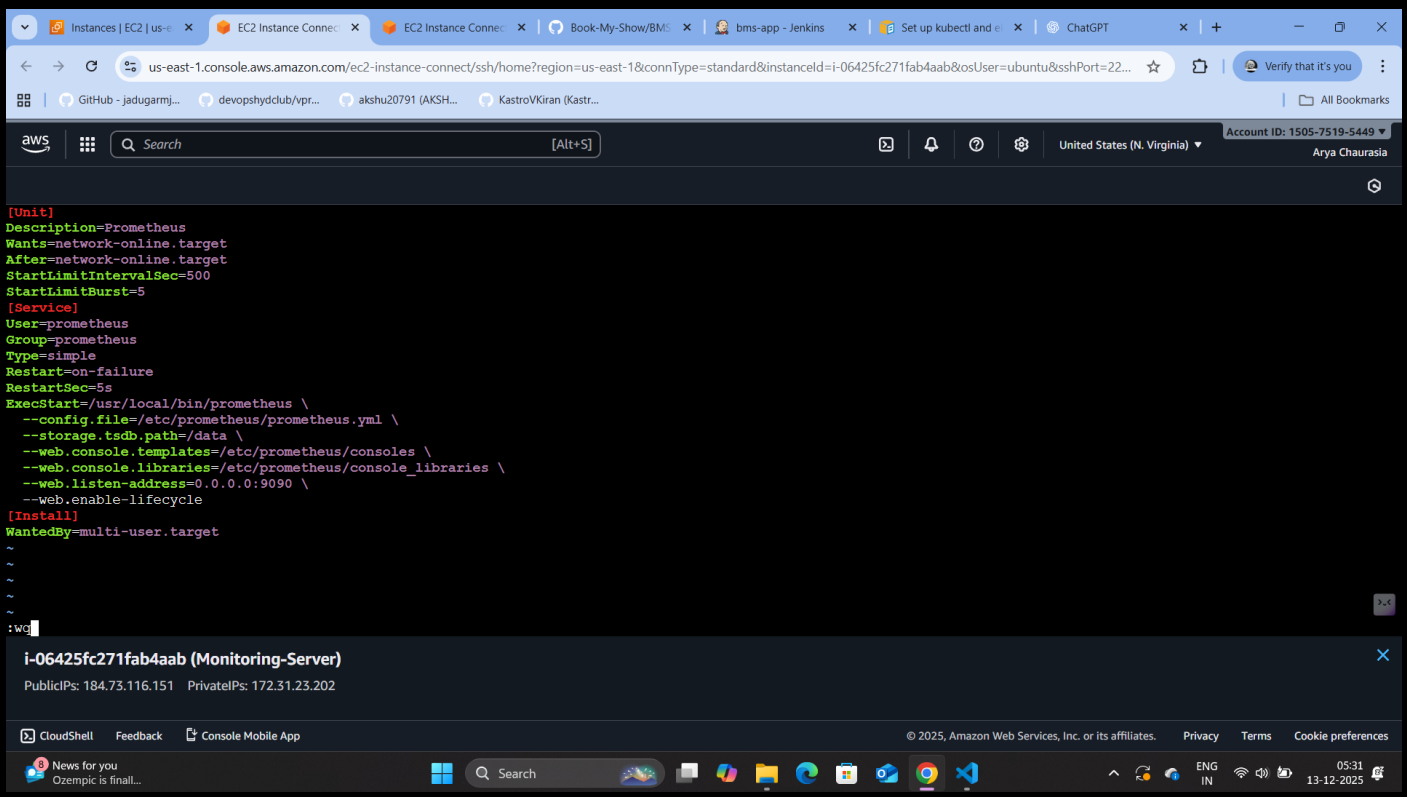
Step-29-Move theprometheus binary and a promtool to the /usr/local/bin/.



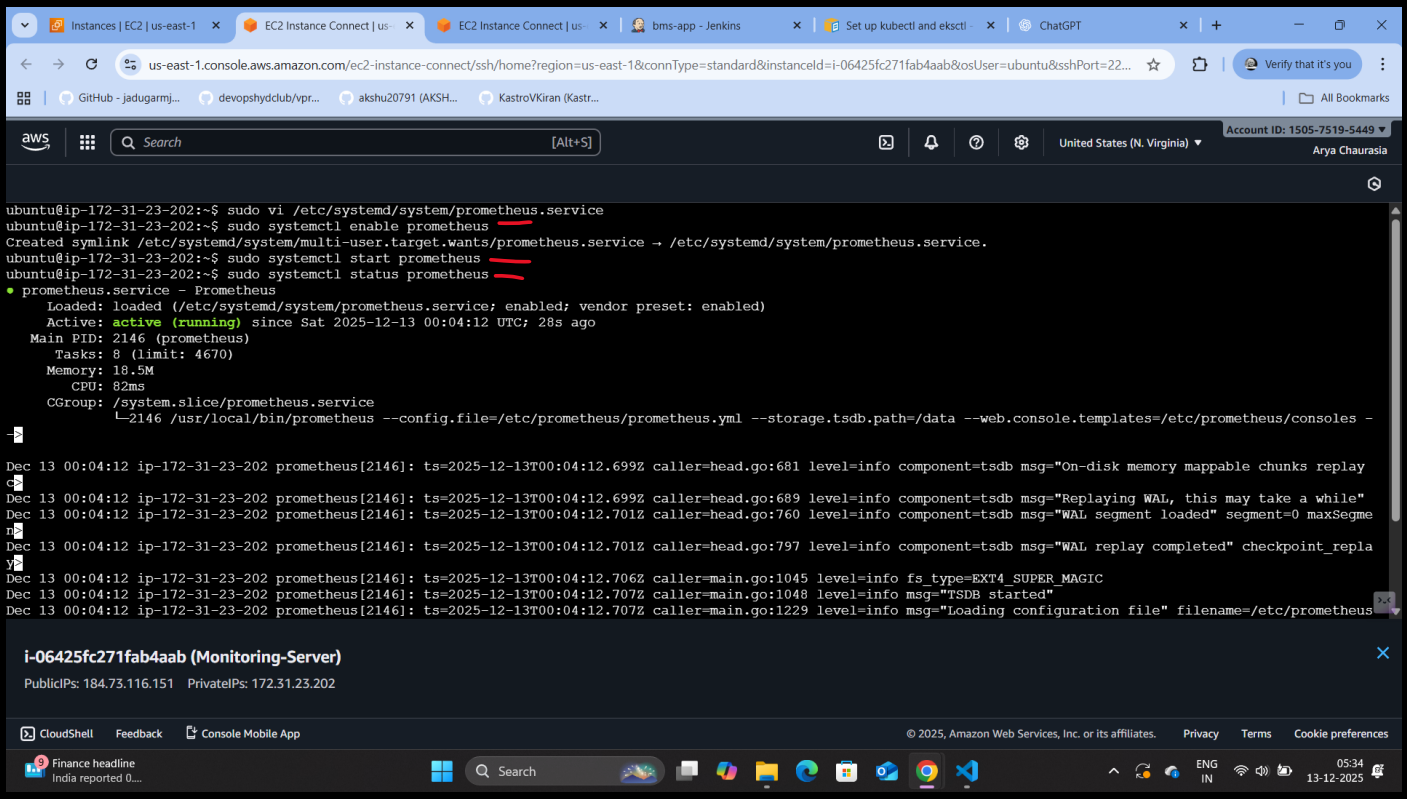
Step-30- Set the correct ownership for the /etc/prometheus/ and data directory



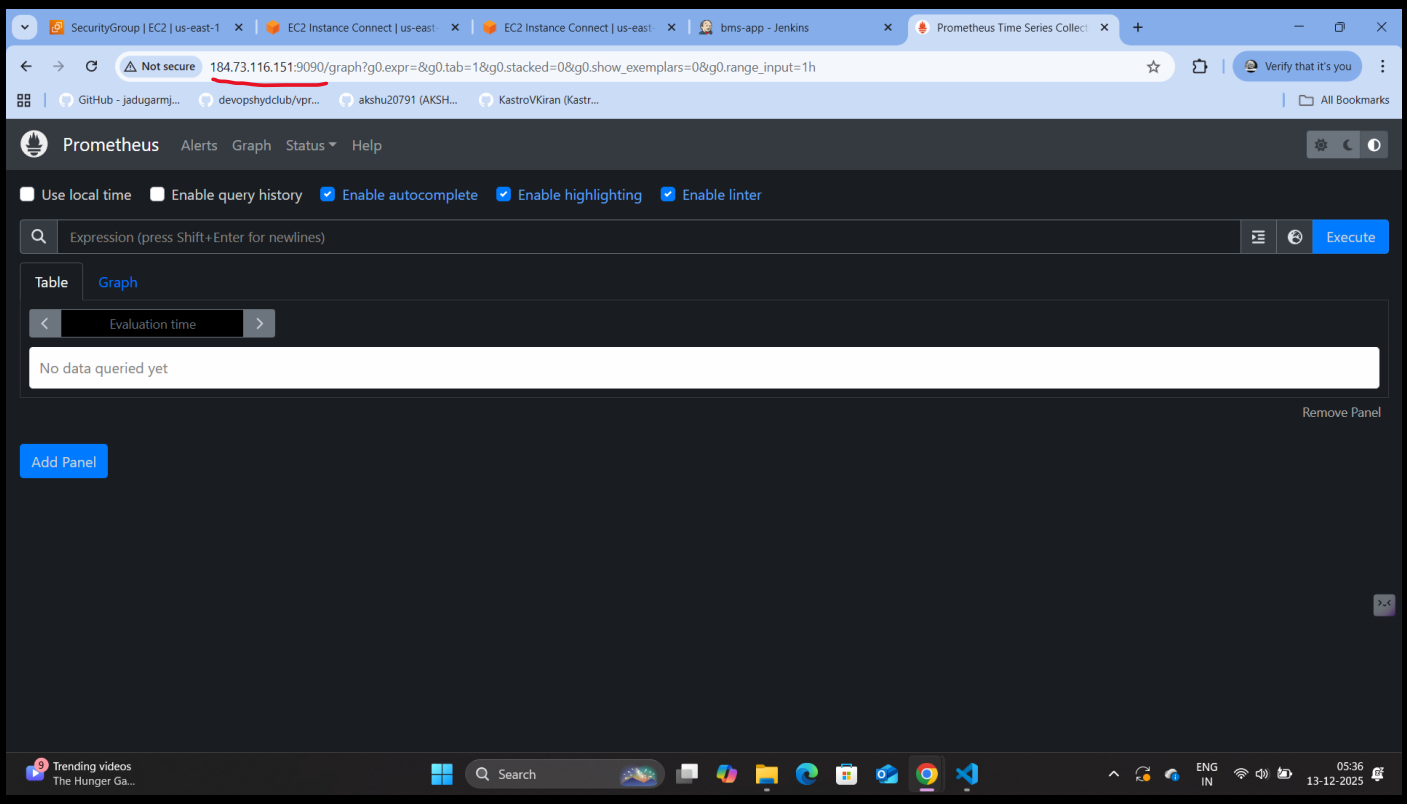
Step-31- We are going to use systemd which is a system and service manager for linux OS, for that we need to create a systemd unit configuration file.



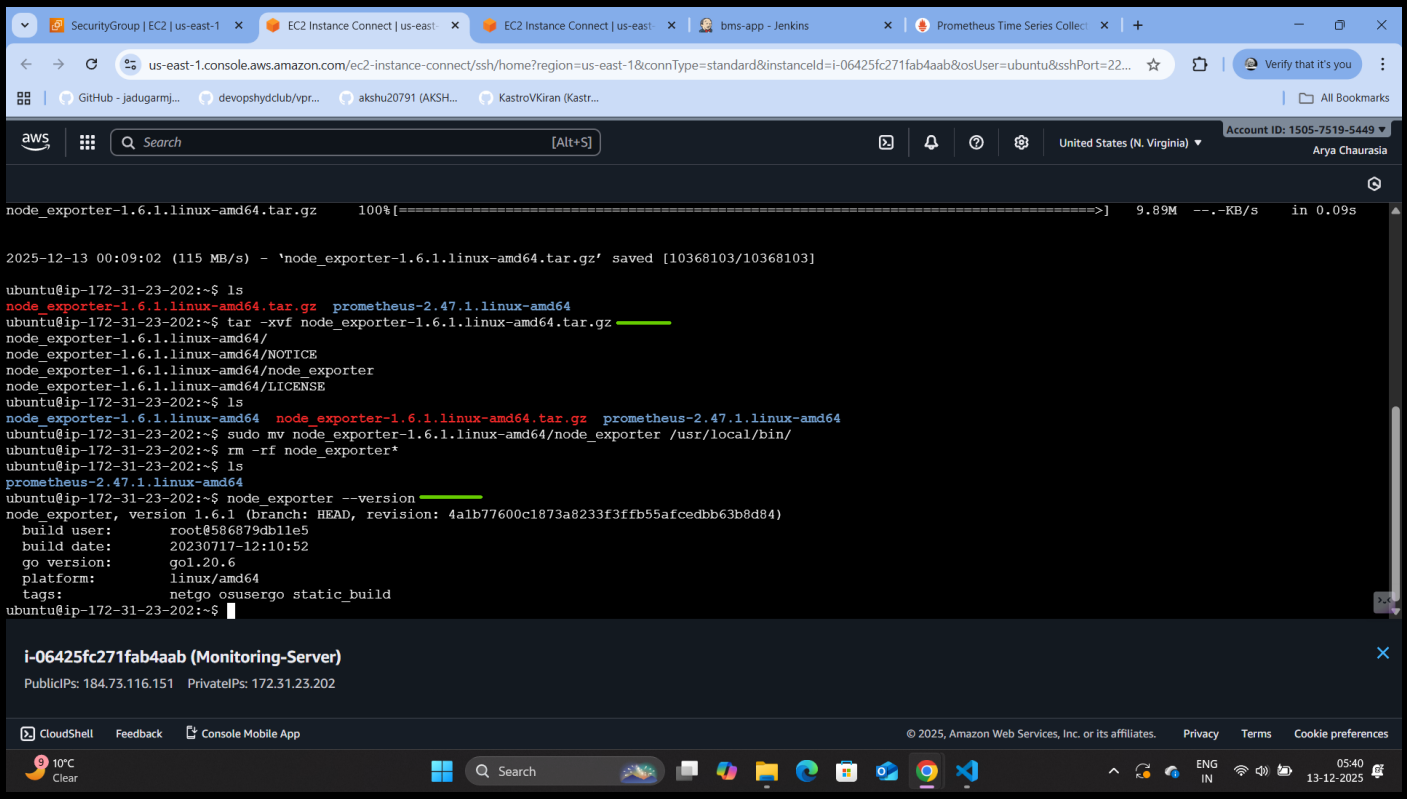
Step-32- Now we will check the status of the prometheus



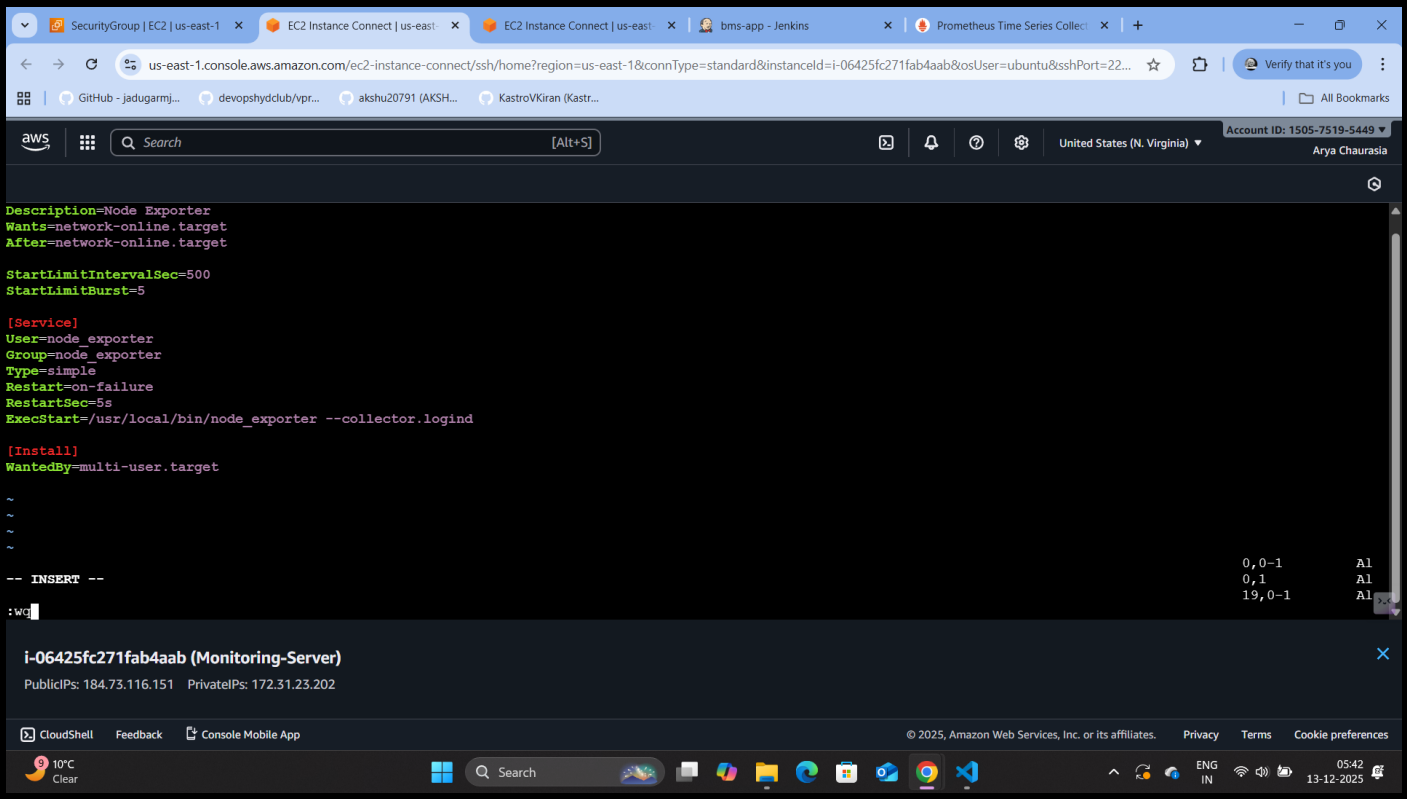
Step-33-<monitoring-server IP> : 9090 to run the prometheus



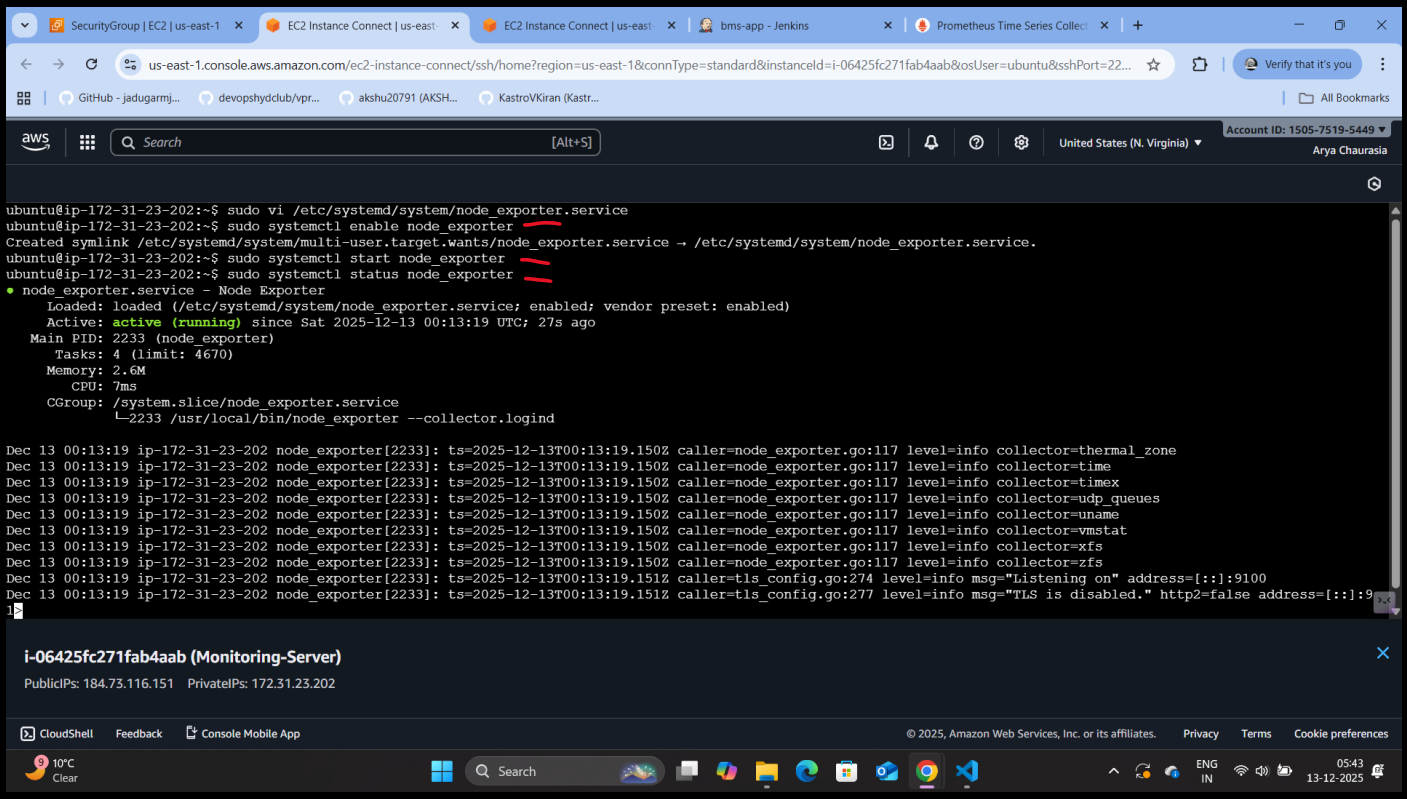
Step-34- Now we will download the node exporter and extract the files.



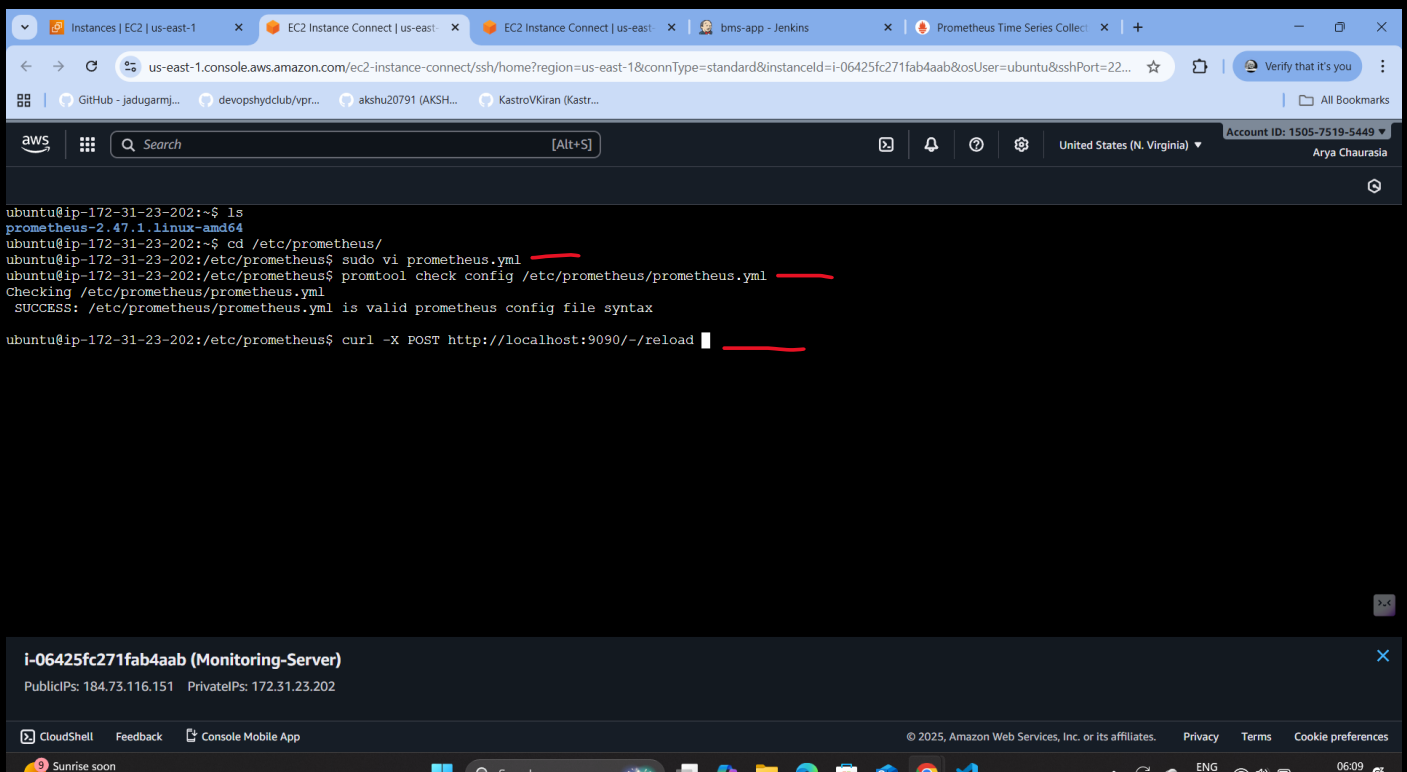
Step-35-Adding the following content to the node_exporter.service file



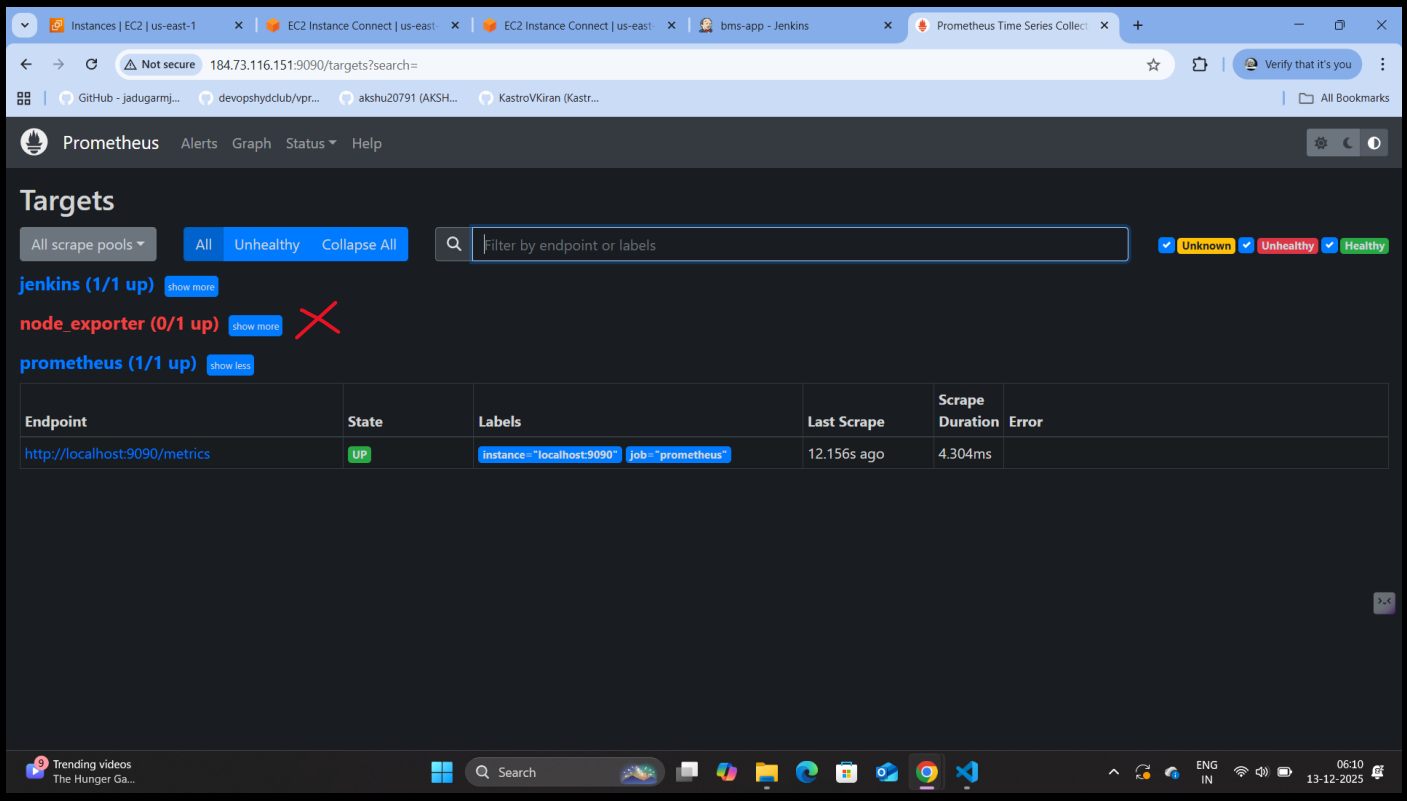
Step-36-Now we will enable and start the node exporter.



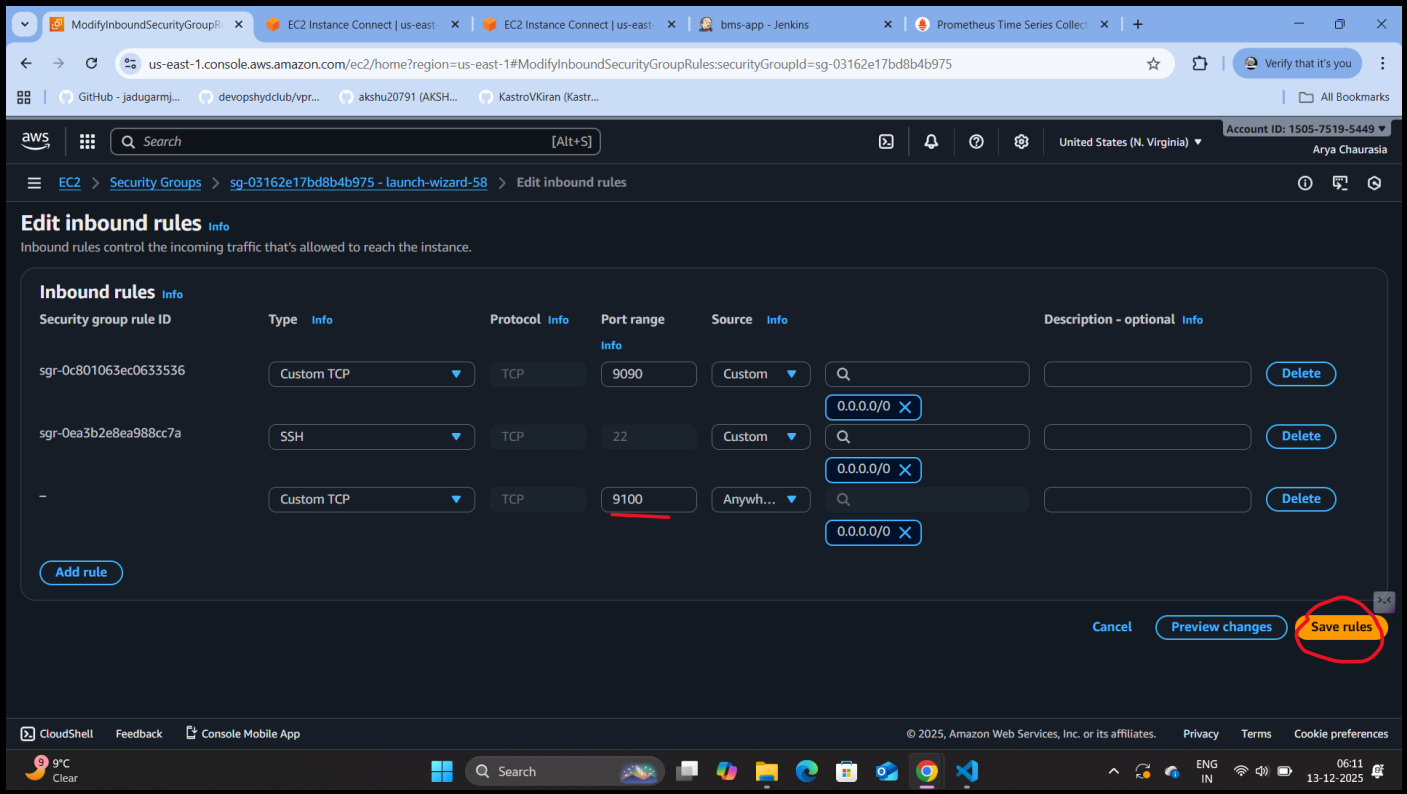
Step-37- To scrape metrics from Node Exporter and Jenkins, we need to modify the prometheus.yml file



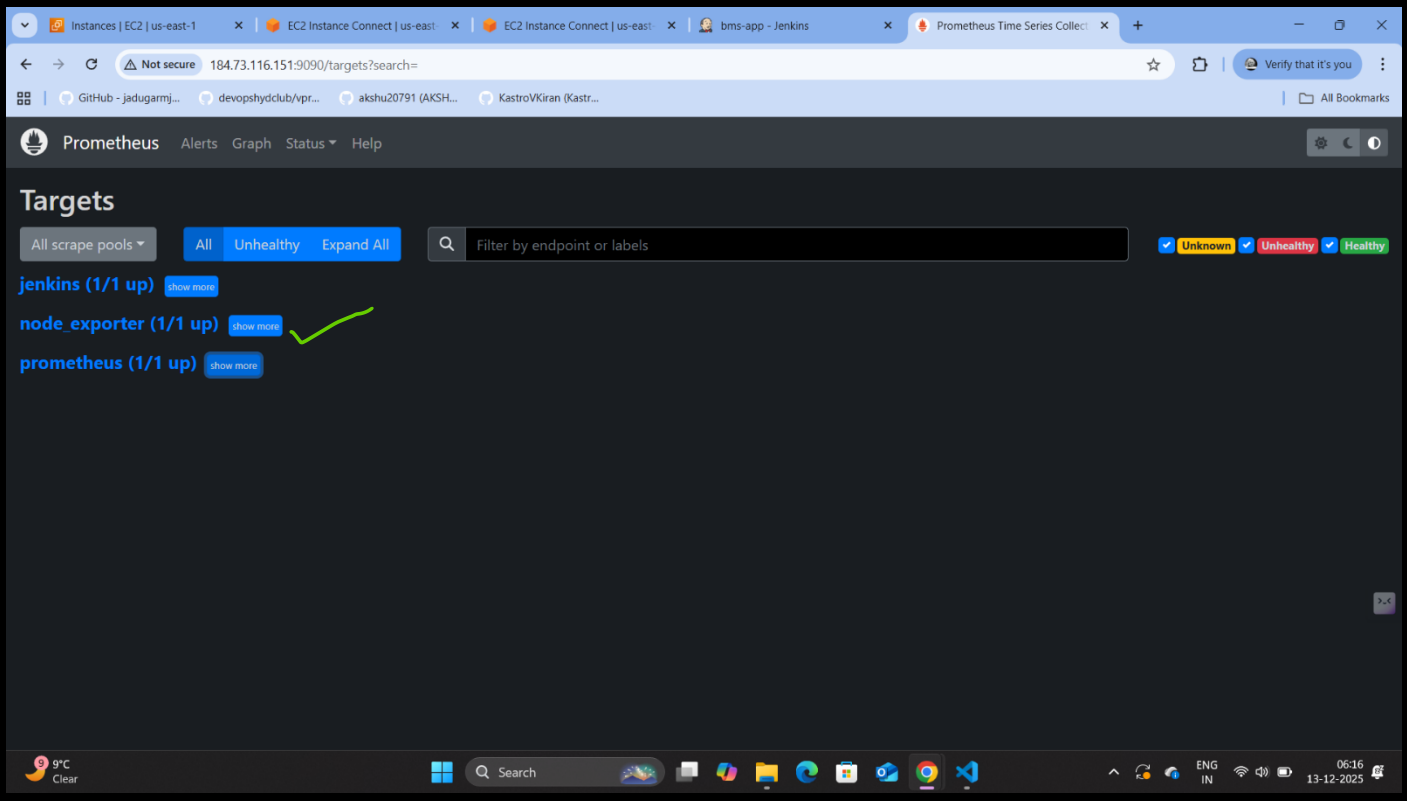
Step-38- Here we see that our node_exporter is not running so we will go to the security group of the monitoring server.



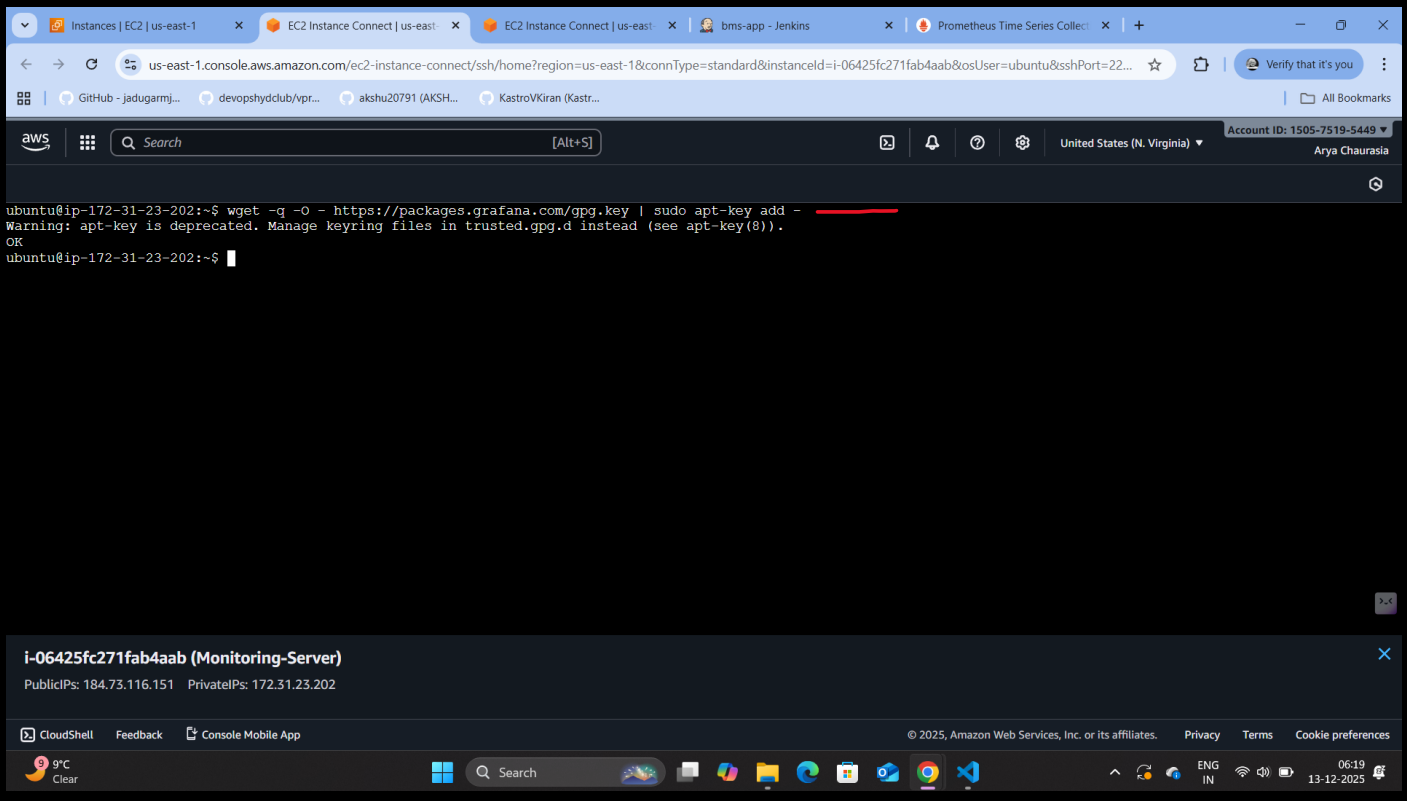
Step-39- We will enable the port 9100 for node_exporter.



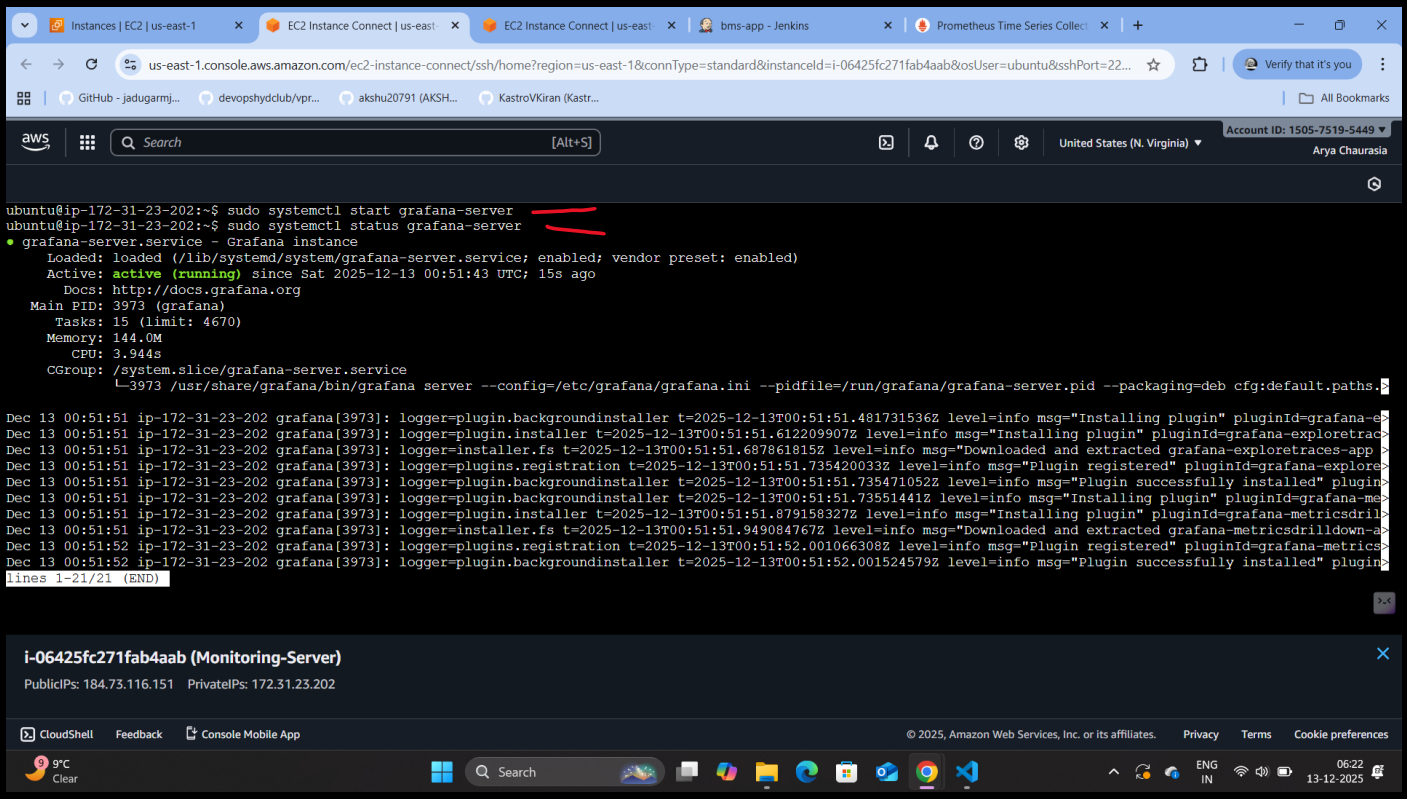
Step-40- Now refresh the page and we will see that the node_exporter is running.



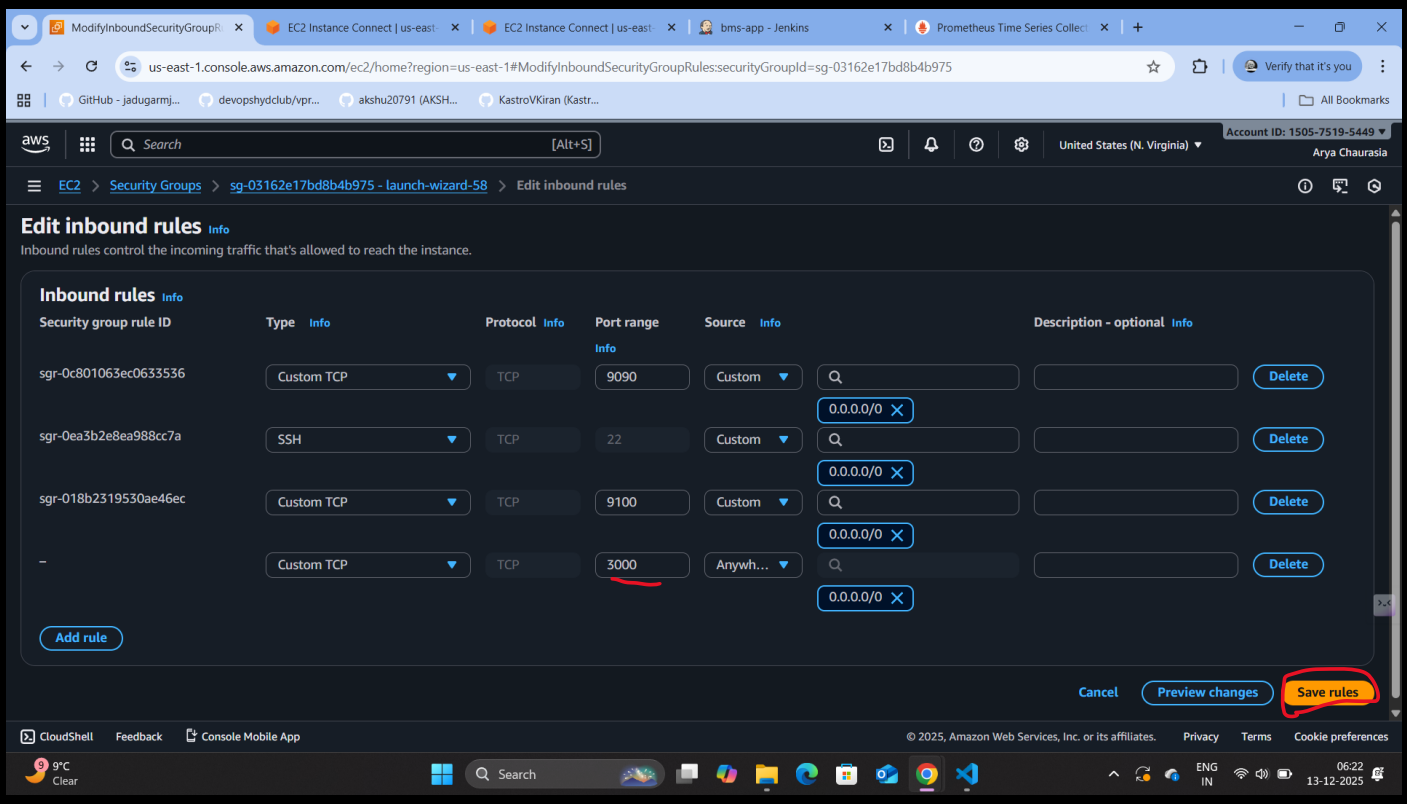
Step-41- Now we will install Grafana and add the gpg key for Grafana and you should see OK when the command is executed.



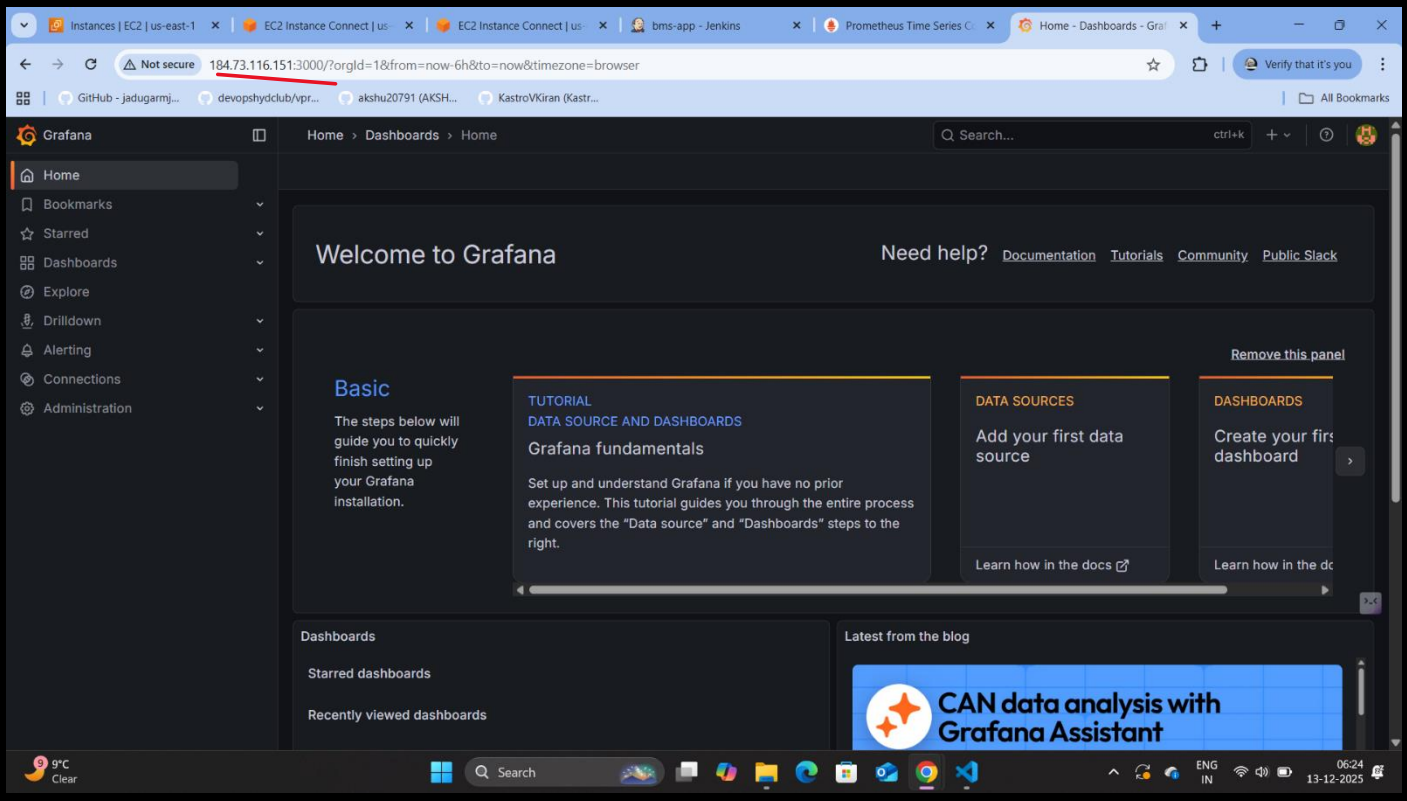
Step-42- Enable and start the Grafan Service and also we will check the Grafana status.



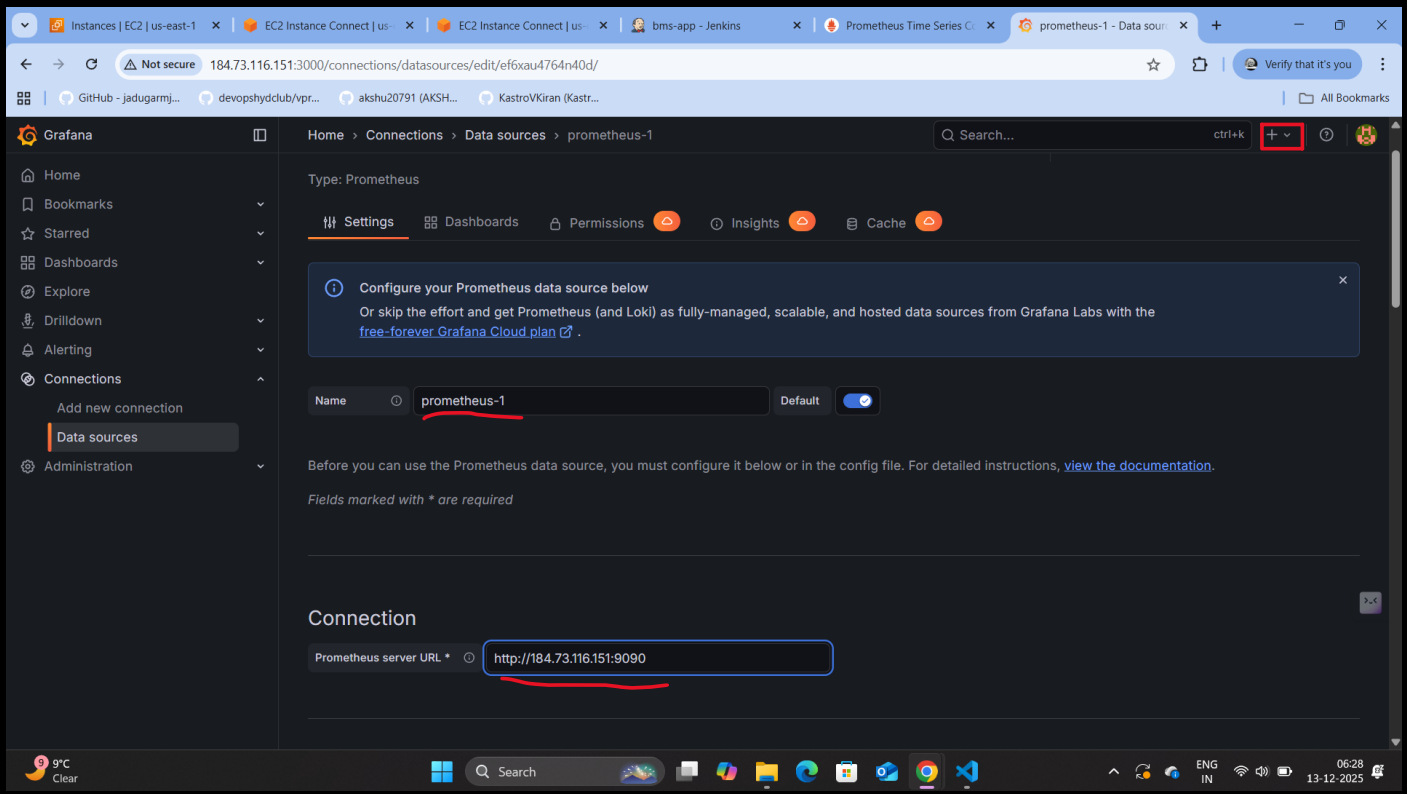
Step-43-We enable the port 3000 for Grafana.



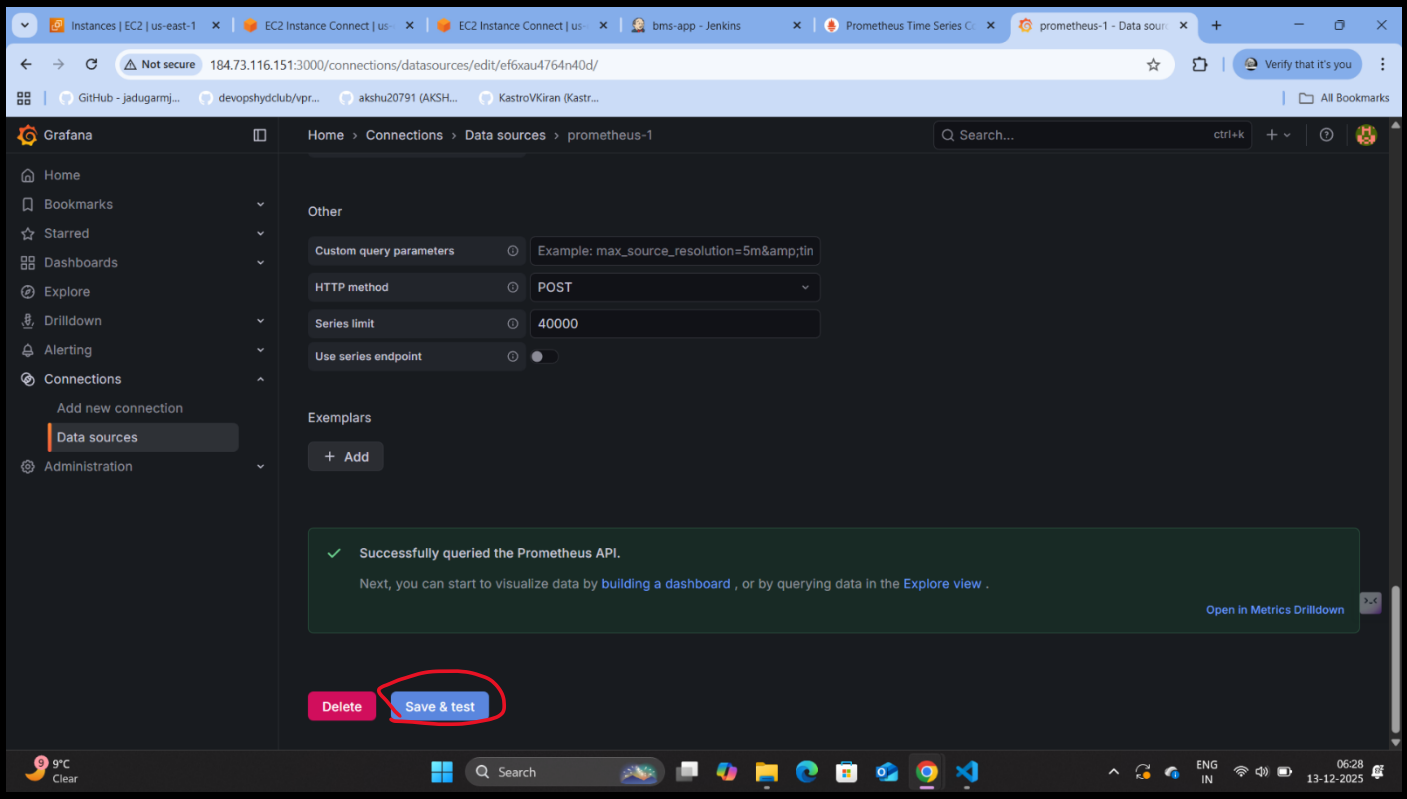
Step-44- <Monitoring Server IP>3000 to open the Grafana.



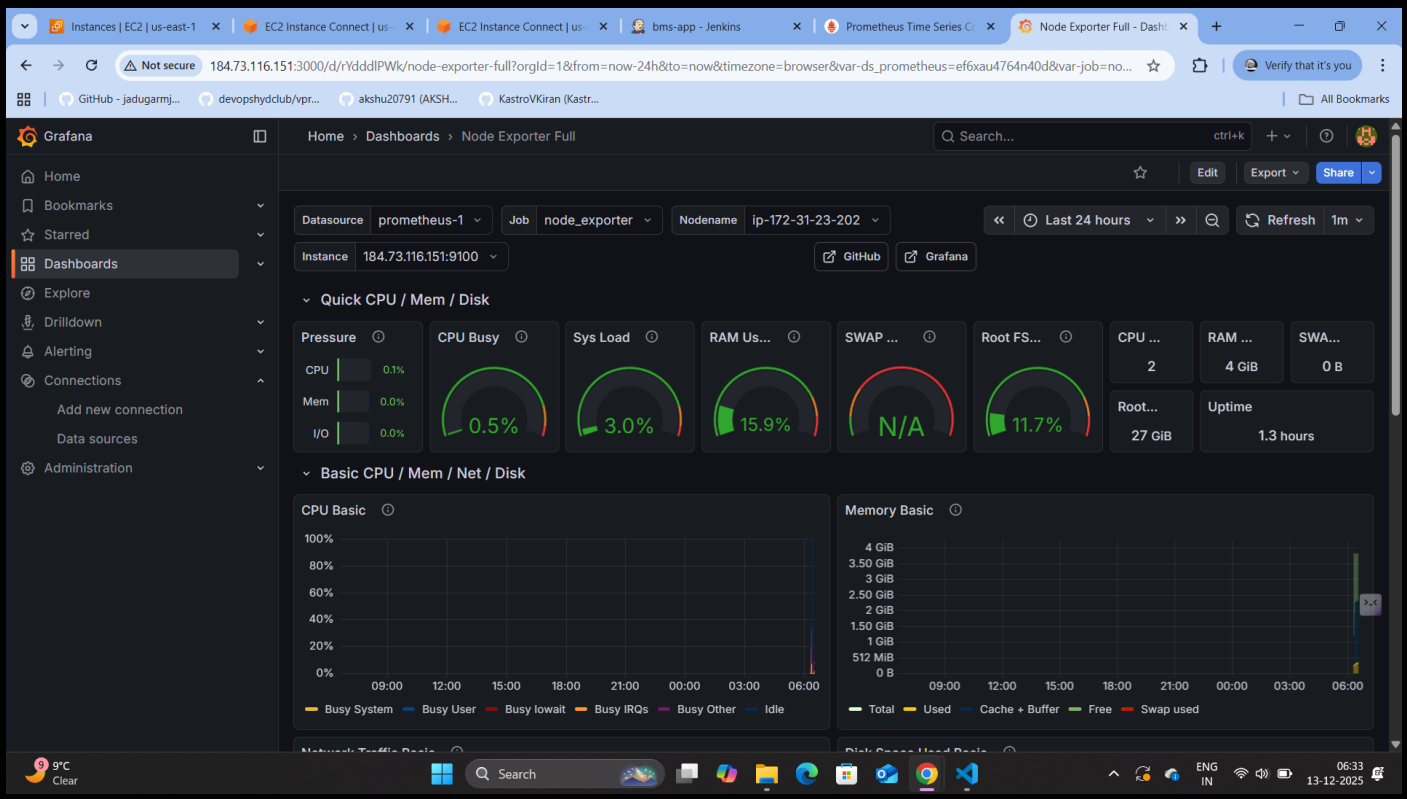
Step-45-On the right site upper corner + sign is there, click there on import dashboards, give name and IP address of the monitoring server.



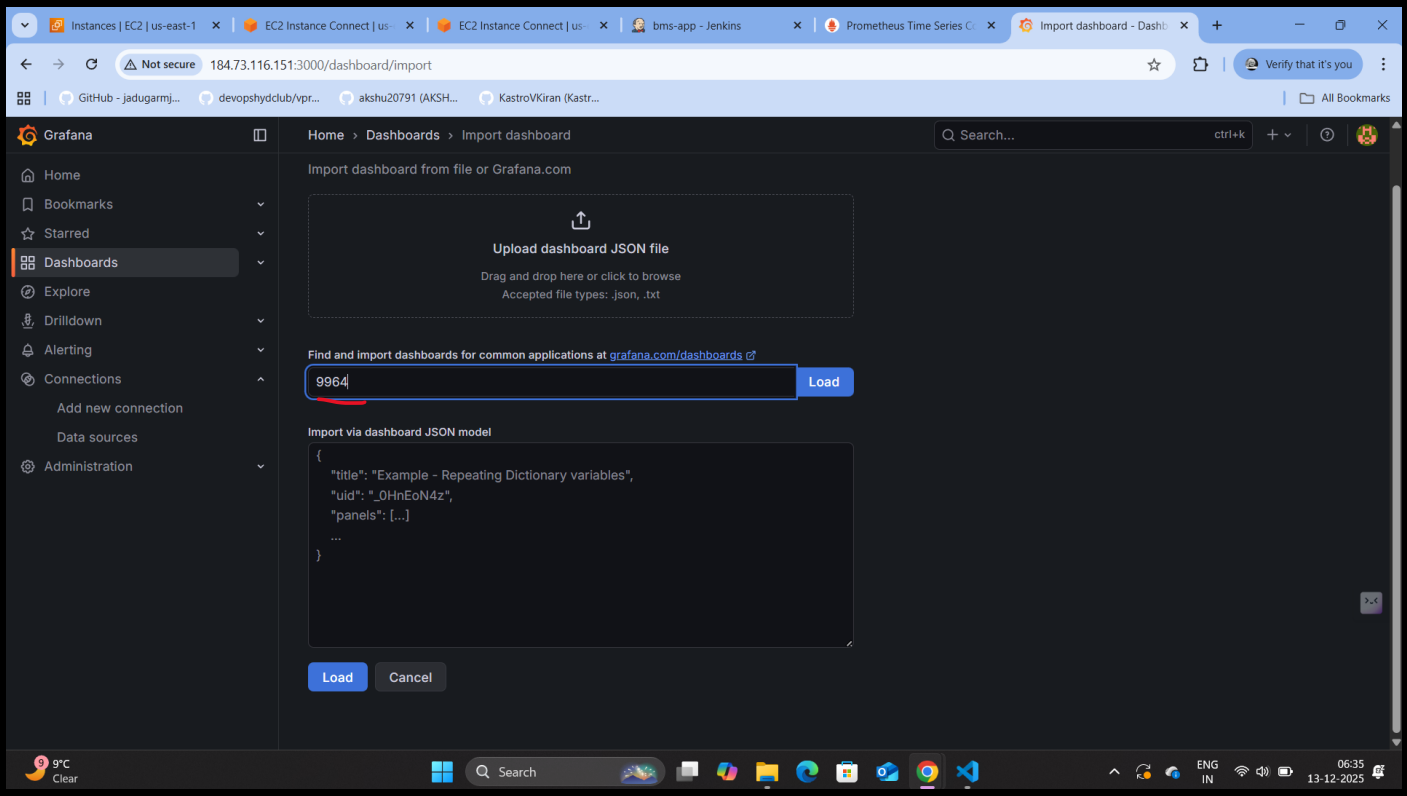
Step-46-Click save to get the dashboard for the prometheus



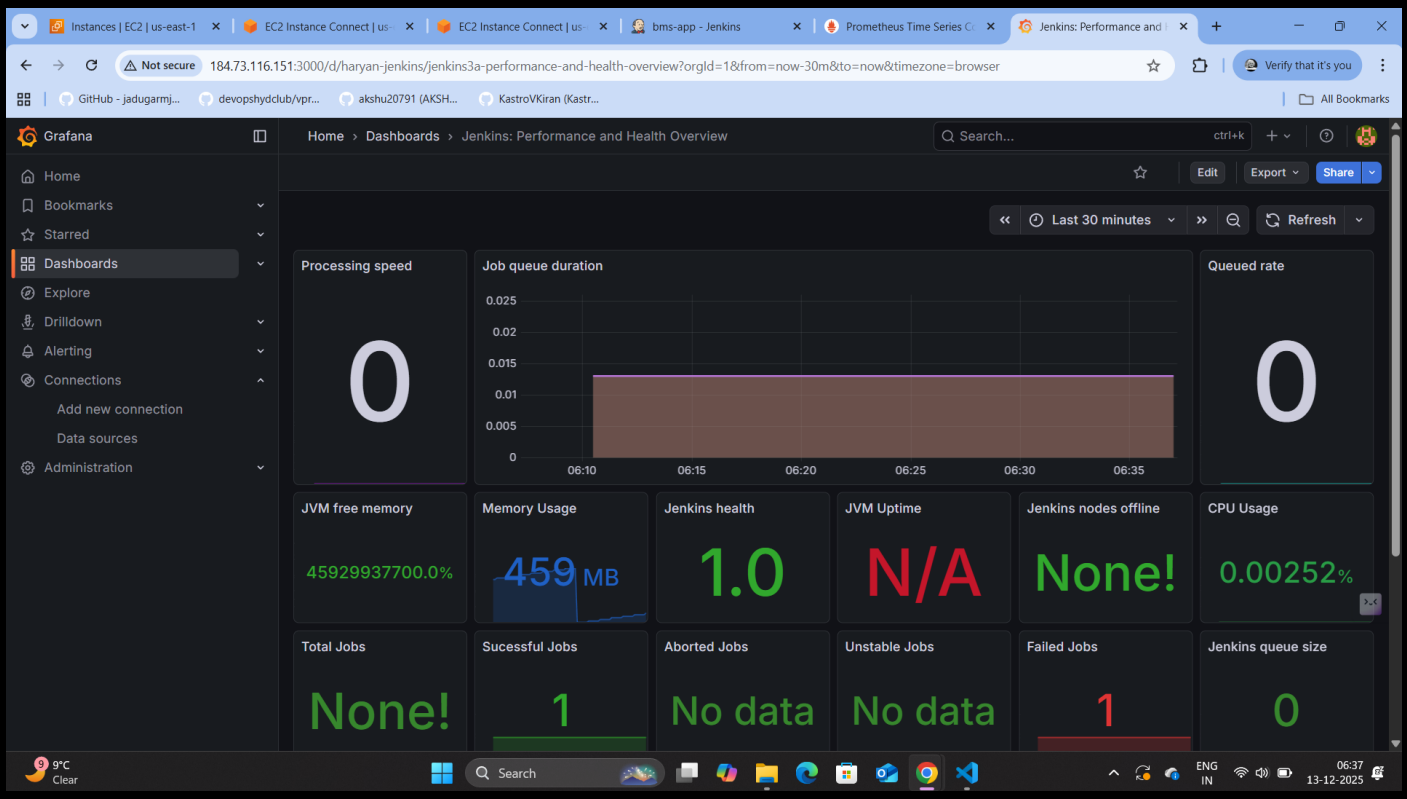
Step-47- Here is our dashboard



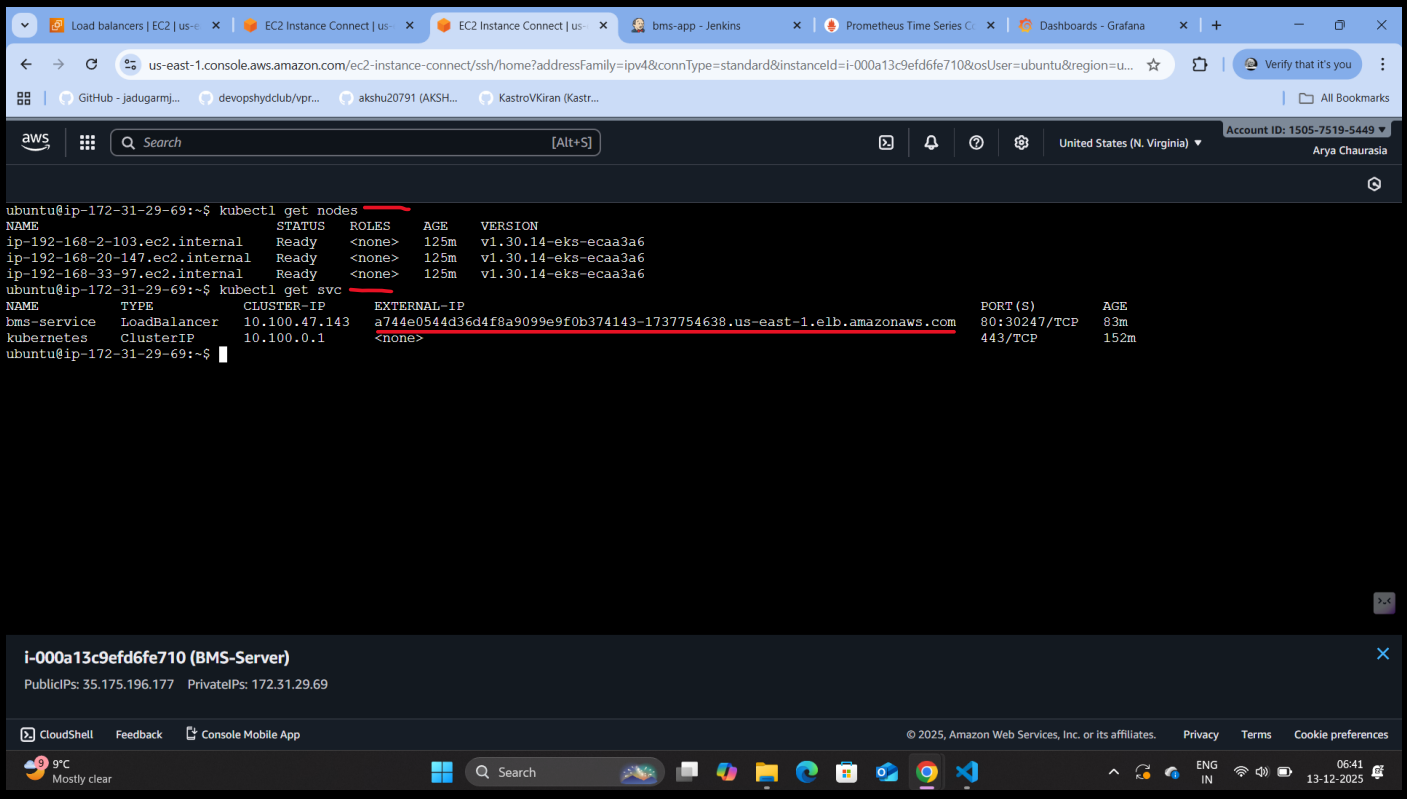
Step-48- 9964 is dashboard number for Jenkins , load it



Step-49- Here we can see the performance of the Jenkins.



Step-50- It shows the worker nodes inside the K8S Cluster, kubectl get svc will give you the URL for your application, copy it and paste it to the browser.



Step-51- Here we seen the deployement of our application.