**PROJECT:2** **Trend Store**

**Project Overview:**

The project uses Docker for containerizing the application and terraform to provision AWS infrastructure using Infrastructure as Code. Jenkins automates the CI/CD pipeline for building and deploying the application. The containerized application is deployed on AWS EKS for scalable Kubernetes orchestration, and Docker Hub is used to store container images. Prometheus and Grafana are implemented for monitoring and visualizing application and infrastructure performance.

**Tech Stack used:**

* Github
* Docker Hub
* Jenkins
* EC2
* Kubernetes
* DockerBuild
* Terraform
* Grafana
* Prometheus

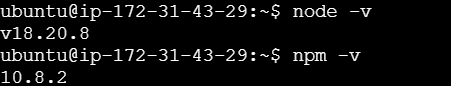
**System Check:**

* sudo apt update -y
* sudo apt upgrade -y

**Installation:**

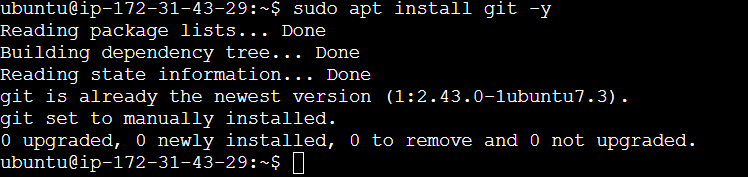
**Nodejs**

* curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash –
* sudo apt install -y nodejs
* node -v
* npm -v



**Git**

* sudo apt install git -y



**Docker**

* sudo apt install docker.io -y

**Enable Docker**

* sudo systemctl start docker
* sudo systemctl enable docker

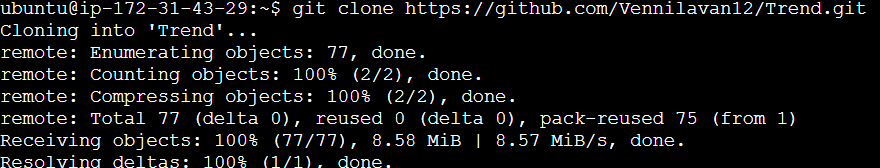
**Docker Permission**

* sudo usermod -aG docker ubuntu
* exit
* docker ps



**Cloning:**

* git clone <https://github.com/Vennilavan12/Trend.git>



**Creating Nano Docker file:**

* cd Trend
* Inside Trend folder creating nano docker file

FROM nginx:alpine

COPY dist /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**Build Docker Image:**

* docker build -t trend app .

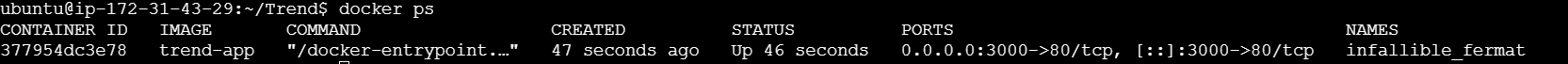


**Run Docker Container:**

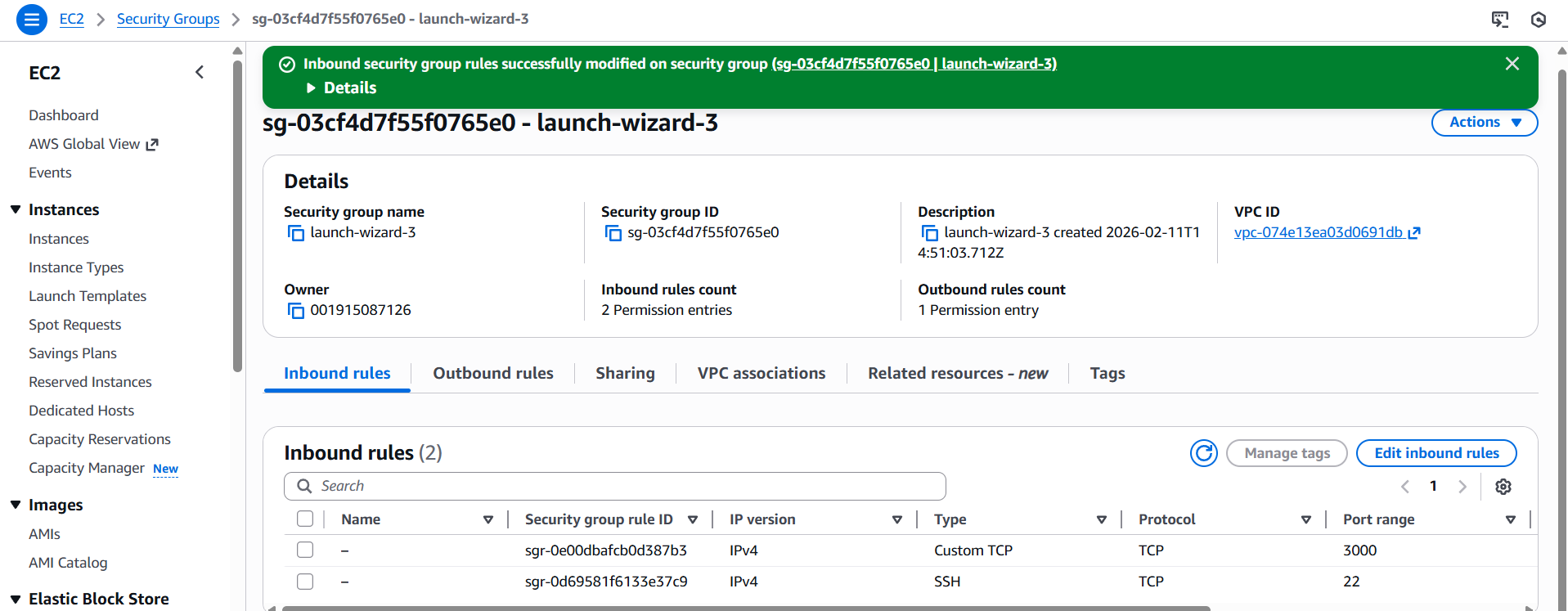
* docker run -d -p 3000:80 trend-app



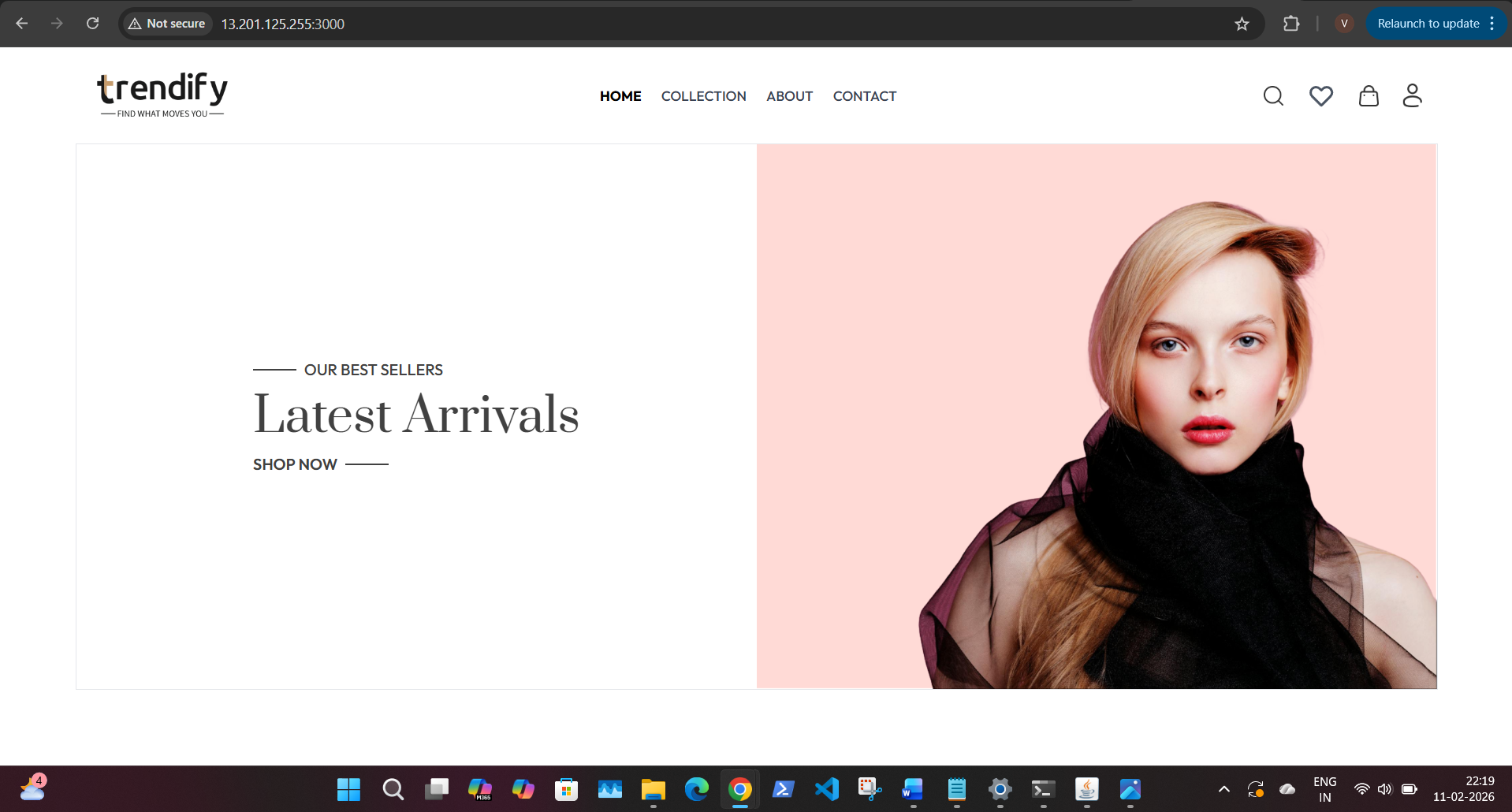
* docker ps



**Security Group:**

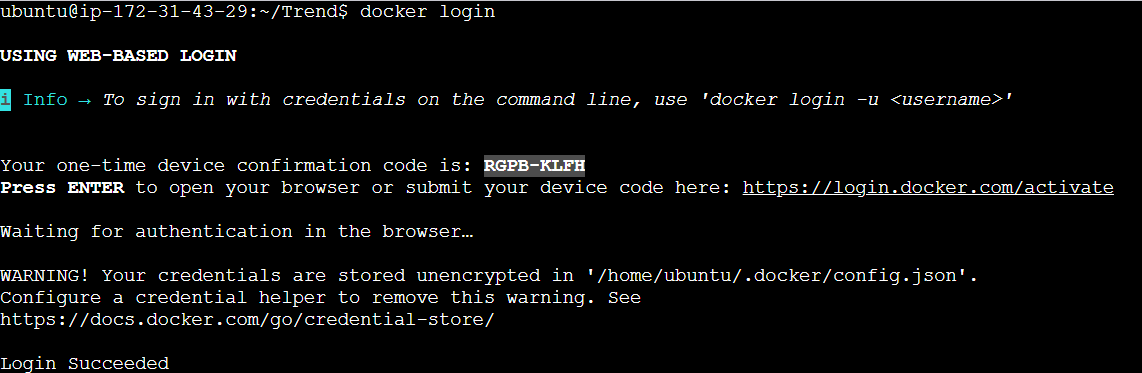


**Ip Address Check:** http://13.201.125.255:3000



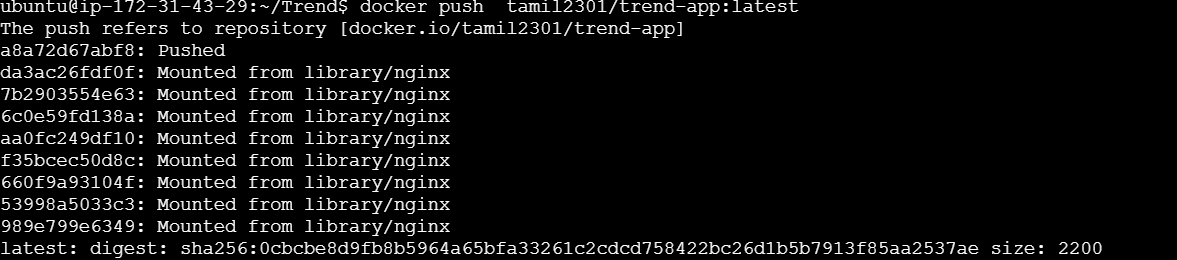
**Docker Login:**

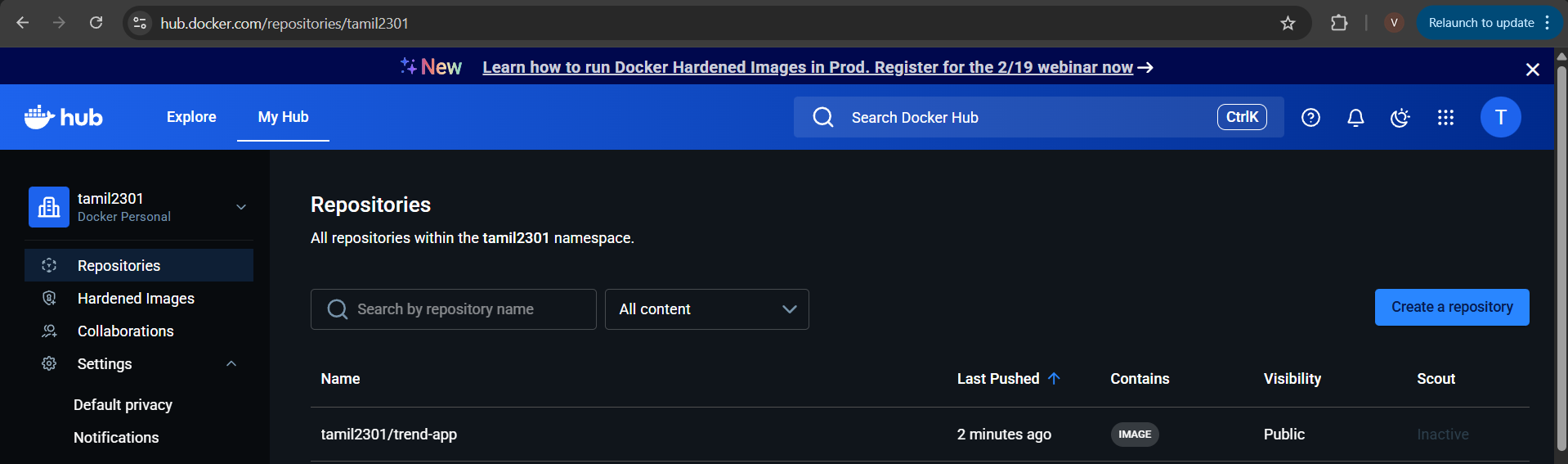
* docker login



**Image Tagging:**

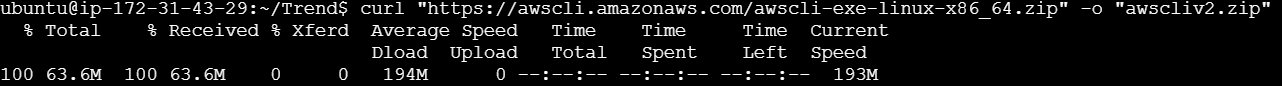
* docker tag trend-app tamil2301/trend-app:latest
* docker push tamil2301/trend-app:latest



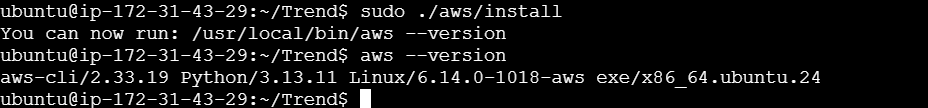


**AWS CLI Installation:**

* sudo apt install unzip -y
* curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

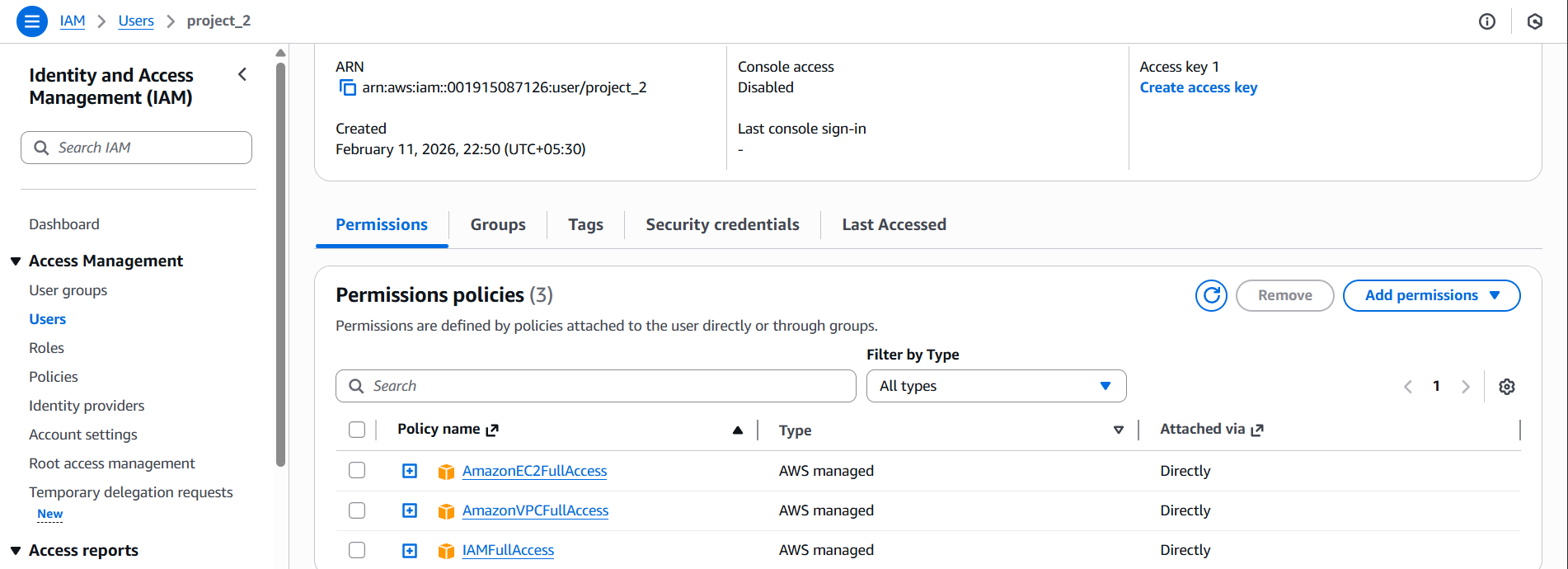


* unzip awscliv2.zip
* sudo ./aws/install
* aws –version



**AWS configure:**

* AWS🡪IAM🡪Create user🡪give name🡪select the required policy and go head🡪save



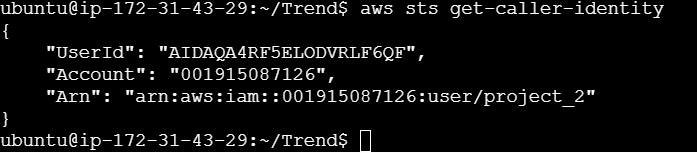
**Go to Security Credential 🡪 Create Access Key 🡪Command Line Interface (CLI) 🡪Next🡪Generate Access Key**

**Access Key: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (i.e. it will be in AWS)**

**Secrete Key: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (i.e. it will be in AWS)**

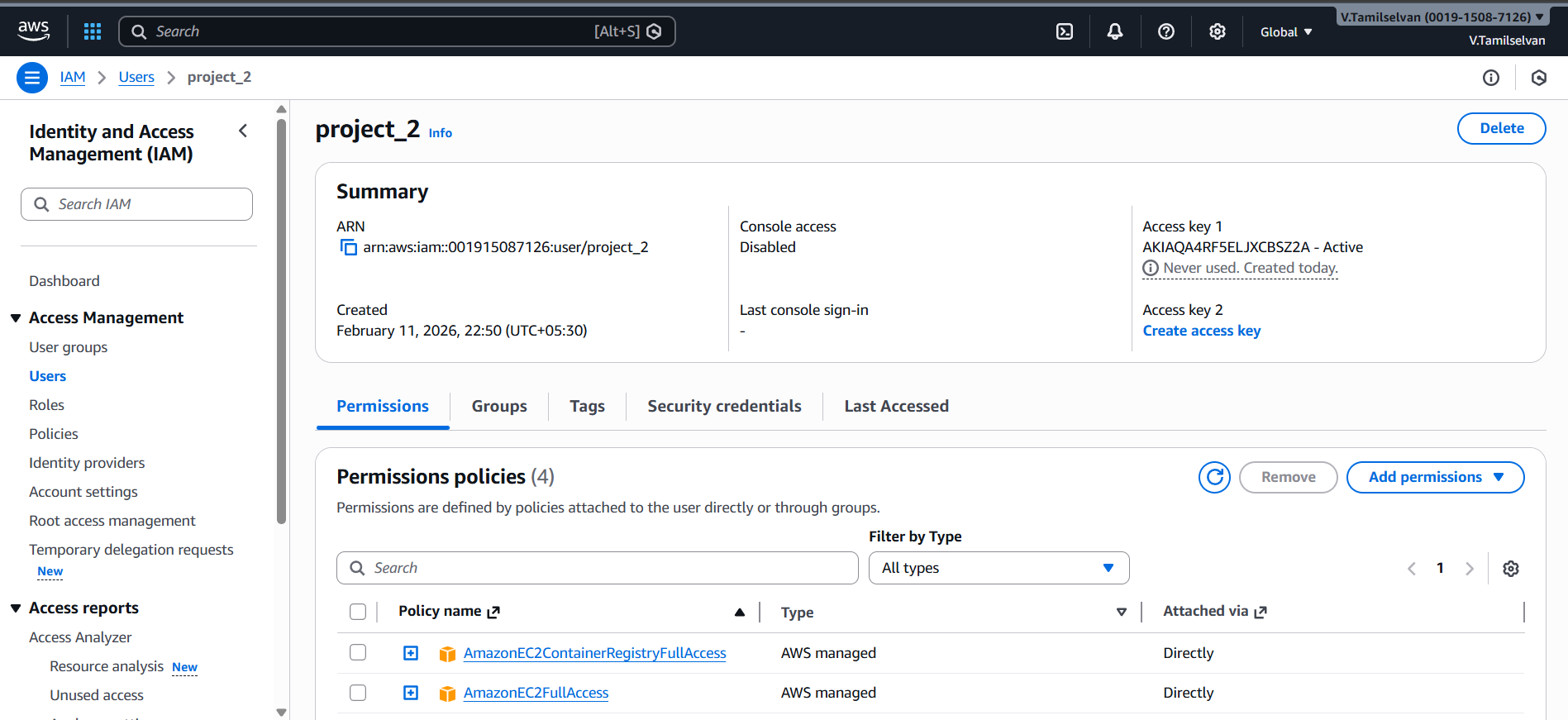
**Verify AWS CLI:**

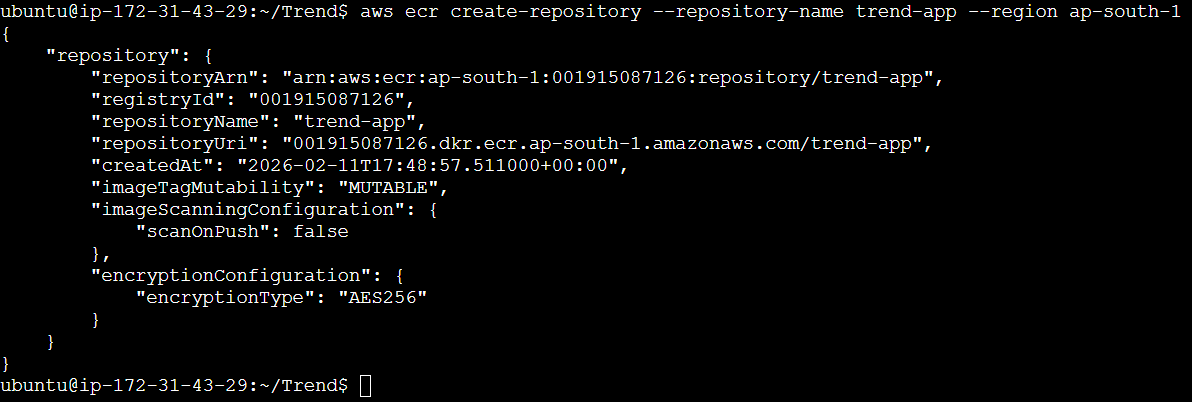
* aws sts get-caller-identity



**ECR Repository:**

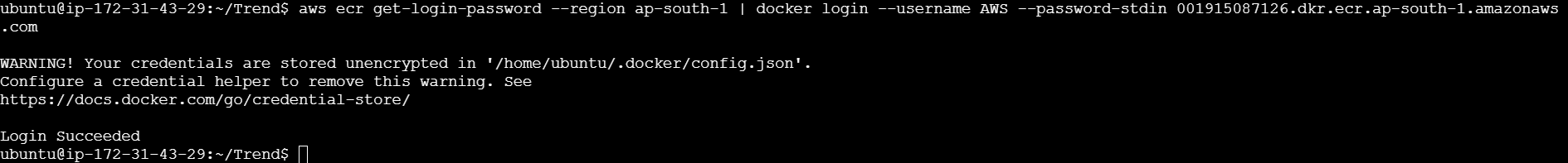
* aws ecr create-repository --repository-name trend-app --region ap-south-1
* Add AmazonEC2containerregistryfullaccess to existing policy
* Then again run 🡪 aws ecr create-repository --repository-name trend-app --region ap-south-1



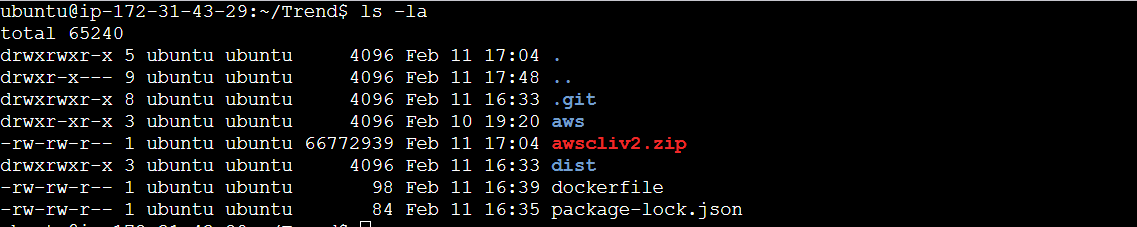


**Docker Login to ECR:**

* aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 001915087126.dkr.ecr.ap-south-1.amazonaws.com



* ls -la



* rm dockerfile
* nano Dockerfile

FROM nginx:alpine

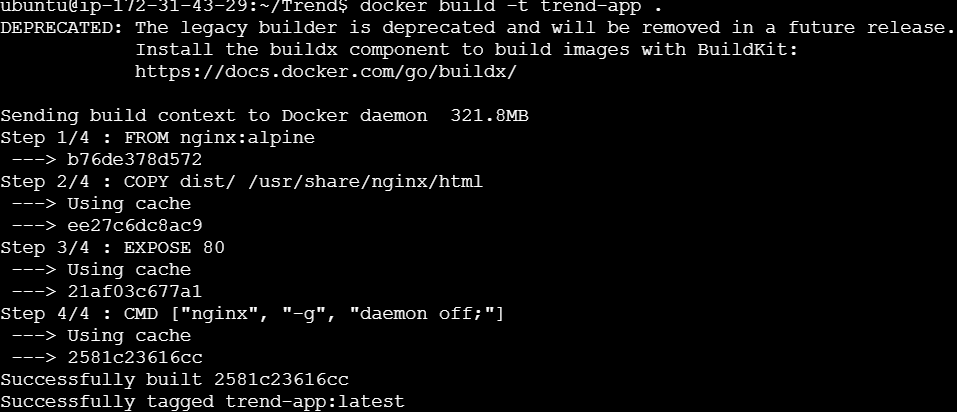
COPY dist/ /usr/share/nginx/html

EXPOSE 80

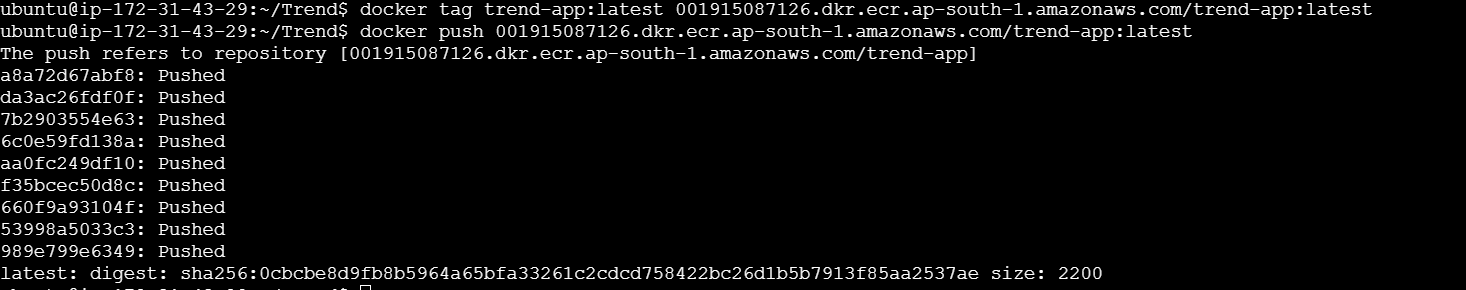
CMD ["nginx", "-g", "daemon off;"]

**Docker Image Build:**

* docker build -t trend-app .

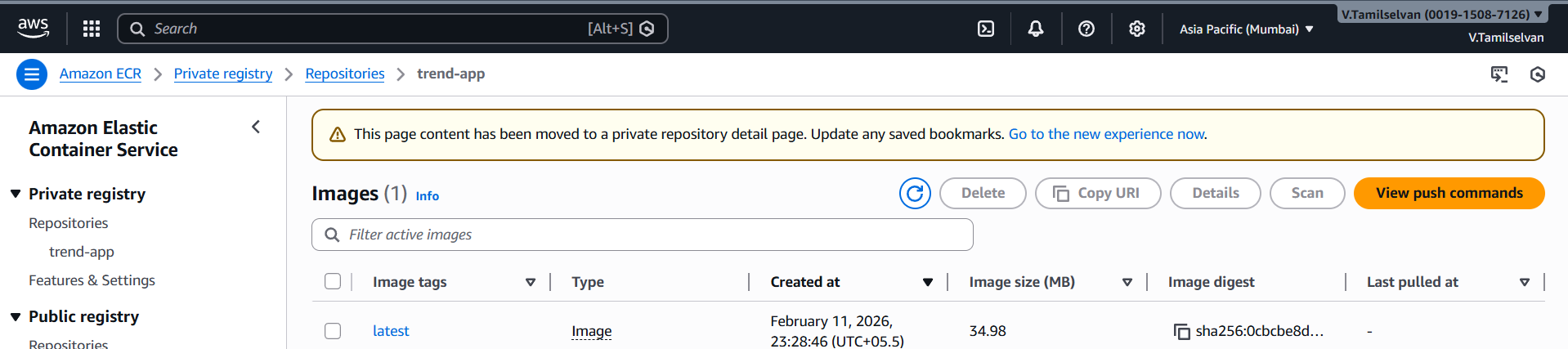


* Tag 🡪docker tag trend-app:latest 001915087126.dkr.ecr.ap-south-1.amazonaws.com/trend-app:latest
* Push 🡪 docker push 001915087126.dkr.ecr.ap-south-1.amazonaws.com/trend-app:latest



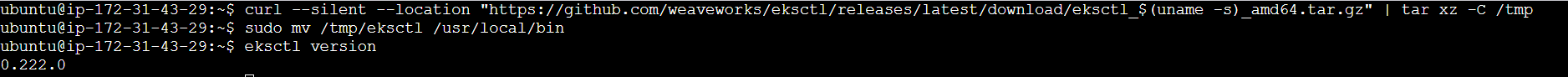
**AWS Verify:**

AWS🡪ECR🡪trend-app🡪images



**EKS cluster creation:**

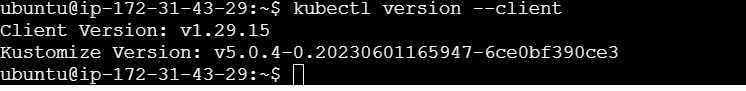
* curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s) \_amd64.tar.gz" | tar xz -C /tmp
* sudo mv /tmp/eksctl /usr/local/bin
* eksctl version



**Kubectl Installation:**

* sudo apt update
* sudo apt install -y kubectl
* Install Dependency 🡪 sudo apt install -y curl apt-transport-https
* Kubectl signing key 🡪 curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
* Adding kubernets repo 🡪 echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
* sudo apt update
* sudo apt install -y kubectl

**Kubernets version:**

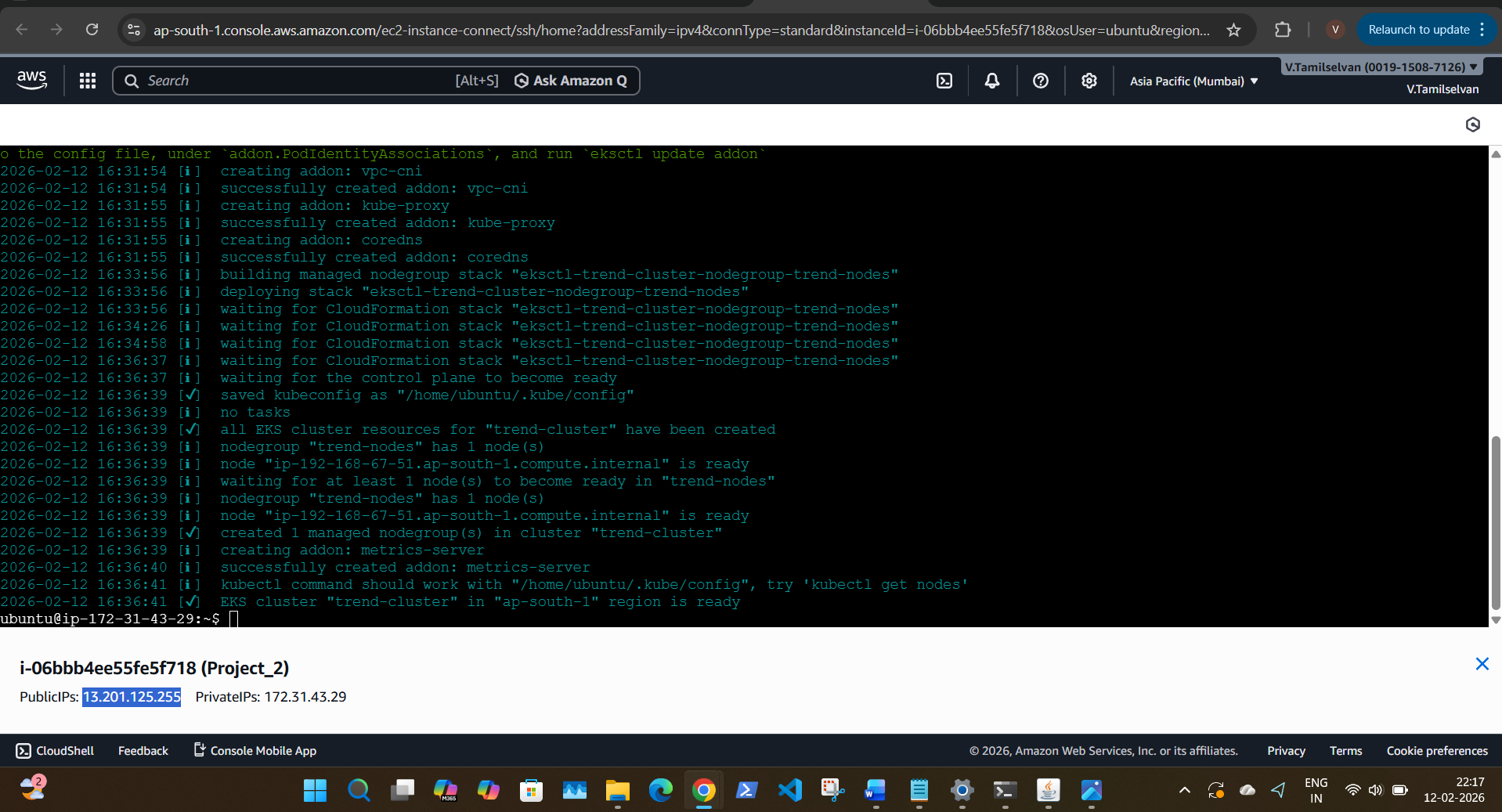


**Eksctl Installation:**

* eksctl version

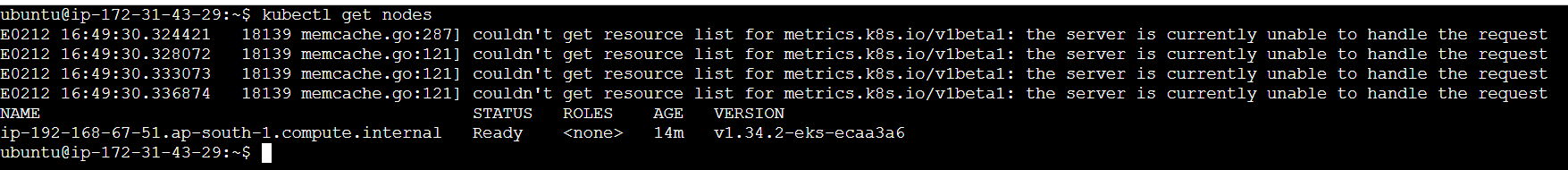


* eksctl create cluster \--name trend-cluster \--region ap-south-1 \--nodegroup-name trend-nodes \--node-type t3.micro \--nodes 1 \--nodes-min 1 \--nodes-max 1 \--managed



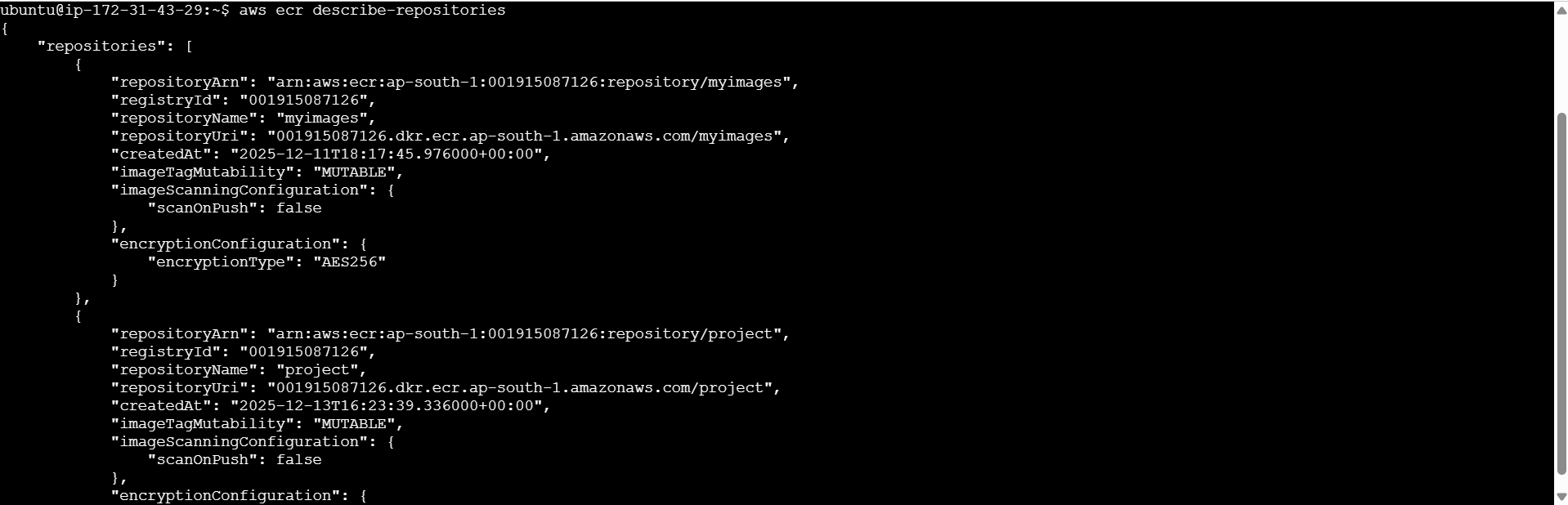
**Verify node:**

* Kubectl get nodes



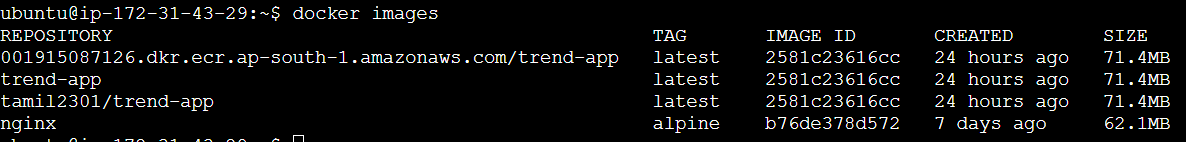
**Creating deployent.yaml:**

* aws ecr describe-repositories



**Docker Image check:**

* docker images

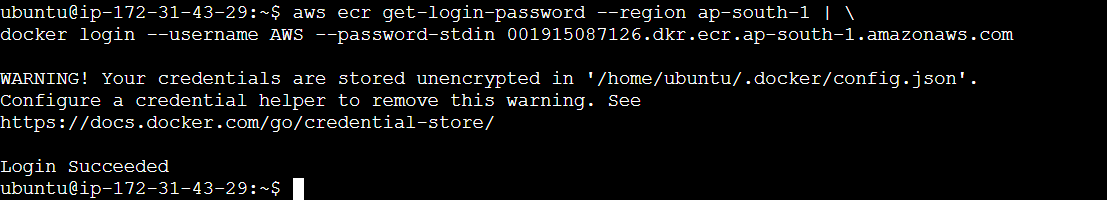


**Repo Creation:**

* **ECR login**

aws ecr get-login-password --region ap-south-1 | \

docker login --username AWS --password-stdin 001915087126.dkr.ecr.ap-south-1.amazonaws.com



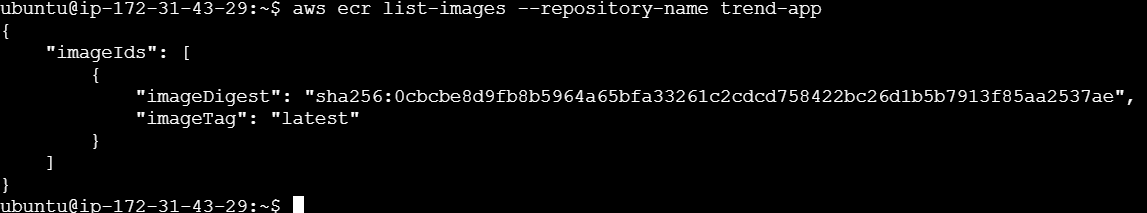
* **Docker push**

docker push 001915087126.dkr.ecr.ap-south-1.amazonaws.com/trend-app:latest

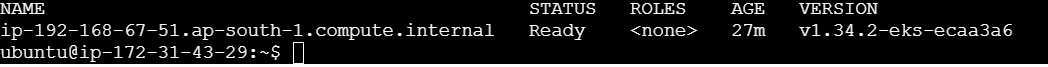


* **Verify image**

aws ecr list-images --repository-name trend-app



* **Verify cluster**



**Creating deployment.yaml:**

* **nano deployment. yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: trend-deployment

spec:

replicas: 1

selector:

matchLabels:

app: trend-app

template:

metadata:

labels:

app: trend-app

spec:

containers:

- name: trend-container

image: 001915087126.dkr.ecr.ap-south-1.amazonaws.com/trend-app:latest

ports:

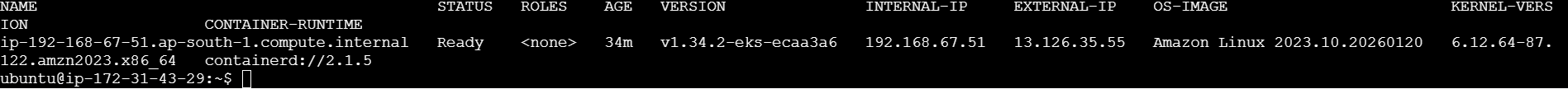
- containerPort: 80

**Apply Deployment:**

* kubectl apply -f deployment.yaml



* kubectl get nodes -o wide



* kubectl describe pod trend-deployment-588bcc6656-tws8b
* kubectl get pods -A

**Deleting Cluster and creating with t3. small**:

* eksctl delete cluster --name trend-cluster --region ap-south-1
* t3. small

eksctl create cluster \

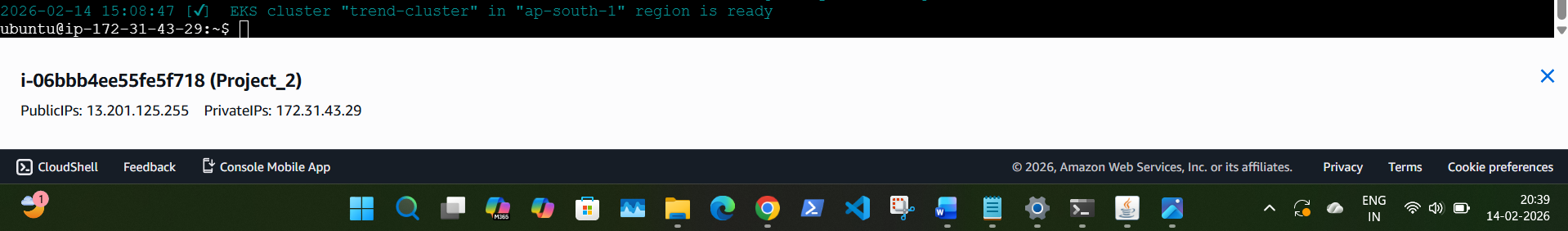
--name trend-cluster \

--region ap-south-1 \

--nodegroup-name trend-nodes \

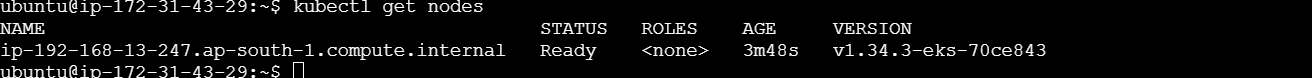
--nodes 1 \

--managed



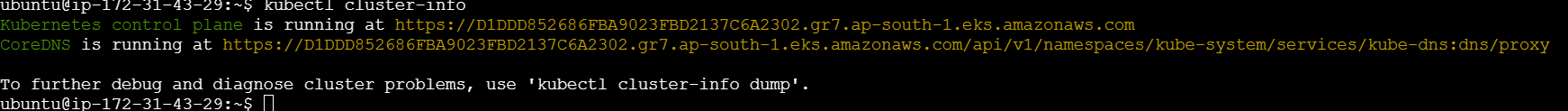
**Verify Cluster is accessible:**

* kubectl get nodes



**Verify Cluster info:**

* kubectl cluster-info



**Deploying Docker App to EKS:**

* **ECR image URL**

aws ecr describe-repositories --region ap-south-1 --query "repositories[0].repositoryUri" --output text



* **nano deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: trend-app-deployment

spec:

replicas: 2

selector:

matchLabels:

app: trend-app

template:

metadata:

labels:

app: trend-app

spec:

containers:

- name: trend-app

image: 001915087126.dkr.ecr.ap-south-1.amazonaws.com/myimages:latest

ports:

- containerPort: 80

* **nano service.yaml**

apiVersion: v1

kind: Service

metadata:

name: trend-app-service

spec:

type: LoadBalancer

selector:

app: trend-app

ports:

- protocol: TCP

port: 80

targetPort: 80

**Deploy:**

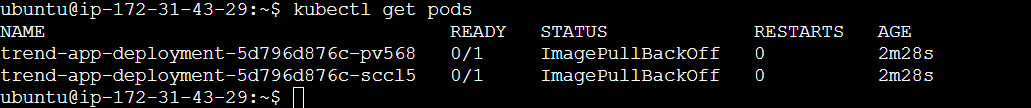
* kubectl apply -f deployment.yaml



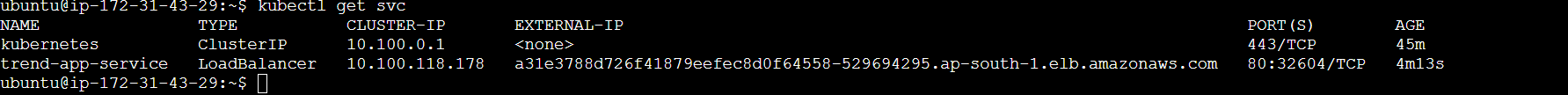
* kubectl apply -f service.yaml



* kubectl get pods (i.e. to check pods)



* kubectl get svc (i.e to check service)



**Create nano Dockerfile:**

* **nano Dockerfile**

FROM nginx:alpine

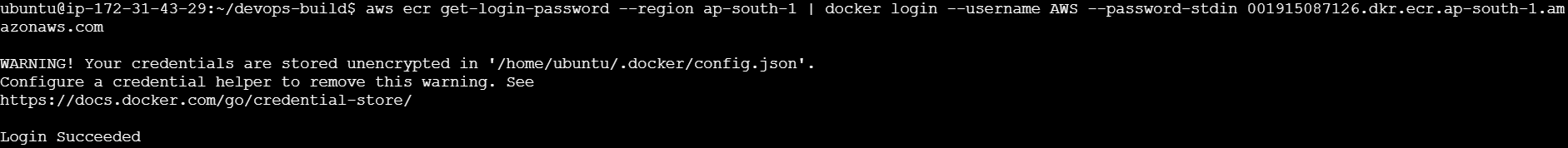
COPY build/ /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**Login to ECR:**

* aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 001915087126.dkr.ecr.ap-south-1.amazonaws.com



**Build Docker Image:**

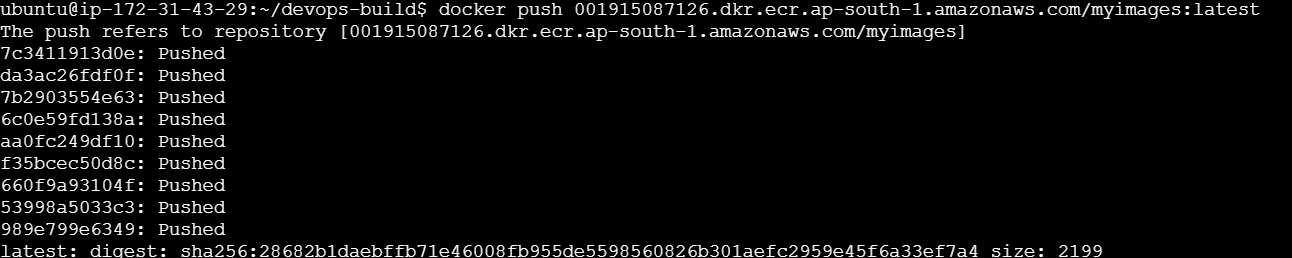
* docker build -t myimages .

**Tag Image:**

* docker tag myimages:latest 001915087126.dkr.ecr.ap-south-1.amazonaws.com/myimages:latest

**Push Image:**

* docker push 001915087126.dkr.ecr.ap-south-1.amazonaws.com/myimages:latest

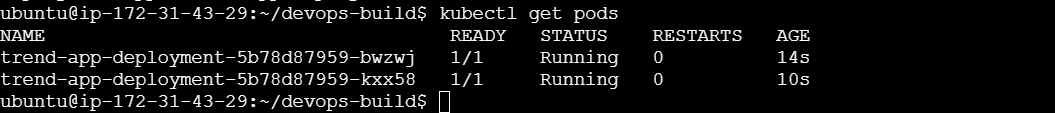


**Restart Deployment:**

* docker push 001915087126.dkr.ecr.ap-south-1.amazonaws.com/myimages:latest

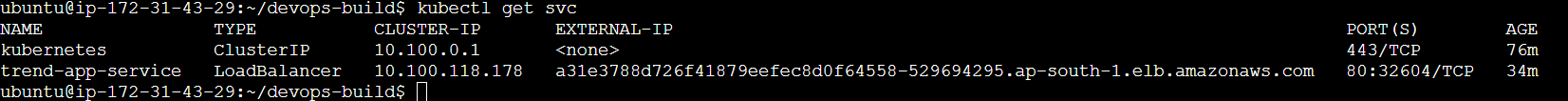
**Check pods:**

* kubectl get pods



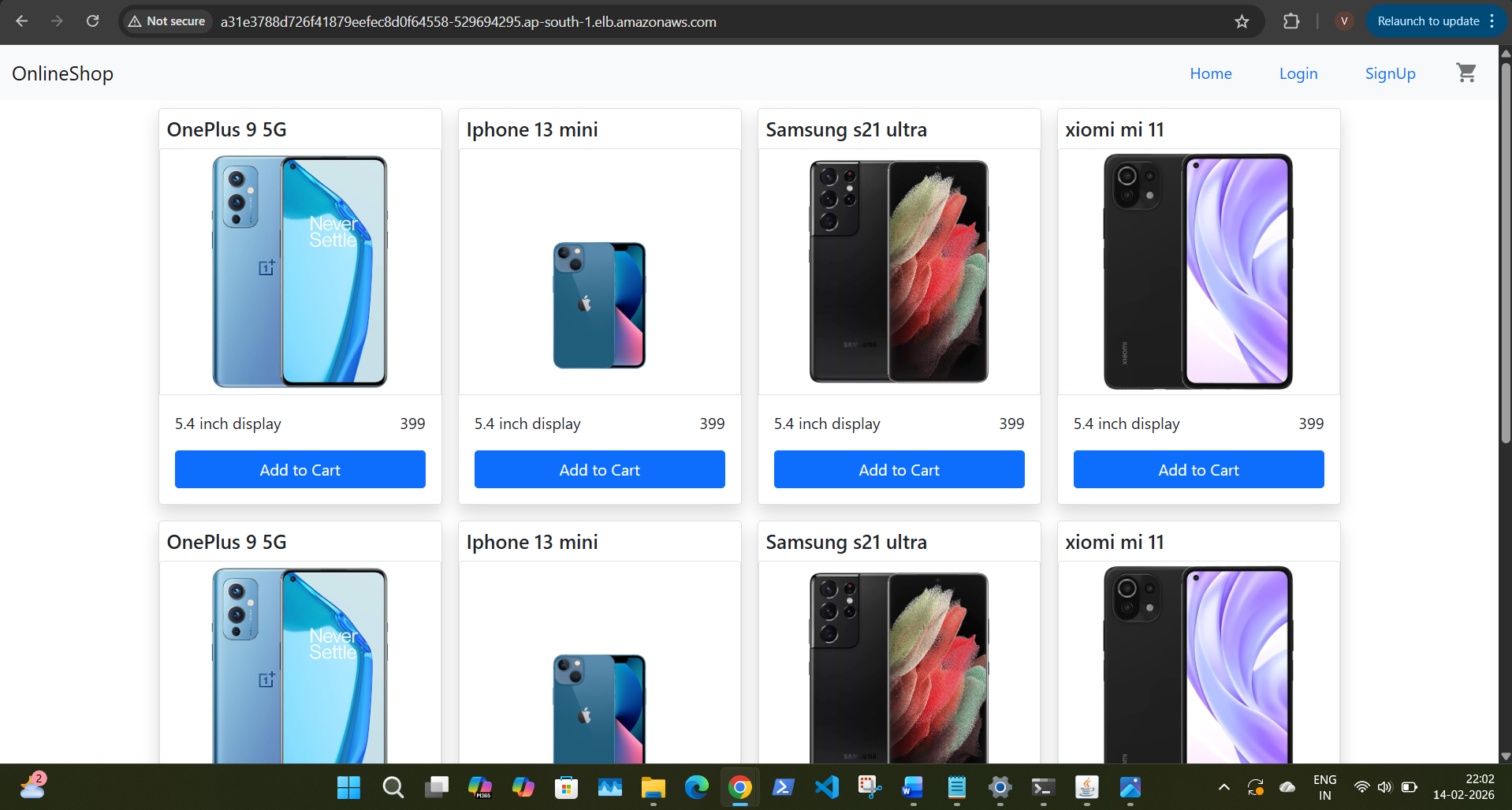
**To get service:**

* kubectl get svc



**Final Output:**

**http://a31e3788d726f41879eefec8d0f64558-529694295.ap-south-1.elb.amazonaws.com/**



**Monitoring:**

**Install Helm:**

* choco install kubernetes-helm -y
* helm version

**Add Prometheus Helm Repo:**

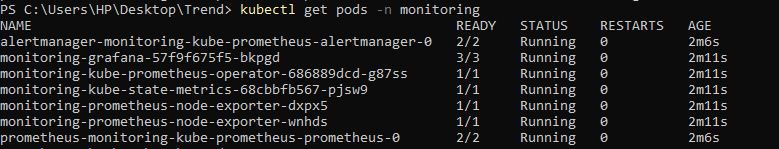
* helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
* helm repo update

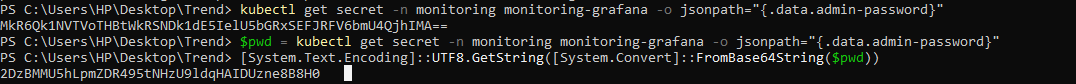
**Install Prometheus + Grafana Stack:**

* kubectl create namespace monitoring
* helm install monitoring prometheus-community/kube-prometheus-stack -n monitoring

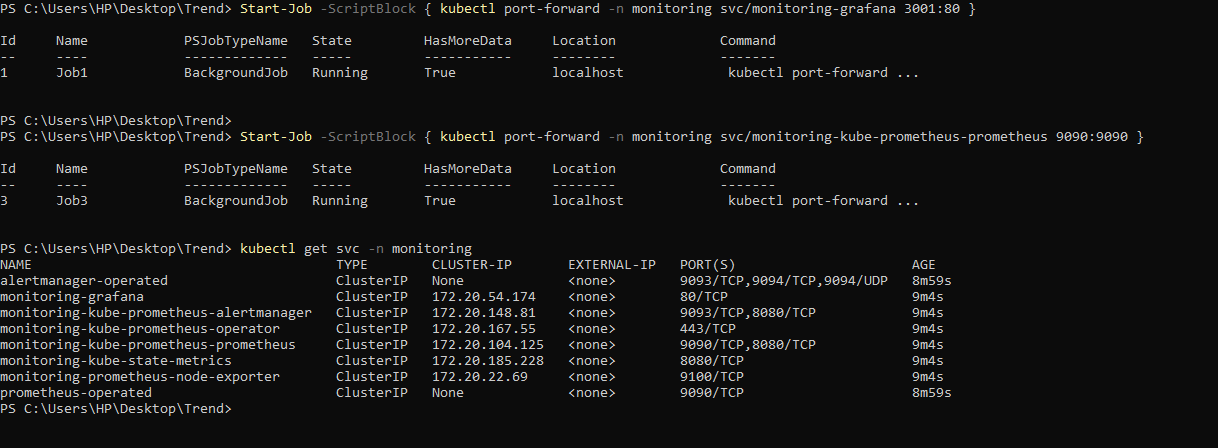
**Checking pods:**

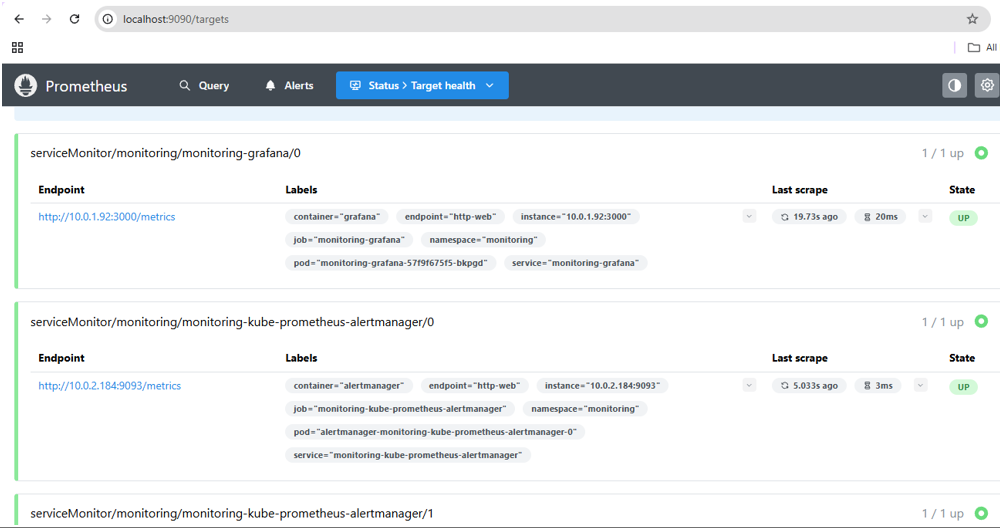
* kubectl get pods -n monitoring

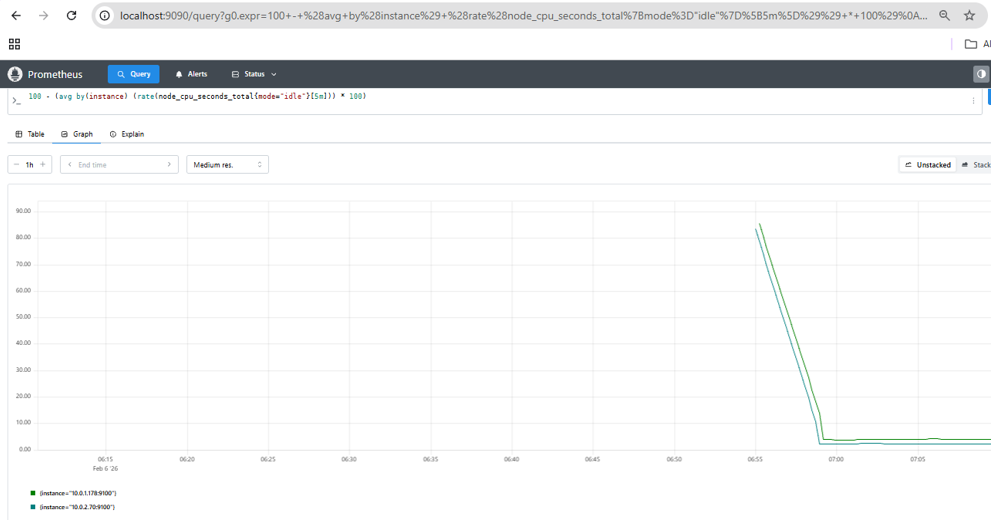


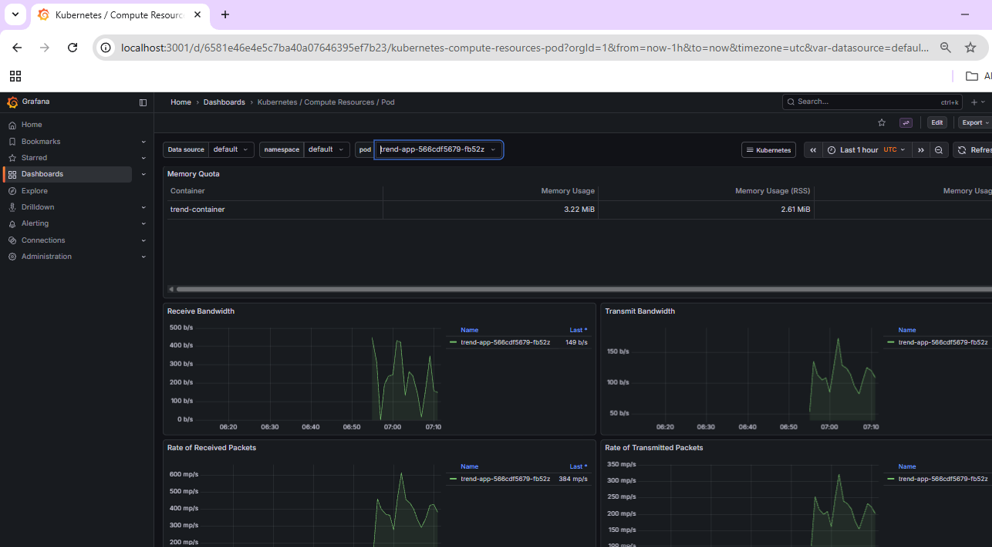


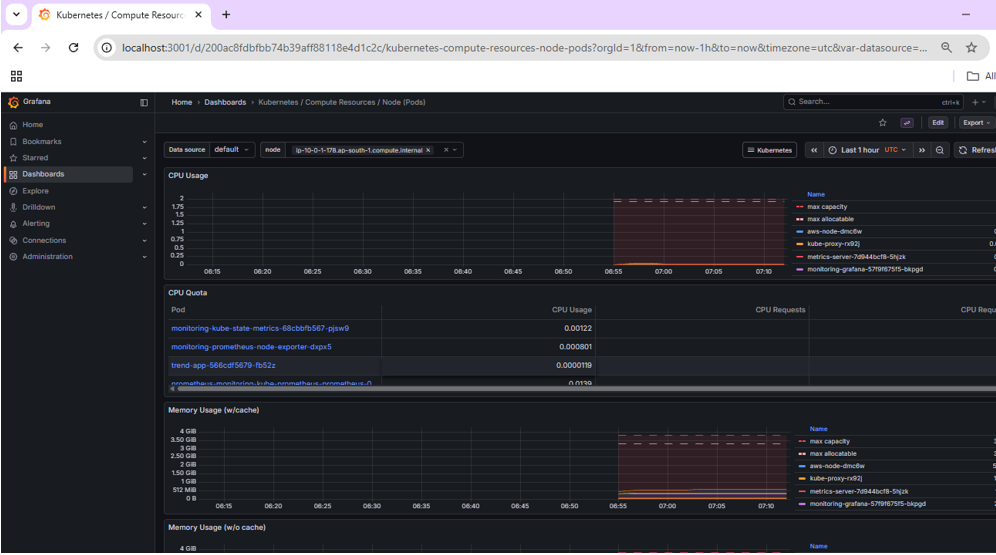
* kubectl get svc -n monitoring
* kubectl get pods -n monitoring

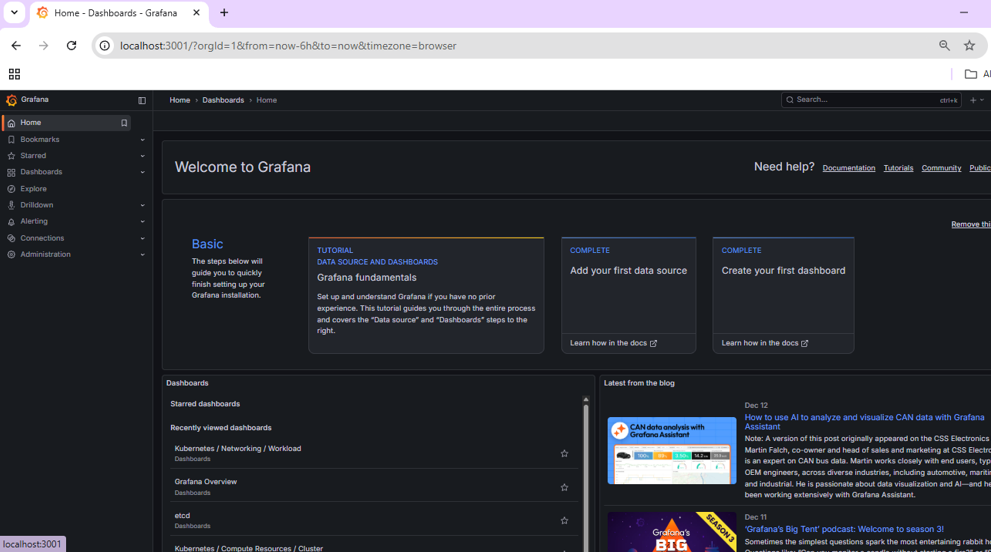
****

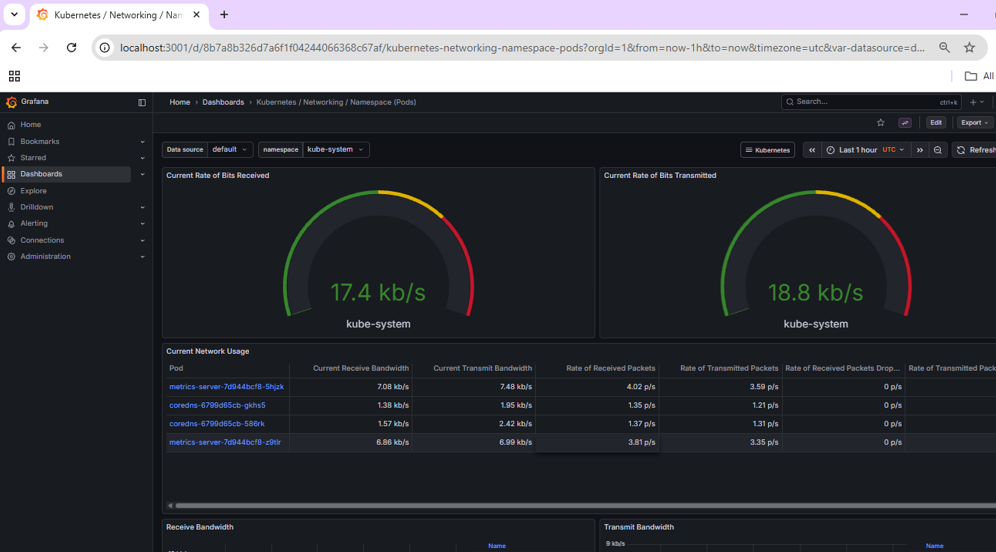


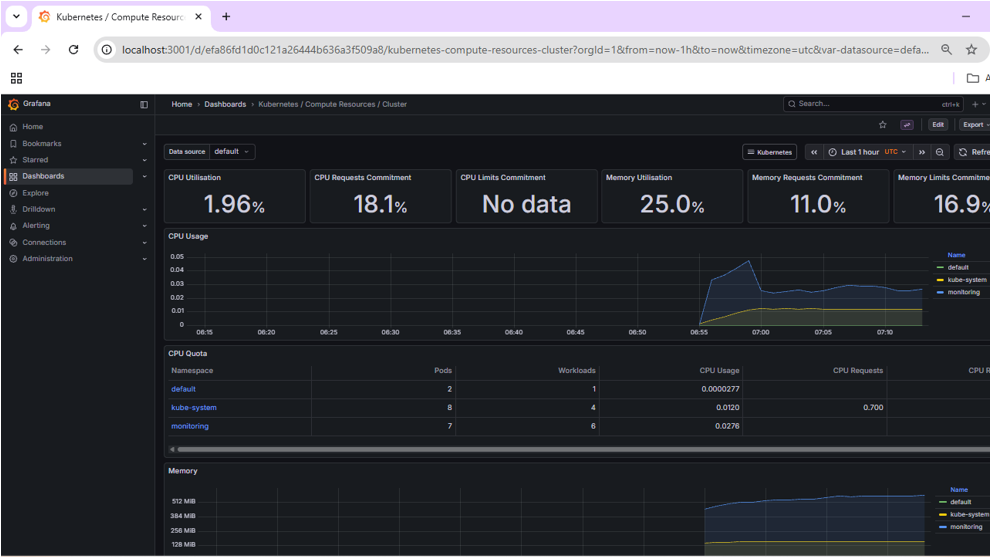


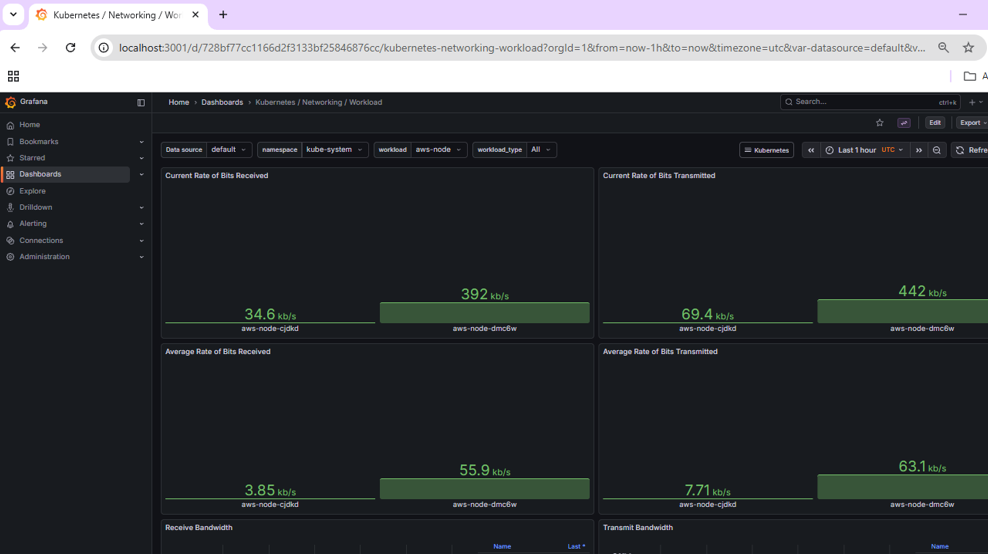




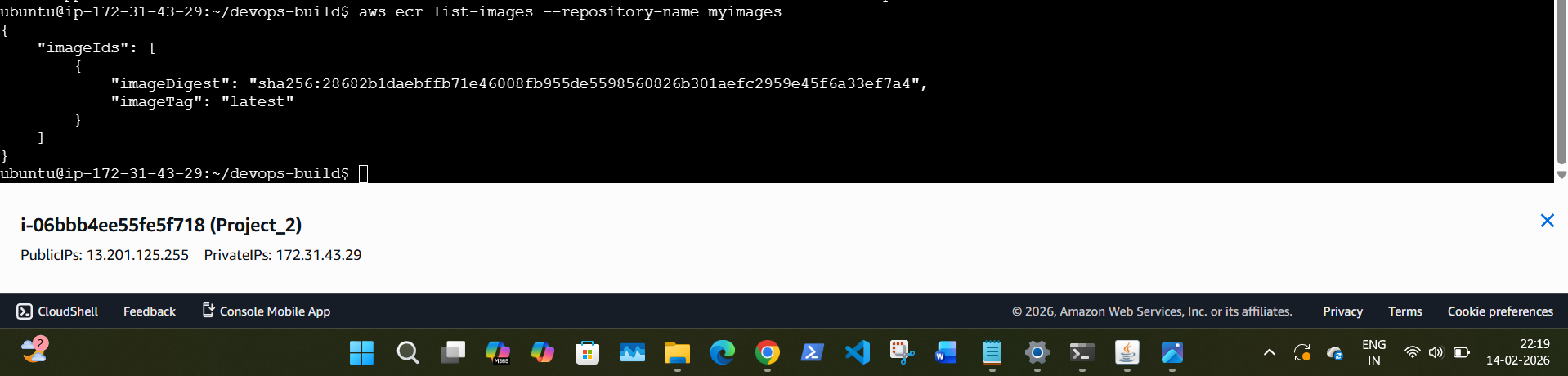


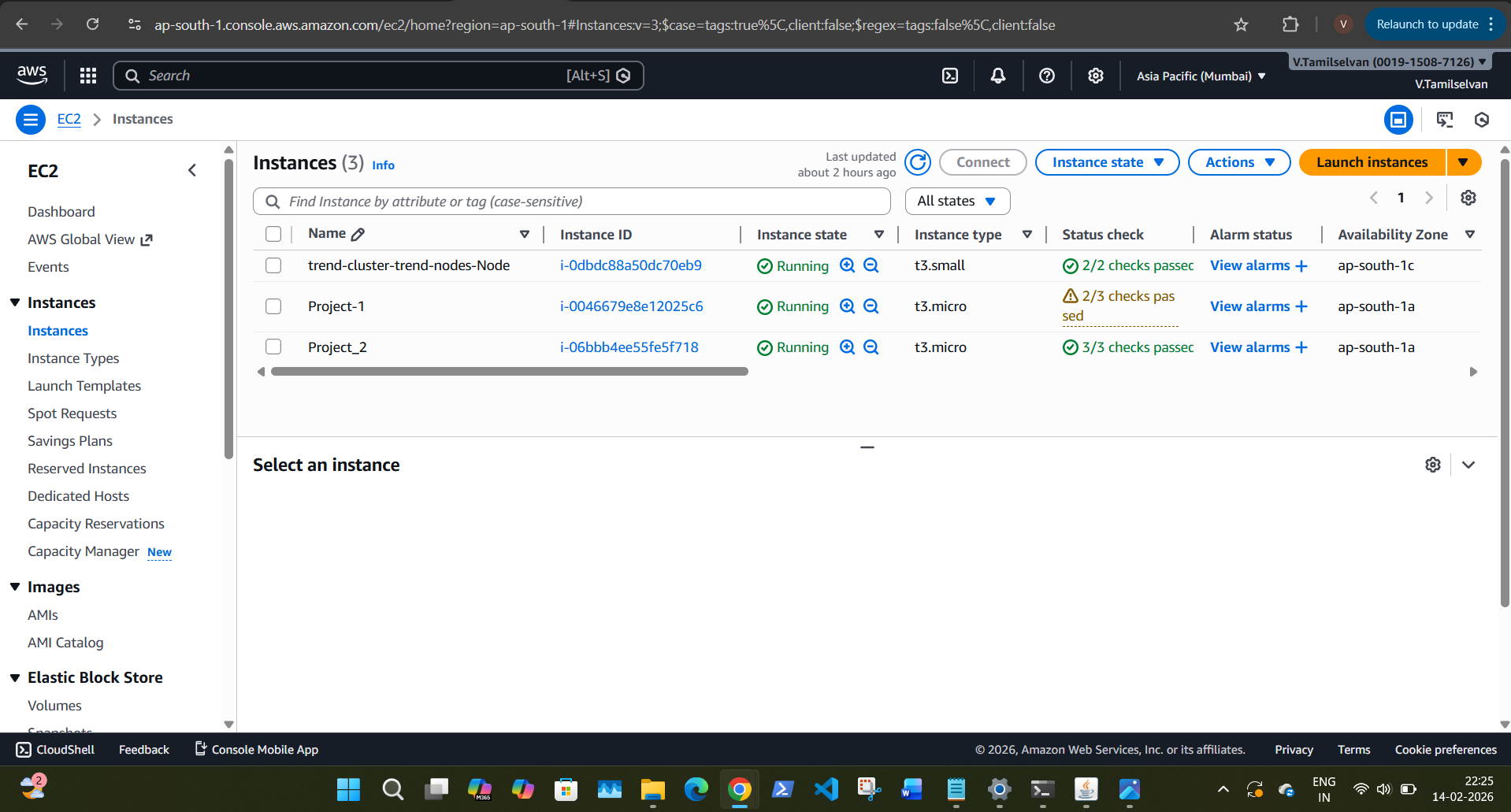


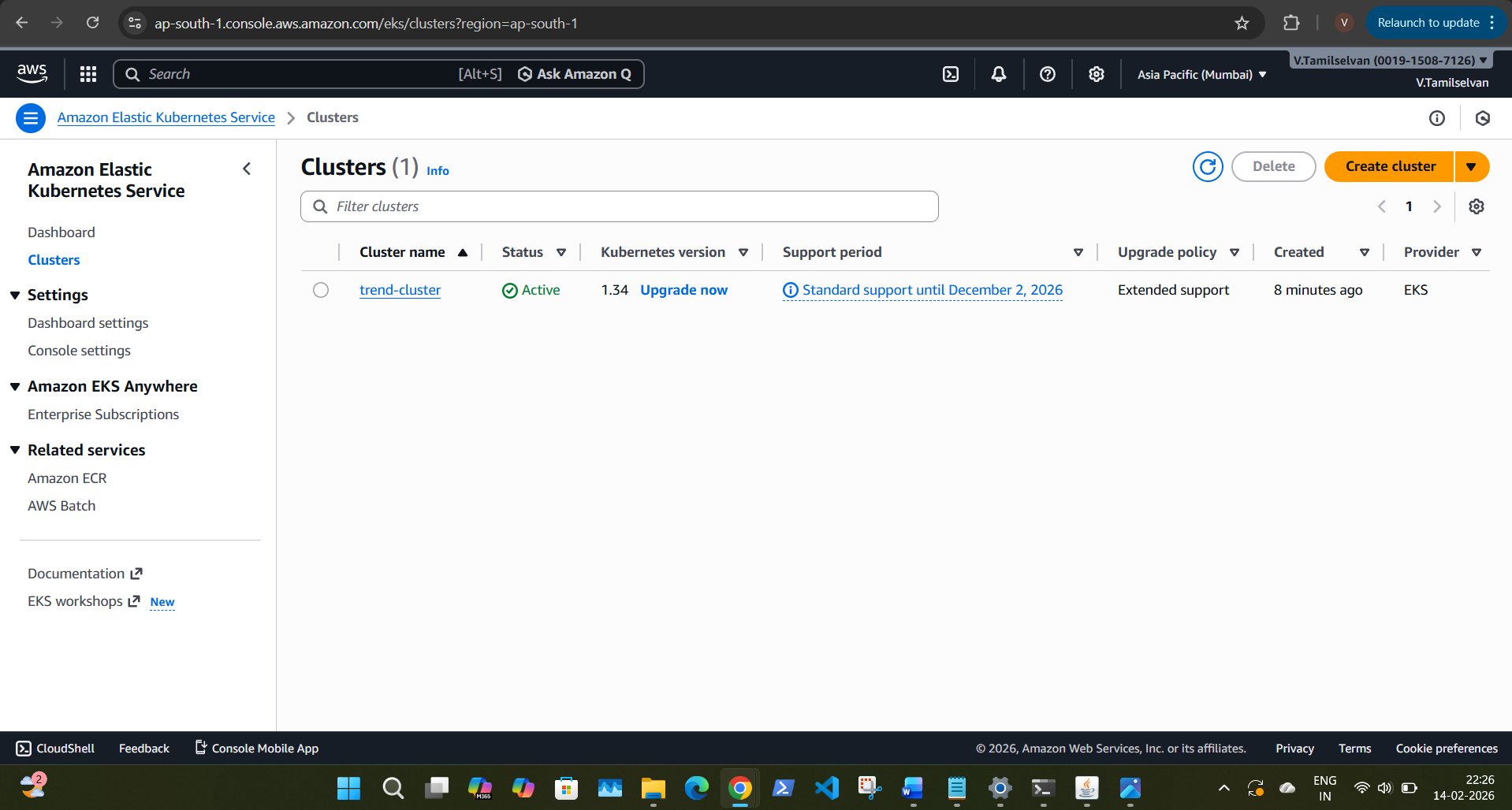


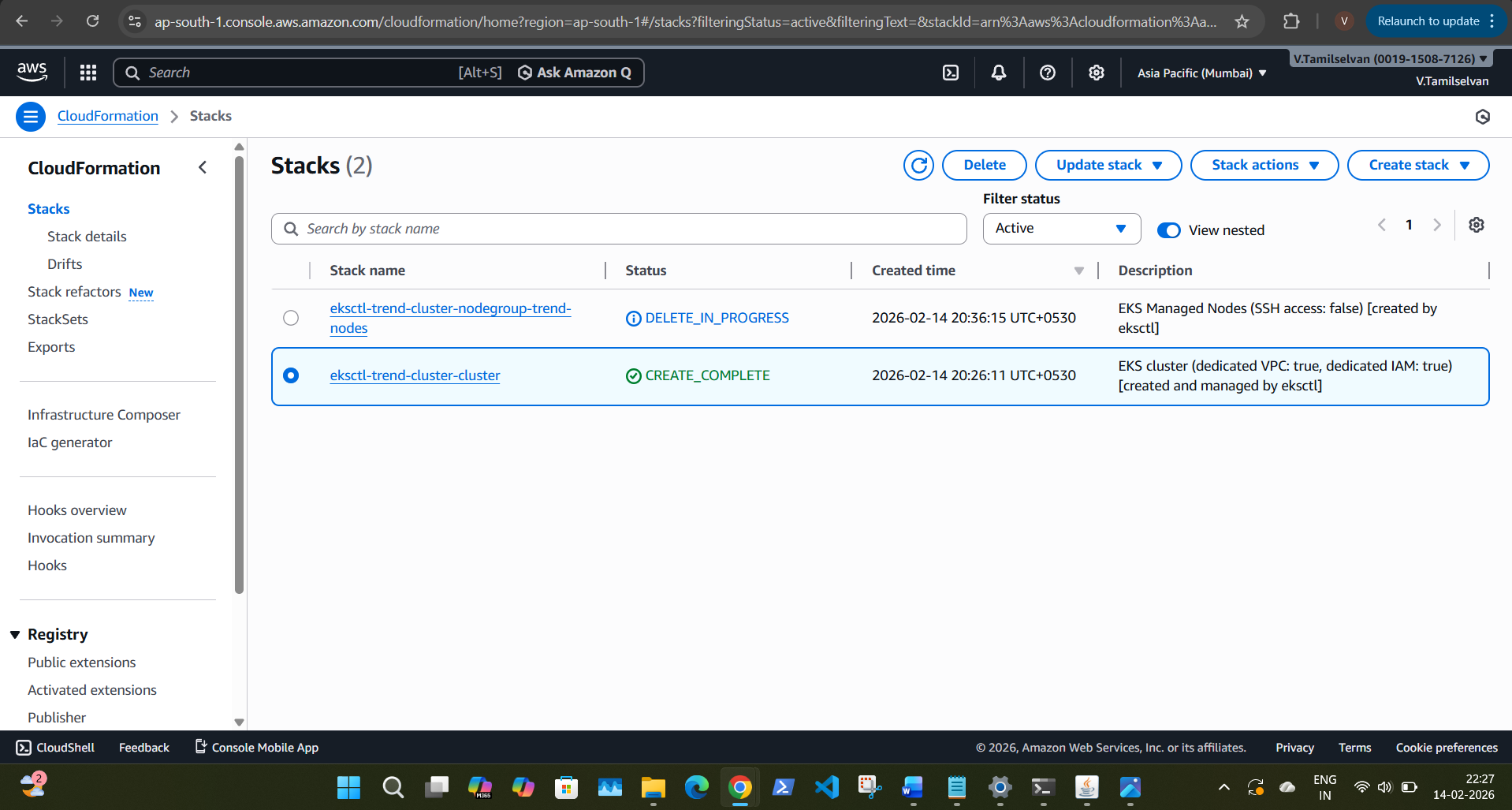


**Add On’s**:









**Conclusion:**

This project successfully demonstrates infrastructure automation using Terraform, CI/CD automation using Jenkins, Kubernetes deployment in AWS EKS, DockerHub integration, and open-source monitoring for production readiness.