# 🚀 Trend React App - CI/CD with Jenkins, Docker, Terraform, EKS, Monitoring

This project demonstrates a production-ready CI/CD pipeline that builds and deploys a React app using:

* **React**
* **Docker**
* **Terraform**
* **AWS EKS**
* **Jenkins**
* **Prometheus + Grafana Monitoring**

## 📁 Project Structure

trend-react-app/  
├── dist/ # React build output  
├── eks-cluster/ # Terraform infrastructure code for EKS  
├── Dockerfile # Container build for React app  
├── Jenkinsfile # Jenkins pipeline configuration  
├── nginx-deployment.yaml # K8s deployment for the app  
├── nginx-service.yaml # K8s LoadBalancer service  
├── prometheus-lb.yaml # Prometheus LB Service  
├── grafana-lb.yaml # Grafana LB Service  
└── README.md

## 🔀 Application Setup Instructions

### ✅ Prerequisites

* AWS CLI configured with EKS permissions
* Terraform installed
* Docker installed
* Jenkins installed (on EC2)
* kubectl installed and configured
* eksctl and helm installed for EKS and monitoring

### 📦 1. Clone the Repository

git clone https://github.com/parthiban4626/trend-react-app.git  
cd trend-react-app

### ☁️ 2. Provision Infrastructure Using Terraform

cd eks-cluster  
terraform init  
terraform apply -auto-approve

Ensure variables such as vpc\_id, subnet\_ids, cluster\_name are configured correctly in terraform.tfvars.

### 🔧 3. Update Kubeconfig

aws eks --region us-east-1 update-kubeconfig --name trend-eks-cluster

### 🚀 4. Deploy the App to EKS

kubectl apply -f nginx-deployment.yaml  
kubectl apply -f nginx-service.yaml

Check service status:

kubectl get svc nginx-service

Look for the EXTERNAL-IP and open it in your browser.

http://abd4a5bdb90be4a25906e86903e991ad-1286492716.us-east-1.elb.amazonaws.com/

## ⚙️ Jenkins CI/CD Pipeline

### 🧹 Jenkinsfile Breakdown

pipeline {  
 agent any  
 environment {  
 CLUSTER\_NAME = "trend-eks-cluster"  
 REGION = "us-east-1"  
 }  
 stages {  
 stage('Clone Repo') {  
 steps {  
 git 'https://github.com/parthiban4626/trend-react-app.git'  
 }  
 }  
 stage('Build Docker Image') {  
 steps {  
 sh 'docker build -t parthiban4626/trend-react-app:latest .'  
 }  
 }  
 stage('Push to DockerHub') {  
 steps {  
 withCredentials([usernamePassword(credentialsId: 'dockerhub-creds', usernameVariable: 'USERNAME', passwordVariable: 'PASSWORD')]) {  
 sh 'echo $PASSWORD | docker login -u $USERNAME --password-stdin'  
 sh 'docker push parthiban4626/trend-react-app:latest'  
 }  
 }  
 }  
 stage('Apply Kubernetes Deployment') {  
 steps {  
 sh 'kubectl apply -f nginx-deployment.yaml'  
 sh 'kubectl apply -f nginx-service.yaml'  
 }  
 }  
 }  
}

### 🔄 Trigger

* Triggered automatically via GitHub Webhook on every push to main.

## 📊 Monitoring Setup

### ✨ Tools Used

* **Helm**: For installing charts
* **Prometheus**: For scraping and alerting
* **Grafana**: For dashboards

### ⚖️ Installation Steps

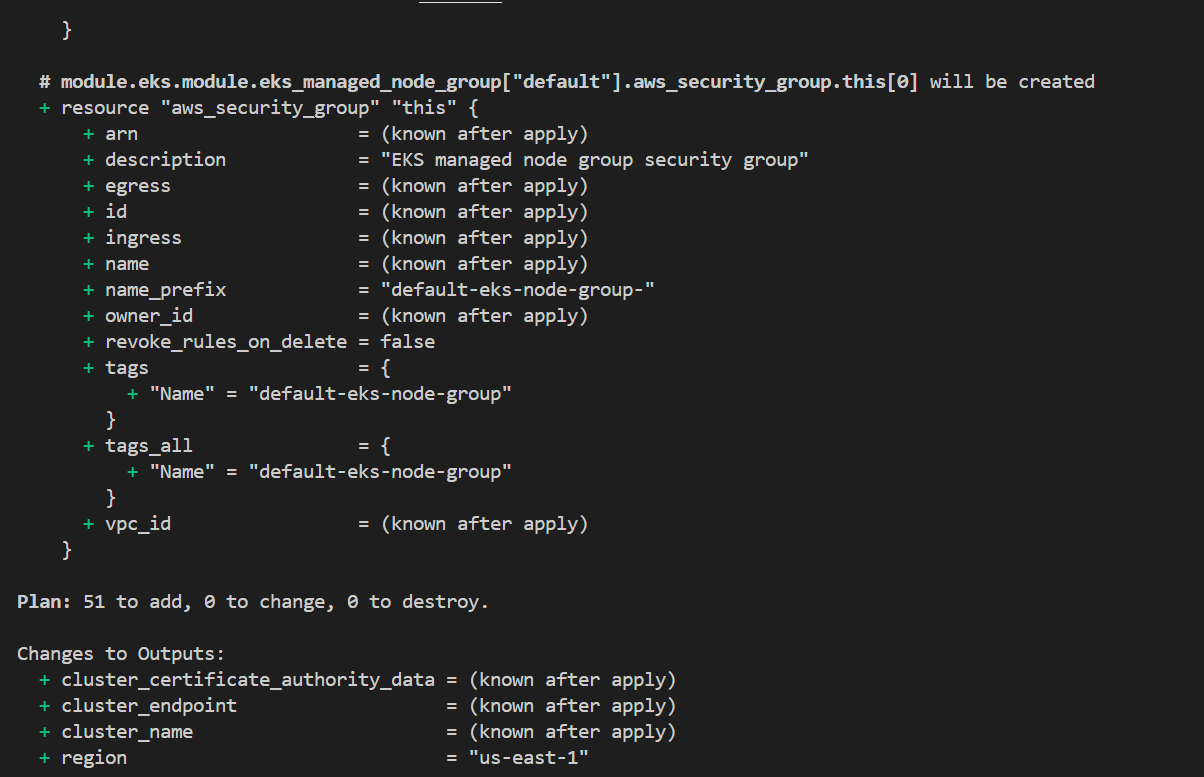
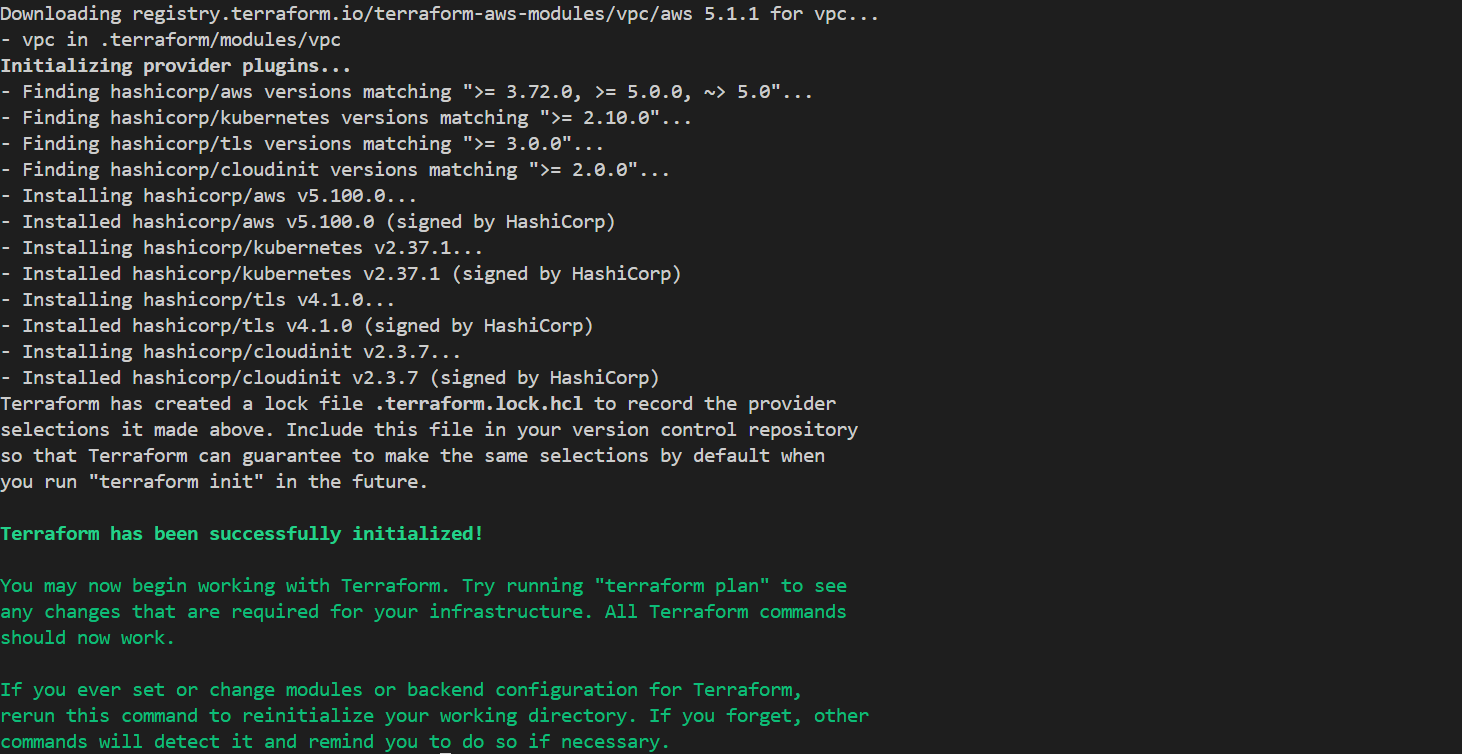
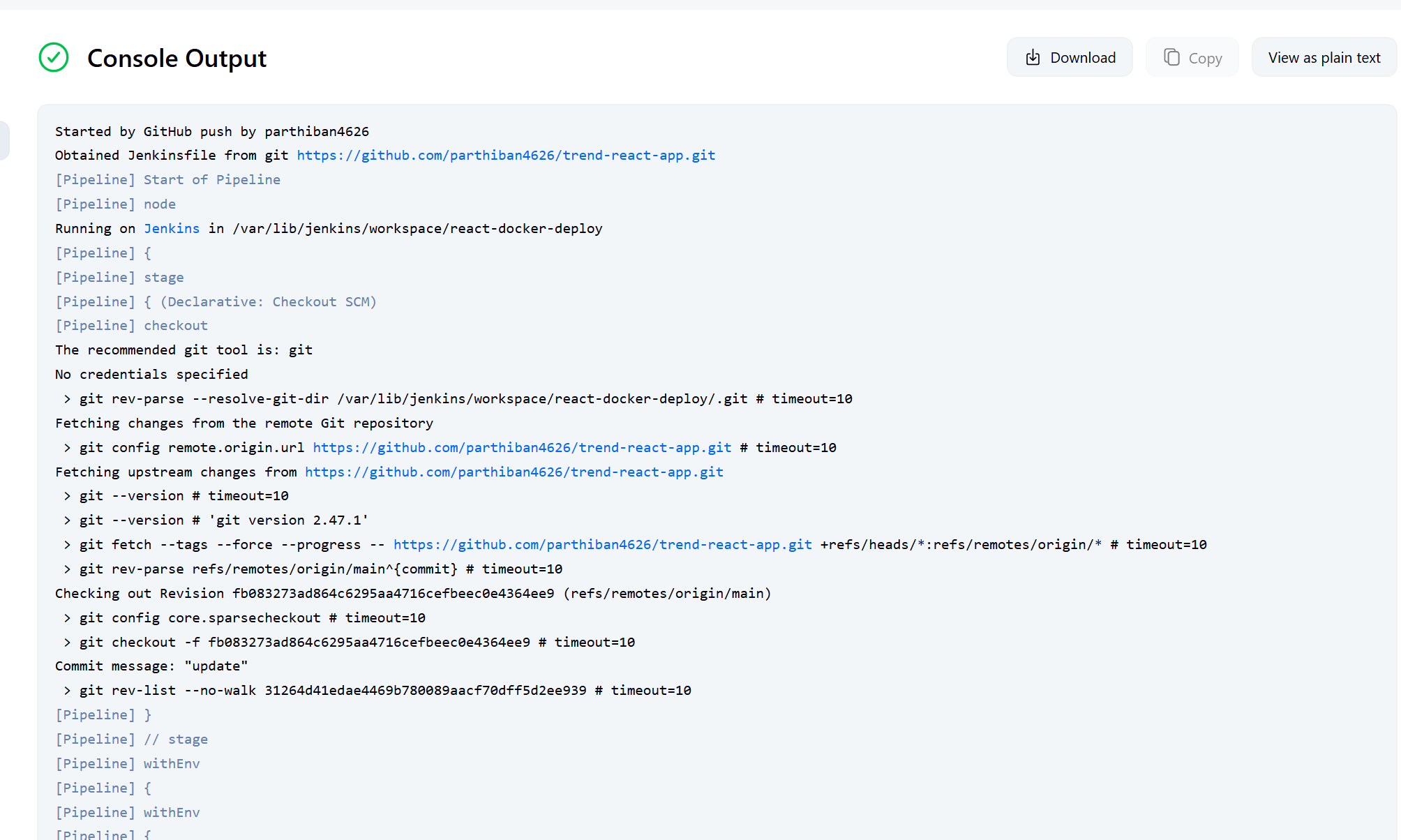
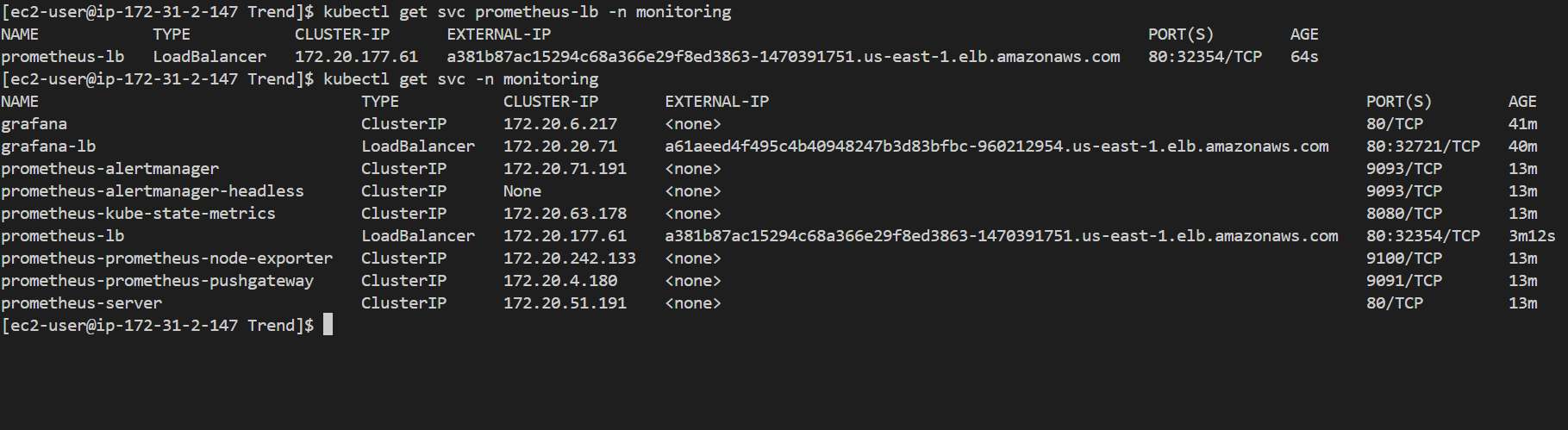
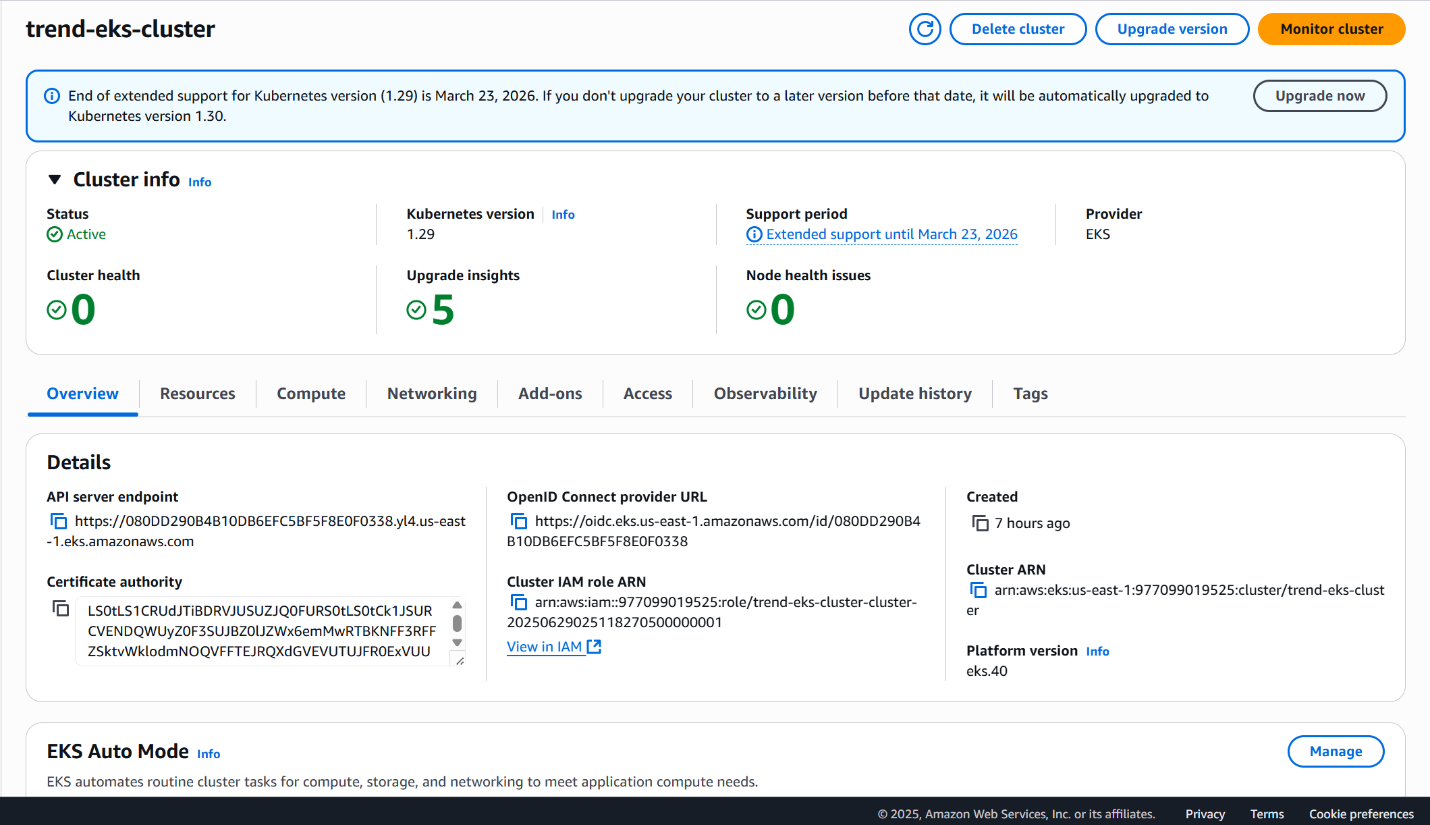
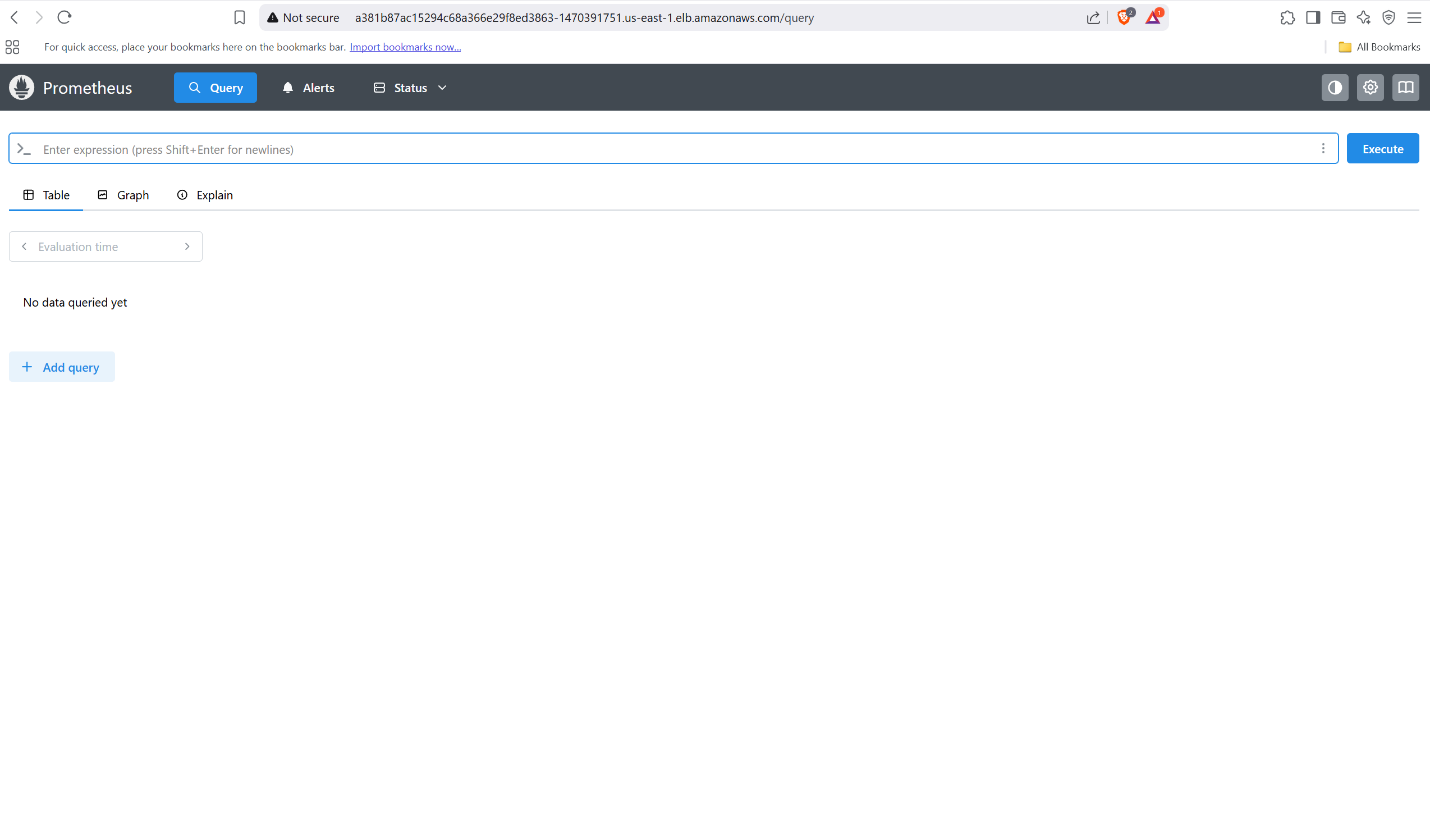
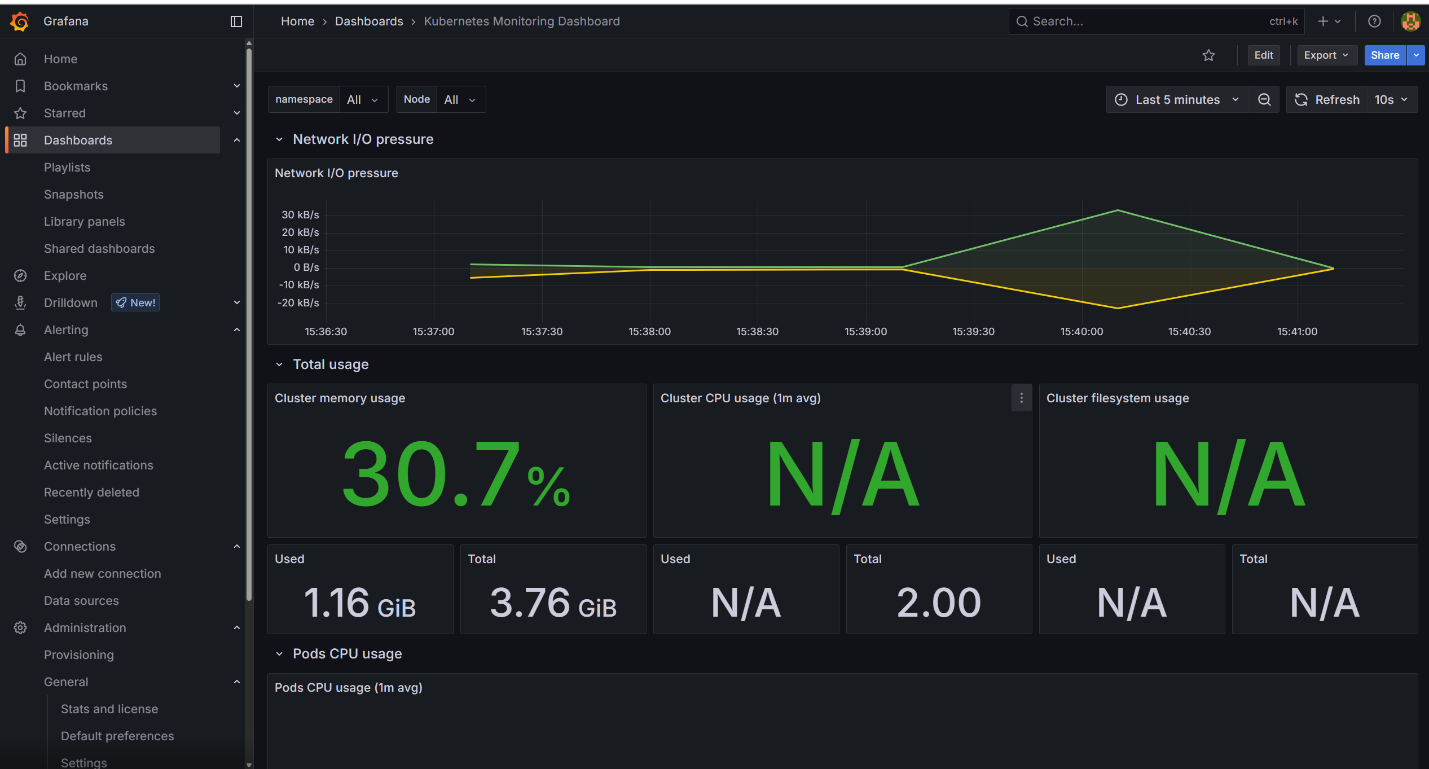
kubectl create ns monitoring  
  
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts  
helm repo update  
  
helm install prometheus prometheus-community/prometheus -n monitoring  
  
helm install grafana prometheus-community/grafana -n monitoring  
  
kubectl apply -f grafana-lb.yaml  
kubectl apply -f prometheus-lb.yaml

### 🔗 Access

kubectl get svc -n monitoring

Use the EXTERNAL-IP of grafana-lb and prometheus-lb to access UI dashboards.

## 📷 Screenshots (Attach in submission)

* Terraform provisioning
* Jenkins Pipeline execution
* EKS cluster and service
* Grafana and Prometheus dashboards

## 👤 Author

* GitHub: [parthiban4626](https://github.com/parthiban4626)