SRE Mini Project – AWS EKS with Terraform, Prometheus, Grafana, and HPA

# 1. Project Overview

This project leverages the AWS Free Tier to provision a Kubernetes (EKS) cluster using Terraform (IaC). The objective is to deploy and configure Prometheus and Grafana monitoring stacks to visualize the state of the cluster and workloads in real time. Additionally, a Horizontal Pod Autoscaler (HPA) is implemented to verify that application pods automatically scale up or down based on CPU utilization.  
  
Keywords: Kubernetes (EKS), Terraform, AWS VPC/EC2/IAM, Helm, Prometheus, Grafana, HPA  
Deliverables: GitHub repository (code + README), execution and verification screenshots, command reference

# 2. Folder Structure

```  
junbeom-sre-terraform-project/  
├── terraform/ # EKS cluster IaC code  
│ ├── main.tf  
│ ├── variables.tf  
│ └── outputs.tf  
├── kubernetes/ # Kubernetes resources  
│ ├── deployments/  
│ │ └── test-app.yaml  
│ ├── services/  
│ │ └── test-app-service.yaml  
│ └── hpa/  
│ └── test-app-hpa.yaml  
├── helm/ # Prometheus & Grafana Helm values  
│ └── monitoring/  
│ └── values.yaml  
├── docs/  
│ └── setup.md # Setup guide (CloudShell execution)  
└── README.md  
```

# 3. Execution Steps

1. 1. Log in to AWS Console, open CloudShell, and set the region to us-east-1
2. 2. Clone the repository and navigate to the terraform directory
3. 3. Provision EKS cluster using Terraform (terraform init → terraform plan → terraform apply)
4. 4. Configure kubectl to connect to EKS (aws eks update-kubeconfig → kubectl get nodes)
5. 5. Deploy test application (Deployment, Service, HPA manifests)
6. 6. Install Prometheus & Grafana with Helm (kube-prometheus-stack chart)
7. 7. Access Grafana (via port-forward or LoadBalancer IP) and validate metrics dashboards
8. 8. Test HPA: run load-generator to generate CPU load and observe scaling events
9. 9. Cleanup: helm uninstall, kubectl delete, terraform destroy to remove resources

# 4. Screenshot Checklist

- terraform apply summary screen  
- kubectl get nodes / get pods output  
- Grafana dashboard (Pods/Nodes resource graphs)  
- kubectl get hpa -w output (scaling events)  
- kubectl top pods output (CPU usage increase)