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

# 1. 프로젝트 개요

이 프로젝트는 AWS 무료티어를 활용하여 Kubernetes(EKS) 클러스터를 Terraform(IaC)으로 자동화 배포하고, Prometheus와 Grafana 모니터링 스택을 설치하여 클러스터 및 애플리케이션 상태를 실시간으로 관찰하는 것을 목표로 한다. 또한 HPA(Horizontal Pod Autoscaler)를 설정하여 CPU 부하에 따라 애플리케이션 파드가 자동으로 스케일링 되는 것을 검증한다.  
  
핵심 키워드: Kubernetes(EKS), Terraform, AWS VPC/EC2/IAM, Helm, Prometheus, Grafana, HPA  
결과물: GitHub 레포지토리(코드, README), 실행 및 검증 스크린샷, 실행 커맨드 정리

# 2. 폴더 구조

```  
junbeom-sre-terraform-project/  
├── terraform/ # EKS 클러스터 IaC 코드  
│ ├── main.tf  
│ ├── variables.tf  
│ └── outputs.tf  
├── kubernetes/ # 배포 리소스  
│ ├── 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 # 실행 가이드 (CloudShell에서 실행)  
└── README.md  
```

# 3. 실행 절차

1. 1. AWS 콘솔 로그인 후 CloudShell 실행, 리전을 us-east-1로 설정
2. 2. 레포지토리 클론 후 terraform 디렉토리로 이동
3. 3. Terraform으로 EKS 클러스터 배포 (terraform init → terraform plan → terraform apply)
4. 4. kubectl을 통해 EKS 클러스터와 연결 (aws eks update-kubeconfig → kubectl get nodes)
5. 5. 테스트 애플리케이션 배포 (Deployment, Service, HPA 매니페스트 적용)
6. 6. Helm으로 Prometheus & Grafana 설치 (kube-prometheus-stack 차트 사용)
7. 7. Grafana 접속 (port-forward 또는 LoadBalancer IP) 후 메트릭 대시보드 확인
8. 8. HPA 테스트: load-generator로 CPU 부하 발생 후 HPA 동작 확인
9. 9. 정리: helm uninstall, kubectl delete, terraform destroy 실행하여 리소스 제거

# 4. 스크린샷 체크리스트

- terraform apply 완료 화면  
- kubectl get nodes / get pods 출력  
- Grafana 대시보드 (Pods/Nodes 리소스 그래프)  
- kubectl get hpa -w 출력 (스케일링 변화)  
- kubectl top pods 출력 (CPU 사용량 상승 확인)