Final Project Proposal

Project Title: End-to-End DevOps Automation Pipeline on AWS EKS with Terraform, GitHub Actions, and GitOps

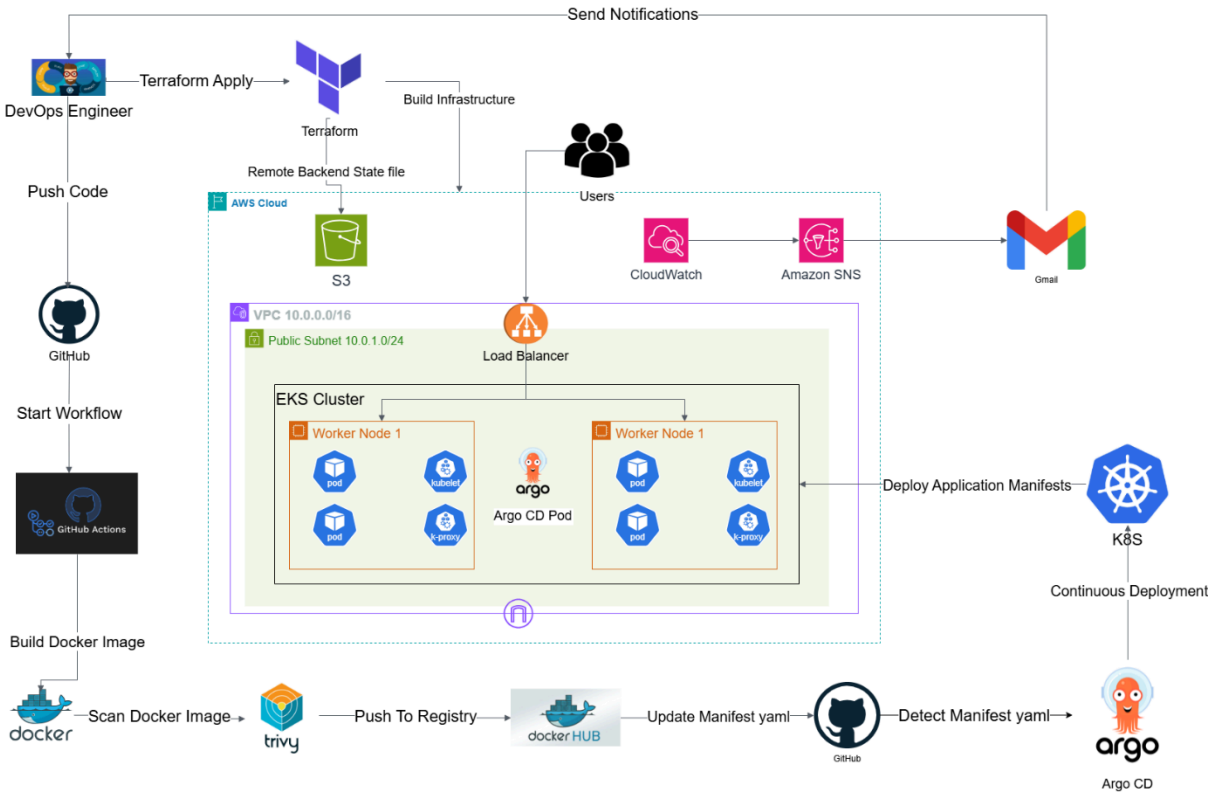
Submission Date: 4/12/2025

1. Project Description

This project aims to design and implement a complete, automated DevOps pipeline for deploying a scalable web application on Amazon Elastic Kubernetes Service (EKS). The pipeline embodies modern DevOps practices, including Infrastructure as Code (IaC), Continuous Integration (CI) via GitHub Actions, Continuous Deployment (CD) via GitOps, and integrated security scanning.

The project will use Terraform to provision the entire AWS foundation (VPC, EKS, S3, CloudWatch, SNS). GitHub Actions will orchestrate the CI pipeline, automating Docker image builds, vulnerability scanning with Trivy, and pushing images to Docker Hub. Crucially, the project implements a GitOps workflow using Argo CD to automatically synchronize and deploy applications to the EKS cluster whenever Kubernetes manifests in Git are updated. The pipeline concludes with automated notifications via Amazon SNS to Gmail.

2. Architecture Diagram



2. Group Members & Roles

Member Name	Role	Responsibilities
Ahmed Mohamed Sabeh Mohamed	Team Leader / Terraform Core Infrastructure Engineer	Design & build AWS infrastructure with Terraform (VPC, EKS, S3, CloudWatch, SNS)
Nader Ibrahim Hamouda Abelkarem	GitHub Actions Engineer	Build and maintain GitHub Actions workflows, integrate Docker Hub, manage CI pipeline
Ahmed Ezzat Hussein Abd Elsalam	Docker & Image Management Engineer	Create Dockerfiles, manage container build process, integrate Trivy scanning
Mina Ibrahim Mikhaiel Gerges	Kubernetes (EKS) Engineer	Develop Kubernetes manifests, install and configure Argo CD for GitOps deployments
Michael Sameih Wilson Garas	Infrastructure & Automation Engineer	Assist with Terraform modules, optimize AWS resource configuration
Mohamed Salah Ismail Mohamed	Monitoring & Documentation Engineer	Configure CloudWatch dashboards and SNS notifications, maintain documentation

3. Team Leader

Ahmed Mohamed Sabeh

4. Objectives

- Automate the provisioning of a secure AWS environment using Terraform with remote S3 backend state management
- Implement a GitHub Actions-based CI pipeline to build, scan, and push Docker images to Docker Hub
- Integrate Trivy for static analysis of Docker images to identify and block vulnerabilities early
- Adopt a GitOps methodology using Argo CD to automate and manage deployments to EKS
- Implement automated notifications via Amazon SNS to Gmail for pipeline events
- Ensure a scalable and resilient application deployment on EKS using managed node groups
- Configure CloudWatch for monitoring and observability

5. Tools & Technologies

Milestone	Description	Deadline
1. Core Infrastructure (Terraform)	Provision VPC, EKS cluster, S3 backend, CloudWatch, and SNS using Terraform	Nov 1, 2025
2. Container & Security Pipeline	Create Dockerfile for the application. Integrate Trivy scanning into GitHub Actions	Nov 3, 2025
3. GitOps Foundation (Argo CD)	Install and configure Argo CD on the EKS cluster. Set up application manifests and sync policies	Nov 10, 2025
4. GitHub Actions CI Pipeline	Build GitHub Actions workflow to: Build → Scan with Trivy → Push to Docker Hub → Update K8s Manifests in Git	Nov 20, 2025
5. Monitoring & Notifications	Configure CloudWatch alerts and SNS notifications to Gmail. Integrate with pipeline events	Nov 25, 2025
6. Validation & Documentation	Perform end-to-end testing. Verify Argo CD auto-deploys changes. Complete project documentation	Dec 4, 2025

Category	**Tools / Services**
Cloud Provider	Amazon Web Services (AWS)
Infrastructure as Code (IaC)	Terraform
Source Code Management	Git, GitHub
CI/CD & Automation	GitHub Actions
Containerization	Docker
Container Registry	Docker Hub
Container Orchestration	Amazon EKS (Kubernetes)
GitOps	Argo CD
Security Scanning	Trivy
Notifications	Amazon SNS
Networking	VPC, Load Balancer
Remote State Management	Amazon S3

6. Milestones & Deadlines

7. Key Performance Indicators (KPIs)

Category	KPI Description
Infrastructure as Code	100% of AWS infrastructure is provisioned and version-controlled via Terraform.
Pipeline Efficiency	CI pipeline (build, scan, push) completes in under 7 minutes.
Security	100% of Docker images are scanned with Trivy; critical vulnerabilities block deployment.
Deployment Automation	All deployments to EKS are managed automatically by Argo CD via Git commits (GitOps).
Reliability	The system achieves zero-downtime deployments and maintains 99.9% service availability.
Monitoring	All critical pipeline events trigger SNS notifications within 2 minutes

8. Conclusion

This project delivers a state-of-the-art, automated DevOps platform on AWS. By leveraging Terraform for IaC, GitHub Actions for CI, and Argo CD for GitOps, it creates a robust, secure, and auditable path from code commit to production. This implementation demonstrates mastery of modern cloud-native tools and establishes a repeatable pattern for efficient and reliable software delivery, fully aligned with the provided architectural vision.