**GitOps Project: Infrastructure and Application Deployment using GitHub Actions**

**Overview**

This project implements a GitOps workflow using GitHub and GitHub Actions for infrastructure provisioning and application deployment. It consists of two repositories:

1. **IaC Repository**: Manages infrastructure as code using Terraform.
2. **Application Repository**: Handles CI/CD pipeline for a Java-based application.

**Architecture Diagram**

A diagram of a software project

AI-generated content may be incorrect.

**Repository Structure**

**1. Infrastructure as Code (IaC) Repository**

* **Branches**:
  + staging: Used by cloud developers for testing infrastructure changes.
  + main: Production-ready infrastructure, reviewed and approved by the architect.
* **Components Provisioned**:
  + VPC
  + Amazon Elastic Container Registry (ECR)
  + Amazon Elastic Kubernetes Service (EKS)
  + Ingress Controller
  + Notifications
* **Workflow Execution**:
  + Developers push changes to staging.
  + Terraform plan is triggered in GitHub Actions.
  + Upon approval, changes are merged into main, triggering Terraform apply.
* **IaC Pipeline Stages**:
  + Checkout source code
  + Terraform setup (init, format, validate)
  + Terraform plan and approval process
  + Terraform apply
  + AWS configuration and EKS setup
  + Install ingress controller

**2. Application Repository**

* **CI/CD Pipeline**:
  + **Testing**:
    - Code checkout
    - Maven test
    - Checkstyle
    - SonarQube static code analysis
  + **Build & Publish**:
    - Build Java application using Maven.
    - Package it as a Docker image.
    - Push the image to Amazon ECR.
  + **Deployment**:
    - Fetch the latest image.
    - Deploy the application to EKS using Helm charts.
* **Application Pipeline Stages**:
  + Code checkout
  + Unit testing and static analysis
  + Build Docker image and push to ECR
  + Configure AWS credentials
  + Deploy to EKS using Helm

**Workflow Execution**

**IaC Pipeline (Terraform)**

* **Terraform Setup**: Initializes Terraform and validates configuration.
* **Plan Execution**: Shows infrastructure changes before applying.
* **Approval Process**: PR is reviewed before merging to main.
* **Terraform Apply**: After merging, GitHub Actions deploys the infrastructure.

**Application Pipeline**

* **Code Checkout & Build**: Fetches code and builds the application.
* **Testing & Quality Checks**: Runs SonarQube, Checkstyle, and unit tests.
* **Dockerization & Push to ECR**: Creates Docker image and stores it in ECR.
* **Deployment via Helm**: Deploys application to EKS cluster.

**Repository Links**

* **IaC Repository**: https://github.com/maheshkoheda/iac-vprofile/actions
* **Application Repository**: <https://github.com/maheshkoheda/vprofile-action/actions>
* **Future Enhancements**
* Implement ArgoCD for enhanced GitOps automation.
* Add security best practices for Terraform and Kubernetes.
* Improve monitoring using Prometheus and Grafana.

**Some Pictures for setup and execution attached**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Author**

[maheshkoheda](https://github.com/maheshkoheda)