Introduction

Welcome to the MultiCloudDevOpsProject! This documentation provides a comprehensive guide to set up and understand the architecture, deployment, and maintenance procedures of the project. The project aims to demonstrate DevOps practices using multiple cloud services such as AWS and OpenShift, along with tools like Terraform, Ansible, Docker, Jenkins, SonarQube, and centralized logging.

Setup Instructions

1. **GitHub Repository Setup:**

* Repository URL: MultiCloudDevOpsProject Repository
* Instructions:
* Clone the repository: git clone <https://github.com/Sara-Fayed/MultiCloudDevOpsProject.git>

1. **Infrastructure Provisioning with Terraform:**

* Terraform scripts are available in the terraform directory.
* Ensure Terraform is installed on your local machine.
* Initialize Terraform: terraform init.
* Review and apply the Terraform scripts to provision AWS resources:
* cd terraform
* terraform apply

1. **Configuration Management with Ansible:**

* Ansible playbooks are available in the Ansible directory.
* Ensure Ansible is installed on your local machine.
* Run Ansible playbooks to configure EC2 instances:
* cd ansible
* ansible-playbook playbook.yml

1. **Containerization with Docker:**

* Dockerfile for building the application image is available in the repository.

1. **Continuous Integration with Jenkins:**

* Follow Jenkins job configuration instructions in the repository for building Docker images on code commits.

1. **Automated Deployment Pipeline:**

* Jenkins pipeline configuration is defined in the Jenkinsfile in the repository.

1. **SonarQube Integration:**
   * SonarQube is used for static code analysis.
   * Ensure SonarQube server is set up and accessible.
   * SonarQube properties should be configured in the project.
2. **Monitoring and Logging:**

* Follow setup instructions for centralized logging on OpenShift for container logs.

1. **AWS Integration:**

* Follow instructions for integrating AWS services provided in the Terraform code.

Architecture Overview

The architecture of the MultiCloudDevOpsProject consists of the following components:

* **GitHub Repository:** Hosts the project codebase and documentation.
* **AWS:** Cloud provider for hosting infrastructure resources such as VPC, Subnets, and EC2 instances.
* **Terraform:** Used for provisioning and managing AWS resources.
* **Ansible:** Configuration management tool for automating software provisioning, configuration, and application deployment.
* **Docker:** Containerization platform for building, shipping, and running applications.
* **Jenkins:** Automation server for continuous integration and continuous deployment (CI/CD) pipelines.
* **SonarQube:** Static code analysis platform for detecting code smells, bugs, and security vulnerabilities.
* **OpenShift:** Container application platform for deploying and managing containerized applications.
* **Centralized Logging:** Utilizes OpenShift for centralized logging of container logs.
* **AWS Services:** Integration with AWS services such as S3 for Terraform backend state and CloudWatch for monitoring.

Troubleshooting Guidelines

1. **Infrastructure Provisioning:** Ensure correct AWS credentials are configured and Terraform is properly initialized.
2. **Ansible Configuration:** Double-check playbook configurations and ensure proper SSH access to EC2 instances.
3. **Docker Build:** Verify Dockerfile for any errors and ensure dependencies are properly installed.
4. **Jenkins Job Configuration:** Check Jenkins job configurations and ensure correct paths and permissions.
5. **Jenkins Pipeline:** Review Jenkinsfile for any syntax errors and ensure proper stage definitions.
6. **SonarQube Integration:** Verify SonarQube server is accessible and project properties are correctly configured.