FluxCD and GitOps Workflow Documentation

Overview

This document provides a detailed guide on setting up and configuring FluxCD as part of a GitOps workflow. It includes the installation steps, configuration of required dependencies, and deployment of applications using FluxCD.

Prerequisites

- A Kubernetes cluster.
- Helm installed on the system.
- Terraform installed for infrastructure provisioning.
- GitHub account with a personal access token.
- Docker installed and running.
- An accessible Git repository for storing FluxCD configurations.

1. Install Dependencies

System Updates and Package Installation

sudo apt update -y

sudo apt-get install -y gnupg software-properties-common

Install HashiCorp Repository

wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | \

sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \

https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \

sudo tee /etc/apt/sources.list.d/hashicorp.list

Install Helm

curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | \

sudo tee /usr/share/keyrings/helm.gpg > /dev/null

sudo apt-get install apt-transport-https --yes

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | \

sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

Install Required Packages

sudo apt update -y

sudo apt install -y git curl helm terraform python3 python3-pip

pip3 install --upgrade pip

pip install ansible-core==2.14.11

pip3 install ansible

2. Install and Configure Docker

sudo apt-get install -y ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

3. Install and Bootstrap FluxCD

curl -s https://fluxcd.io/install.sh | sudo bash

Set up authentication with GitHub:

export GITHUB_TOKEN=<your_github_token>

export GITHUB_USER=<your_github_username>

flux bootstrap github --owner=$GITHUB_USER --repository=fluxcd --branch=master --path=clusters/my-cluster --personal

4. Helm Chart Deployment

Create a Helm chart for deploying a sample application:

CHART_NAME="pizza-chart"

rm -rf $CHART_NAME

helm create $CHART_NAME

Update Chart.yaml:

```
apiVersion: v2
name: pizza-chart
description: A Helm chart for deploying Pizza Frontend and Backend
version: 1.0.0
appVersion: "1.0"
```

Update values.yaml:

```
replicaCount: 1

image:
  frontend: <frontend_image>
  backend: <backend_image>

environment: Development

service:
  frontend:
    port: 80
  backend:
    port: 8080

ingress:
  enabled: true
  host: <your_domain>
  path:
    frontend: /frontend
```

```
  backend: /backend
```

Deploy the Helm chart:

```
helm install pizza-app ./$CHART_NAME
```

5. Configure FluxCD to Manage Helm Releases

Create fluxcd-helmrelease.yaml:

```
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: blazing-pizza
spec:
  interval: 5m
  chart:
    spec:
      chart: ./charts/blazing-pizza
      sourceRef:
        kind: GitRepository
        name: flux-system
  valuesFrom:
  - kind: ConfigMap
    name: blazing-pizza-values
```

Commit and push the changes to the Git repository:

```
git add .
git commit -m "Add Blazing Pizza Helm chart to FluxCD"
git push origin master
```

6. Configure Notifications

Create fluxcd-alert.yaml:

```
apiVersion: notification.toolkit.fluxcd.io/v1beta1
kind: Alert
metadata:
  name: blazing-pizza-alerts
  namespace: flux-system
spec:
  eventSeverity: info
  eventSources:
    - kind: HelmRelease
      name: blazing-pizza
  providerRef:
    name: slack
```

7. Verify FluxCD Deployment

Check if FluxCD components are running:

```
kubectl get pods -n flux-system
```

Check Helm releases:

```
flux get helmreleases --all-namespaces
```

This completes the setup and configuration of FluxCD with the GitOps workflow.