**Crossplane and Helm Documentation**

**1. Introduction**

This document outlines how Crossplane and Helm are used to provision and manage infrastructure in the project. Crossplane enables declarative infrastructure management within Kubernetes, while Helm is used for package management and deployment.

**2. Crossplane Setup**

Crossplane extends Kubernetes capabilities to provision cloud resources like AWS VPCs and EKS clusters.

**2.1 Installing Crossplane**

Crossplane can be installed using Helm:

helm repo add crossplane-stable https://charts.crossplane.io/stable

helm repo update

helm install crossplane --namespace crossplane-system crossplane-stable/crossplane

**Verify installation:**

kubectl get pods -n crossplane-system

**2.2 Configuring AWS Provider**

Crossplane requires a provider to manage AWS resources. Install the AWS provider:

kubectl crossplane install provider crossplane/provider-aws:v0.37.0

**Verify installation:**

kubectl get providers

**Create a ProviderConfig to authenticate Crossplane with AWS:**

apiVersion: aws.crossplane.io/v1beta1

kind: ProviderConfig

metadata:

name: aws-config

spec:

credentials:

source: Secret

secretRef:

namespace: crossplane-system

name: aws-creds

key: creds

**Create the secret aws-creds with AWS credentials:**

kubectl create secret generic aws-creds -n crossplane-system \

--from-file=creds=./aws-creds.txt

**3. Provisioning Infrastructure with Crossplane**

**3.1 Creating a VPC**

Define a VPC using Crossplane:

apiVersion: ec2.aws.crossplane.io/v1beta1

kind: VPC

metadata:

name: my-vpc

spec:

providerConfigRef:

name: aws-config

forProvider:

cidrBlock: 10.0.0.0/16

enableDnsSupport: true

enableDnsHostnames: true

deletionPolicy: Delete

Apply the manifest:

kubectl apply -f vpc.yaml

**3.2 Creating an EKS Cluster**

Define an EKS cluster:

apiVersion: eks.aws.crossplane.io/v1beta1

kind: Cluster

metadata:

name: my-eks-cluster

spec:

providerConfigRef:

name: aws-config

forProvider:

region: us-west-2

roleArn: arn:aws:iam::123456789012:role/EKSClusterRole

resourcesVpcConfig:

securityGroupIds:

- sg-12345678

subnetIds:

- subnet-abcdef01

- subnet-abcdef02

version: "1.27"

deletionPolicy: Delete

Apply the manifest:

kubectl apply -f eks-cluster.yaml

**3.3 Creating an EKS Node Group**

Define a node group for the EKS cluster:

apiVersion: eks.aws.crossplane.io/v1beta1

kind: NodeGroup

metadata:

name: my-node-group

spec:

providerConfigRef:

name: aws-config

forProvider:

clusterName: my-eks-cluster

scalingConfig:

minSize: 1

maxSize: 3

desiredSize: 2

instanceTypes:

- t3.medium

subnets:

- subnet-abcdef01

- subnet-abcdef02

deletionPolicy: Delete

Apply the manifest:

kubectl apply -f eks-nodegroup.yaml

**4. Helm for Kubernetes Package Management**

**4.1 Installing Helm**

Helm is used to manage Kubernetes applications. Install Helm:

curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash

Verify installation:

Helm version

**4.2 Deploying an Application with Helm**

Example: Deploying NGINX using Helm:

helm repo add bitnami https://charts.bitnami.com/bitnami

helm install my-nginx bitnami/nginx

Verify deployment:

kubectl get pods

**4.3 Managing Helm Releases**

List installed releases:

helm list

Upgrade an existing release:

helm upgrade my-nginx bitnami/nginx

Uninstall a release:

helm uninstall my-nginx