############################# Pre-requisite ####################################

# # **Management Kubernetes cluster**

* This is the management cluster and is used to host Crossplane and ArgoCD
* This cluster can be provisioned by any approach, on cloud (EKS, AKS) or using Kind or Minikube
* In this hands-on we will create an EKS cluster to host Crossplane. You can refer to the Create EKS-Using-eksctl or Create-EKS-Using-UI file for instructions on creating an EKS cluster.

A screenshot of a computer

Description automatically generated

# # **Crossplane installed on Management Kubernetes Cluster**

* You can refer to the Intall-Crossplane-on-EKS file.

#################### Provision EKS on AWS using Crossplane ##################

https://github.com/hoanglecao/argocd-crossplane/tree/main/infrastructure/crossplane-config/aws

1. **Install provider-family-aws**  
   Create a provider in yaml file. Refer to [upbound/provider-family-aws@v1.3.0 | Upbound Marketplace](mailto:upbound/provider-family-aws@v1.3.0%20|%20Upbound%20Marketplace)

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Apply this configuration with ‘kubectl’: kubectl apply -f 1-provider-aws.yaml





1. **Create an AWS secret**  
   Make sure you have an AWS user. Create a credentials file in the format:  
   A black screen with white text

   Description automatically generated

Run command: kubectl create secret generic aws-secret -n crossplane-system --from-file=creds=aws-credentials.txt



A screenshot of a computer

Description automatically generated

1. **Create provider config for AWS**

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 2-provider-aws-config.yaml



1. **Install provider-aws-ec2 provider**This provider containscomponents such as a VPC, subnets, and route tables.

A screenshot of a computer

Description automatically generated

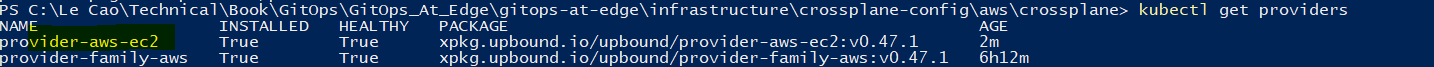
A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 3-provider-aws-ec2.yaml



Check and wait until the provider become healthy.



1. **Create VPC**

Refer to[VPC - upbound/provider-aws-ec2@v1.3.1 | Upbound Marketplace](https://marketplace.upbound.io/providers/upbound/provider-aws-ec2/v1.3.1/resources/ec2.aws.upbound.io/VPC/v1beta1)

**A screen shot of a computer

Description automatically generated**

Suggest using a prefix for all your components. For example, let’s name it dev-main vpc. This is useful in case you host multiple environments such as dev, staging, and production in a single AWS account. Using an environment prefix will help you better manage your resources.

Apply it: kubectl apply -f 0-vpc.yaml

****

A close-up of a computer screen

Description automatically generated

1. **Create Internet gateways**

A screenshot of a computer program

Description automatically generated

Apply it: kubectl apply -f 1-igw.yaml



A screenshot of a computer

Description automatically generated

1. **Create Subnets**

Create four subnets in two different availability zones. The first will be the private zone.

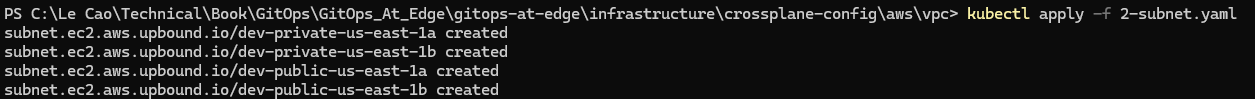
A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Apply it: kubectl apply -f 2-subnet.yaml



A screenshot of a computer

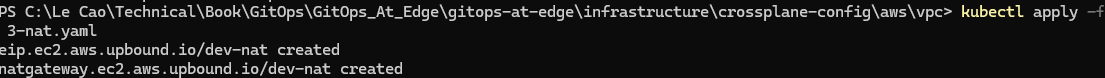
Description automatically generated

1. **Create NATGateway**

**A screenshot of a computer program

Description automatically generated**

Apply it: kubectl apply -f 3-nat.yaml



**A screenshot of a computer

Description automatically generated**

**A close-up of a computer screen

Description automatically generated**

1. **Create Routes**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

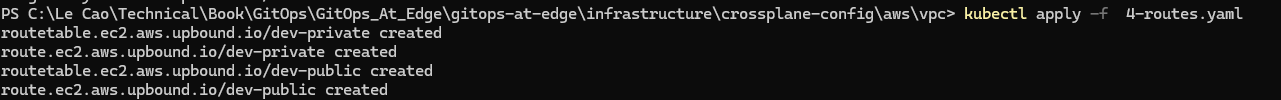
A screen shot of a computer program

Description automatically generated

A computer screen with text

Description automatically generated

Apply it: kubectl apply -f 4-routes.yaml



A close-up of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Install provider-aws-iam**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 4-provider-aws-iam.yaml



1. **Install provider-aws-eks**

A screenshot of a computer

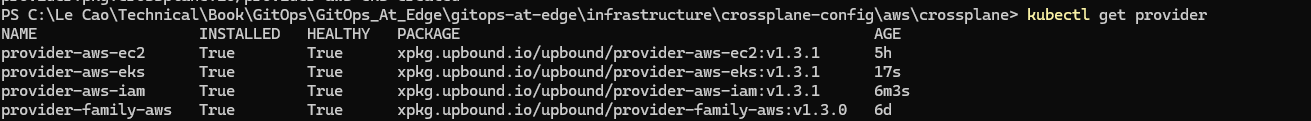
Description automatically generated

A screen shot of a computer

Description automatically generated

Apply: kubectl apply -f 5-provider-aws-eks.yaml





1. **Create Role for cluster**

A screenshot of a computer program

Description automatically generated

Apply: kubectl apply -f 0-eks-iam-role.yaml



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Create EKS**

A screenshot of a computer program

Description automatically generated

kubectl apply -f 1-eks.yaml



Although the command says that the Cluster created but in AWS UI might be in Creating status. Need to wait until it completed.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

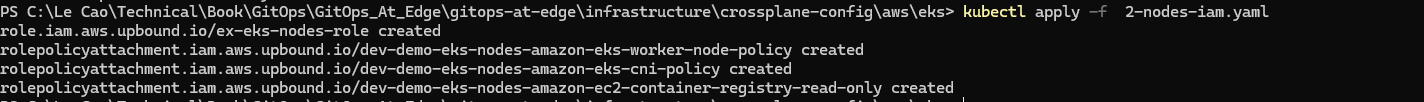
1. **Create Role for Node Groups**

One reason why node group needs its own IAM role is to be able to join your EKS cluster. We can share this role among different node groups. Like the EKS, we need to create an IAM role with assumeRolePolicy. But in this case, the principal service will be EC2.

A screen shot of a computer program

Description automatically generated

Apply it: kubectl apply -f 2-nodes-iam.yaml

****

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Create NodeGroup**

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Apply it: kubectl apply -f 3-nodes.yaml



###################Provision AKS on Azure using Crossplane ##################

https://github.com/hoanglecao/argocd-crossplane/tree/main/infrastructure/crossplane-config/azure

1. **Install Azure Provider**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 1-provider-family-azure.yaml



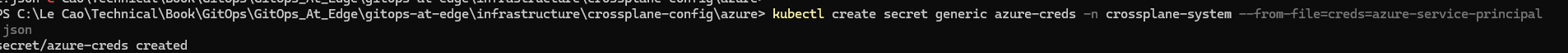
1. **Create an Azure secret**

Create a azure-service-principal.json credentials file in the format:

A computer screen with green text

Description automatically generated

Run command: kubectl create secret generic azure-creds -n crossplane-system --from-file=creds= azure-service-principal.json



A screenshot of a computer

Description automatically generated

1. **Install Azure Provider Config**

A screenshot of a computer program

Description automatically generated

Apply it: kubectl apply -f 2-provider-azure-config.yaml

1. **Create Resource Group**

A screenshot of a computer program

Description automatically generated

Apply it: kubectl apply -f resource-group.yaml



A close-up of a computer screen

Description automatically generated

1. **Install provider-azure-network**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 3-provider-azure-network.yaml



1. **Create Virtual Network**

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 2-virtual-network.yaml



A screenshot of a chat

Description automatically generated

1. **Create Subnet**

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 3-subnet.yaml



A white screen with text

Description automatically generated with medium confidence

1. **Install provider-azure-containerservice**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 5-provider-azure-containerservice.yaml

1. **Create AKS**

A screen shot of a computer

Description automatically generated

Apply it: kubectl apply -f 4-aks.yaml