Terraform for AKS

Project directory should look like following

project

└── terraform

├── main.tf

└── variables.tf

In main.tf, define the AKS cluster and required configurations:

provider "azurerm" {

features {}

}

resource "azurerm\_resource\_group" "example" {

name = "dev-resources"

location = "East US"

}

resource "azurerm\_kubernetes\_cluster" "example" {

name = "dev-aks-cluster"

location = azurerm\_resource\_group.example.location

resource\_group\_name = azurerm\_resource\_group.example.name

dns\_prefix = "dev-aks-dns"

default\_node\_pool {

name = "default"

node\_count = 1

vm\_size = "Standard\_DS2\_v2"

}

tags = {

Environment = "Development "

}

}

In variables.tf, define variables as following

variable "client\_id" {}

variable "client\_secret" {}

variable "tenant\_id" {}

Run Terraform commands:

terraform init

terraform plan # Review the plan

terraform apply # Apply changes to create AKS cluster

**CI/CD Pipeline using GitHub Actions:**

1. Create a .github/workflows directory:

Inside this directory, create a YAML file for your workflow, for example, deploy.yml:

name: CI/CD for AKS Deployment

on:

push:

branches:

- main # Change this according to your branch

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Login to Azure

uses: azure/login@v1

with:

creds: ${{ secrets.AZURE\_CREDENTIALS }}

- name: Set up kubectl

uses: azure/setup-kubectl@v1

with:

version: 'latest'

- name: Azure CLI script

run: |

kubectl apply -f path/to/kubernetes-configs # Apply your Kubernetes configs here

Configurations that needed to be done

Set up Azure Service Principal and GitHub Secrets:

Create an Azure Service Principal with the necessary permissions and obtain its credentials (client ID, client secret, tenant ID).

Add these credentials as secrets in your GitHub repository (Settings -> Secrets).

Commit the workflow file:

Commit the .github/workflows/deploy.yml file to your GitHub repository.

This workflow will trigger on every push to the specified branch, login to Azure using the provided service principal credentials, set up kubectl, and apply the Kubernetes configuration to the AKS cluster.