DevOps Project

**Overview**

This project provisions a Linux VM on Azure using Terraform, configures it with Ansible, and automates the entire pipeline through Jenkins running in Docker on Windows.

1. **Install Required Tools on Windows**

* Docker Desktop
* Git Bash or PowerShell
* Azure CLI
* Chocolatey (https://chocolatey.org/install)
* Terraform (choco install terraform)
* SSH (ssh-keygen)

1. **Generate and Copy SSH Keys**

* ssh-keygen -t rsa -b 4096 -f C:\Users\pc\.ssh\id\_rsa
* copy C:\Users\pc\.ssh\id\_rsa.pub E:\Git\devops-project\terraform\id\_rsa.pub

1. **Create Azure Service Principal**

* az login
* az ad sp create-for-rbac --name jenkins-sp --role Contributor --scopes /subscriptions/f32912ed-d29f-4020-8bc7-0d5b7cf7e7f6

1. **Start Jenkins in Docker**

docker run -d --name jenkins -p 8080:8080 -p 50000:50000 -v jenkins\_home:/var/jenkins\_home -v "E:\Git\devops-project":/workspace jenkins/jenkins:lts

1. **Install Jenkins Plugins**

* Pipeline
* SSH Agent
* AnsiColor
* Credentials Binding

1. **Install Terraform & Ansible in Jenkins Container**

* docker exec -u root -it jenkins bash
* apt update && apt install -y curl unzip
* curl -o terraform.zip https://releases.hashicorp.com/terraform/1.6.6/terraform\_1.6.6\_linux\_amd64.zip
* unzip terraform.zip
* mv terraform /usr/local/bin/
* terraform version
* apt install -y software-properties-common
* apt install -y ansible
* ansible –version

1. **Store Azure Credentials in Jenkins**

* apt-get install azure-cli
* az ad sp create-for-rbac --name jenkins-sp --role Contributor --scopes /subscriptions/f32912ed-d29f-4020-8bc7-0d5b7cf7e7f6
* Store like this for groovy:

ARM\_CLIENT\_ID = 'c58d071f-7512-46a8-8538-41b05534e5b0'

ARM\_CLIENT\_SECRET = '0HD8Q~1B54~DW6rHKODZmXXvP\_UY5UbSRTzrcOu'

ARM\_SUBSCRIPTION\_ID = 'f32912ed-d29f-4020-8bc7-0d5b7cf7e7f6'

ARM\_TENANT\_ID = 'a0b05d0e-54f3-4dbb-bdbe-9ec3982b25e2'

1. **Add Private Key as Jenkins SSH Credential**

* Kind: SSH Username with Private Key
* ID: ansible-key
* Username: azureuser
* Paste contents of id\_rsa

1. **Terraform Files:**

**main.tf:**

provider "azurerm" {

features {}

subscription\_id = "f32912ed-d29f-4020-8bc7-0d5b7cf7e7f6"

}

resource "azurerm\_resource\_group" "main" {

name = var.resource\_group\_name

location = var.location

}

resource "azurerm\_virtual\_network" "vnet" {

name = "devops-vnet"

address\_space = ["10.0.0.0/16"]

location = var.location

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

}

resource "azurerm\_subnet" "subnet" {

name = "devops-subnet"

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

virtual\_network\_name = azurerm\_virtual\_network.vnet.name

address\_prefixes = ["10.0.1.0/24"]

}

resource "azurerm\_network\_security\_group" "nsg" {

name = "devops-nsg"

location = var.location

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

security\_rule {

name = "AllowSSH"

priority = 1001

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "22"

source\_address\_prefix = "\*"

destination\_address\_prefix = "\*"

}

security\_rule {

name = "AllowHTTP"

priority = 1002

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "80"

source\_address\_prefix = "\*"

destination\_address\_prefix = "\*"

}

}

resource "azurerm\_subnet\_network\_security\_group\_association" "subnet\_nsg\_assoc" {

subnet\_id = azurerm\_subnet.subnet.id

network\_security\_group\_id = azurerm\_network\_security\_group.nsg.id

}

resource "azurerm\_public\_ip" "public\_ip" {

name = "devops-ip"

location = var.location

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

allocation\_method = "Static"

sku = "Standard"

}

resource "azurerm\_network\_interface" "nic" {

name = "devops-nic"

location = var.location

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

ip\_configuration {

name = "internal"

subnet\_id = azurerm\_subnet.subnet.id

private\_ip\_address\_allocation = "Dynamic"

public\_ip\_address\_id = azurerm\_public\_ip.public\_ip.id

}

}

resource "azurerm\_linux\_virtual\_machine" "vm" {

name = "devops-vm"

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

location = var.location

size = var.vm\_size

admin\_username = var.admin\_username

network\_interface\_ids = [azurerm\_network\_interface.nic.id]

admin\_ssh\_key {

username = var.admin\_username

public\_key = file("${path.module}/id\_rsa.pub")

}

os\_disk {

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

}

output "public\_ip" {

value = azurerm\_public\_ip.public\_ip.ip\_address

}

**variables.tf:**

variable "resource\_group\_name" {

default = "project"

}

variable "location" {

default = "East US"

}

variable "vm\_size" {

default = "Standard\_B1s"

}

variable "admin\_username" {

default = "azureuser"

}

variable "windows\_username" {

description = "Windows username for path resolution"

default = "pc"

}

1. Jenkinsfile

pipeline {

agent any

environment {

TF\_DIR = "/workspace/terraform"

INVENTORY\_FILE = "inventory"

ARM\_CLIENT\_ID = 'c58d071f-7512-46a8-8538-41b05534e5b0'

ARM\_CLIENT\_SECRET = '0HD8Q~1B54~D-W6rHKODZmXXvP\_UY5UbSRTzrcOu'

ARM\_SUBSCRIPTION\_ID = 'f32912ed-d29f-4020-8bc7-0d5b7cf7e7f6'

ARM\_TENANT\_ID = 'a0b05d0e-54f3-4dbb-bdbe-9ec3982b25e2'

}

stages {

stage('Terraform Init') {

steps {

dir("/workspace/terraform") {

sh 'terraform init'

}

}

}

stage('Terraform Apply') {

steps {

dir("/workspace/terraform") {

sh 'terraform apply -auto-approve'

}

}

}

stage('Get IP') {

steps {

script {

env.VM\_IP = sh(script: "cd ${env.TF\_DIR} && terraform output -raw public\_ip", returnStdout: true).trim()

echo "Extracted Public IP: ${env.VM\_IP}"

writeFile file: "${env.INVENTORY\_FILE}", text: """[web]

${env.VM\_IP} ansible\_user=azureuser ansible\_ssh\_private\_key\_file=/root/.ssh/id\_rsa

"""

}

}

}

stage('Run Ansible') {

steps {

sshagent(['ansible\_ssh\_key']) {

sh '''

ANSIBLE\_HOST\_KEY\_CHECKING=False \

ansible-playbook -i inventory /workspace/ansible/install\_web.yml

'''

}

}

}

stage('Verify') {

steps {

sh 'echo "Verifying deployed web app"'

sh "curl http://${env.VM\_IP}"

}

}

}

}

1. Ansible Playbook

- hosts: all

become: true

tasks:

- name: Install Apache

apt:

name: apache2

state: present

update\_cache: yes

- name: Start Apache

systemd:

name: apache2

state: started

enabled: true

- name: Deploy index.html

copy:

src: ../app/index.html

dest: /var/www/html/index.html

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Project</title>

</head>

<body>

<h1>Hello Devops Automated Pipeline Project</h1>

<h4>By Syed Haris Ahmed</h4>

</body>

</html>

1. Final Output





















