Terraform :

Tag v01

#### S1 : Create a VM in Azure and prepare for the app

- Take care about the sshkey and network security group

#### S2 : Login (ssh) into the machine using the key

#### S3: Prepare the machine for python and flask

sudo apt update && sudo apt upgrade -y

sudo apt install python3 python3-pip -y

pip3 install flask

sudo apt install git -y

Tag v02

#### S4: Create app dir and flask app as follows:

mkdir flask\_app && cd flask\_app

vi app.py

**Type in the following code:**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

return 'Hello, World!'

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=5000)

**Save and Run the application**

python3 app.py &

**verify it**

curl http://127.0.0.1:5000

#### S5: Download Hashicorp ‘s key install it and get terraform

wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) 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

sudo apt update && sudo apt install terraform

#### S6: create the dir for the tf programs and create a main.tf file

mkdir ~/tf && cd ~/tf

vi main.tf

provider "azurerm" {

features {}

subscription\_id = "<Your subscription id here>"

}

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

name = "FlaskDemoGroup"

location = "centralindia"

}

resource "azurerm\_virtual\_network" "example" {

name = "FlaskVNet"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.example.location

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

}

resource "azurerm\_subnet" "example" {

name = "FlaskSubnet"

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

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

address\_prefixes = ["10.0.1.0/24"]

}

# Create a Public IP

resource "azurerm\_public\_ip" "example" {

name = "FlaskPublicIP"

location = azurerm\_resource\_group.example.location

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

allocation\_method = "Dynamic" # Dynamic IP address (can also be "Static" if you prefer)

sku = "Basic"

}

# Update the Network Interface to Associate the Public IP

resource "azurerm\_network\_interface" "example" {

name = "FlaskNIC"

location = azurerm\_resource\_group.example.location

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

ip\_configuration {

name = "internal"

subnet\_id = azurerm\_subnet.example.id

private\_ip\_address\_allocation = "Dynamic"

public\_ip\_address\_id = azurerm\_public\_ip.example.id # Associate the public IP

}

}

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

name = "FlaskVM"

location = azurerm\_resource\_group.example.location

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

admin\_username = "azureuser"

size = "Standard\_B1s"

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

os\_disk {

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

# Specify the image source

source\_image\_reference {

publisher = "Canonical"

offer = "0001-com-ubuntu-server-jammy"

sku = "22\_04-LTS"

version = "latest"

}

# Add SSH key for authentication

admin\_ssh\_key {

username = "azureuser"

public\_key = file("~/.ssh/azure\_id\_rsa.pub") # Path to your public key

}

}

# Output the Public IP address

output "public\_ip" {

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

depends\_on = [azurerm\_public\_ip.example]

}

###########

Run following commands and resolve errors (key, azcli)

terraform init

terraform plan

#### S7: Resolve issues and login into Azure

# Update your system

sudo apt-get update

# Install prerequisites

sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

# Import the Microsoft repository key

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

az –version

ssh-keygen -t rsa -b 4096 -f ~/.ssh/azure\_id\_rsa

az login

Follow instructions and Ensure a similar out appears

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No Subscription name Subscription ID Tenant

----- -------------------- ------------------------------------ -------------

#### S8: Rerun terraform and verify the output (New VM created)

terraform plan

terraform apply

------------------------------------------------------------------------------------------------------------------------

#### S9: install ansible, create scripts and run them – check for error in script

sudo apt install ansible -y

Set the ip addresses, ansible\_user and private key for the user in the hosts file

vi hosts

[flask\_vm]

<ip of machine> ansible\_user=azureuser ansible\_ssh\_private\_key\_file=~/.ssh/azure\_id\_rsa

vi deploy\_flask.yml

- name: Deploy Flask App

hosts: flask\_app\_vm

become: yes

tasks:

- name: Install Python3 and Pip

apt:

update\_cache: yes

- name: Install Python3 and Pip

apt:

name:

- python3

- python3-pip

state: present

- name: Install Flask

pip:

name: flask

- name: Copy Flask App to VM

copy:

src: ../flask\_app/app.py

dest: /home/azureuser/flask\_app/

- name: Start Flask App

shell: python3 /home/azureuser/flask\_app/app.py

ansible-playbook -i hosts deploy\_flask.yml

Test on <public ip:5000>

#### S10: Shutdown app and Cleanup the resources

Press Ctrl-C in the running ansible command to shutdown flask application

cd ../tf

terraform destroy

If req, shutdown the machine

sudo shutdown -h now