#### Chapter 5 Automating Infrastructure Deployments in Azure Using Terraform

### Authenticating to Azure

## 2. Configuring the service principal in Terraform

#Setting Terraform properties in Terraform configuration

# variables.tf

variable "client\_secret" {

description = "Enter your CLient Secret. Please make sure you do not store the value of your client secret in the SCM repository"

}

variable "client\_id" {

description = "Your client Id."

default = "<SERVICE-PRINCIPAL-CLIENT\_ID>"

}

variable "subscription\_id" {

description = "Your subscription id."

default = "<AZURE-SUBSCRIPTION\_ID>"

}

variable "tenant\_id" {

description = "Your tenant id."

default = "<AZURE-TENANT\_ID>"

}

## 3. Azure Provider configuration in Terraform

# main.tf

# Azure Provider source and version being used

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "=2.46.0"

}

}

}

# Configure the Microsoft Azure Provider

provider "azurerm" {

features {}

subscription\_id = var.subscription\_id

client\_id = var.client\_id

client\_secret = var.client\_secret

tenant\_id = var.tenant\_id

}

### Configuring Terraform state management

## 1. Configuring a storage account

# create\_azure\_storage.sh

#!/bin/bash

# create\_azure\_storage.sh

RESOURCE\_GROUP\_NAME=tstate

STORAGE\_ACCOUNT\_NAME=tstate$RANDOM

CONTAINER\_NAME=tstate

# Create resource group

az group create --name $RESOURCE\_GROUP\_NAME --location germanywestcentral

# Create storage account

az storage account create --resource-group $RESOURCE\_GROUP\_NAME --name $STORAGE\_ACCOUNT\_NAME --sku Standard\_LRS --encryption-services blob

# Get storage account key

ACCOUNT\_KEY=$(az storage account keys list --resource-group $RESOURCE\_GROUP\_NAME --account-name $STORAGE\_ACCOUNT\_NAME --query '[0].value' -o tsv)

# Create blob container

az storage container create --name $CONTAINER\_NAME --account-name $STORAGE\_ACCOUNT\_NAME --account-key $ACCOUNT\_KEY

echo "storage\_account\_name: $STORAGE\_ACCOUNT\_NAME"

echo "container\_name: $CONTAINER\_NAME"

echo "access\_key: $ACCOUNT\_KEY"

## 2. Configuring Terraform to use the Storage account

# main.tf

# Azure Provider source and version being used

terraform {

backend "azurerm" {

resource\_group\_name = "tstate"

storage\_account\_name = "tstate16782"

container\_name = "tstate"

key = "terraform.tfstate"

}

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "=2.46.0"

}

}

}

# Configure the Microsoft Azure Provider

provider "azurerm" {

features {}

subscription\_id = var.subscription\_id

client\_id = var.client\_id

client\_secret = var.client\_secret

tenant\_id = var.tenant\_id

}

# Create a resource group

resource "azurerm\_resource\_group" "test-rg" {

name = "test-resources"

location = "West Europe"

}

### Creating a virtual network in Azure

## 1.

#variables.tf

variable "client\_secret" {

description = "Enter your CLient Secret. Please make sure you do not store the value of your client secret in the SCM repository"

}

variable "client\_id" {

description = "Your client Id."

}

variable "subscription\_id" {

description = "Your subscription id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "tenant\_id" {

description = "Your tenant id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "prefix" {

description = "The prefix which should be used for all resources in this Chapter"

default = "test"

}

variable "location" {

description = "The Azure Region in which all resources in this Chapter should be created."

default = "West Europe"

}

variable "dbpassword" {

description = "Database Password."

}

##2. Create virtual\_network.tf and use the following code to create a VNET:

# virtual\_network.tf

# Virtual network

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

name = "${var.prefix}-vnet"

address\_space = ["10.0.0.0/16"]

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

}

# Subnet

resource "azurerm\_subnet" "subnet" {

name = "${var.prefix}Subnet"

resource\_group\_name = azurerm\_resource\_group.test-rg.name

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

address\_prefixes = ["10.0.1.0/24"]

}

# Public ip

resource "azurerm\_public\_ip" "publicip" {

name = "pip1"

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

allocation\_method = "Dynamic"

sku = "Basic"

}

# Network interface

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

name = "${var.prefix}-nic"

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

ip\_configuration {

name = "ipconfig1"

subnet\_id = azurerm\_subnet.subnet.id

private\_ip\_address\_allocation = "Dynamic"

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

}

}

### Creating network security groups in Azure

# virtual\_network.tf

# Security Group

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

name = "${var.prefix}SecurityGroup"

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

tags = {

environment = "Production"

}

}

# Security rules

resource "azurerm\_network\_security\_rule" "httprule" {

name = "${var.prefix}http"

priority = 100

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "8080"

source\_address\_prefix = "\*"

destination\_address\_prefix = "10.0.1.0/24"

resource\_group\_name = azurerm\_resource\_group.test-rg.name

network\_security\_group\_name = azurerm\_network\_security\_group.frontendnsg.name

}

# Security rules

resource "azurerm\_network\_security\_rule" "sshrule" {

name = "${var.prefix}ssh"

priority = 101

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "22"

source\_address\_prefix = "\*"

destination\_address\_prefix = "10.0.1.0/24"

resource\_group\_name = azurerm\_resource\_group.test-rg.name

network\_security\_group\_name = azurerm\_network\_security\_group.frontendnsg.name

}

# Subnet – Security Group association

resource "azurerm\_subnet\_network\_security\_group\_association" "subnetsecuritygroup" {

subnet\_id = azurerm\_subnet.subnet.id

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

}

### Creating a virtual machine in Azure

##1. Create the virtual\_machine.tf file with the following code:

# virtual\_machine.tf

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

# name = "${var.prefix}-vm"

name = "${var.prefix}-vm"

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

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

vm\_size = "Standard\_DS1\_v2"

storage\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "16.04-LTS"

version = "latest"

}

storage\_os\_disk {

name = "${var.prefix}osdisk1"

caching = "ReadWrite"

create\_option = "FromImage"

managed\_disk\_type = "Standard\_LRS"

}

os\_profile {

computer\_name = "hostname"

admin\_username = "${var.prefix}admin"

admin\_password = "Password1234!"

}

os\_profile\_linux\_config {

disable\_password\_authentication = false

}

tags = {

environment = "Production"

}

}

### Creating a virtual machine instance with Packer AMI

## 2. Create a file named packer-ami.json and paste the following packer configuration in this JSON file:

{

"variables": {

"az\_client\_id": "",

"az\_client\_secret": ""

},

"builders": [{

"type": "azure-arm",

"client\_id": "{{user `az\_client\_id`}}",

"client\_secret": "{{user `az\_client\_secret`}}",

"tenant\_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

"subscription\_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

"managed\_image\_resource\_group\_name": "ami-rg-store",

"managed\_image\_storage\_account\_type": "Standard\_LRS",

"managed\_image\_name": "jenkins-ami-{{timestamp}}",

"os\_type": "Linux",

"image\_publisher": "Canonical",

"image\_offer": "0001-com-ubuntu-server-focal",

"image\_sku": "20\_04-lts-gen2",

"azure\_tags": {

"dept": "Production",

"task": "VM Image deployment"

},

"location": "West Europe",

"vm\_size": "Standard\_DS2\_v2"

}],

"provisioners": [{

"execute\_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo -E sh '{{ .Path }}'",

"inline": [

"apt update -y",

"sudo apt install -y zip curl wget apt-transport-https openjdk-11-jre openjdk-11-jdk",

"echo '----Install Jenkins ----'",

"curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null",

"echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null",

"sudo apt-get update -y",

"sudo apt-get install -y jenkins"

],

"inline\_shebang": "/bin/sh -x",

"type": "shell"

}]

}

## 5. Create a Terraform virtual machine configuration to use this AMI

# virtual\_machine.tf

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

name = "${var.prefix}-vm"

location = var.location

resource\_group\_name = azurerm\_resource\_group.test-rg.name

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

vm\_size = "Standard\_DS1\_v2"

storage\_image\_reference {

id = "/subscriptions/${var.subscription\_id}/resourceGroups/ami-rg-store/providers/Microsoft.Compute/images/jenkins-ami-1624752554"

}

storage\_os\_disk {

name = "${var.prefix}osdisk"

caching = "ReadWrite"

create\_option = "FromImage"

managed\_disk\_type = "Standard\_LRS"

}

os\_profile {

computer\_name = "hostname"

admin\_username = "${var.prefix}admin"

admin\_password = "Password1234!"

}

os\_profile\_linux\_config {

disable\_password\_authentication = false

}

tags = {

environment = "Production"

}

}

### Configuring Azure Machine scale set

## 1. Create the vitual\_machine\_scale\_set.tf file and paste the following code in it:

# virtual\_machine\_scale\_set.tf

resource "azurerm\_linux\_virtual\_machine\_scale\_set" "vmss" {

name = "${var.prefix}-vmss"

resource\_group\_name = azurerm\_resource\_group.test-rg.name

location = var.location

sku = "Standard\_DS2\_v2"

instances = 3

admin\_username = "adminuser"

admin\_password = "p@ssword1234"

source\_image\_id = "/subscriptions/${var.subscription\_id}/resourceGroups/ami-rg-store/providers/Microsoft.Compute/images/{ManagedImageName}"

admin\_ssh\_key {

username = "adminuser"

public\_key = file("~/.ssh/id\_rsa.pub")

}

network\_interface {

name = "internalNI"

primary = true

ip\_configuration {

name = "internal"

primary = true

subnet\_id = azurerm\_subnet.subnet.id

}

}

os\_disk {

storage\_account\_type = "Standard\_LRS"

caching = "ReadWrite"

}

}

# AutoScale configuration

resource "azurerm\_monitor\_autoscale\_setting" "vmssautoscale" {

name = "autoscale-config"

resource\_group\_name = azurerm\_resource\_group.test-rg.name

location = var.location

target\_resource\_id = azurerm\_linux\_virtual\_machine\_scale\_set.vmss.id

profile {

name = "AutoScale"

capacity {

default = 3

minimum = 1

maximum = 5

}

rule {

metric\_trigger {

metric\_name = "Percentage CPU"

metric\_resource\_id = azurerm\_linux\_virtual\_machine\_scale\_set.vmss.id

time\_grain = "PT1M"

statistic = "Average"

time\_window = "PT5M"

time\_aggregation = "Average"

operator = "GreaterThan"

threshold = 75

}

scale\_action {

direction = "Increase"

type = "ChangeCount"

value = "1"

cooldown = "PT1M"

}

}

rule {

metric\_trigger {

metric\_name = "Percentage CPU"

metric\_resource\_id = azurerm\_linux\_virtual\_machine\_scale\_set.vmss.id

time\_grain = "PT1M"

statistic = "Average"

time\_window = "PT5M"

time\_aggregation = "Average"

operator = "LessThan"

threshold = 25

}

scale\_action {

direction = "Decrease"

type = "ChangeCount"

value = "1"

cooldown = "PT1M"

}

}

}

}

### Creating and deploying an App Service for Java web application

## 1. Create all the variables in variables.tf:

# variables.tf

variable "client\_secret" {

description = "Enter your CLient Secret. Please make sure you do not store the value of your client secret in the SCM repository"

}

variable "client\_id" {

description = "Your client Id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "subscription\_id" {

description = "Your subscription id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "tenant\_id" {

description = "Your tenant id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "prefix" {

description = "The prefix which should be used for all resources in this Chapter"

default = "prod"

}

variable "location" {

description = "The Azure Region in which all resources in this Chapter should be created."

default = "West Europe"

}

variable "environment" {

description = "The Azure Region in which all resources in this Chapter should be created."

default = "Production"

}

variable "dbpassword" {

description = "Enter a strong Database password with at-least one capitals, one small letters, one number and a special character. Example : P@ssw0rd in default"

default = "P@ssw0rd"

}

## 2. Create main.tf file

# main.tf

provider "azurerm" {

features {}

subscription\_id = var.subscription\_id

client\_id = var.client\_id

client\_secret = var.client\_secret

tenant\_id = var.tenant\_id

}

# Azure Provider source and version being used

terraform {

backend "azurerm" {

resource\_group\_name = "tstate"

storage\_account\_name = "tstate16782"

container\_name = "tstate"

key = "terraform\_app\_service.tfstate"

}

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "=2.46.0"

}

}

}

## 1. Create a new rg.tf file

# rg.tf

# Create a resource group north

resource "azurerm\_resource\_group" "north-prod-rg" {

name = "${var.prefix}-north-resources"

location = "North Europe"

tags = {

Environment = var.environment

}

}

# Create a resource group west

resource "azurerm\_resource\_group" "west-prod-rg" {

name = "${var.prefix}-west-resources"

location = "West Europe"

tags = {

Environment = var.environment

}

}

# Create a Resource Groups global

resource "azurerm\_resource\_group" "global-prod-rg" {

name = "${var.prefix}-global-rg"

location = "Germany West Central"

tags = {

Environment = var.environment

}

}

## 4. Create a network.tf file with the following Terraform code to create networks, subnets, and public IPs:

# network.tf

# Virtual network North

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

name = "${var.prefix}-north-vnet"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.north-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

}

# Virtual network West

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

name = "${var.prefix}-west-vnet"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.west-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

}

# Subnet North

resource "azurerm\_subnet" "north-subnet" {

name = "${var.prefix}-north-subnet"

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

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

address\_prefixes = ["10.0.1.0/24"]

}

# Subnet West

resource "azurerm\_subnet" "west-subnet" {

name = "${var.prefix}-west-subnet"

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

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

address\_prefixes = ["10.0.1.0/24"]

}

# Public ip North

resource "azurerm\_public\_ip" "north-publicip" {

name = "${var.prefix}-north-pip"

location = azurerm\_resource\_group.north-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

allocation\_method = "Dynamic"

}

# Public ip West

resource "azurerm\_public\_ip" "west-publicip" {

name = "${var.prefix}-west-pip"

location = azurerm\_resource\_group.west-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

allocation\_method = "Dynamic"

}

### Managing Application Insights for Java web application

## Create a app\_insight.tf file and paste the following code in it to create Application Insight:

# app\_insight.tf

resource "azurerm\_application\_insights" "north-appinsights" {

name = "north-${var.prefix}-appinsights"

location = var.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

application\_type = "java"

}

output "instrumentation\_key\_north" {

value = azurerm\_application\_insights.north-appinsights.instrumentation\_key

sensitive = true

}

output "app\_id\_north" {

value = azurerm\_application\_insights.north-appinsights.app\_id

}

resource "azurerm\_application\_insights" "west-appinsights" {

name = "west-${var.prefix}-appinsights"

location = var.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

application\_type = "java"

}

output "instrumentation\_key\_west" {

value = azurerm\_application\_insights.west-appinsights.instrumentation\_key

sensitive = true

}

output "app\_id\_west" {

value = azurerm\_application\_insights.west-appinsights.app\_id

}

### Creating a database for Java web application

# mysql\_db.tf

resource "azurerm\_mysql\_server" "mysqlserver" {

name = "${var.prefix}-app-mysqlserver"

location = var.location

resource\_group\_name = azurerm\_resource\_group.global-prod-rg.name

administrator\_login = "mysqladmin"

administrator\_login\_password = var.dbpassword

sku\_name = "GP\_Gen5\_2"

storage\_mb = 5120

version = "5.7"

auto\_grow\_enabled = true

backup\_retention\_days = 7

ssl\_enforcement\_enabled = true

ssl\_minimal\_tls\_version\_enforced = "TLS1\_2"

tags = {

Environment = var.environment

}

}

resource "azurerm\_mysql\_database" "mysqldb" {

name = "${var.prefix}-app-db"

resource\_group\_name = azurerm\_resource\_group.global-prod-rg.name

server\_name = azurerm\_mysql\_server.mysqlserver.name

charset = "utf8"

collation = "utf8\_unicode\_ci"

}

### Creating App Service

## 1. Create a file named app\_service.tf for the creation of App service plans for North and West Europe:

# app\_service.tf

# App Servie plan North

resource "azurerm\_app\_service\_plan" "north-asp" {

name = "${var.prefix}-north-asp"

location = azurerm\_resource\_group.north-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

sku {

tier = "Standard"

size = "S1"

}

tags = {

Environment = var.environment

}

}

# App Servie plan West

resource "azurerm\_app\_service\_plan" "west-asp" {

name = "${var.prefix}-west-asp"

location = azurerm\_resource\_group.west-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

sku {

tier = "Standard"

size = "S1"

}

tags = {

Environment = var.environment

}

}

## 2. Now, we can create App Services (in North and West Europe)

# app\_service.tf

# App Service North

resource "azurerm\_app\_service" "north-appservice" {

name = "${var.prefix}-north-java-appservice"

location = azurerm\_resource\_group.north-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

app\_service\_plan\_id = azurerm\_app\_service\_plan.north-asp.id

site\_config {

java\_version = "1.8"

java\_container = "TOMCAT"

java\_container\_version = "9.0"

}

tags = {

Environment = var.environment

}

connection\_string {

name = "Database"

type = "MySQL"

value = "Server=${var.prefix}-app-mysqlserver;Port=3306;Database=${var.prefix}-app-db;User=mysqladmin;SSLMode=1;UseSystemTrustStore=0;Password=${var.dbpassword}"

}

app\_settings = {

APPINSIGHTS\_INSTRUMENTATIONKEY = azurerm\_application\_insights.north-appinsights.instrumentation\_key

APPLICATIONINSIGHTS\_CONNECTION\_STRING = azurerm\_application\_insights.north-appinsights.connection\_string

}

}

# App Service West

resource "azurerm\_app\_service" "west-appservice" {

name = "${var.prefix}-west-java-appservice"

location = azurerm\_resource\_group.west-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

app\_service\_plan\_id = azurerm\_app\_service\_plan.west-asp.id

site\_config {

java\_version = "1.8"

java\_container = "TOMCAT"

java\_container\_version = "9.0"

}

tags = {

Environment = var.environment

}

connection\_string {

name = "Database"

type = "MySQL"

value = "Server=${var.prefix}-app-mysqlserver;Port=3306;Database=${var.prefix}-app-db;User=mysqladmin;SSLMode=1;UseSystemTrustStore=0;Password=${var.dbpassword}"

}

app\_settings = {

APPINSIGHTS\_INSTRUMENTATIONKEY = azurerm\_application\_insights.west-appinsights.instrumentation\_key

APPLICATIONINSIGHTS\_CONNECTION\_STRING = azurerm\_application\_insights.west-appinsights.connection\_string

}

}

### Creating Application Gateway for Java web application

# app\_gateway.tf

# App Gateway North

resource "azurerm\_application\_gateway" "north-gateway" {

name = "${var.prefix}-north-appgateway"

location = azurerm\_resource\_group.north-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.north-prod-rg.name

sku {

name = "WAF\_Medium"

tier = "WAF"

capacity = 2

}

waf\_configuration {

enabled = "true"

firewall\_mode = "Detection"

rule\_set\_type = "OWASP"

rule\_set\_version = "3.0"

}

gateway\_ip\_configuration {

name = "${var.prefix}-north-gateway-ip-configuration"

subnet\_id = azurerm\_subnet.north-subnet.id

}

frontend\_port {

name = "${azurerm\_virtual\_network.north-vnet.name}-feport"

port = 80

}

frontend\_ip\_configuration {

name = "${azurerm\_virtual\_network.north-vnet.name}-feip"

public\_ip\_address\_id = azurerm\_public\_ip.north-publicip.id

}

backend\_address\_pool {

name = "${azurerm\_virtual\_network.north-vnet.name}-beap"

fqdns = ["${azurerm\_app\_service.north-appservice.name}.azurewebsites.net"]

}

probe {

name = "north-probe"

protocol = "http"

path = "/"

host = "${azurerm\_app\_service.north-appservice.name}.azurewebsites.net"

interval = "30"

timeout = "30"

unhealthy\_threshold = "3"

}

backend\_http\_settings {

name = "${azurerm\_virtual\_network.north-vnet.name}-be-htst"

cookie\_based\_affinity = "Disabled"

port = 80

protocol = "Http"

request\_timeout = 60

probe\_name = "north-probe"

pick\_host\_name\_from\_backend\_address = true

}

http\_listener {

name = "${azurerm\_virtual\_network.north-vnet.name}-httplstn"

frontend\_ip\_configuration\_name = "${azurerm\_virtual\_network.north-vnet.name}-feip"

frontend\_port\_name = "${azurerm\_virtual\_network.north-vnet.name}-feport"

protocol = "Http"

}

request\_routing\_rule {

name = "${azurerm\_virtual\_network.north-vnet.name}-rqrt"

rule\_type = "Basic"

http\_listener\_name = "${azurerm\_virtual\_network.north-vnet.name}-httplstn"

backend\_address\_pool\_name = "${azurerm\_virtual\_network.north-vnet.name}-beap"

backend\_http\_settings\_name = "${azurerm\_virtual\_network.north-vnet.name}-be-htst"

}

tags = {

Environment = var.environment

}

}

# App Gateway West

resource "azurerm\_application\_gateway" "west-gateway" {

name = "${var.prefix}-west-appgateway"

location = azurerm\_resource\_group.west-prod-rg.location

resource\_group\_name = azurerm\_resource\_group.west-prod-rg.name

sku {

name = "WAF\_Medium"

tier = "WAF"

capacity = 2

}

waf\_configuration {

enabled = "true"

firewall\_mode = "Detection"

rule\_set\_type = "OWASP"

rule\_set\_version = "3.0"

}

gateway\_ip\_configuration {

name = "${var.prefix}-west-gateway-ip-configuration"

subnet\_id = azurerm\_subnet.west-subnet.id

}

frontend\_port {

name = "${azurerm\_virtual\_network.west-vnet.name}-feport"

port = 80

}

frontend\_ip\_configuration {

name = "${azurerm\_virtual\_network.west-vnet.name}-feip"

public\_ip\_address\_id = azurerm\_public\_ip.west-publicip.id

}

backend\_address\_pool {

name = "${azurerm\_virtual\_network.west-vnet.name}-beap"

fqdns = ["${azurerm\_app\_service.west-appservice.name}.azurewebsites.net"]

}

probe {

name = "west-probe"

protocol = "http"

path = "/"

host = "${azurerm\_app\_service.west-appservice.name}.azurewebsites.net"

interval = "30"

timeout = "30"

unhealthy\_threshold = "3"

}

backend\_http\_settings {

name = "${azurerm\_virtual\_network.west-vnet.name}-be-htst"

cookie\_based\_affinity = "Disabled"

port = 80

protocol = "Http"

request\_timeout = 60

probe\_name = "west-probe"

pick\_host\_name\_from\_backend\_address = true

}

http\_listener {

name = "${azurerm\_virtual\_network.west-vnet.name}-httplstn"

frontend\_ip\_configuration\_name = "${azurerm\_virtual\_network.west-vnet.name}-feip"

frontend\_port\_name = "${azurerm\_virtual\_network.west-vnet.name}-feport"

protocol = "Http"

}

request\_routing\_rule {

name = "${azurerm\_virtual\_network.west-vnet.name}-rqrt"

rule\_type = "Basic"

http\_listener\_name = "${azurerm\_virtual\_network.west-vnet.name}-httplstn"

backend\_address\_pool\_name = "${azurerm\_virtual\_network.west-vnet.name}-beap"

backend\_http\_settings\_name = "${azurerm\_virtual\_network.west-vnet.name}-be-htst"

}

tags = {

Environment = var.environment

}

}

### Creating Traffic Manager for multi-location web app

##create a Traffic Manager in a new file called traffic\_manager.tf:

# traffic\_manager.tf

# Traffic manager for Java Web App

resource "azurerm\_traffic\_manager\_profile" "traffic-manager" {

name = "${var.prefix}-java-webapp-tm"

resource\_group\_name = azurerm\_resource\_group.global-prod-rg.name

traffic\_routing\_method = "Performance"

dns\_config {

relative\_name = "${var.prefix}-java-webapp"

ttl = 300

}

monitor\_config {

protocol = "http"

port = 80

path = "/"

interval\_in\_seconds = 30

timeout\_in\_seconds = 9

tolerated\_number\_of\_failures = 3

}

tags = {

Environment = var.environment

}

}

# Endpoint North

resource "azurerm\_traffic\_manager\_endpoint" "north-tm-endpoint" {

name = "${var.prefix}-north-global-tm"

resource\_group\_name = azurerm\_resource\_group.global-prod-rg.name

profile\_name = azurerm\_traffic\_manager\_profile.traffic-manager.name

target = azurerm\_public\_ip.north-publicip.fqdn

endpoint\_location = azurerm\_public\_ip.north-publicip.location

type = "externalEndpoints"

}

# Endpoint West

resource "azurerm\_traffic\_manager\_endpoint" "west-tm-endpoint" {

name = "${var.prefix}-west-global-tm"

resource\_group\_name = azurerm\_resource\_group.global-prod-rg.name

profile\_name = azurerm\_traffic\_manager\_profile.traffic-manager.name

target = azurerm\_public\_ip.west-publicip.fqdn

endpoint\_location = azurerm\_public\_ip.west-publicip.location

type = "externalEndpoints"

}

### Deploying a Java web application in App Services

## github workflow configuration YML

# Docs for the Azure Web Apps Deploy action: https://github.com/Azure/webapps-deploy

# More GitHub Actions for Azure: https://github.com/Azure/actions

name: Build and deploy WAR app to Azure Web App - prod-north-java-appservice

on:

push:

branches:

- master

workflow\_dispatch:

jobs:

build:

runs-on: windows-latest

steps:

- uses: actions/checkout@v2

- name: Set up Java version

uses: actions/setup-java@v1

with:

java-version: '1.7'

- name: Build with Maven

run: mvn clean install

- name: Upload artifact for deployment job

uses: actions/upload-artifact@v2

with:

name: java-app

path: '${{ github.workspace }}/target/\*.war'

deploy:

runs-on: windows-latest

needs: build

environment:

name: 'production'

url: ${{ steps.deploy-to-webapp.outputs.webapp-url }}

steps:

- name: Download artifact from build job

uses: actions/download-artifact@v2

with:

name: java-app

- name: Deploy to Azure Web App

id: deploy-to-webapp

uses: azure/webapps-deploy@v2

with:

app-name: 'prod-north-java-appservice'

slot-name: 'production'

publish-profile: ${{ secrets.AzureAppService\_PublishProfile\_xxxxxxxxxxxxxxxxxxxxxxxxxxxx }}

package: '\*.war'

### Managing Azure Spring Cloud

## 1. new file var.tf

# var.tf

variable "client\_secret" {

description = "Enter your CLient Secret. Please make sure you do not store the value of your client secret in the SCM repository"

}

variable "client\_id" {

description = "Your client Id."

default = " xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx "

}

variable "subscription\_id" {

description = "Your subscription id."

default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

variable "tenant\_id" {

description = "Your tenant id."

default = " xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx "

}

variable "prefix" {

description = "The prefix which should be used for all resources in this Chapter"

default = "prod"

}

variable "environment" {

description = "The Azure Region in which all resources in this Chapter should be created."

default = "Production"

}

variable "resource\_group\_name" {

type = string

description = "Core Infrastructure Resource Group"

default = "sc-rg"

}

variable "location" {

type = string

default = "East US"

}

## 3.

# rg.tf

# Create a Spring Cloud Resource Groups global

resource "azurerm\_resource\_group" "sc-prod-rg" {

name = var.resource\_group\_name

location = var.location

tags = {

Environment = var.environment

}

}

## 4.

# spring\_cloud.tf

# Application insights for spring cloud service

resource "azurerm\_application\_insights" "sc\_app\_insights" {

name = "sc\_insights"

location = var.location

resource\_group\_name = azurerm\_resource\_group.sc-prod-rg.name

application\_type = "web"

}

resource "azurerm\_spring\_cloud\_service" "sc" {

name = "sc-service"

resource\_group\_name = azurerm\_resource\_group.sc-prod-rg.name

location = var.location

timeouts {

create = "60m"

delete = "2h"

}

trace {

instrumentation\_key = azurerm\_application\_insights.sc\_app\_insights.instrumentation\_key

}

}

## 5.

# spring\_cloud.tf

# Spring Cloud App

resource "azurerm\_spring\_cloud\_app" "sc-app" {

name = "hellospring"

resource\_group\_name = azurerm\_resource\_group.sc-prod-rg.name

service\_name = azurerm\_spring\_cloud\_service.sc.name

}