## Setting up Azure

### Step 1: Create a service principal:

You can create a service principle in two ways. Either using the Azure CLI or using the portal.

To create the service principal using the CLI, Run the below command with your service principal name and subscription id.

***#az ad sp create-for-rbac -n "<serviceprincipal-name>" --role="Contributor" --scopes="/subscriptions/<subscription-id>***

***Output:***

*{*

*"appId": "f32006e3-d030-49e6-935e-214821ded64c",*

*"displayName": "Terraform-Sk",*

*"name": "*[*http://Terraform-Sk*](http://terraform-sk)*",*

*"password": "ry7dIPY~F.oN4~1HIxQrKrSIzzfE\_s4LGc",*

*"tenant": "19b25223-4653-4976-90ab-252b35d5c969"*

*}*

### Step 2: Create a terraform configuration file with *.tf* extension

Update the client\_id with the app\_id which you got in the above output. Similarly, Update client\_secret and tenant\_id with password and tenant values respectively.

Get the subscription id from the azure portal and update it in the below file as well.

# Configure the Microsoft Azure Provider

provider "azurerm" {

subscription\_id = "ab8e394e-1b63-4a20-8caf-ad72b67af795"

client\_id = "b1c046a3-ee83-490a-826a-731488f834db"

client\_secret = "~3yi-Y8N2teK..z\_g-Q\_~y0e9zx2UhNSui"

tenant\_id = "19b25223-4653-4976-90ab-252b35d5c969"

features {}

}

# Create a resource group if it doesn't exist

resource "azurerm\_resource\_group" "myterraformgroup" {

name = "myResourceGroup"

location = "eastus"

tags = {

environment = "Terraform Demo"

}

}

# Create virtual network

resource "azurerm\_virtual\_network" "myterraformnetwork" {

name = "myVnet"

address\_space = ["10.0.0.0/16"]

location = "eastus"

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

tags = {

environment = "Terraform Demo"

}

}

# Create subnet

resource "azurerm\_subnet" "myterraformsubnet" {

name = "mySubnet"

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

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

address\_prefixes = ["10.0.1.0/24"]

}

# Create public IPs

resource "azurerm\_public\_ip" "myterraformpublicip" {

name = "myPublicIP"

location = "eastus"

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

allocation\_method = "Dynamic"

tags = {

environment = "Terraform Demo"

}

}

# Create Network Security Group and rule

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

name = "myNetworkSecurityGroup"

location = "eastus"

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

security\_rule {

name = "SSH"

priority = 1001

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "22"

source\_address\_prefix = "\*"

destination\_address\_prefix = "\*"

}

tags = {

environment = "Terraform Demo"

}

}

# Create network interface

resource "azurerm\_network\_interface" "myterraformnic" {

name = "myNIC"

location = "eastus"

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

ip\_configuration {

name = "myNicConfiguration"

subnet\_id = azurerm\_subnet.myterraformsubnet.id

private\_ip\_address\_allocation = "Dynamic"

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

}

tags = {

environment = "Terraform Demo"

}

}

# Connect the security group to the network interface

resource "azurerm\_network\_interface\_security\_group\_association" "example" {

network\_interface\_id = azurerm\_network\_interface.myterraformnic.id

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

}

# Generate random text for a unique storage account name

resource "random\_id" "randomId" {

keepers = {

# Generate a new ID only when a new resource group is defined

resource\_group = azurerm\_resource\_group.myterraformgroup.name

}

byte\_length = 8

}

# Create storage account for boot diagnostics

resource "azurerm\_storage\_account" "mystorageaccount" {

name = "diag${random\_id.randomId.hex}"

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

location = "eastus"

account\_tier = "Standard"

account\_replication\_type = "LRS"

tags = {

environment = "Terraform Demo"

}

}

# Create (and display) an SSH key

resource "tls\_private\_key" "example\_ssh" {

algorithm = "RSA"

rsa\_bits = 4096

}

output "tls\_private\_key" { value = tls\_private\_key.example\_ssh.private\_key\_pem }

# Create virtual machine

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

name = "myVM"

location = "eastus"

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

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

size = "Standard\_B1s"

os\_disk {

name = "myOsDisk"

caching = "ReadWrite"

storage\_account\_type = "Premium\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

computer\_name = "myvm"

admin\_username = "azureuser"

disable\_password\_authentication = true

admin\_ssh\_key {

username = "azureuser"

public\_key = "tls\_private\_key.example\_ssh.public\_key\_openssh"

}

boot\_diagnostics {

storage\_account\_uri = azurerm\_storage\_account.mystorageaccount.primary\_blob\_endpoint

}

tags = {

environment = "Terraform Demo"

}

}

### Step 3: Download and initialize terraform

1. Download and extract the terraform zip file depending upon the os type.
2. Update the terraform directory path in the system path environment.
3. Run below commands in sequence.

***#terraform --version*** (To check the terraform version)

***#terraform init*** (initializing terraform)

***#terraform validate*** (validate the code)

***#terraform plan*** (performs dry run before provisioning)

***#terraform apply*** (provision the infrastructure on azure)