## Steps To Create Server From Custom Image

Create two files called ***variables.tf*** and ***main.tf***. While we can write terraform deployment script in main.tf file, We will declare all the variables used in variables.tf

### Step 1: Authentication with Azure

Declare four variables called subscription\_id, client\_id, client\_secret, tenant\_id in your variables.tf file with their values.

*variable "****subscription\_id****" {*

*type = string*

*default = "f452bc3d-xxxx-4b53-bfa4-568cf5acd868"*

*}*

*variable "****client\_id****" {*

*type = string*

*default = "b1c046a3-ee83-490a-826a-731488f834db"*

*}*

*variable "****client\_secret****" {*

*type = string*

*default = "90~cXnn73rM.8334\_3-HGPcXMBkQ1-T1EP"*

*}*

*variable "****tenant\_id****" {*

*type = string*

*default = "19b25223-4653-4976-90ab-252b35d5c969"*

*}*

Now in “*main.tf*”, We will use terraform’s ***azurerm*** provider.

Write the below authentication code,

**provider "azurerm" {**

**subscription\_id = var.subscription\_id**

**client\_id = var.client\_id**

**client\_secret = var.client\_secret**

**tenant\_id = var.tenant\_id**

**features {}**

**}**

Notice, Here we are passing the values from the variables file which we have declared previously.

### Step 2: Creating public IP

We will use terrafom’s ***azurerm\_public\_ip*** resource to create a public IP which we will be using while creating a network interface

# Creating Public IP

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

*name =* ***var.new\_public\_ip\_name***

*location =* ***var.resource\_group\_location***

*resource\_group\_name =* ***var.resource\_group\_name***

*allocation\_method = "Dynamic"*

*tags = {*

*environment = "production"*

*}*

*}*

Note, We need to declare 3 more variables, new public ip name, resource group name and location and pass them to this resource.

So add the below code to your variables.tf

*variable "****resource\_group\_name****" {*

*type = string*

*description = "RG name in Azure"*

*default = "image-rg"*

*}*

*variable "****resource\_group\_location****" {*

*type = string*

*description = "RG location in Azure"*

*default = "East US"*

*}*

*variable "****new\_public\_ip\_name****"{*

*type = string*

*default = "image-vm-ip3"*

*}*

### Step 3: Creating a Network Interface

We can use terraform ***azurerm\_network\_interface*** resource to create a Network interface which we will assign to VM later

#### Getting subnet id

For creating the Network interface, We need subnet id. Here we can use terraforms data block to fetch the details of the subnet.

*data "azurerm\_subnet" "subnet" {*

*name =* ***var.subnet***

*resource\_group\_name = var.resource\_group\_name*

*virtual\_network\_name =* ***var.virtual\_network***

*}*

Notice, We need to pass, Subnet name and Virtual Network name to this block. So define those variables as well.

*variable "****subnet****"{*

*type = string*

*default = "image-subnet"*

*}*

*variable "****virtual\_network****"{*

*type = string*

*default = "image-vnet"*

*}*

After writing the above data block, You can use ***${data.azurerm\_subnet.subnet.id}*** to fetch the subnet id.

Now that we got the subnet id, let's write the resource block to create a network interface.

# Creating a network interface

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

*name =* ***var.new\_network\_interface\_name***

*location = var.resource\_group\_location*

*resource\_group\_name = var.resource\_group\_name*

*ip\_configuration {*

*name =* ***var.new\_network\_interface\_ip\_name***

*subnet\_id = "${data.azurerm\_subnet.subnet.id}"*

*private\_ip\_address\_allocation = "Dynamic"*

*public\_ip\_address\_id = "${azurerm\_public\_ip.public\_ip.id}"*

*}*

*tags = {*

*environment = "production"*

*}*

*}*

Since this resource block needs a new network interface name and network interface ip name, Let’s define those variables as well.

*variable "****new\_network\_interface\_name****" {*

*type = string*

*default = "image-nic3"*

*}*

*variable "****new\_network\_interface\_ip\_name****" {*

*type = string*

*default = "nic-ip"*

*}*

### Step 4: Creating VM from custom image

#### Getting image id

We need the id of the image from which we are creating the VM. We can again use terraforms data block to fetch the details of the image.

*data "****azurerm\_shared\_image****" "search" {*

*name =* ***var.custom\_image\_name***

*gallery\_name = "myGallery"*

*resource\_group\_name = var.resource\_group\_name*

*}*

We need to pass custom image name to this block which we can declare in the variable file

*variable "****custom\_image\_name****" {*

*type = string*

*default = "custom-image"*

*}*

#### Creating VM

Now that we got the image id, We can use terraforms ***azurerm\_virtual\_machine***resource to create the VM

# Creating VM from custom image

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

*name =* ***var.new\_vm\_name***

*location = var.resource\_group\_location*

*resource\_group\_name = var.resource\_group\_name*

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

*vm\_size = "Standard\_DS12\_v2"*

*delete\_os\_disk\_on\_termination = true*

*delete\_data\_disks\_on\_termination = true*

*storage\_image\_reference {*

*id = "${data.azurerm\_shared\_image.search.id}"*

*}*

*storage\_os\_disk {*

*name =* ***var.new\_storage\_disk\_name***

*caching = "ReadWrite"*

*create\_option = "FromImage"*

*managed\_disk\_type = "Standard\_LRS"*

*}*

*os\_profile {*

*computer\_name =* ***var.new\_computer\_name***

*admin\_username =* ***var.new\_user\_name***

*admin\_password =* ***var.new\_password***

*}*

*os\_profile\_linux\_config {*

*disable\_password\_authentication = false*

*}*

*}*

We need to pass 5 new variables here for new VM name, storage disk name, computer name, username and password for the computer respectively. Let’s define them

*variable "****new\_user\_name****"{*

*type = string*

*default = "basawaraj"*

*}*

*variable "****new\_password****"{*

*type = string*

*default = "admin@2020"*

*}*

*variable "****new\_computer\_name****"{*

*type = string*

*default = "New Computer"*

*}*

*variable "****new\_vm\_name****" {*

*type = string*

*default = "new-vm-3"*

*}*

*variable "****new\_storage\_disk\_name****" {*

*type = string*

*default = "new-vm-storage-disk"*

*}*

### Step 5: Deploying the VM and Infrastructure

Now that our script files are ready, We can run terraform command to deploy the VM

***terraform init*** - Initialize current working directory containing Terraform configuration files.

***terraform validate*** - Validates the configuration files in current directory.

***terraform apply*** - Apply the changes required to reach the desired state of the configuration.

Optionally you can use terraform’s output block to display the VM id to console. Create a new file called outputs.tf in the same directory and add the below code

*output "****new\_vm\_id****" {*

*value = "${azurerm\_virtual\_machine.vm.id}"*

*}*

Your complete ***main.tf*** file should look like below

*provider "azurerm" {*

*subscription\_id = var.subscription\_id*

*client\_id = var.client\_id*

*client\_secret = var.client\_secret*

*tenant\_id = var.tenant\_id*

*features {}*

*}*

*data "azurerm\_shared\_image" "search" {*

*name = var.custom\_image\_name*

*gallery\_name = "myGallery"*

*resource\_group\_name = var.resource\_group\_name*

*}*

*data "azurerm\_subnet" "subnet" {*

*name = var.subnet*

*resource\_group\_name = var.resource\_group\_name*

*virtual\_network\_name = var.virtual\_network*

*}*

*# Creating Public IP*

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

*name = var.new\_public\_ip\_name*

*location = var.resource\_group\_location*

*resource\_group\_name = var.resource\_group\_name*

*allocation\_method = "Dynamic"*

*tags = {*

*environment = "production"*

*}*

*}*

*# Creating a network interface*

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

*name = var.new\_network\_interface\_name*

*location = var.resource\_group\_location*

*resource\_group\_name = var.resource\_group\_name*

*ip\_configuration {*

*name = var.new\_network\_interface\_ip\_name*

*subnet\_id = "${data.azurerm\_subnet.subnet.id}"*

*private\_ip\_address\_allocation = "Dynamic"*

*public\_ip\_address\_id = "${azurerm\_public\_ip.public\_ip.id}"*

*}*

*tags = {*

*environment = "production"*

*}*

*}*

*# Creating VM from custom image*

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

*name = var.new\_vm\_name*

*location = var.resource\_group\_location*

*resource\_group\_name = var.resource\_group\_name*

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

*vm\_size = "Standard\_DS12\_v2"*

*delete\_os\_disk\_on\_termination = true*

*delete\_data\_disks\_on\_termination = true*

*storage\_image\_reference {*

*id = "${data.azurerm\_shared\_image.search.id}"*

*}*

*storage\_os\_disk {*

*name = var.new\_storage\_disk\_name*

*caching = "ReadWrite"*

*create\_option = "FromImage"*

*managed\_disk\_type = "Standard\_LRS"*

*}*

*os\_profile {*

*computer\_name = var.new\_computer\_name*

*admin\_username = var.new\_user\_name*

*admin\_password = var.new\_password*

*}*

*os\_profile\_linux\_config {*

*disable\_password\_authentication = false*

*}*

*}*