Terraform Modules Lab 1.11

Expected Outcome

#### In this challenge, you will create a module to contain a scalable virtual machine deployment, then create an environment where you will call the module.

Steps

# Create Folder Structure

#### Create a directory lab1.11 and change directory into it.

Mkdir lab1.11

Cd lab1.11

In order to organize your code inside the lab\_exercise directory, create the following folder structure with main.tf, variables.tf and terraform.tfvars files.

Lab1.11

├── main.tf

├── modules

│ └── my\_linux\_vm

| └── main.tf

├── terraform.tfvars

└── variables.tf

# Create the Module

#### Inside the my\_linux\_vm module folder create a main.tf file with the following contents:

variable "prefix" {}

variable "location" {}

variable "username" {}

variable "vm\_size" {}

provider "azurerm" {

features {}

}

resource "random\_password" "password" { length = 16

special = true override\_special = "!"

}

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

location = var.location

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

}

resource "azurerm\_virtual\_network" "main" {

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

location = azurerm\_resource\_group.main.location

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

address\_space = ["10.0.0.0/16"]

}

resource "azurerm\_subnet" "main" {

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

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

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

address\_prefixes = ["10.0.1.0/24"]

}

resource "azurerm\_network\_interface" "main" {

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

location = azurerm\_resource\_group.main.location

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

ip\_configuration { #check docs for changes

name = "config1"

subnet\_id = azurerm\_subnet.main.id private\_ip\_address\_allocation = "dynamic"

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

}

}

resource "azurerm\_virtual\_machine" "main" {

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

location = azurerm\_resource\_group.main.

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

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

vm\_size = var.vm\_size

storage\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

storage\_os\_disk {

name = "${var.prefix}myvm-osdisk"

caching = "ReadWrite"

create\_option = "FromImage"

managed\_disk\_type = "Standard\_LRS"

}

os\_profile\_linux\_config {

disable\_password\_authentication = false

}

os\_profile {

computer\_name = "${var.prefix}myvm"

admin\_username = var.username

admin\_password = random\_password.password.result

}

}

resource "azurerm\_public\_ip" "main" {

name = "${var.prefix}-my-pubip"

location = azurerm\_resource\_group.main.location

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

allocation\_method = "Static"

}

output "vm-password" {

value = random\_password.password.result

description = "Dynamically generated password to access the VM."

}

output "private-ip" {

value = azurerm\_network\_interface.main.private\_ip\_address

description = "Private IP Address"

}

output "public-ip" {

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

description = "Public IP Address"

}

# Create Variables in Root

In your root directory, there should be a variables.tf file.

#### Create "prefix", "location", and "username" variables without defaults. This will result in them being required.

variable "prefix" {}

variable "location" {}

variable "username" {}

Extra credit: How many other variables can you extract?

## Pass in Variables

#### Create a file called 'terraform.tfvars' and add the following variables: Make sure to replace "xxx" with your initials

prefix = "xxx"

location = "uksouth"

username = "terrauser"

# Create the Module declaration in Root

#### Update the root main.tf to declare your module, it should look similar to this:

module "myawesomelinuxvm-a" {

source = "./modules/my\_linux\_vm"

}

Notice the relative module sourcing.

# Terraform Init

**Run terraform init**.

Initializing modules...

- module.myawesomewindowsvm

Getting source "./modules/my\_linux\_vm"

## Terraform Plan

**Run terraform plan.**

Error: Missing required argument

on main.tf line 1, in module "myawesomelinuxvm-a": 1: module "myawesomelinuxvm-a" {

The argument "prefix" is required, but no definition was found.

Error: Missing required argument

on main.tf line 1, in module "myawesomelinuxvm-a": 1: module "myawesomelinuxvm-a" {

The argument "location" is required, but no definition was found.

Error: Missing required argument

on main.tf line 1, in module "myawesomelinuxvm-a": 1: module "myawesomelinuxvm-a" {

The argument "username" is required, but no definition was found.

Error: Missing required argument

on main.tf line 1, in module "myawesomelinuxvm-a": 1: module "myawesomelinuxvm-a" {

The argument "vm\_size" is required, but no definition was found.

#### We have a problem! We didn't set **required variables** for our module.

**Update the main.tf file:**

module "myawesomelinuxvm-a" { source = "./modules/my\_linux\_vm" prefix = "${var.prefix}a" location = var.location

username = var.username vm\_size = "Standard\_A2\_v2"

}

#### **Run terraform** plan again, this time there should not be any errors and you should see your VM built from your module.

+ module.myawesomelinuxvm-a.azurerm\_resource\_group.module id: <computed>

location: "uksouth"

...

Plan: 7 to add, 0 to change, 0 to destroy.

### Add Another Module

#### Add another module block describing another set of Virtual Machines:

module "myawesomelinuxvm-b" { source = "./modules/my\_linux\_vm" prefix = "${var.prefix}b" location = var.location

username = var.username vm\_size = "Standard\_A2\_v2"

}

### Terraform Plan

Since we added another module call, we must run terraform init again before running terraform plan.

#### We should see twice as much infrastructure in our plan.

# module.myawesomewindowsvm.azurerm\_resource\_group.main will be created

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

+ id = (known after apply)

+ location = "uksouth"

}

...

# module.differentwindowsvm.azurerm\_resource\_group.main will be created

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

+ id = (known after apply)

+ location = "uksouth"

}

...

Plan: 14 to add, 0 to change, 0 to destroy.

More Variables

#### In your main.tf file at the root we can see some **duplication.**

Extract "vm\_size" into a local variable to your environment main.tf and reference in each module.

locals {

vm\_size = "Standard\_A2\_v2"

}

#### Now reference them in the module blocks:

module "myawesomelinuxvm-a" {

...

vm\_size = local.vm\_size

}

module "myawesomelinuxvm-b" {

...

vm\_size = local.vm\_size

}

### Terraform Plan

Run **terraform plan** and verify that your plan succeeds and looks the same. Run **terraform apply** to build the infrastructure.

### Module Outputs

#### In your **main.tf** file at the root we query for the public IPs of the servers built within the module by creating output references to the public-ip value of the respective modules.

output "public\_ip\_vm\_a" {

value = module.myawesomelinuxvm-a.public-ip

}

output "public\_ip\_vm\_b" {

value = module.myawesomelinuxvm-b.public-ip

}

Run **terraform refresh** to view these outputs.

### Cleanup

Once complete, run a **terraform destroy** to tear down the infrastructure.

### Advanced areas to explore

1. Extract the Resource Group, Virtual Network, and Subnet into a "Networking" module and the Network Interface and Virtual Machine into a "VM" module and reference them with module declarations.
2. Update the VM module to use SSH instead of password authentication.
3. Add a reference to the Public Terraform Module for [Azure Compute](https://registry.terraform.io/modules/Azure/compute/azurerm/latest)

### Resources

* + [Using Terraform Modules](https://www.terraform.io/docs/language/modules/syntax.html)
  + [Source Terraform Modules](https://www.terraform.io/docs/language/modules/sources.html)
  + [Public Module Registry](https://www.terraform.io/docs/registry/index.html)
  + [locals](https://learn.hashicorp.com/tutorials/terraform/locals)