INTERNAL

**Lab** **Guide** **-** **Terraform** **Foundations** **for** **Azure**

**Downloading** **Terraform**

Visit the Downloads page at https://www.terraform.io/downloads.html and download the relevant executable for your operating system. This course will assume a Windows environment for all the demos and lab exercises

OR

Take the terraform.zip file located in the root of the repository cloned. This contains the terraform.exe for Windows.

**Setting** **up** **Terraform** **for** **the** **lab**

**1.** **Copy** **the** **exe** **file** **downloaded** **above** **into** **a** **working** **folder** **of** **your** **choice** **2.** **Set** **the** **path** **variable** **to** **ensure** **that** **terraform.exe** **is** **in** **the** **path**

If using Windows Command Prompt

path=%path%;<path to terraform exe>

If using Windows Power Shell

$env:Path += “;<path to terraform exe>”

Azure Authentication using Azure CLI and Service Principal

1. Login to your Personal Azure Account (Portal.azure.com) 2. On top right, click the cloud shell icon (first icon)

3. It opens the command editor and select the Bash flow, mount a storage for this. 4. Type **az** **login,** authenticate using the code displayed.

5. A json with complete details of the Azure subscription will be displayed. 6. Extract the **id** from it.

7. Type the following command, and replace the SUBSCRIPTION\_ID with the **id** you copied in the previous step,

az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/SUBSCRIPTION\_ID"

8. It outputs a json with **appId,** **password** and **tenant** which are mapped to the respective attributes client\_id, client\_secret and tenant\_id in the following command,

INTERNAL

az login --tenant

--service-principal -u CLIENT\_ID -p CLIENT\_SECRET TENANT\_ID

9. Run the above command to test existence of service principal.

**Assignment** **#1**

**Problem** **statement:** Create a Resource Group with a specific region in the Azure portal. Start provisioning a Virtual Network and subsequent components of the VNET as part of the same resource group and launch a VM which displays “Hello World from Azure VM” in the Browser.

**Instruction** **steps:**

1.Copy the skeleton files from the folder “**assignment1**” of the lab repository into your working directory for this course

2. Open the file “**terraform.tfvars**” and paste the respective values from your Azure account as given below,

azure\_subscription\_id = "xxxxxx" azure\_client\_id = "xxxxxxxxx" azure\_client\_secret = "xxxxxxxxxx" azure\_tenant\_id = "xxxxxxxx"

3. Open the file **“provider-variables.tf”** and paste the below code into the file.

variable "azure\_subscription\_id" { type = string

description = "Azure Subscription ID" }

variable "azure\_client\_id" { type = string

description = "Azure Client ID" }

variable "azure\_client\_secret" { type = string

description = "Azure Client Secret" }

variable "azure\_tenant\_id" { type = string

INTERNAL

description = "Azure Tenant ID" }

4. Open the file **“provider-main.tf”** and add the following code inside **provider** **"azurerm"** block after the **environment** attribute.

subscription\_id = client\_id = client\_secret = tenant\_id =

var.azure\_subscription\_id var.azure\_client\_id var.azure\_client\_secret var.azure\_tenant\_id

5. Open “**network-main.tf**” and add the following set of codes one by one under each resource blocks,

i. For the block, resource **"azurerm\_resource\_group"** **"network-rg"** add

name= "${var.app\_name}-${var.environment}-rg" location = var.location

tags = {

application = var.app\_name environment = var.environment

}

ii. For the block resource **"azurerm\_virtual\_network"** **"network-vnet"** add

name = "${var.app\_name}-${var.environment}-vnet" address\_space = [var.network-vnet-cidr] resource\_group\_name= azurerm\_resource\_group.network-rg.name

location = azurerm\_resource\_group.network-rg.location

tags = {

application = var.app\_name environment = var.environment

}

iii. For the block resource **"azurerm\_subnet"** **"network-subnet"** add

name = "${var.app\_name}-${var.environment}-subnet" address\_prefix = var.network-subnet-cidr virtual\_network\_name=azurerm\_virtual\_network.network-vnet.name

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

6. Open **“linux-vm-variables.tf”** file and provide your **email-id** which you used to login to your Microsoft Azure account in the **variable** **"web-linux-admin-username"** section. Add,

INTERNAL

default = “Your Azure Account Email-id”

7. Open **“linux-vm-main.tf”** and add the following codes under respective blocks,

i. Under resource **"azurerm\_network\_security\_group"** **"web-linux-vm-nsg"**, add

depends\_on=[azurerm\_resource\_group.network-rg]

name = "${lower(replace(var.app\_name," ","-"))}-${var.environment}-web-linux-vm-nsg"

location = azurerm\_resource\_group.network-rg.location

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

security\_rule {

name = description = priority = direction = access = protocol = source\_port\_range = destination\_port\_range = source\_address\_prefix = destination\_address\_prefix =

}

security\_rule {

name = description = priority = direction = access = protocol = source\_port\_range = destination\_port\_range = source\_address\_prefix = destination\_address\_prefix =

}

"allow-ssh" "allow-ssh" 100 "Inbound" "Allow" "Tcp"

"\*" "22"

"Internet" "\*"

"allow-http" "allow-http" 110 "Inbound" "Allow" "Tcp"

"\*" "80"

"Internet" "\*"

tags = {

application = var.app\_name environment = var.environment

}

ii. Under resource **"azurerm\_subnet\_network\_security\_group\_association"** **"web-linux-vm-nsg-association"**, add

INTERNAL

depends\_on=[azurerm\_network\_security\_group.web-linux-vm-nsg]

subnet\_id = azurerm\_subnet.network-subnet.id network\_security\_group\_id = azurerm\_network\_security\_group.web-linux-vm-nsg.id

iii. Under resource "azurerm\_public\_ip" "web-linux-vm-ip", add

depends\_on=[azurerm\_resource\_group.network-rg]

name= "linux-${random\_string.random-linux-vm.result}-vm-ip"

location = azurerm\_resource\_group.network-rg.location resource\_group\_name = azurerm\_resource\_group.network-rg.name

allocation\_method = "Static"

tags = {

application = var.app\_name environment = var.environment

}

iv. Under **data** **"template\_file"** **"linux-vm-cloud-init"** block, add the code

template = file("azure-user-data.sh")

8. Executing the assignment

a. Open Command Prompt / Power Shell and navigate to the assignment working directory

b. Run the following command to download and setup the AWS Provider Plug-in,

terraform init

c. Execute the terraform plan and output the plan to a file using the following command,

terraform plan -out assignment1.tfplan

var.app\_name – The application name used to build resources

Enter a value : **MyVM**

var.company – The company name used to build resources

Enter a value : **SCB**

INTERNAL

var.environment – The environment to be built

Enter a value : **Development**

Var.netwrok-subnet-cidr – The CIDR for the network subnet

Enter a value : **10.0.1.0/24**

Var.netwrok-vnet-cidr – the CIDR of the network VNET

Enter a value : **10.0.0.0/16**

Var.web-linux-vm-size – Size (SKU) of the virtual machine to create

Enter a value: **Standard\_D1\_v2**

d. Execute the terraform apply

terraform apply assignment1.tfplan

e. Open the Azure Portal of your account and verify the resources created. Get the

Public IP from the Virtual Machines resource. Paste the output in your browser and check whether it displays the following output,

“**Hello** **world** **from** **Azure** **Virtual** **Machine**” 9. Destroy the resources

a. Execute the terraform destroy by providing the same values given in step 8.

terraform destroy

**Assignment** **#2**

**Problem** **statement:** Create a Resource Group with a specific region in the Azure portal. Start provisioning a Virtual Network and subsequent components of the VNET as part of the same

INTERNAL

resource group and launch a Virtual Machine. Provision the VM with Managed Disks (create from a standard Image) for storing application data and application log with a default volume.

**Instruction** **steps:**

1.Copy the skeleton files from the folder “**assignment2**” of the lab repository into your working directory for this course

2. Open the file “**terraform.tfvars**” and paste the respective values from your Azure account as given below,

azure\_subscription\_id = "xxxxxx" azure\_client\_id = "xxxxxxxxx" azure\_client\_secret = "xxxxxxxxxx" azure\_tenant\_id = "xxxxxxxx"

3. Open the file **“assignment2.tf”** and paste the below code into the file under **Azure** **Authentication** **variables**.

variable "azure\_subscription\_id" { type = string

description = "Azure Subscription ID" }

variable "azure\_client\_id" { type = string

description = "Azure Client ID" }

variable "azure\_client\_secret" { type = string

description = "Azure Client Secret" }

variable "azure\_tenant\_id" { type = string

description = "Azure Tenant ID" }

variable "prefix" { default = "terraformVMEBS" }

4. Paste the below code inside the Provider **“azurerm”** block,

subscription\_id = client\_id = client\_secret = tenant\_id =

var.azure\_subscription\_id var.azure\_client\_id var.azure\_client\_secret var.azure\_tenant\_id

5. Under resource **"azurerm\_resource\_group"** **"** **main"**, add

INTERNAL

name = "${var.prefix}-resources" location = "West Europe"

6. Under resource **"azurerm\_virtual\_network"** **"mainVnet",** add

name address\_space location

resource\_group\_name

= "${var.prefix}-network" = ["10.0.0.0/16"]

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

7. Under resource **"azurerm\_subnet"** **"internal"**, add

name = resource\_group\_name = virtual\_network\_name = address\_prefixes =

"internal" azurerm\_resource\_group.main.name azurerm\_virtual\_network.main.name ["10.0.2.0/24"]

8. Under resource **"azurerm\_network\_interface"** **"mainNIC"**, add

name = location = resource\_group\_name =

"${var.prefix}-nic" azurerm\_resource\_group.main.location azurerm\_resource\_group.main.name

ip\_configuration {

name = "testconfiguration1" subnet\_id = azurerm\_subnet.internal.id private\_ip\_address\_allocation = "Dynamic"

}

9. Under resource **"azurerm\_virtual\_machine"** **"mainVM"**, add the full code as given below

name = "${var.prefix}-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 = "Standard\_DS1\_v2"

# Uncomment this line to delete the OS disk automatically when deleting the VM delete\_os\_disk\_on\_termination = true

# Uncomment this line to delete the data disks automatically when deleting the VM delete\_data\_disks\_on\_termination = true

storage\_image\_reference { publisher = "Canonical"

INTERNAL

offer = "UbuntuServer" sku = "16.04-LTS" version = "latest"

} storage\_os\_disk {

name = "myosdisk1" caching = "ReadWrite"

create\_option = "FromImage" managed\_disk\_type = "Standard\_LRS"

} os\_profile {

computer\_name = "hostname" admin\_username = "testadmin" admin\_password = "Password1234!" }

os\_profile\_linux\_config { disable\_password\_authentication = false

} tags = {

environment = "staging" }

10. Executing the assignment

a. Open Command Prompt / Power Shell and navigate to the assignment working directory

b. Run the following command to download and setup the AWS Provider Plug-in,

terraform init

c. Execute the terraform plan and output the plan to a file using the following command,

terraform plan -out assignment2.tfplan

d. Execute the terraform apply

terraform apply assignment2.tfplan

**Assignment** **#3**

**Problem** **Statement:** Experiment with a few functions using the Terraform Console

**Instruction** **steps:**

Use the following command to enter the Terraform console

INTERNAL

terraform console

Use the min function to find the minimum of 53, 45, 23 and 34

min(53,45,23,34)

Use the abs function to find the absolute value of -4

abs(-4)

Use the format function to print Hello World ! by inserting "World" into "Hello %s!"

format("Hello, %s!", "World")

Use split function to split the words in "foo,bar,horn" based on comma separator

split(",", "foo,bar,horn")

Use the substr function to print first 4 characters from "hello world"

substr("hello world", 0, 4)

Use concat function to concatenate two lists - ["a",""] and ["b","c"]

concat(["a", ""], ["b", "c"])

Use merge function to merge two maps - {a="b", c="d"} and {e="f", c="z"}

merge({a="b", c="d"}, {e="f", c="z"})

use slice function to print first three items of the list ["a", "b", "c", "d"]

slice(["a", "b", "c", "d"], 0, 3)

print current timestamp

timestamp()

use formatdate to print the following date in MMM DD,YYYY format 2018-01-02T23:12:01Z

formatdate("MMM DD, YYYY", "2018-01-02T23:12:01Z")

Use cidrsubnet function to create the subnet 172.18.0.0/16 from the n/w address -

172.16.0.0/12

INTERNAL

cidrsubnet("172.16.0.0/12", 4, 2)

use tostring function to convert the boolean true to a string

tostring(true)

**Assignment** **#4**

**Problem** **Statement:** In this assignment, you will be using a single configuration script and use terraform workspaces coupled with the use of variables to provision resources to 2 environments.

**Instructional** **Steps:**

1. Open your Azure account and **login** **to** **Azure** **CLI** using Service Principal using the following Command by clicking the cloud shell icon located in the top right,

az login

2. Create **ARN\_Access\_Key** using the following steps, a. From the Azure CLI, type the command

code

b. Paste the following code into the code editor and save the file as **backend.sh**

#!/bin/bash

RESOURCE\_GROUP\_NAME=multistategroup STORAGE\_ACCOUNT\_NAME=multistatestorage CONTAINER\_NAME=multistatecontainer

# Create resource group

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

# 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

INTERNAL

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"

3. Go back to Azure CLI and type the command,

sh backend.sh You will get the following entities,

a. Storage Account Name: b. Storage Contaner Name: c. Access Key:

4. Get into the “**Assignment4**” folder from your Windows Powershell session and set up the following Environment Variables,

a.$env:ARM\_CLIENT\_ID=””

b.$env: ARM\_SUBSCRIPTION\_ID=”” c.$env:ARM\_CLIENT\_SECRET=“ “

d.$env: ARM\_TENANT\_ID=” “

e. $env: ARM\_ACCESS\_KEY=” “(Paste the value of the access obtained in step 3)

5. Navigate to the labskeleton and open the **assignment4** folder.

a. Open the “**terraform.tfvars**” file and paste the following code, location="eastus"

rg-prefix="rg-multienv-demo"

storage-account-name="staticwebsite111112233" index\_document="index.html"

source\_content="<h1>This is static content coming from the Terraform</h1>"

b. Open the “**backend.tf**” file and paste the following code inside backend “**azurerm**” block,

resource\_group\_name = "multitstategroup" storage\_account\_name = "mutistatestorage" container\_name = "multitstatecontainer" key = "terraform.tfstate"

c. Open the **“main.tf”** file and add the codes as given,

i. Under **resource** **“azurerm\_resource\_group”** **“resource\_group”**, add name = var.rg-name

location = var.location

ii. Under resource **"azurerm\_storage\_account"** **"storage\_account"**, add name = var.storage-account-name

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

INTERNAL

location = account\_tier = account\_replication\_type = account\_kind =

var.location "Standard" "LRS" "StorageV2"

static\_website {

index\_document = var.index\_document }

iii. Under **resource** **"azurerm\_storage\_blob"** **"example"**, add

name = var.index\_document storage\_account\_name = azurerm\_storage\_account.storage\_account.name storage\_container\_name = "$web"

type = "Block" content\_type = "text/html" source\_content = var.source\_content

6. Navigate to the **Powershell** and start executing the terraform,

a. Run the following command to download and setup the AWS Provider Plug-in terraform init

b. Execute the terraform plan and output the plan to a file using the following command,

terraform plan -out assignment4.tfplan c. Var.rg.name : multitstate

d. Execute the terraform apply

terraform apply assignment4.tfplan

1. If it says resource group already exists, then execute the following command,

terraform import azurere\_resource\_group.resource\_group /subscriptions/yoursubscriptionid/resourceGroups/multitstate

It asks Var.rg.name : multitstate

Then execute terraform apply once again.

e. Navigate to your Azure Portal and search the storage accounts. f. Select “**staticwebsite111112233**” storage account

**g.** On the left pane, under Data Management Section and click the option “**Static** **website”**

h. This gives a primary endpoint. Copy and paste the endpoint in another browser tab and observe the output,

**“This** **is** **static** **content** **coming** **from** **the** **Terraform”**

7. Navigate back to the Powershell and type the following commands, a.terraform workspace new development

b.terraform plan --var-file=environments/terraform-dev.tfvars

c.terraform apply --var-file=environments/terraform-dev.tffvars

8. Give the value “**yes**” when it asks and visit the azure portal and check the storage account “**staticwebsite100dev**” for the static website from dev environment.

9. Repeat the step 7 for prod environment and verify the output.

INTERNAL

**Assignment** **#5**

**Problem** **Statement:** In this assignment, you will be deploying Azure virtual network using terraform parent and child modules instead of a simple configuration file.

**Instructional** **Steps:**

**1.** Create a storage account “**moduledemo**” from Azure Portal.

**2.** Create a container “**modulecontainer**” within the moduledemo storage account and allow container public access.

**3.** On the left pane of the storage account, under **security+networking** tab, click the **Access** **Keys.**

**4.** Click show keys button and get the **Key1** value and use it as an **ARN\_ACCESS\_KEY** environment variable.

**5.** Get into the “**Assignment5**” folder from your Windows Powershell session and set up the following Environment Variables,

i. $env: ARM\_CLIENT\_ID= “ “

ii. $env: ARM\_SUBSCRIPTION\_ID=” “ iii. $env: ARM\_CLIENT\_SECRET=“ “

iv. $env: ARM\_TENANT\_ID=” “

v. $env: ARM\_ACCESS\_KEY=” “(Paste the value of the access obtained in step 4)

**6.** Open the folder **assignment5** from lab skeleton.

**7.** Navigate to the file “**main.tf**” and paste the following code,

i. Under the **terraform** **{** **backend** **"azurerm"**, add the code resource\_group\_name="NetworkWatcherRG"

storage\_account\_name="moduledemo" contaner\_name="modulecontainer" key="terraform.tfstate"

ii. Under the **resource** **"azurerm\_resource\_group"** **"rg"**, add name = var.resource\_group\_name location = var.location

iii. Under **module** **"virtual-network"**, add

source = "./modules/virtual-network" virtual\_network\_name= var.virtual\_network\_name resource\_group\_name = var.resource\_group\_name location = var.location virtual\_network\_address\_space=

var.virtual\_network\_address\_space subnet\_name = var.subnet\_name

subnet\_address\_prefix = var.subnet\_address\_prefix iv. Under **output** **“azure\_subnet\_id”** ,add

value= module.virtual-network.subnet\_id

INTERNAL

**8.** Open “**terraform.tfvars**” and place the values in the appropriate attributes,

resource\_group\_name = "terrform-demo-rg" virtual\_network\_name = "tfdemo-vnet-01" virtual\_network\_address\_space = ["172.16.0.0/16"] location = "eastUS"

subnet\_name = "subnet-01" subnet\_address\_prefix = "172.16.0.0/24"

**9.** Navigate to the folder **modules/virtual-network** and open the file **modulemain.tf** i. Under **resource** **"azurerm\_virtual\_network"** **"vnet**", add the below code,

name = var.virtual\_network\_name resource\_group\_name = var.resource\_group\_name location = var.location

address\_space = var.virtual\_network\_address\_space

ii. Under **resource** **"azurerm\_subnet"** **"subnet"**, add name = var.subnet\_name

resource\_group\_name = var.resource\_group\_name virtual\_network\_name= azurerm\_virtual\_network.vnet.name address\_prefix = var.subnet\_address\_prefix

**10.** Open the file **outputs.tf** and paste the following code into it, description = "id of the subnet" value = azurerm\_subnet.subnet.id

**11.** Executing the assignment

a. Open Command Prompt / Power Shell and navigate to the **assignment5** working directory

b. Run the following command to download and setup the AWS Provider Plug-in,

terraform init

c. Execute the terraform plan and output the plan to a file using the following command,

terraform plan -out assignment5.tfplan

d. Execute the terraform apply

terraform apply assignment5.tfplan

INTERNAL

**Appendix** **A:** **Useful** **commands**

**\*\*** **TO** **ADD** **TERRAFORM** **TO** **WINDOWS** **PATH** **in** **COMMAND** **PROMPT** **\*\***

path=%path%;<path to terraform exe>

**\*\*** **TO** **ADD** **TERRAFORM** **TO** **WINDOWS** **PATH** **in** **WINDOWS** **POWER** **SHELL** **\*\***

$env:Path += ";<path to terraform exe>"

**\*\*** **TO** **CHECH** **TERRAFORM** **VERSION** **\*\***

terraform version

**\*\*** **TO** **INITIALIZE** **TERRAFORM** **PLUG** **INS** **\*\***

terraform init

**\*\*** **TO** **CREATE** **TERRAFORM** **PLAN** **\*\***

terraform plan -out <plan file name>

**\*\*** **TO** **SHOW** **THE** **PLAN** **\*\***

terraform show <plan file name>

**\*\*** **TO** **RUN** **THE** **PLAN** **\*\***

terraform apply <plan file name>

**\*\*** **TO** **DESTROY** **THE** **RESOURCES** **\*\***

terraform destroy

**\*\*** **TO** **CREATE** **A** **NEW** **WORKSPACE.** **E.g.** **development** **\*\***

terraform workspace new development

**\*\*** **TO** **SWITCH** **WORKSPACE.** **E.g.** **development** **\*\***

terraform workspace select development INTERNAL

INTERNAL

**\*\*** **TO** **OPEN** **TERRAFORM** **CONSOLE** **\*\***

terraform console

**\*\*** **TO** **SET** **ENV** **VARIABLES** **FOR** **ACCESS** **KEY** **AND** **SECRET** **KEY** **\*\***

$env:ARM\_CLIENT\_ID = "value of the client id"

$env:ARM\_CLIENT\_SECRET = "value of the secret key"