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,

```
az login --service-principal -u CLIENT_ID -p CLIENT_SECRET
--tenant 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
```

```
        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 = var.azure_subscription_id
client_id       = var.azure_client_id
client_secret   = var.azure_client_secret
tenant_id       = 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,

```
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                       = "allow-ssh"
    description                = "allow-ssh"
    priority                   = 100
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "22"
    source_address_prefix      = "Internet"
    destination_address_prefix = "*"
  }

  security_rule {
    name                       = "allow-http"
    description                = "allow-http"
    priority                   = 110
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "80"
    source_address_prefix      = "Internet"
    destination_address_prefix = "*"
  }

  tags = {
    application = var.app_name
    environment = var.environment
  }
```

   ii. Under resource **"azurerm_subnet_network_security_group_association" "web-linux-vm-nsg-association"**, add

```
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**

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 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 = var.azure_subscription_id
client_id       = var.azure_client_id
client_secret   = var.azure_client_secret
tenant_id       = var.azure_tenant_id
```

5. Under resource **"azurerm_resource_group" " main"**, add

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

6. Under resource **"azurerm_virtual_network" "mainVnet",** add

```
name                = "${var.prefix}-network"
address_space       = ["10.0.0.0/16"]
location            = azurerm_resource_group.main.location
resource_group_name = azurerm_resource_group.main.name
```

7. Under resource **"azurerm_subnet" "internal"**, add

```
name                 = "internal"
resource_group_name  = azurerm_resource_group.main.name
virtual_network_name = azurerm_virtual_network.main.name
address_prefixes     = ["10.0.2.0/24"]
```

8. Under resource **"azurerm_network_interface" "mainNIC"**, add

```
name                = "${var.prefix}-nic"
location            = azurerm_resource_group.main.location
resource_group_name = 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"
    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

```
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

```
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
```

```
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
      ```

```
location                 = var.location
account_tier             = "Standard"
account_replication_type = "LRS"
account_kind             = "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. Prefix for the environment: Public
   d. Execute the terraform apply

   ```
   terraform apply assignment4.tfplan
   ```

   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.

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
```

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
```

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
```

** 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"
```