**First execute only Part1 code & then execute Part2 code::**

//\*\*\*Part1:: Creation of RG using terraform scripts\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

  subscription\_id = "870f9e29-b3a9-4826-b012-f6e1e456427b"

  client\_id = "8d79f9f7-1d58-48f4-b198-1ec2e3f79236"

  client\_secret = "OQN8Q~qYLBBiLcPeZrUGEhNb9uAudCx-5VaP\_b-F"

  tenant\_id = "8f89f11b-2714-4c64-afaf-8dba532aa5fa"

  features {

  }

}

resource "azurerm\_resource\_group" "RGCreations" {

    name = "NareshRG"

    location = "North Europe"

}

//\*\*\*Part2::Creation of Storage Account using terraform scripts\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

  subscription\_id = "870f9e29-b3a9-4826-b012-f6e1e456427b"

  client\_id = "8d79f9f7-1d58-48f4-b198-1ec2e3f79236"

  client\_secret = "OQN8Q~qYLBBiLcPeZrUGEhNb9uAudCx-5VaP\_b-F"

  tenant\_id = "8f89f11b-2714-4c64-afaf-8dba532aa5fa"

  features {

  }

}

resource "azurerm\_storage\_account" "SACreations" {

  name                     = "nareshsa1971"

  resource\_group\_name      = "MyPracticeRG"

  location                 = "North Europe"

  account\_tier             = "Standard"

  account\_replication\_type = "GRS"

  tags = {

    environment = "staging"

  }

}

//\*\*\*Terraform script for Vnet deployment in Azure cloud computing\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

subscription\_id = "870f9e29-b3a9-4826-b012-f6e1e456427b"

  client\_id = "8d79f9f7-1d58-48f4-b198-1ec2e3f79236"

  client\_secret = "OQN8Q~qYLBBiLcPeZrUGEhNb9uAudCx-5VaP\_b-F"

  tenant\_id = "8f89f11b-2714-4c64-afaf-8dba532aa5fa"

  features {

  }

}

resource "azurerm\_virtual\_network" "VnetCreations" {

  name                = "MyVnet1515"

  address\_space       = ["10.0.0.0/16"]

  location            = "North Europe"

  resource\_group\_name = "NareshRG"

}

//\*\*\*Terraform script for Subnet deployment in Azure cloud computing\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

subscription\_id = "870f9e29-b3a9-4826-b012-f6e1e456427b"

  client\_id = "8d79f9f7-1d58-48f4-b198-1ec2e3f79236"

  client\_secret = "OQN8Q~qYLBBiLcPeZrUGEhNb9uAudCx-5VaP\_b-F"

  tenant\_id = "8f89f11b-2714-4c64-afaf-8dba532aa5fa"

  features {

  }

}

resource "azurerm\_subnet" "Subnetcreations" {

  name                 = "Subnet1"

  resource\_group\_name  = "NareshRG"

  virtual\_network\_name = "MyVnet1515"

  address\_prefixes     = ["10.0.1.0/24"]

  service\_endpoints    = ["Microsoft.Sql", "Microsoft.Storage"]

}

//\*\*\*terraform script for deployment of multiple resources in Azure cloud computing\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

  subscription\_id = "870f9e29-b3a9-4826-b012-f6e1e456427b"

  client\_id = "8d79f9f7-1d58-48f4-b198-1ec2e3f79236"

  client\_secret = "OQN8Q~qYLBBiLcPeZrUGEhNb9uAudCx-5VaP\_b-F"

  tenant\_id = "8f89f11b-2714-4c64-afaf-8dba532aa5fa"

  features {

  }

}

resource "azurerm\_virtual\_network" "VnetCreations" {

  name                = "MyVnet1717"

  address\_space       = ["10.0.0.0/24"]

  location            = "North Europe"

  resource\_group\_name = "NareshRG"

}

resource "azurerm\_storage\_account" "SACreations" {

  name                = "nareshsa1996"

  resource\_group\_name = "NareshRG"

  location            = "North Europe"

  account\_tier        = "Standard"

  account\_replication\_type = "LRS"

  tags = {

    environment = "staging"

  }

}

//\*\*\*terraform script for deployment of multiple resources in Azure cloud computing\*\*\*//

terraform {

  required\_providers {

    azurerm = {

      source = "hashicorp/azurerm"

      version = "2.93.0"

    }

  }

}

provider "azurerm" {

  subscription\_id = "5e2de06f-6710-472b-a097-90a2cee4c5c5"

  client\_id = "27b1cb1e-440d-4e0f-bf4a-60d434c5897f"

  client\_secret = "Fo48Q~QY.HKmq.ysuyJaSobfJdM9CzelK40IudA\_"

  tenant\_id = "5f6c3ffe-5945-4c72-aec8-fed83bf8744e"

  features {

  }

}

resource "azurerm\_resource\_group" "RGCreations" {

    name = "IzzuRG"

    location = "North Europe"

}

resource "azurerm\_virtual\_network" "VnetCreations" {

  name                = "MyVnet1515"

  address\_space       = ["10.0.0.0/16"]

  location            = "North Europe"

  resource\_group\_name = "IzzuRG"

}

resource "azurerm\_subnet" "Subnetcreations" {

  name                 = "Subnet1"

  resource\_group\_name  = "IzzuRG"

  virtual\_network\_name = "MyVnet1515"

  address\_prefixes     = ["10.0.1.0/24"]

  service\_endpoints    = ["Microsoft.Sql", "Microsoft.Storage"]

}

resource "azurerm\_storage\_account" "SACreations" {

  name                = "khidashsa1996"

  resource\_group\_name = "IzzuRG"

  location                 = "North Europe"

  account\_tier             = "Standard"

  account\_replication\_type = "LRS"

  tags = {

    environment = "staging"

  }

}

**Terraform Scripts execution directly from Azure Portal:**

**Steps to perform in Azure Portal:**

**Step1:** Login to Azure portal

**Step2:** Click on cloudshell.

**Step3:** create the Storage account & wait for some 5-10 mins till the bash terminal gets ready.

**Step4:** mkdir Terraform>>enter>>this command is to create a directory.

**Step5:** cd Terraform>>enter>>this command is to get into the directory.

**Step6:** vi main.tf>>enter>>this command is to create VM file for Terraform.

**Step7:** i >>this command is to insert the code for our infrastructure resources.

**Step8:** paste the VM creation and other resources creation code>>

**Step9:** esc >>this command is to come out of the file.

**Step10:** :wq >>this command is to save the file.

**Step11:** press enter.

**Step12:** terraform init>>this command is to initialize the terraform.

**Step13:** press enter.

**Step14:** terraform plan -out main.tfplan>>this command is used to create the execution plan.

**Step15:** press enter.

**Step16:** terraform apply main.tfplan>>this command is to apply the execution plan to our cloud infrastructure.

**Step17:** press enter and wait for some 8-10 mins then here will see the Azure resources are getting deployed/created in cloud platform as per the terraform script we have written.

**Terraform script:**

# Configure the Microsoft Azure Provider

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "~>2.0"

}

}

}

provider "azurerm" {

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

sensitive = true

}

# Create virtual machine

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

name = "TerraformVM"

location = "eastus"

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

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

size = "Standard\_DS1\_v2"

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 = " TerraformVM "

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"

}

}