# Terraform Script For Linux VM deploy:

# Configure the Microsoft Azure Provider

*provider* "azurerm" {

    # The "feature" block is required for AzureRM provider 2.x.

    # If you're using version 1.x, the "features" block is not allowed.

    version = "~>2.0"

    features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

# 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\_prefix       = "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 virtual machine

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

    name                  = "myVM"

    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       = "16.04.0-LTS"

        version   = "latest"

    }

    computer\_name  = "myvm"

    admin\_username = "azureuser"

    admin\_password = "Linux@123456"

    disable\_password\_authentication = false

    boot\_diagnostics {

        storage\_account\_uri = azurerm\_storage\_account.mystorageaccount.primary\_blob\_endpoint

    }

    tags = {

        environment = "Terraform Demo"

    }

}

# Terraform Script For Windows VM deploy:

MAIN.TF file>>

# Configure the Microsoft Azure Provider

*provider* "azurerm" {

    # The "feature" block is required for AzureRM provider 2.x.

    # If you're using version 1.x, the "features" block is not allowed.

    version = "~>2.0"

    features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

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

  name     = "${var.prefix}-resources"

  location = var.location

}

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

  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

}

*resource* "azurerm\_subnet" "internal" {

  name                 = "internal"

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

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

  address\_prefix       = "10.0.2.0/24"

}

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

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

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

  location            = azurerm\_resource\_group.main.location

  ip\_configuration {

    name                          = "internal"

    subnet\_id                     = azurerm\_subnet.internal.id

    private\_ip\_address\_allocation = "Dynamic"

  }

}

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

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

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

  location                        = azurerm\_resource\_group.main.location

  size                            = "Standard\_F2"

  admin\_username                  = "adminuser"

  admin\_password                  = "P@ssw0rd1234!"

  network\_interface\_ids = [

    azurerm\_network\_interface.main.id,

  ]

  source\_image\_reference {

    publisher = "MicrosoftWindowsServer"

    offer     = "WindowsServer"

    sku       = "2016-Datacenter"

    version   = "latest"

  }

  os\_disk {

    storage\_account\_type = "Standard\_LRS"

    caching              = "ReadWrite"

  }

}

VARIABLES.TF File>>

*variable* "prefix" {

  description = "The prefix which should be used for all resources in this example"

}

*variable* "location" {

  description = "The Azure Region in which all resources in this example should be created."

}

# Terraform Script For multiple Linux VM deploy:

*provider* "azurerm" {

    # The "feature" block is required for AzureRM provider 2.x.

    # If you're using version 1.x, the "features" block is not allowed.

    version = "=2.0.0"

    features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

# Create a resource group if it doesn't exist

*resource* "azurerm\_resource\_group" "myterraformgroup" {

    name     = var.prefix

    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"

    }

    depends\_on = ["azurerm\_resource\_group.myterraformgroup" ]

}

# Create subnet

*resource* "azurerm\_subnet" "myterraformsubnet" {

    count                = tonumber(var.temp)

    name                 = "mySubnet${count.index}"

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

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

    address\_prefix       = "10.0.${count.index}.0/24"

depends\_on = ["azurerm\_virtual\_network.myterraformnetwork"]

}

# data "azurerm\_subnet" "myterraformsubnet" {

#   for\_each             = var.nsg\_ids

#   name                 = each.key

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

#     virtual\_network\_name = azurerm\_virtual\_network.myterraformnetwork.name

#     depends\_on = ["azurerm\_subnet.myterraformsubnet"]

# }

# Create public IPs

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

    count                        = tonumber(var.temp)

    name                         = "myPublicIP${count.index}"

    location                     = "eastus"

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

    allocation\_method            = "Dynamic"

    tags = {

        environment = "Terraform Demo"

    }

    depends\_on = ["azurerm\_resource\_group.myterraformgroup" ]

}

# Create Network Security Group and rule

*resource* "azurerm\_network\_security\_group" "myterraformnsg" {

    count               = tonumber(var.temp)

    name                = "nsg${count.index}"

    location            = var.location

    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"

    }

    depends\_on = ["azurerm\_resource\_group.myterraformgroup" ]

}

# Create network interface

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

    count               = tonumber(var.temp)

    name                      = "myNIC"

    location                  = "eastus"

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

    ip\_configuration {

        name                          = "myNicConfiguration"

        subnet\_id                     = element(azurerm\_subnet.myterraformsubnet.\*.id,count.index)

        private\_ip\_address\_allocation = "Dynamic"

        public\_ip\_address\_id          = element(azurerm\_public\_ip.myterraformpublicip.\*.id,count.index)

    }

    tags = {

        environment = "Terraform Demo"

    }

    depends\_on = ["azurerm\_subnet.myterraformsubnet","azurerm\_public\_ip.myterraformpublicip" ]

}

# resource "azurerm\_subnet\_network\_security\_group\_association" "example" {

#   count                     = tonumber(var.temp)

#   subnet\_id                 = each.key

#   network\_security\_group\_id = azurerm\_network\_security\_group.myterraformnsg[each.key].id

#   depends\_on = ["data.azurerm\_subnet.myterraformsubnet"]

#}

*resource* "azurerm\_subnet\_network\_security\_group\_association" "example" {

count               = tonumber(var.temp)

  subnet\_id                 = element(azurerm\_subnet.myterraformsubnet.\*.id,count.index)

  network\_security\_group\_id = element(azurerm\_network\_security\_group.myterraformnsg.\*.id,count.index)

  depends\_on = ["azurerm\_subnet.myterraformsubnet"]

}

# Connect the security group to the network interface

# resource "azurerm\_network\_interface\_security\_group\_association" "example" {

#     count               = tonumber(var.temp)

#    network\_interface\_id      = element(azurerm\_network\_interface.myterraformnic.\*.id,count.index)

#     network\_security\_group\_id = element(azurerm\_network\_security\_group.myterraformnsg.\*.id,count.index)

#     depends\_on = ["azurerm\_network\_interface.myterraformnic"]

# }

# Generate random text for a unique storage account name

*resource* "random\_id" "randomId" {

   count                       = tonumber(var.temp)

    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" {

    count                       = tonumber(var.temp)

    name                        = "diag${element(random\_id.randomId.\*.hex,count.index)}"

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

    location                    = "eastus"

    account\_tier                = "Standard"

    account\_replication\_type    = "LRS"

    tags = {

        environment = "Terraform Demo"

    }

    depends\_on = ["random\_id.randomId"]

}

# Create virtual machine

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

    count                       = tonumber(var.temp)

    name                  = "VM-${count.index}"

    location              = "eastus"

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

    network\_interface\_ids = [element(azurerm\_network\_interface.myterraformnic.\*.id,count.index)]

    size                  = "Standard\_DS1\_v2"

    os\_disk {

        name              = "myOsDisk${count.index}"

        caching           = "ReadWrite"

        storage\_account\_type = "Premium\_LRS"

    }

    source\_image\_reference {

        publisher = "Canonical"

        offer     = "UbuntuServer"

        sku       = "16.04.0-LTS"

        version   = "latest"

    }

    computer\_name  = "myvm"

    admin\_username = "azureuser"

    admin\_password = "Linux@123456"

    disable\_password\_authentication = false

    boot\_diagnostics {

        storage\_account\_uri = element(azurerm\_storage\_account.mystorageaccount.\*.primary\_blob\_endpoint,count.index)

    }

    tags = {

        environment = "Terraform Demo"

    }

    depends\_on = ["azurerm\_subnet\_network\_security\_group\_association.example","azurerm\_storage\_account.mystorageaccount"]

}

Terraform scripts for multiple windows VM deploy:

# Configure the Microsoft Azure Provider

*provider* "azurerm" {

    # The "feature" block is required for AzureRM provider 2.x.

    # If you're using version 1.x, the "features" block is not allowed.

    version = "~>2.0"

    features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

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

  name     = "${var.prefix}-resources"

  location = var.location

}

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

  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

}

*resource* "azurerm\_subnet" "internal" {

    count              = tonumber(var.temp)

  name                 = "internal${count.index}"

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

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

  address\_prefix       = "10.0.${count.index}.0/24"

}

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

    count              = tonumber(var.temp)

  name                = "${var.prefix}-nic${count.index}"

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

  location            = azurerm\_resource\_group.main.location

  ip\_configuration {

    name                          = "internal${count.index}"

    subnet\_id                     = element(azurerm\_subnet.internal.\*.id,count.index)

    private\_ip\_address\_allocation = "Dynamic"

  }

}

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

    count              = tonumber(var.temp)

  name                            = "${var.prefix}-vm${count.index}"

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

  location                        = azurerm\_resource\_group.main.location

  size                            = "Standard\_F2"

  admin\_username                  = "adminuser"

  admin\_password                  = "Password@1234"

  network\_interface\_ids = [

    element(azurerm\_network\_interface.main.\*.id,count.index)

  ]

  source\_image\_reference {

    publisher = "MicrosoftWindowsServer"

    offer     = "WindowsServer"

    sku       = "2016-Datacenter"

    version   = "latest"

  }

  os\_disk {

    storage\_account\_type = "Standard\_LRS"

    caching              = "ReadWrite"

  }

}

# MULTIPLE VM WITH MULTI TIER ARCHITECTURE:

Main.tf

# Configure the Microsoft Azure Provider

*provider* "azurerm" {

    # The "feature" block is required for AzureRM provider 2.x.

    # If you're using version 1.x, the "features" block is not allowed.

    version = "~>2.0"

    features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

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

  name     = var.prefix

  location = var.location

}

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

  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

}

*resource* "azurerm\_subnet" "internal" {

    count              =  3

  name                 = var.subnet[count.index]

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

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

  address\_prefix       = "10.0.${count.index}.0/24"

}

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

    count              = tonumber(var.temp)

  name                = "${var.prefix}-nic${count.index}"

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

  location            = azurerm\_resource\_group.main.location

  ip\_configuration {

    name                          = "internal${count.index}"

    subnet\_id                     = "/subscriptions/b616044f-0c6e-46e7-87e8-3ba295723db7/resourceGroups/${azurerm\_resource\_group.main.name}/providers/Microsoft.Network/virtualNetworks/${azurerm\_virtual\_network.main.name}/subnets/${var.subnet\_ids[count.index]}"

    private\_ip\_address\_allocation = "Dynamic"

  }

}

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

    count                          = tonumber(var.temp)

  name                            = "${var.prefix}-vm${count.index}"

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

  location                        = azurerm\_resource\_group.main.location

  size                            = "Standard\_F2"

  admin\_username                  = "adminuser"

  admin\_password                  = "Password@1234"

  network\_interface\_ids = [

    element(azurerm\_network\_interface.main.\*.id,count.index)

  ]

  source\_image\_reference {

    publisher = "MicrosoftWindowsServer"

    offer     = "WindowsServer"

    sku       = "2016-Datacenter"

    version   = "latest"

  }

  os\_disk {

    storage\_account\_type = "Standard\_LRS"

    caching              = "ReadWrite"

  }

}

Variable.tf:

*variable* "prefix" {

  description = "The prefix which should be used for all resources in this example"

  default = "terraharsh"

}

*variable* "location" {

  description = "The Azure Region in which all resources in this example should be created."

  default = "eastus"

}

*variable* "temp" {

    description = "enter total no of vm to be deploy"

    default = 6

}

# variable "eachsub" {

#   description = "enter no of vm to be deploy in each subnet"

#     default = 2

# }

*variable* "subnet" {

    description = "the name of subnet you need to identify"

    default = ["web", "app" , "db" ]

}

*variable* "substr" {

    default = "/subscriptions/b616044f-0c6e-46e7-87e8-3ba295723db7resourceGroups/terraharsh/providers/Microsoft.Network/virtualNetworks/terraharsh-network/subnets/"

}

*variable* "subnet\_ids" {

    default = [

       "app","app","web","web","db","db",

           ]

}