**Azure Linux Virtual Machine with Public IP Using Terraform Assignment #1**

**Create a new directory for your Terraform project.**

1. **Open the Windows Explorer.**
2. **Navigate to the location where you want to create the directory.**
3. **Right-click and select "New" > "Folder".**
4. **Enter a name for the folder.**
5. **Press "Enter" to create the folder.**

**Create main.tf file in Terraform folder and write this code.**

# Create a resource group, This code creates an Azure Resource Group named "learning" in the "West Europe" location.

resource "azurerm\_resource\_group" "learning" {

  name     = "learning"

  location = "West Europe"

}

# This code creates an Azure virtual network with the name "avn-1" and an address space of "10.0.0.0/16". It is located in the "learning" resource group and region

resource "azurerm\_virtual\_network" "avn-1" {

  name                = "avn-1"

  address\_space       = ["10.0.0.0/16"]

  location            = azurerm\_resource\_group.learning.location

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

}

# This code creates a subnet named "internal" with an address prefix of "10.0.2.0/24". It is assigned to the virtual network named "avn-1" within the "learning" resource group.

resource "azurerm\_subnet" "subnet-1" {

  name                 = "internal"

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

  virtual\_network\_name = azurerm\_virtual\_network.avn-1.name

  address\_prefixes     = ["10.0.2.0/24"]

}

# This code creates a static public IP address resource named "mypublicip" in the "learning" resource group located in the "West Europe" region. It is tagged with "environment: Production".

resource "azurerm\_public\_ip" "mypublicip" {

  name                = "mypublicip"

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

  location            = azurerm\_resource\_group.learning.location

  allocation\_method   = "Static"

  tags = {

    environment = "Production"

  }

}

# This code creates a network interface named "nic-1" within the "learning" resource group and located in the "West Europe" region.

# It is configured with an IP configuration named "internal" that uses a dynamically allocated private IP address and is associated with a public IP address named "mypublicip".

resource "azurerm\_network\_interface" "nic-1" {

  name                = "nic-1"

  location            = azurerm\_resource\_group.learning.location

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

  ip\_configuration {

    name                          = "internal"

    subnet\_id                     = azurerm\_subnet.subnet-1.id

    private\_ip\_address\_allocation = "Dynamic"

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

  }

}

# This code block creates an Azure virtual machine using a Linux image, sets its configuration options such as size and login credentials, and connects it to the specified network interface.

# It also specifies the resource group and location.

resource "azurerm\_linux\_virtual\_machine" "linux-vm" {

  name                            = "linux-vm"

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

  location                        = azurerm\_resource\_group.learning.location

  size                            = "Standard\_F2"

  admin\_username                  = "adminuser"

  admin\_password                  = "Pass$devops123"

  disable\_password\_authentication = false

  network\_interface\_ids = [

    azurerm\_network\_interface.nic-1.id,

  ]

  # This block specifies the OS disk settings with caching and storage account type.

  # The source\_image\_reference block is used to specify the details of the OS image, such as publisher, offer, SKU, and version.

  os\_disk {

    caching              = "ReadWrite"

    storage\_account\_type = "Standard\_LRS"

  }

  source\_image\_reference {

    publisher = "Canonical"

    offer     = "0001-com-ubuntu-server-focal"

    sku       = "20\_04-lts"

    version   = "latest"

  }

}

The terraform plan command is used to create an execution plan which previews the changes that Terraform will make to the infrastructure when the configuration is applied. This command is typically used to check the state of the infrastructure and the changes that will be made before actually applying the changes using terraform apply.

Command: terraform validate

Command: terraform plan

Command: terraform apply