Nikhil

Github repo link:- https://github.com/nikhil58530/ABC-1611.git

5th question

Main.tf

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "4.20.0"

}

}

}

provider "azurerm" {

features {}

subscription\_id = var.subscription\_id

client\_id = var.client\_id

client\_secret = var.client\_secret

tenant\_id = var.tenant\_id

}

resource "azurerm\_resource\_group" "rg" {

name = "rg-devops-demo"

location = var.location

}

# NSG for VM1

resource "azurerm\_network\_security\_group" "nsg\_vm1" {

name = "nsg-vm1"

location = azurerm\_resource\_group.rg.location

resource\_group\_name = azurerm\_resource\_group.rg.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 = "\*"

}

}

# NSG for VM2

resource "azurerm\_network\_security\_group" "nsg\_vm2" {

name = "nsg-vm2"

location = azurerm\_resource\_group.rg.location

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

security\_rule {

name = "AllowInternal"

priority = 1001

direction = "Inbound"

access = "Allow"

protocol = "Tcp"

source\_port\_range = "\*"

destination\_port\_range = "\*"

source\_address\_prefix = var.internal\_cidr

destination\_address\_prefix = var.internal\_cidr

}

}

# VNet1 (for public VM)

resource "azurerm\_virtual\_network" "vnet1" {

name = "vnet1"

address\_space = [var.vnet1\_address\_space]

location = azurerm\_resource\_group.rg.location

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

}

resource "azurerm\_subnet" "subnet1" {

name = "subnet1"

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

virtual\_network\_name = azurerm\_virtual\_network.vnet1.name

address\_prefixes = [var.subnet1\_prefix]

}

# VNet2 (for private VM)

resource "azurerm\_virtual\_network" "vnet2" {

name = "vnet2"

address\_space = [var.vnet2\_address\_space]

location = azurerm\_resource\_group.rg.location

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

}

resource "azurerm\_subnet" "subnet2" {

name = "subnet2"

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

virtual\_network\_name = azurerm\_virtual\_network.vnet2.name

address\_prefixes = [var.subnet2\_prefix]

}

# NSG Associations

resource "azurerm\_subnet\_network\_security\_group\_association" "assoc\_subnet1" {

subnet\_id = azurerm\_subnet.subnet1.id

network\_security\_group\_id = azurerm\_network\_security\_group.nsg\_vm1.id

}

resource "azurerm\_subnet\_network\_security\_group\_association" "assoc\_subnet2" {

subnet\_id = azurerm\_subnet.subnet2.id

network\_security\_group\_id = azurerm\_network\_security\_group.nsg\_vm2.id

}

# VNet Peering

resource "azurerm\_virtual\_network\_peering" "peer1to2" {

name = "peer-vnet1-to-vnet2"

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

virtual\_network\_name = azurerm\_virtual\_network.vnet1.name

remote\_virtual\_network\_id = azurerm\_virtual\_network.vnet2.id

allow\_virtual\_network\_access = true

}

resource "azurerm\_virtual\_network\_peering" "peer2to1" {

name = "peer-vnet2-to-vnet1"

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

virtual\_network\_name = azurerm\_virtual\_network.vnet2.name

remote\_virtual\_network\_id = azurerm\_virtual\_network.vnet1.id

allow\_virtual\_network\_access = true

}

# Public IP for VM1

resource "azurerm\_public\_ip" "vm1\_public\_ip" {

name = "vm1-public-ip"

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

location = azurerm\_resource\_group.rg.location

allocation\_method = "Static"

sku = "Standard"

}

# NIC for VM1

resource "azurerm\_network\_interface" "nic\_vm1" {

name = "nic-vm1"

location = azurerm\_resource\_group.rg.location

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

ip\_configuration {

name = "internal"

subnet\_id = azurerm\_subnet.subnet1.id

private\_ip\_address\_allocation = "Static"

private\_ip\_address = var.vm1\_private\_ip

public\_ip\_address\_id = azurerm\_public\_ip.vm1\_public\_ip.id

}

}

# NIC for VM2

resource "azurerm\_network\_interface" "nic\_vm2" {

name = "nic-vm2"

location = azurerm\_resource\_group.rg.location

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

ip\_configuration {

name = "internal"

subnet\_id = azurerm\_subnet.subnet2.id

private\_ip\_address\_allocation = "Static"

private\_ip\_address = var.vm2\_private\_ip

}

}

# Public VM (VM1)

resource "azurerm\_linux\_virtual\_machine" "vm1" {

name = "vm-public"

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

location = azurerm\_resource\_group.rg.location

size = "Standard\_B1s"

admin\_username = var.admin\_username

network\_interface\_ids = [azurerm\_network\_interface.nic\_vm1.id]

disable\_password\_authentication = true

admin\_ssh\_key {

username = var.admin\_username

public\_key = file(var.ssh\_public\_key\_path)

}

os\_disk {

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

}

# Private VM (VM2)

resource "azurerm\_linux\_virtual\_machine" "vm2" {

name = "vm-private"

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

location = azurerm\_resource\_group.rg.location

size = "Standard\_B1s"

admin\_username = var.admin\_username

network\_interface\_ids = [azurerm\_network\_interface.nic\_vm2.id]

disable\_password\_authentication = true

admin\_ssh\_key {

username = var.admin\_username

public\_key = file(var.ssh\_public\_key\_path)

}

os\_disk {

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

}

---------------------------------------------------------

Terraform.tfvars

subscription\_id = "1d1879cd-7c93-4f97-b9eb-d7546b5a2066"

client\_id = "5d58be33-1ee7-4326-9eee-013ef2a5c5b0"

client\_secret = "Q6q8Q~brBj3gfBhjNtHVsLEGpgm9oc4J2txZ6bzp"

tenant\_id = "ae7b4c65-9e93-4d7f-95a3-1cdd01063a6f"

location = "West US"

vnet1\_address\_space = "10.5.0.0/16"

vnet2\_address\_space = "10.15.0.0/16"

subnet1\_prefix = "10.5.1.0/24"

subnet2\_prefix = "10.15.1.0/24"

internal\_cidr = "10.0.0.0/8"

vm1\_private\_ip = "10.5.1.4"

vm2\_private\_ip = "10.15.1.4"

admin\_username = "azureuser"

ssh\_public\_key\_path = "~/.ssh/id\_rsa.pub"

variables.tf

variable "subscription\_id" {

description = "Azure subscription ID"

type = string

}

variable "client\_id" {

description = "Azure client (application) ID"

type = string

}

variable "client\_secret" {

description = "Azure client secret"

type = string

sensitive = true

}

variable "tenant\_id" {

description = "Azure tenant ID"

type = string

}

variable "location" {

description = "Azure region to deploy resources"

type = string

default = "East US"

}

variable "vnet1\_address\_space" {

description = "Address space for VNet1"

type = string

default = "10.5.0.0/16"

}

variable "vnet2\_address\_space" {

description = "Address space for VNet2"

type = string

default = "10.15.0.0/16"

}

variable "subnet1\_prefix" {

description = "Subnet prefix for subnet1 (VNet1)"

type = string

default = "10.0.1.0/24"

}

variable "subnet2\_prefix" {

description = "Subnet prefix for subnet2 (VNet2)"

type = string

default = "10.1.1.0/24"

}

variable "internal\_cidr" {

description = "Internal CIDR block for communication between VMs"

type = string

default = "10.0.0.0/8"

}

variable "vm1\_private\_ip" {

description = "Static private IP for VM1 (public VM)"

type = string

default = "10.0.1.4"

}

variable "vm2\_private\_ip" {

description = "Static private IP for VM2 (private VM)"

type = string

default = "10.1.1.4"

}

variable "admin\_username" {

description = "Admin username for both VMs"

type = string

default = "azureuser"

}

variable "ssh\_public\_key\_path" {

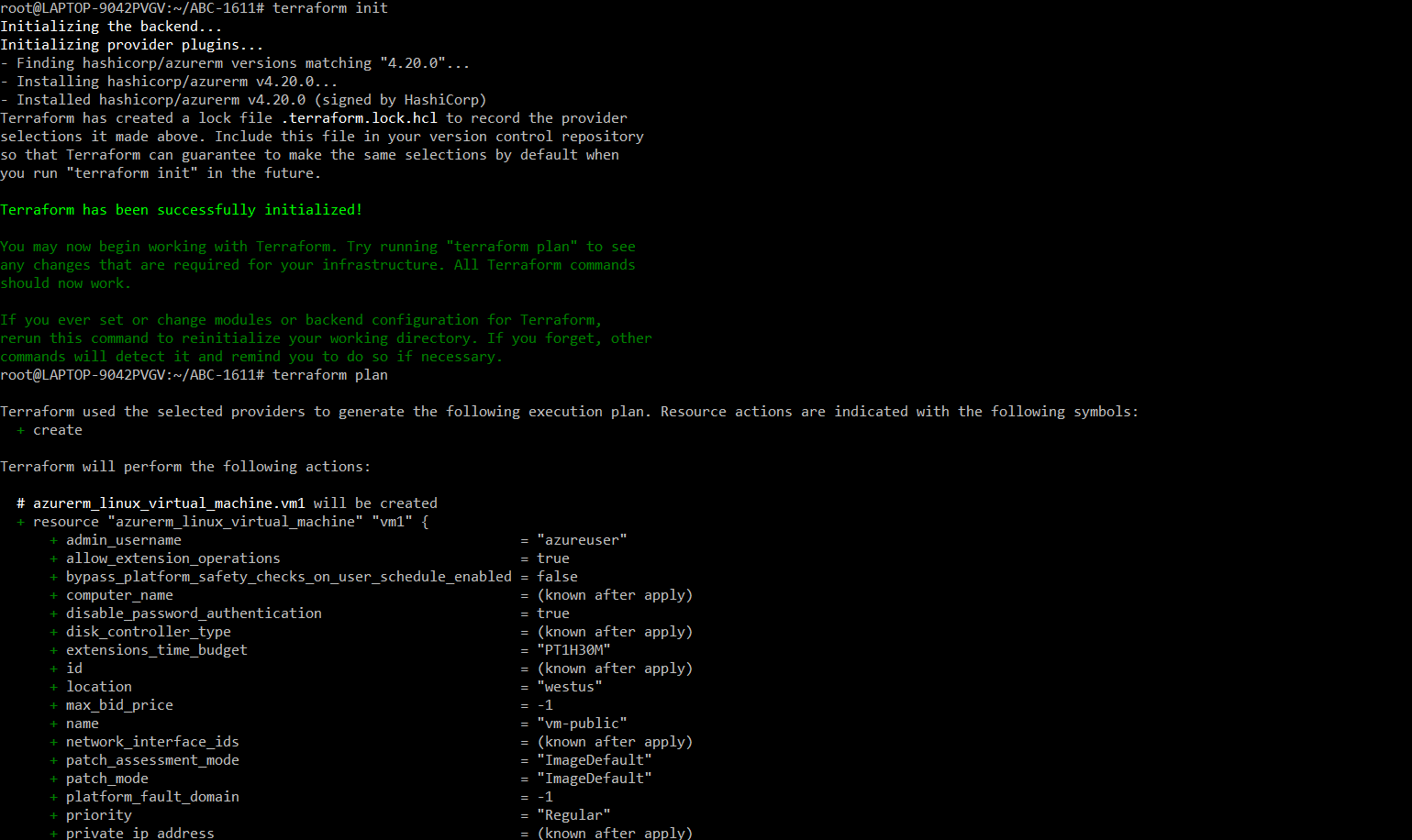
description = "Path to SSH public key for VM login"

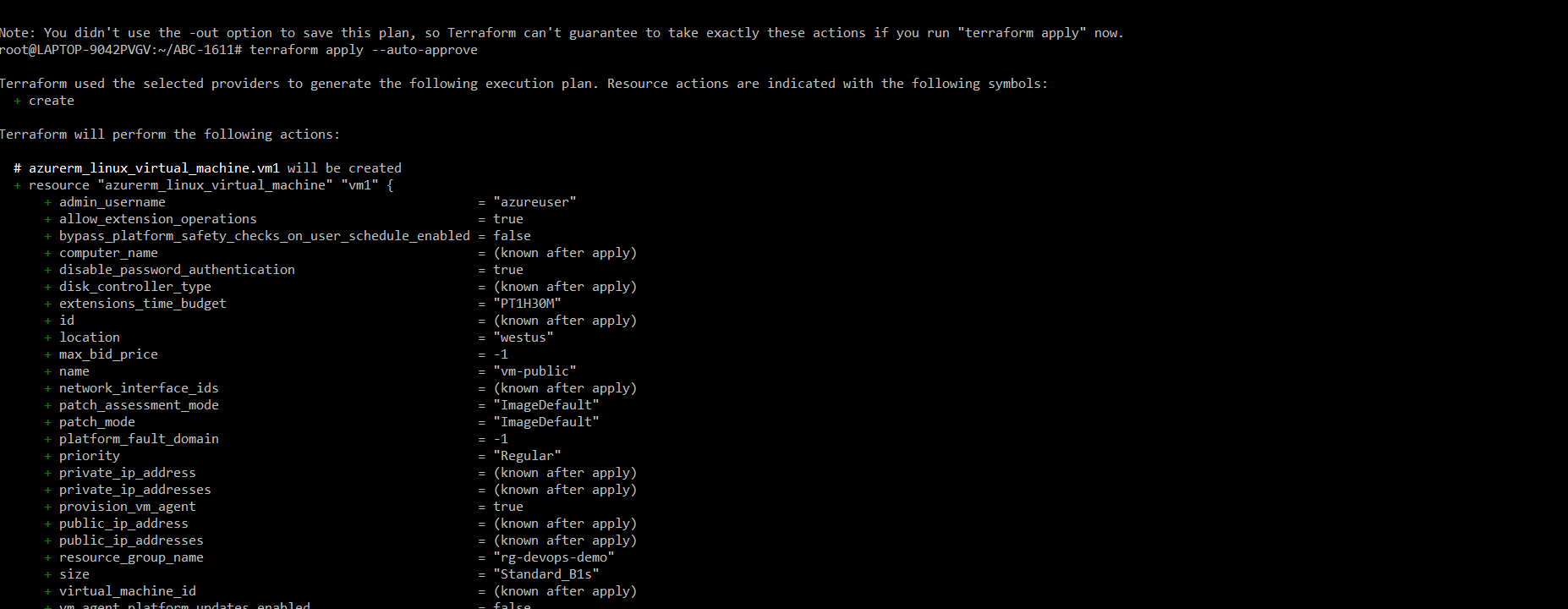
type = string

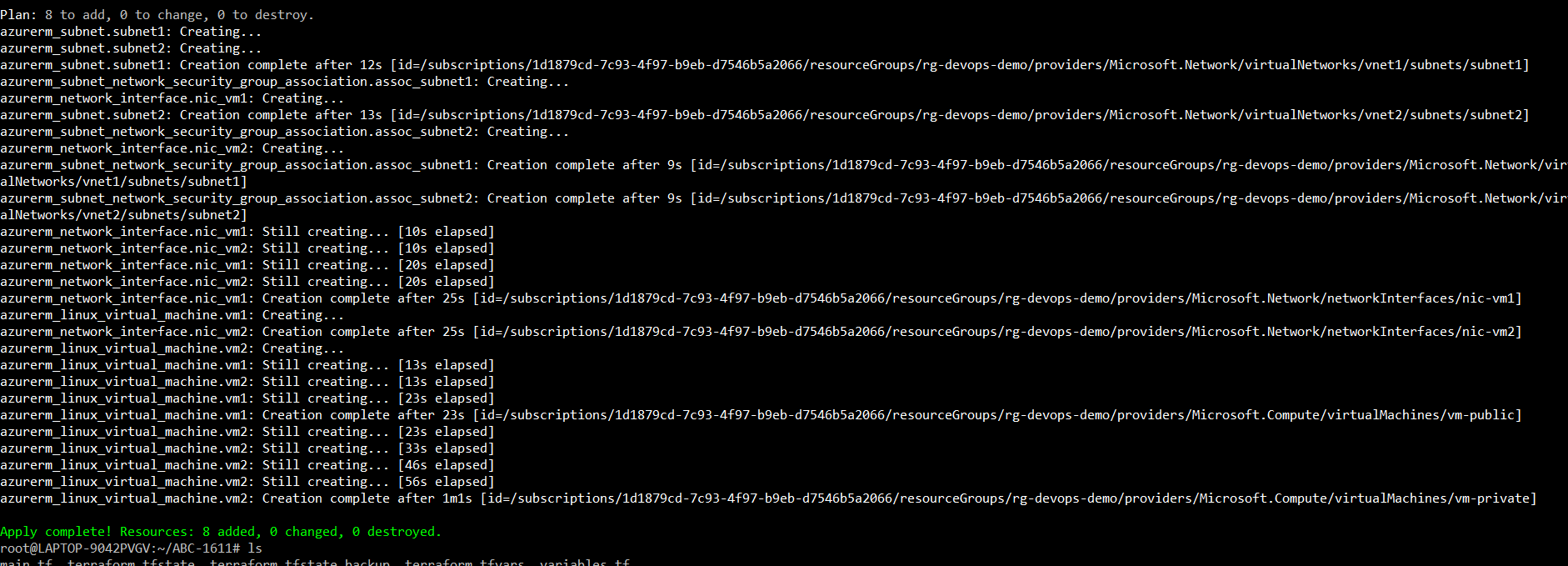
default = "~/.ssh/id\_rsa.pub"

}

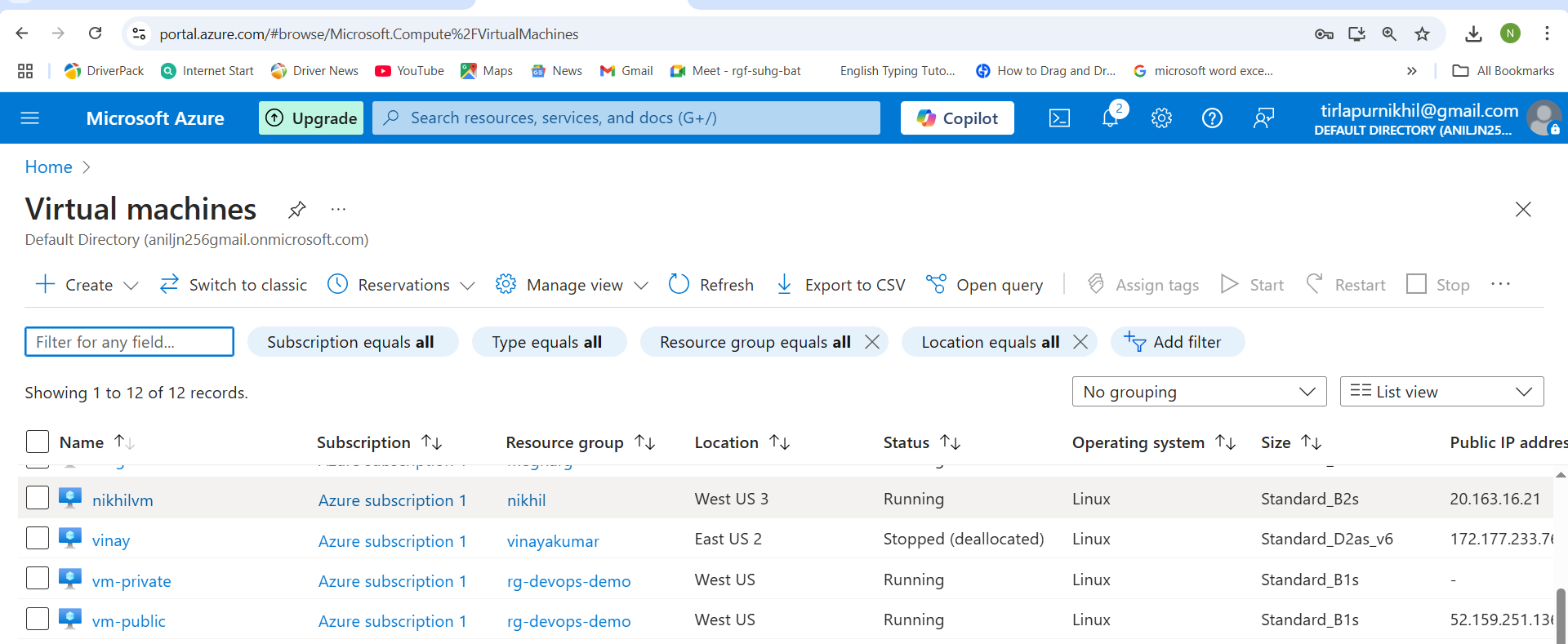
Terraform.init and terraform.plan is shown



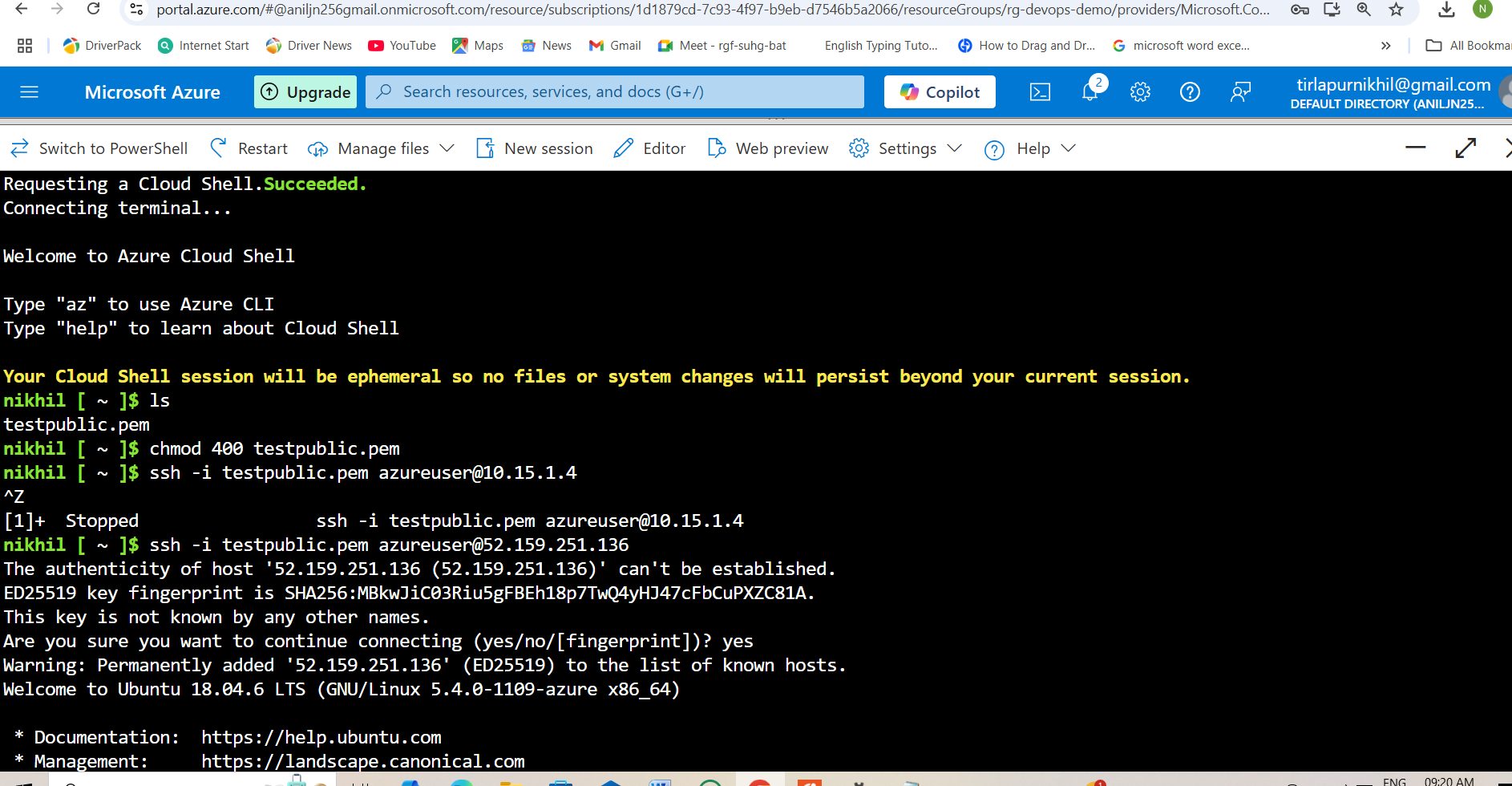




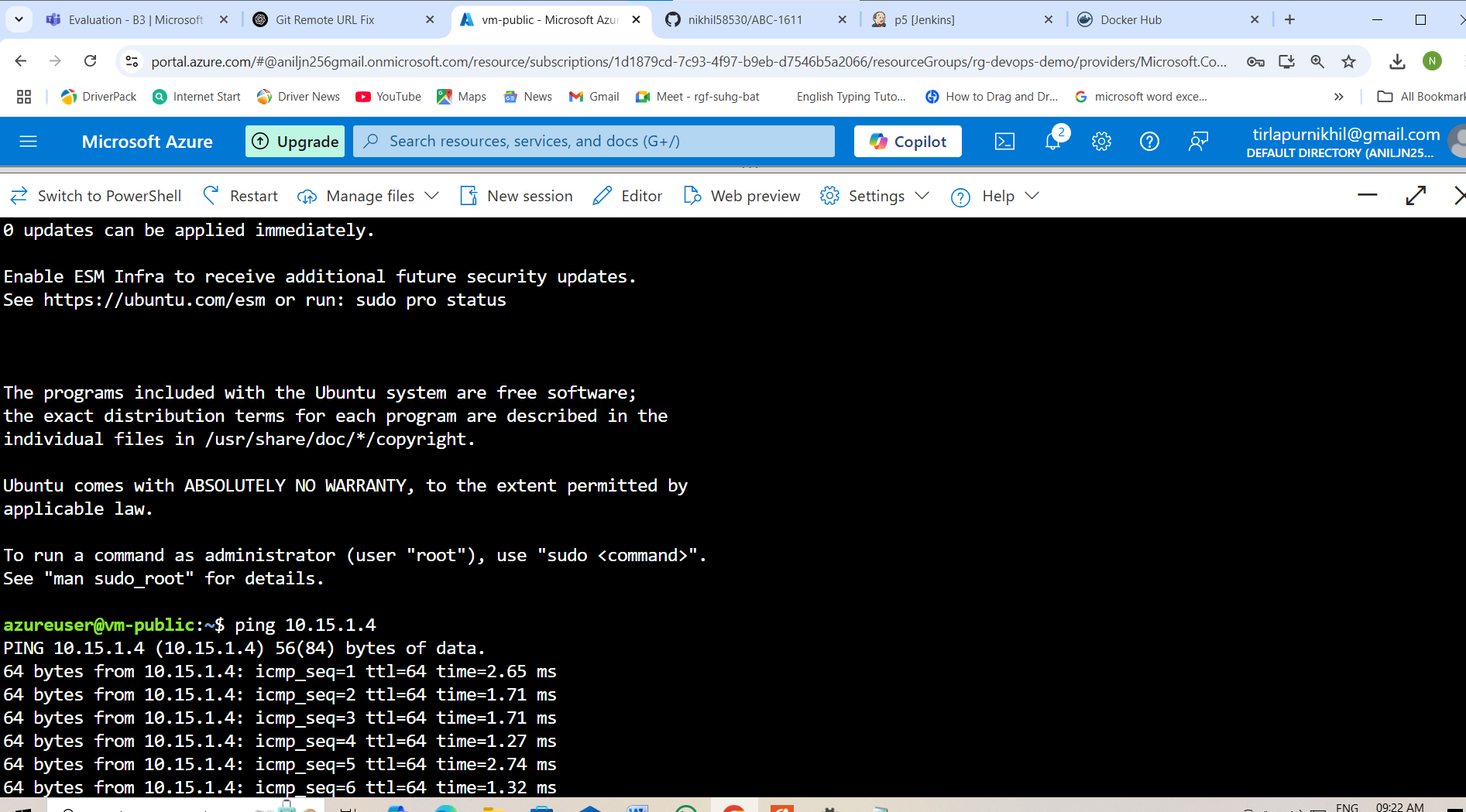
Two vm got created



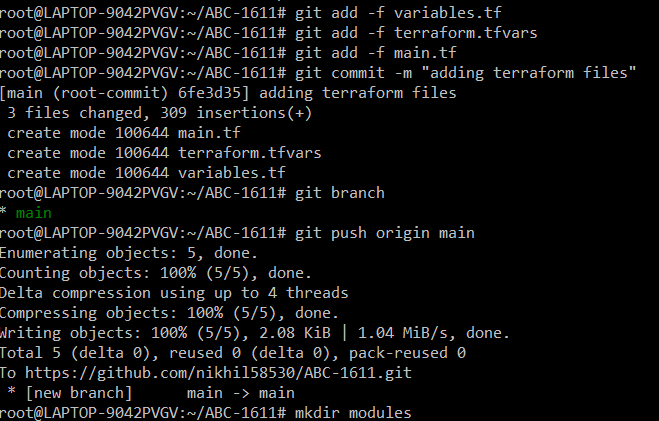
Entering in public vm



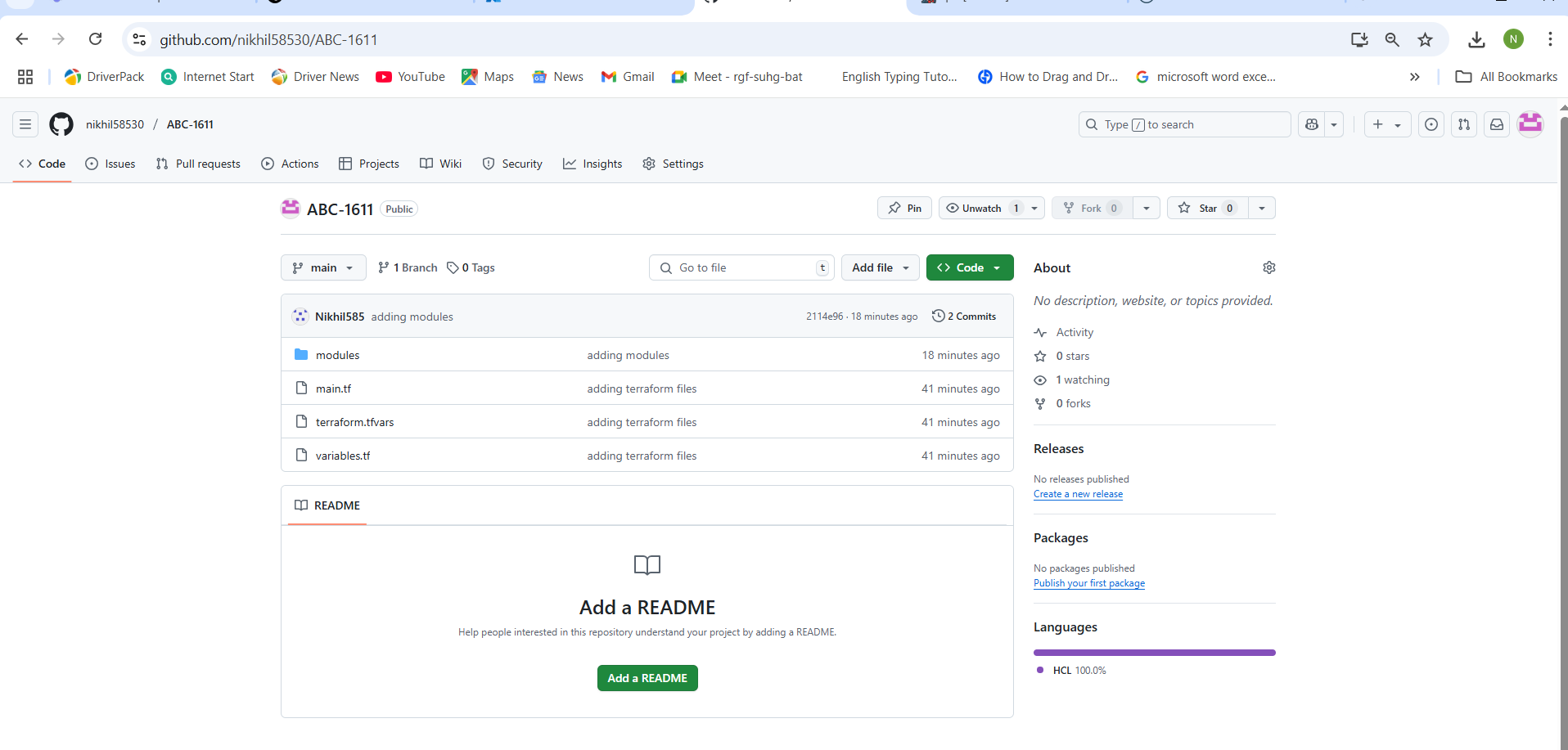
To ensure peering is done in have made ping form vm\_public to vm\_private



Moving all files to github repo



Created repo ABC-1611 and cloned it and pushed all the code in it by removing the secretes form it



Module is created

