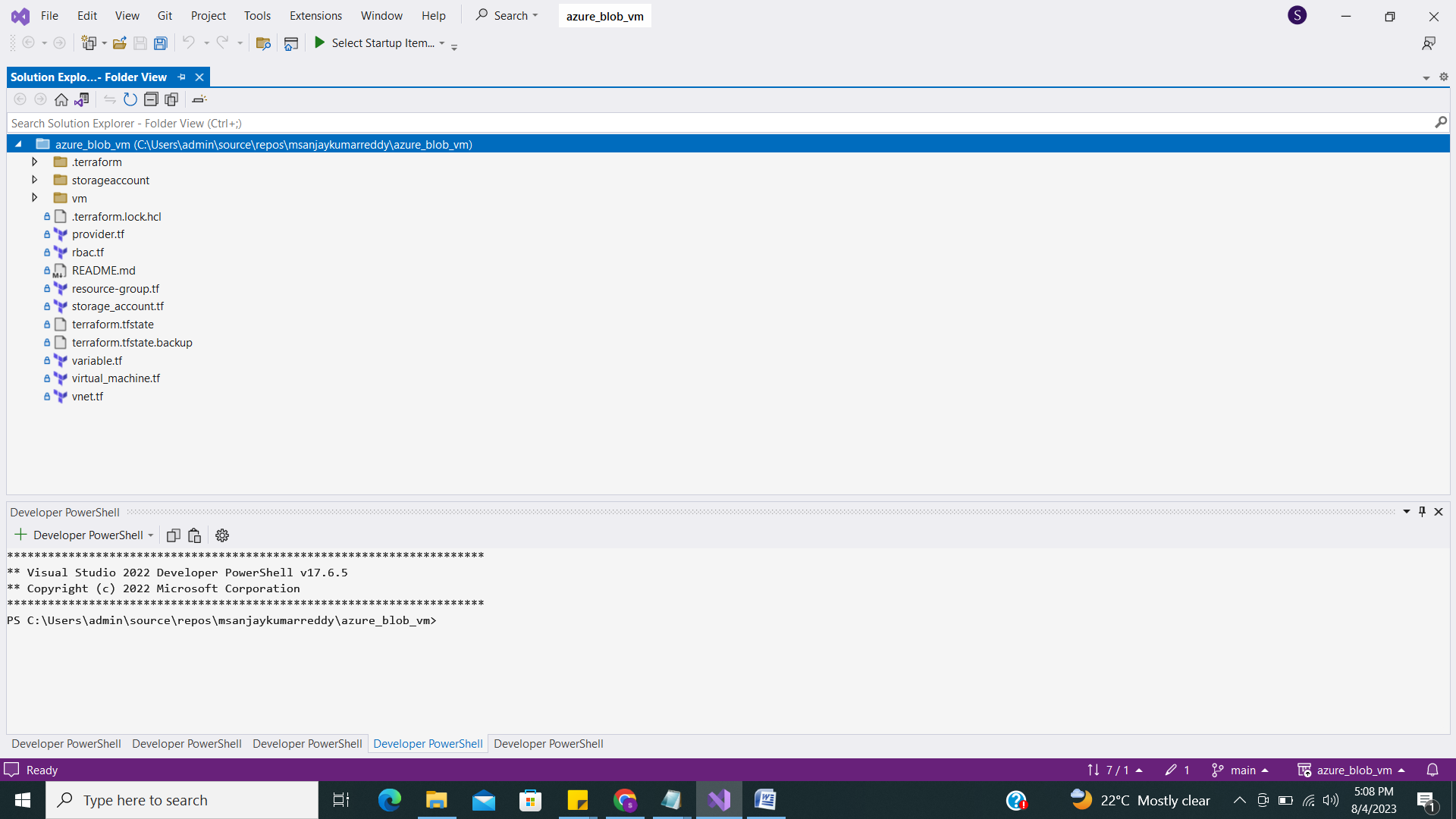
Vm with storage accounts and block devices.  
file structure :  
  
  
  
  
code :  
------------------------------------------------------------------------------------------------------------------------------------------

**Provider.tf**

provider "azurerm" {

features {

}

/\*

client\_id = "129d9ebf-70d0-4caf-b27c-aa35924ce0fe"

client\_secret = "pmv8Q~wVfHrAiQz8r2x.Fb6KuMmzIxvZW6~FEbZc"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

\*/

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

terraform {

required\_providers {

azurerm = {

source = "azurerm"

version = "2.88.1"

}

}

}

**Rbac.tf**

resource "azurerm\_role\_assignment" "storage" {

scope = module.storage\_account.storage\_account\_id

# using azure defined role

role\_definition\_name = "Reader"

principal\_id = module.vm.vm\_pricipal\_id

}

resource "azurerm\_role\_assignment" "container" {

scope = azurerm\_storage\_container.container2.resource\_manager\_id

# using azure defined role

# role\_definition\_name = "Storage Blob Data Contributor"

role\_definition\_name = "Reader"

principal\_id = module.vm.vm\_pricipal\_id

}

**Resourcegroup.tf**  
  
resource "azurerm\_resource\_group" "example" {

name = var.resource\_group\_name

location = "East US"

}  
  
**storageaccounts.tf**  
module "storage\_account" {

source = "./storageaccount"

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

location = azurerm\_resource\_group.example.location

name = "demo12"

white\_list\_ip = ["157.48.67.54"]

whitelist\_subnet\_ids = [azurerm\_subnet.public\_subnet.id]

}

resource "azurerm\_storage\_container" "container" {

name = "demo"

storage\_account\_name = module.storage\_account.storage\_account\_name

container\_access\_type = "private"

depends\_on = [

module.storage\_account

]

}

resource "azurerm\_storage\_container" "container2" {

name = "demo1"

storage\_account\_name = module.storage\_account.storage\_account\_name

container\_access\_type = "private"

depends\_on = [

module.storage\_account

]

}

**Variable.tf**  
variable "resource\_group\_name" {

type = string

default = "qwerty12344321"

}  
  
**virtual\_machine.tf**  
module "vm" {

source = "./vm/"

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

location = azurerm\_resource\_group.example.location

public\_key\_path = "C:/Users/wv3cxq/.ssh/id\_rsa.pub"

name = "demo"

subnet\_id = azurerm\_subnet.public\_subnet.id

}  
  
**vnet.tf**  
resource "azurerm\_virtual\_network" "example" {

name = "example-network"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.example.location

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

}

resource "azurerm\_subnet" "public\_subnet" {

name = "public\_subnet"

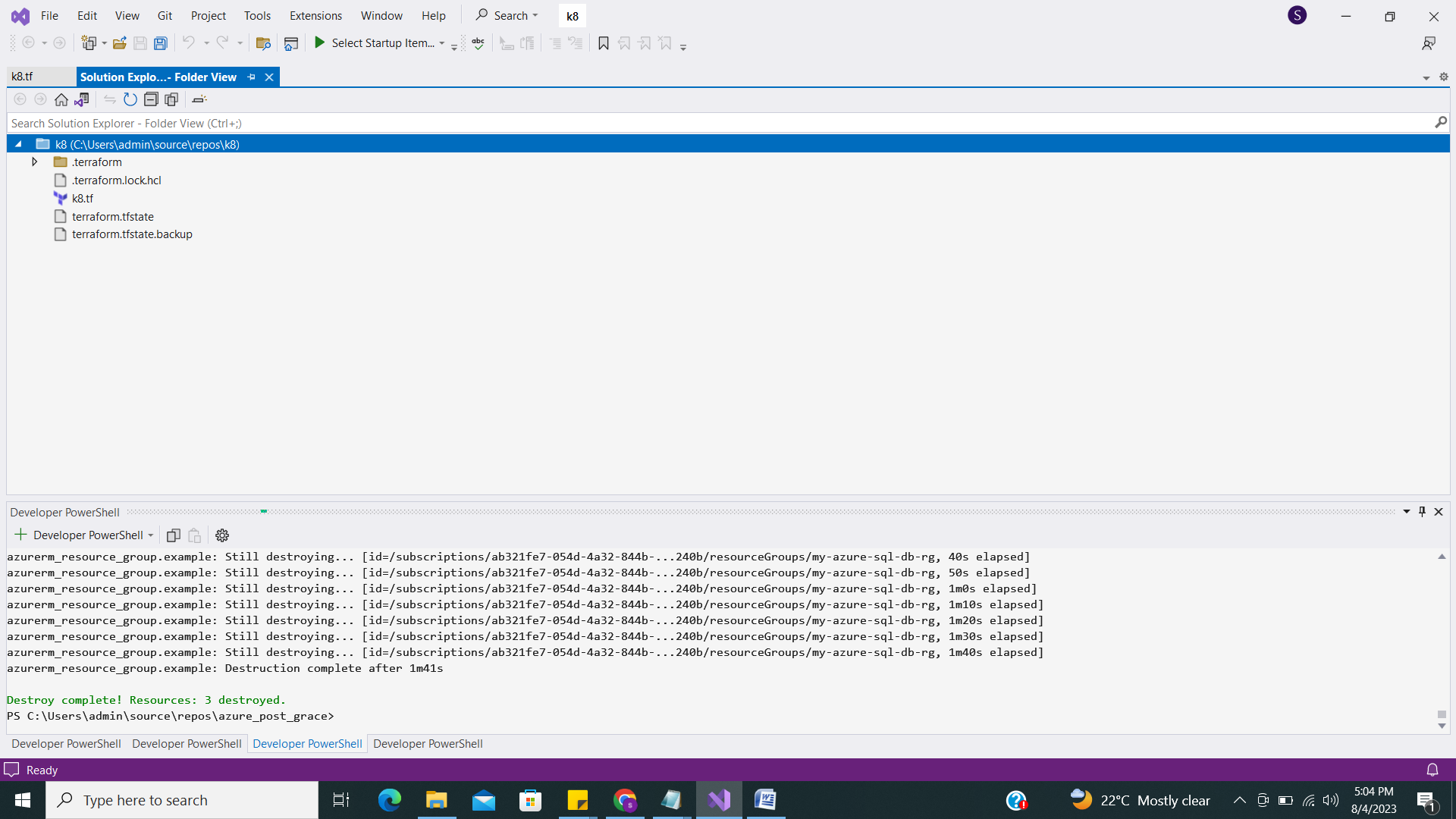
resource\_group\_name = azurerm\_resource\_group.example.name

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

address\_prefixes = ["10.0.1.0/24"]

service\_endpoints = ["Microsoft.Storage"]

}  
------------------------------------------------------------------------------------------------------------------------------------------

k8   
file structure   
  
  
code :  
provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "rgkuber" {

name = "kuber"

location = "Central India" # Change this to your desired location

}

resource "azurerm\_kubernetes\_cluster" "example" {

name = "my-k8s-cluster"

location = azurerm\_resource\_group.rgkuber.location

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

dns\_prefix = "my-k8s-cluster" # Replace with your desired DNS prefix

kubernetes\_version = "1.25.5" # Replace with your desired Kubernetes version

default\_node\_pool {

name = "default"

node\_count = 2

vm\_size = "Standard\_DS2\_v2" # Replace with your desired VM size

}

service\_principal {

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda" # Replace with your Service Principal's client\_id

client\_secret = "BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR" # Replace with your Service Principal's client\_secret

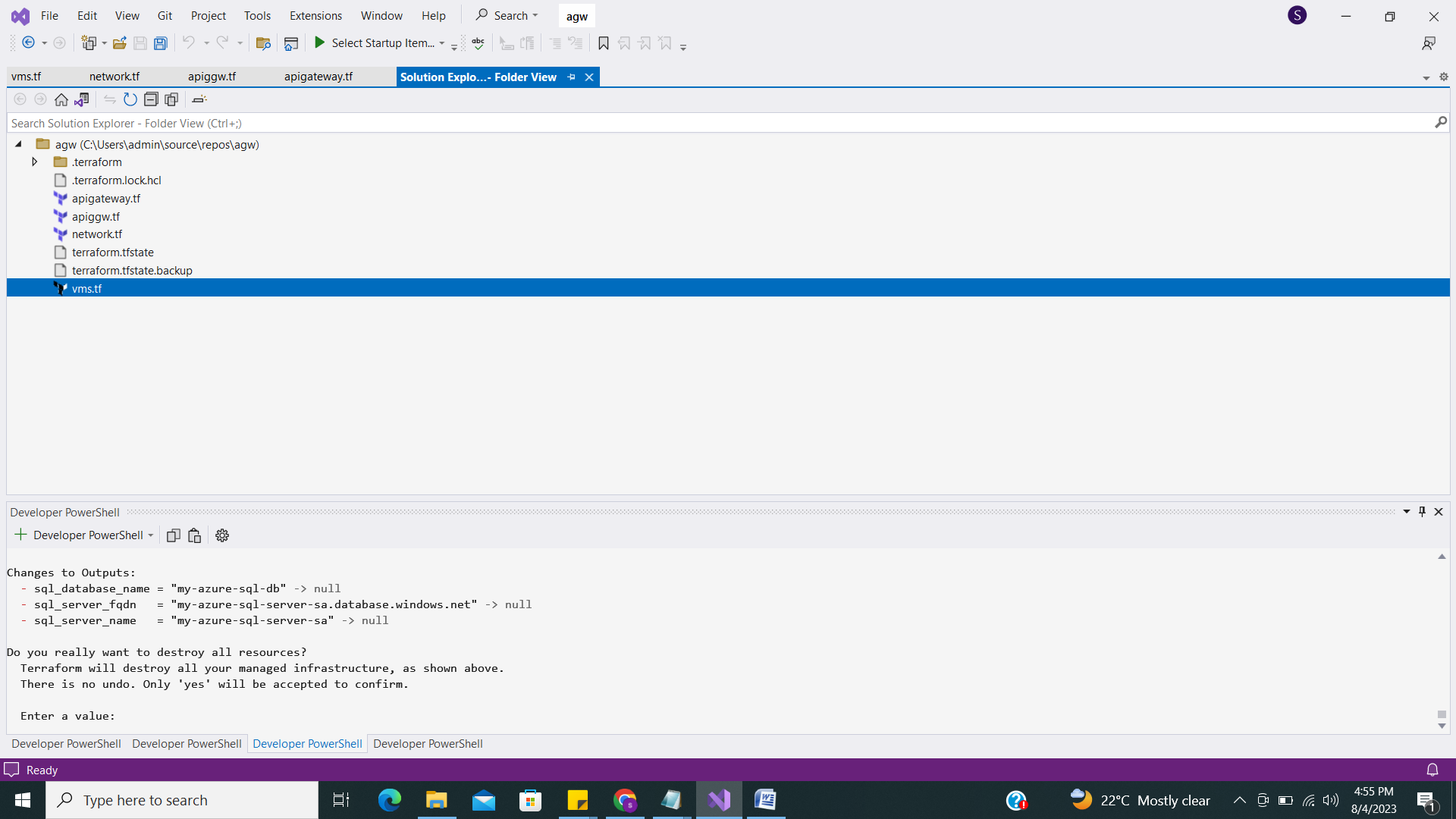
}

tags = {

environment = "dev"

}

}

Application Gateway  
  
file structure:   


Code :   
-----------------------------------------------------------------------------------------------------------------------------------------

Main.tf  
apigatway.tf  
resource "azurerm\_application\_gateway" "example" {

name = "exampleAppGateway"

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

location = azurerm\_resource\_group.example.location

sku {

name = "Standard\_Small"

tier = "Standard"

capacity = 2

}

gateway\_ip\_configuration {

name = "exampleGatewayConfig"

subnet\_id = azurerm\_subnet.example.id

}

frontend\_port {

name = "exampleFrontendPort"

port = 80

}

frontend\_ip\_configuration {

name = "exampleFrontendIP"

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

}

http\_listener {

name = "exampleHTTPListener"

frontend\_ip\_configuration\_name = "exampleFrontendIP"

frontend\_port\_name = "exampleFrontendPort"

protocol = "Http"

}

request\_routing\_rule {

name = "exampleRoutingRule"

rule\_type = "Basic"

http\_listener\_name = azurerm\_application\_gateway.example.http\_listener[0].name

backend\_address\_pool\_name = azurerm\_application\_gateway.example.backend\_address\_pool[0].name

backend\_http\_settings\_name = azurerm\_application\_gateway.example.http\_settings[0].name

}

backend\_address\_pool {

name = "exampleBackendAddressPool"

}

backend\_http\_settings {

name = "exampleHTTPSettings"

cookie\_based\_affinity = "Disabled"

path = "/"

port = 80

protocol = "Http"

request\_timeout = 30

}

http\_request\_routing\_rule {

name = "exampleReqRoutingRule"

rule\_type = "PathBasedRouting"

backend\_address\_pool\_name = azurerm\_application\_gateway.example.backend\_address\_pool[0].name

backend\_http\_settings\_name = azurerm\_application\_gateway.example.http\_settings.name

path\_based\_routing\_configuration {

path\_rule {

name = "examplePathRule"

paths = ["/app1/\*"]

backend\_address\_pool\_name = azurerm\_application\_gateway.example.backend\_address\_pool[0].name

backend\_http\_settings\_name = azurerm\_application\_gateway.example.http\_settings.name

}

}

}

}  
  
**apiggw.tf.**  
  
provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "example" {

name = "exampleResourceGroup"

location = "East US"

}

resource "azurerm\_virtual\_network" "example" {

name = "exampleVNET"

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

location = azurerm\_resource\_group.example.location

address\_space = ["10.0.0.0/16"]

}

resource "azurerm\_subnet" "example" {

name = "exampleSubnet"

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

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

address\_prefixes = ["10.0.1.0/24"]

}

resource "azurerm\_public\_ip" "example" {

name = "examplePublicIP"

location = azurerm\_resource\_group.example.location

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

allocation\_method = "Static"

}

**network.tf**  
resource "azurerm\_network\_interface" "music" {

name = "music-nic"

location = azurerm\_resource\_group.example.location

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

ip\_configuration {

name = "music-ip-config"

subnet\_id = azurerm\_subnet.example.id

private\_ip\_address\_allocation = "Dynamic"

}

}

resource "azurerm\_network\_interface" "movie" {

name = "movie-nic"

location = azurerm\_resource\_group.example.location

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

ip\_configuration {

name = "movie-ip-config"

subnet\_id = azurerm\_subnet.example.id

private\_ip\_address\_allocation = "Dynamic"

}

}

**vms.tf**

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

name = "music"

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

location = azurerm\_resource\_group.example.location

size = "Standard\_B1s"

admin\_username = "adminuser"

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

admin\_ssh\_key {

username = "adminuser"

public\_key = file("~/.ssh/id\_rsa.pub") # Replace with the path to your public SSH key

}

os\_disk {

name = "musicOSDisk"

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

computer\_name = "music-vm"

}

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

name = "movie"

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

location = azurerm\_resource\_group.example.location

size = "Standard\_B1s"

admin\_username = "adminuser"

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

admin\_ssh\_key {

username = "adminuser"

public\_key = file("~/.ssh/id\_rsa.pub") # Replace with the path to your public SSH key

}

os\_disk {

name = "movieOSDisk"

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

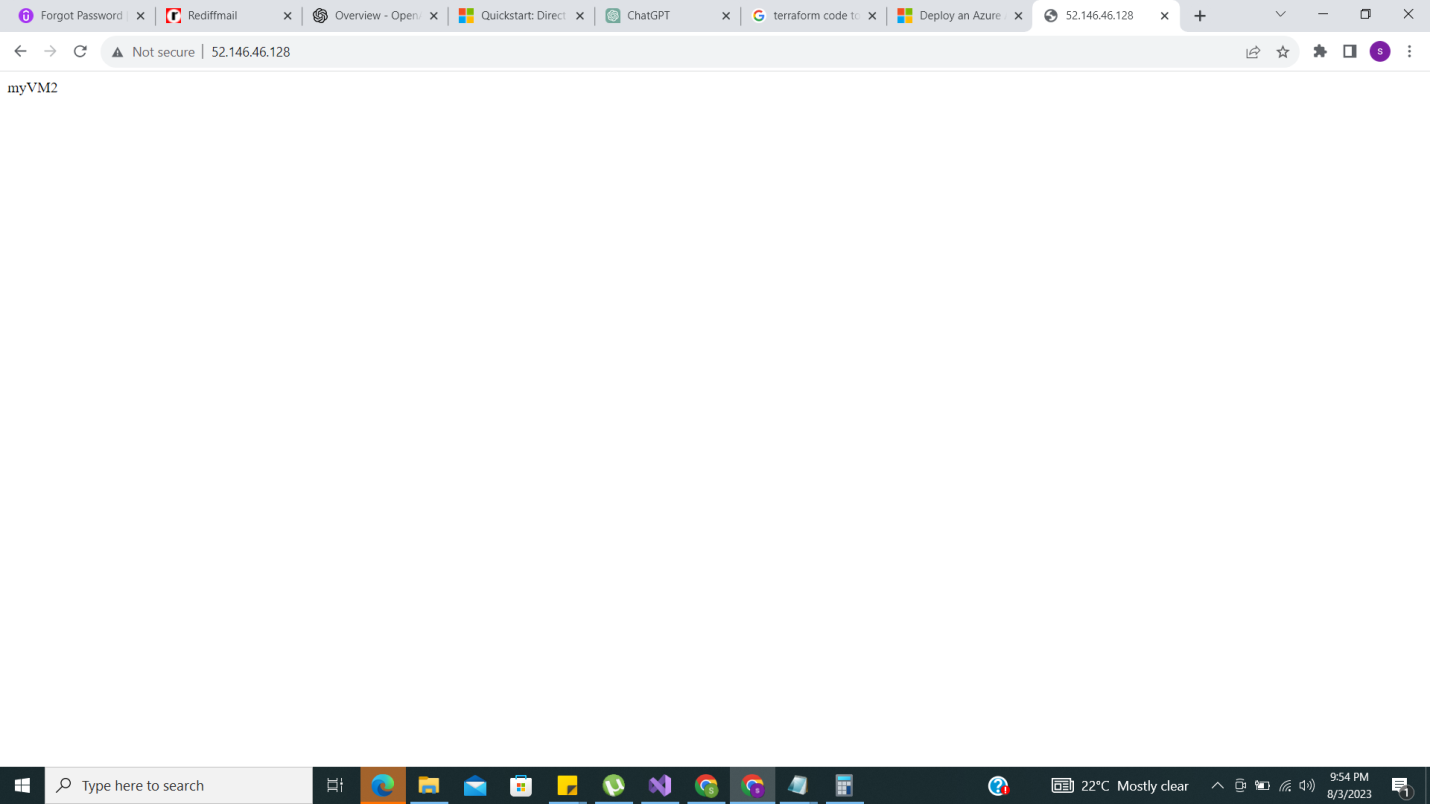
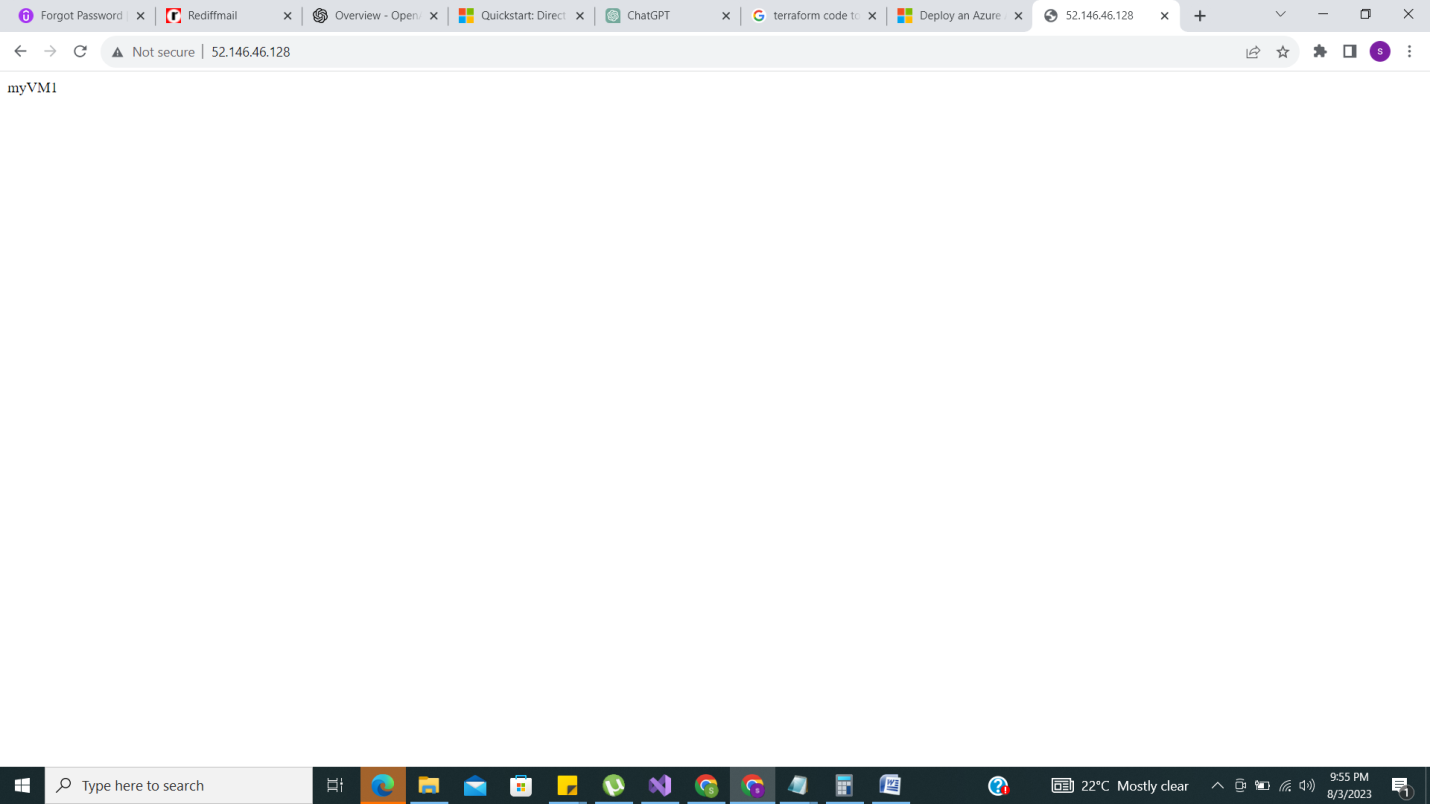
sku = "18.04-LTS"

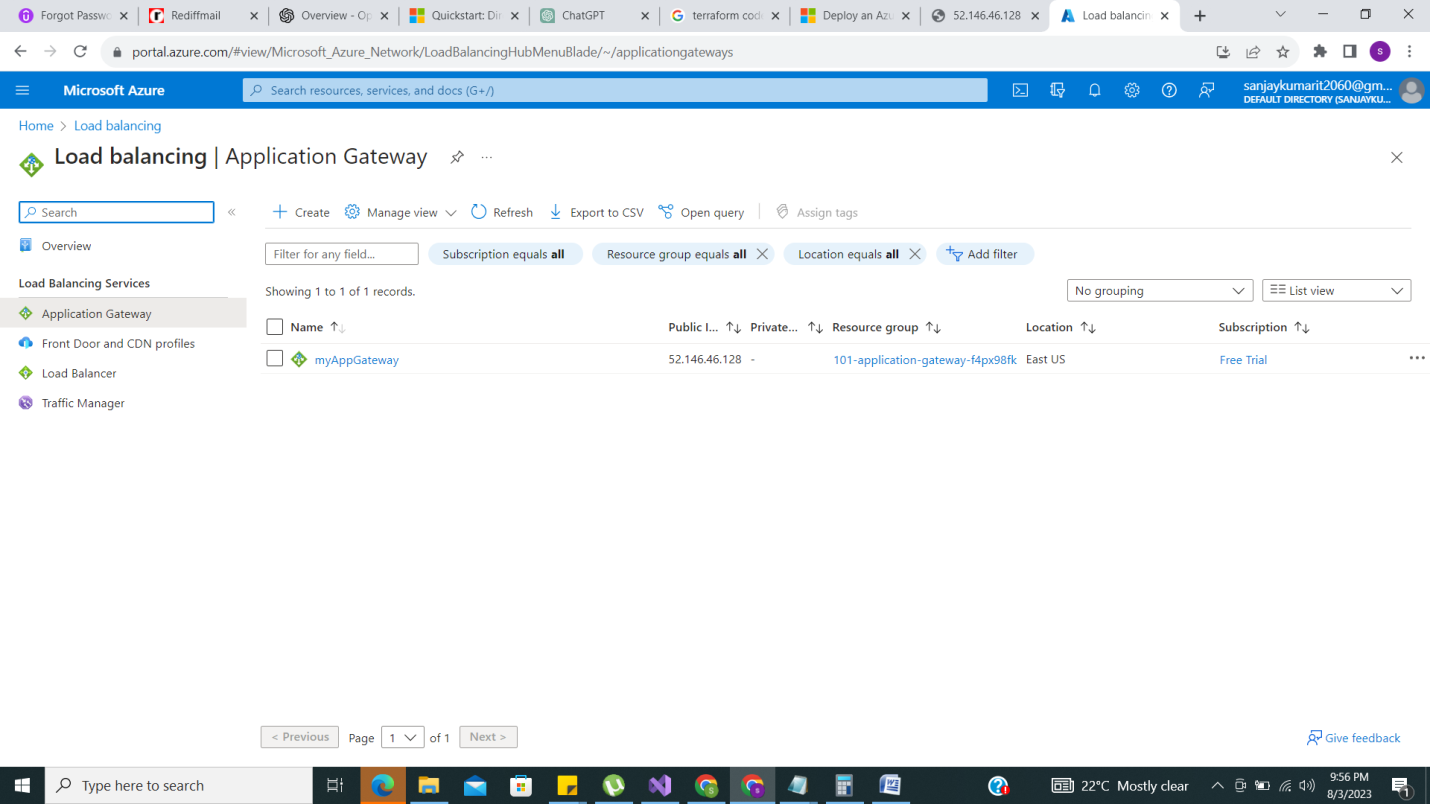
version = "latest"

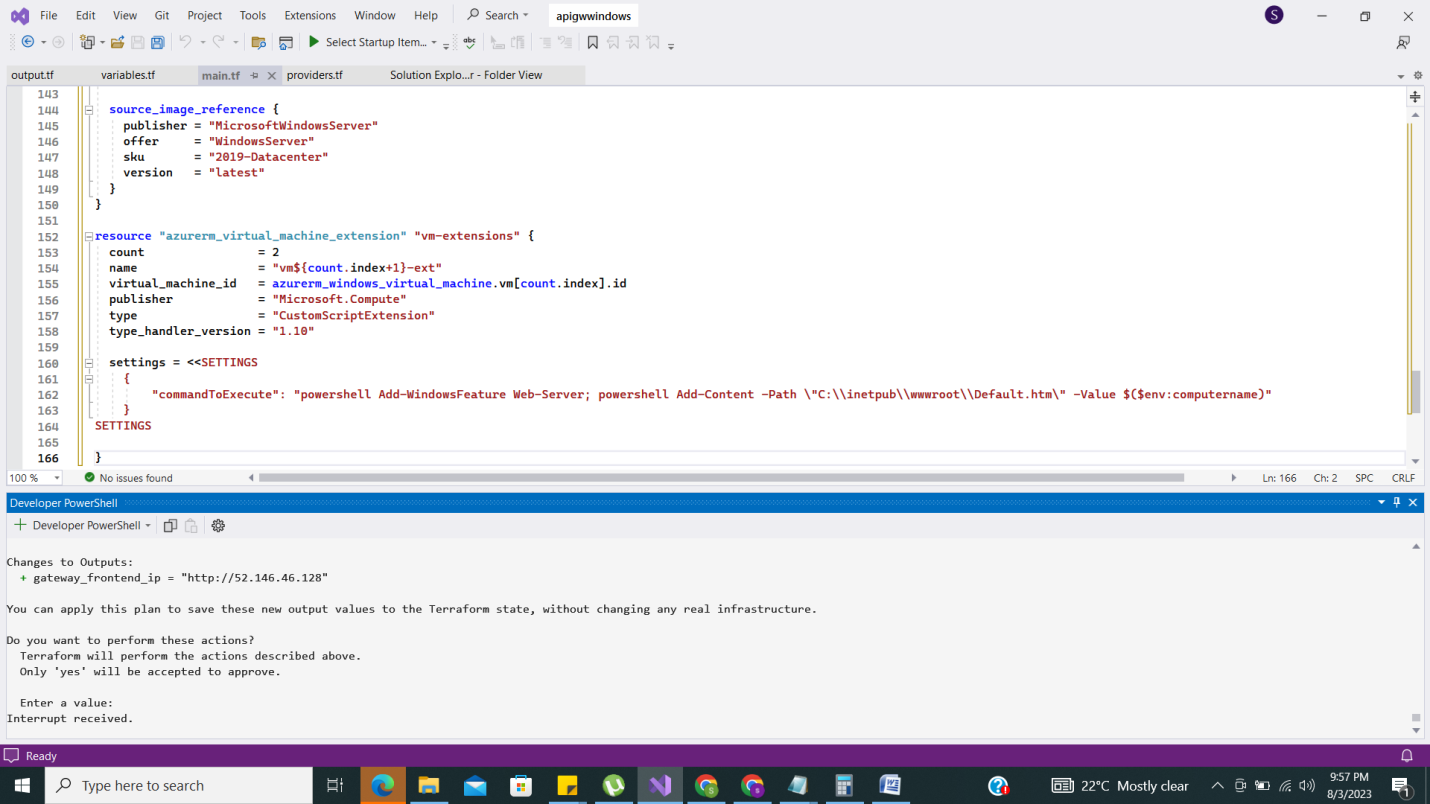
}

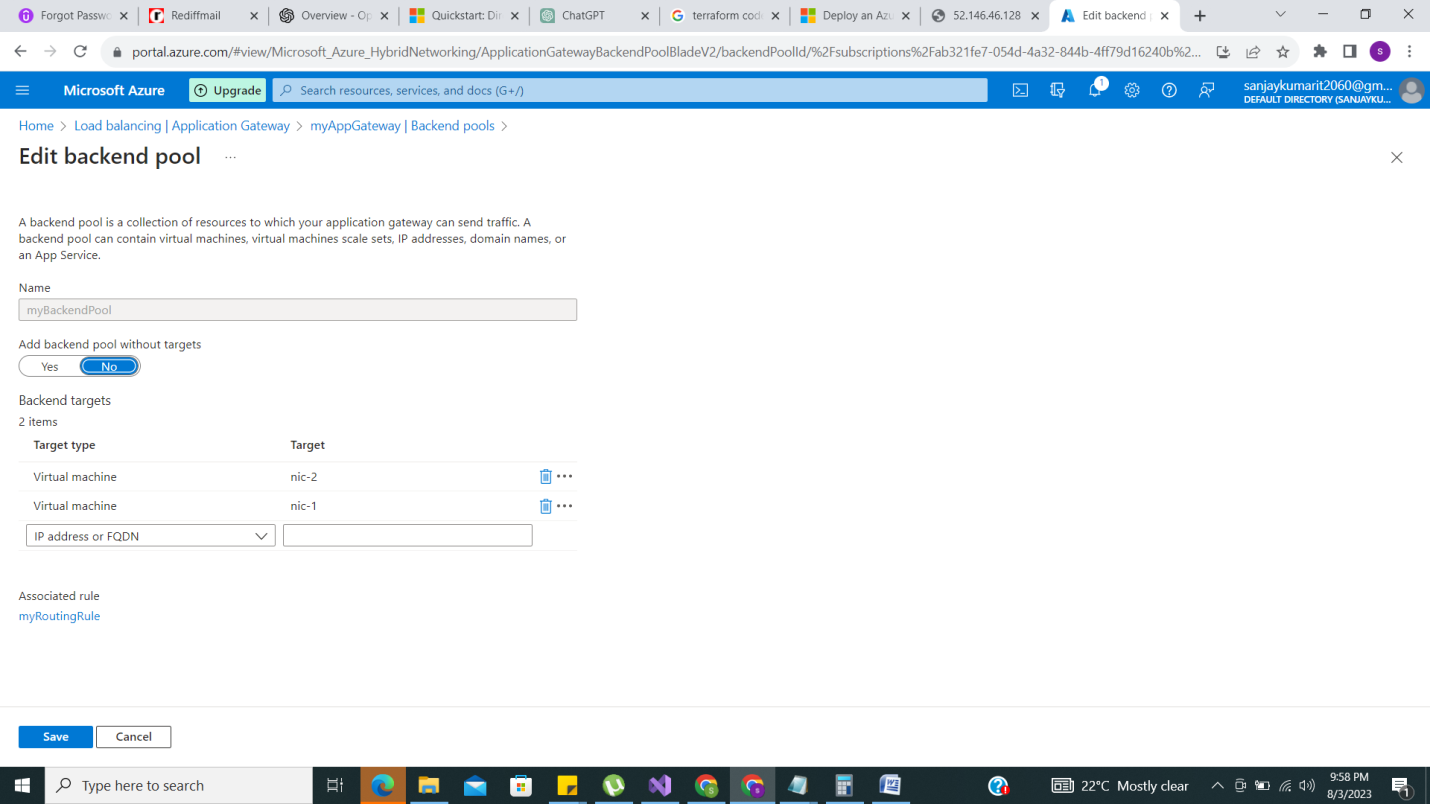
computer\_name = "movie-vm"

}

Output  
  
  
  




azurerm\_sql\_server.

Terraform Code .  
  
------------------------------------------------------------------------------------------------------------------------------------------

provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "example" {

name = "my-azure-sql-db-rg"

location = "East US" # Change this to your preferred location

}

resource "azurerm\_mssql\_server" "example" {

name = "my-azure-sql-server-sa"

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

location = azurerm\_resource\_group.example.location

version = "12.0" # Change to the desired SQL Server version

administrator\_login = "sqladmin"

administrator\_login\_password = "P@ssw0rd123!" # Change this to your desired password

/\*

sku {

name = "GP\_Gen5\_2"

tier = "GeneralPurpose"

capacity = 2

family = "Gen5"

}

\*/

}

resource "azurerm\_mssql\_database" "example" {

name = "my-azure-sql-db"

server\_id = azurerm\_mssql\_server.example.id

collation = "SQL\_Latin1\_General\_CP1\_CI\_AS" # Change to your desired collation

}

output "sql\_server\_name" {

value = azurerm\_mssql\_server.example.name

}

output "sql\_server\_fqdn" {

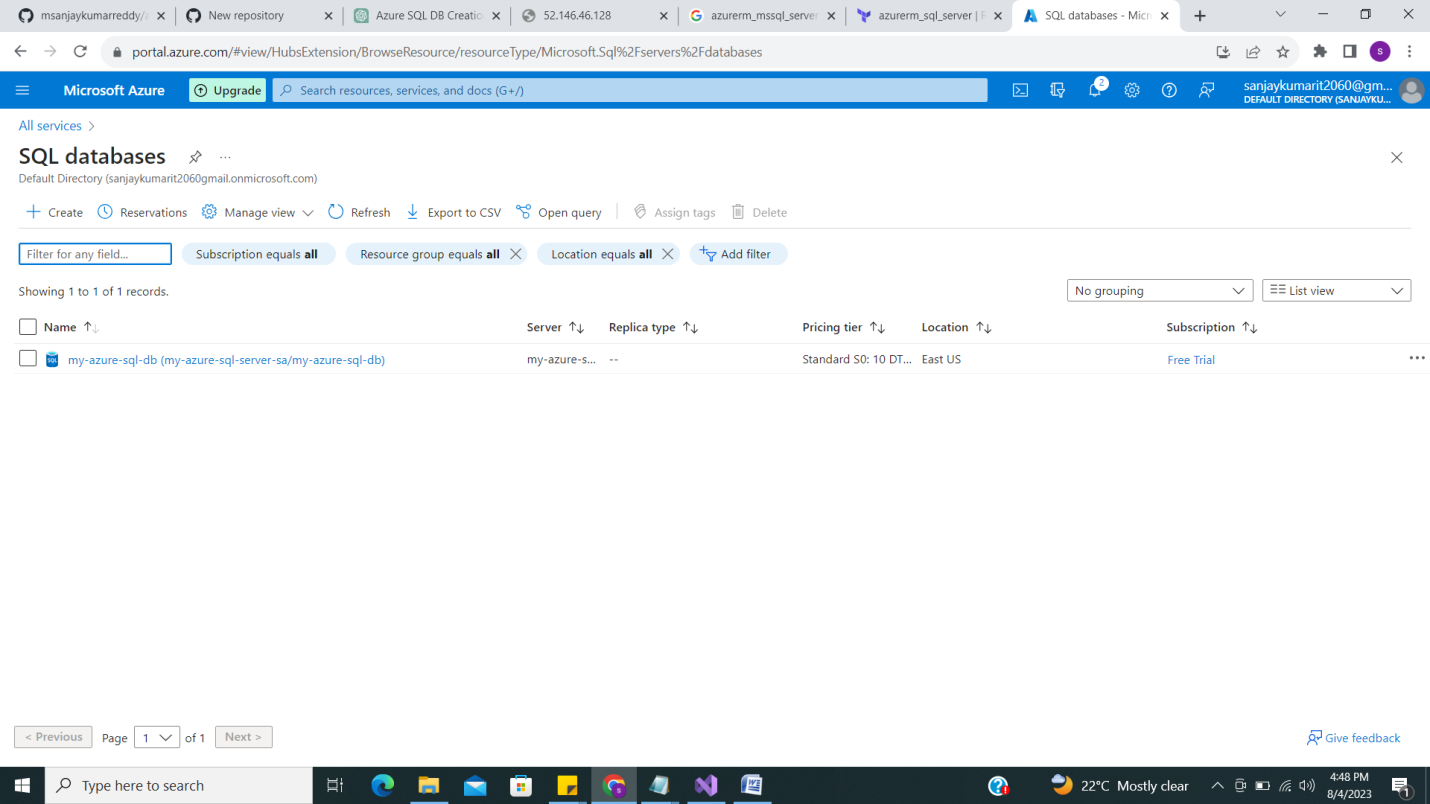
value = azurerm\_mssql\_server.example.fully\_qualified\_domain\_name

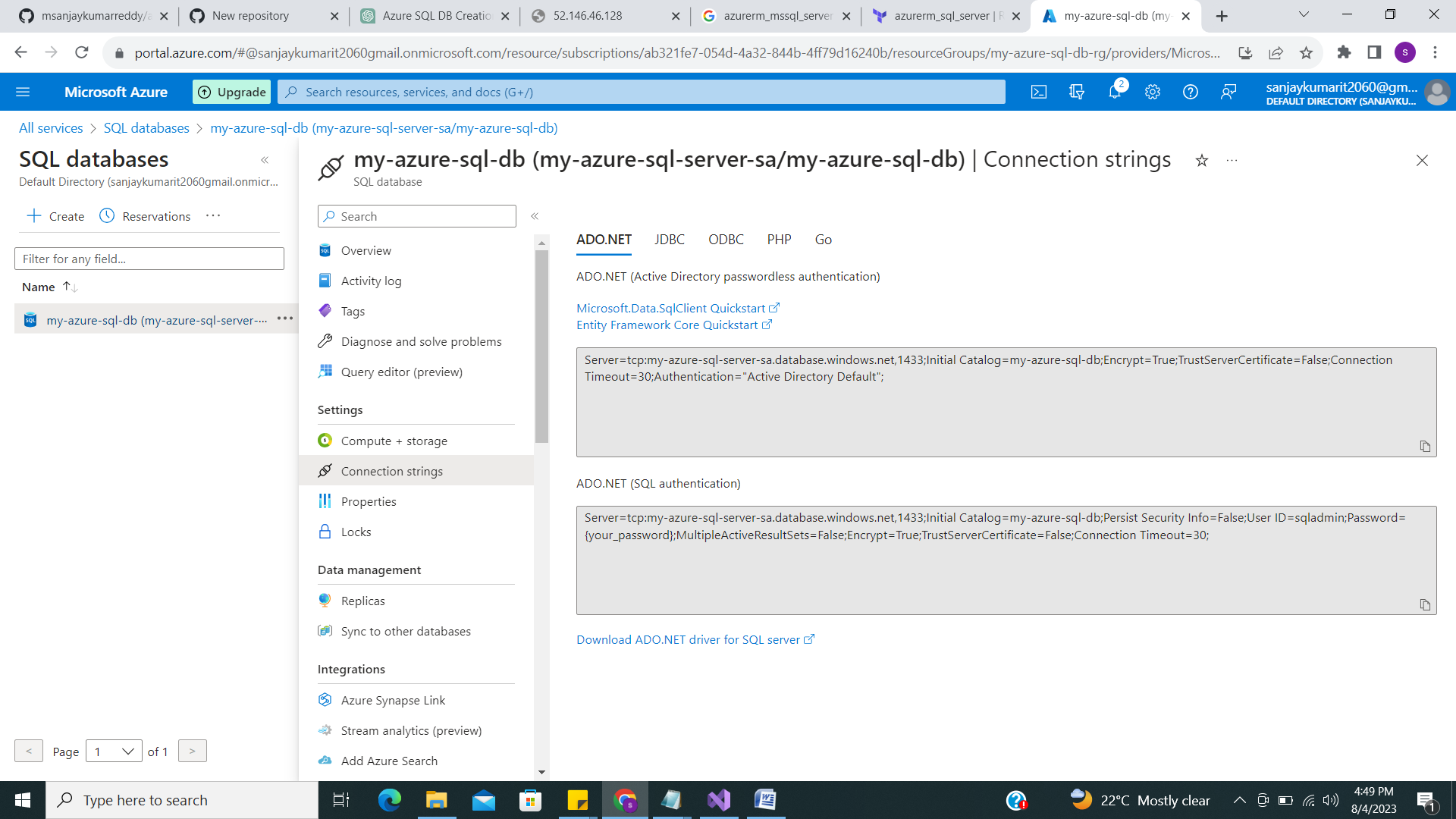
}

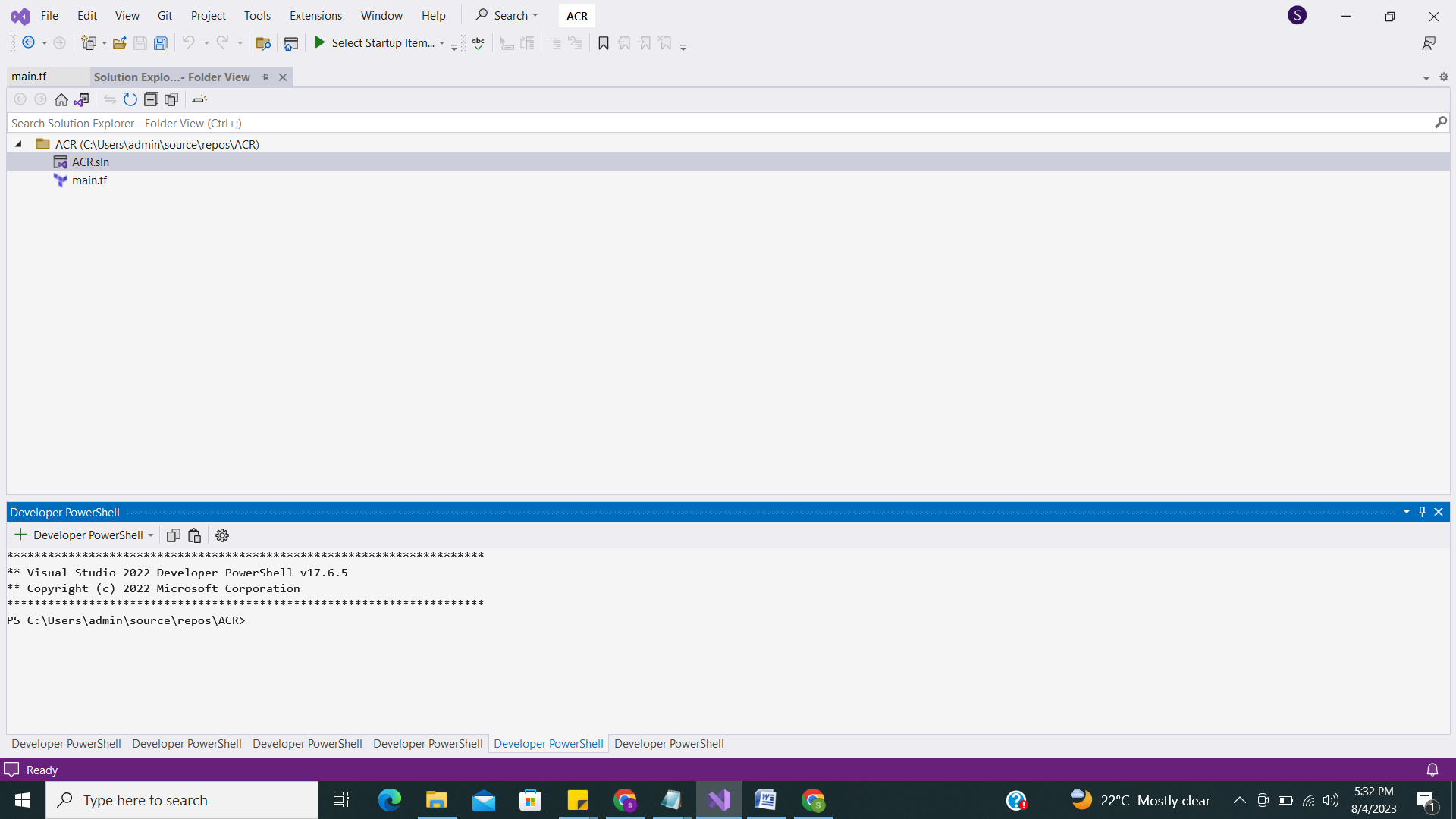
output "sql\_database\_name" {

value = azurerm\_mssql\_database.example.name

}

Screen shots :  
  




**azure container registry :  
  
  
  
code :**provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "example" {

name = "my-container-registry-rg"

location = "East US" # Change this to your preferred location

}

resource "azurerm\_container\_registry" "example" {

name = "mycontainerregistrypip"

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

location = azurerm\_resource\_group.example.location

sku = "Basic"

admin\_enabled = true

}

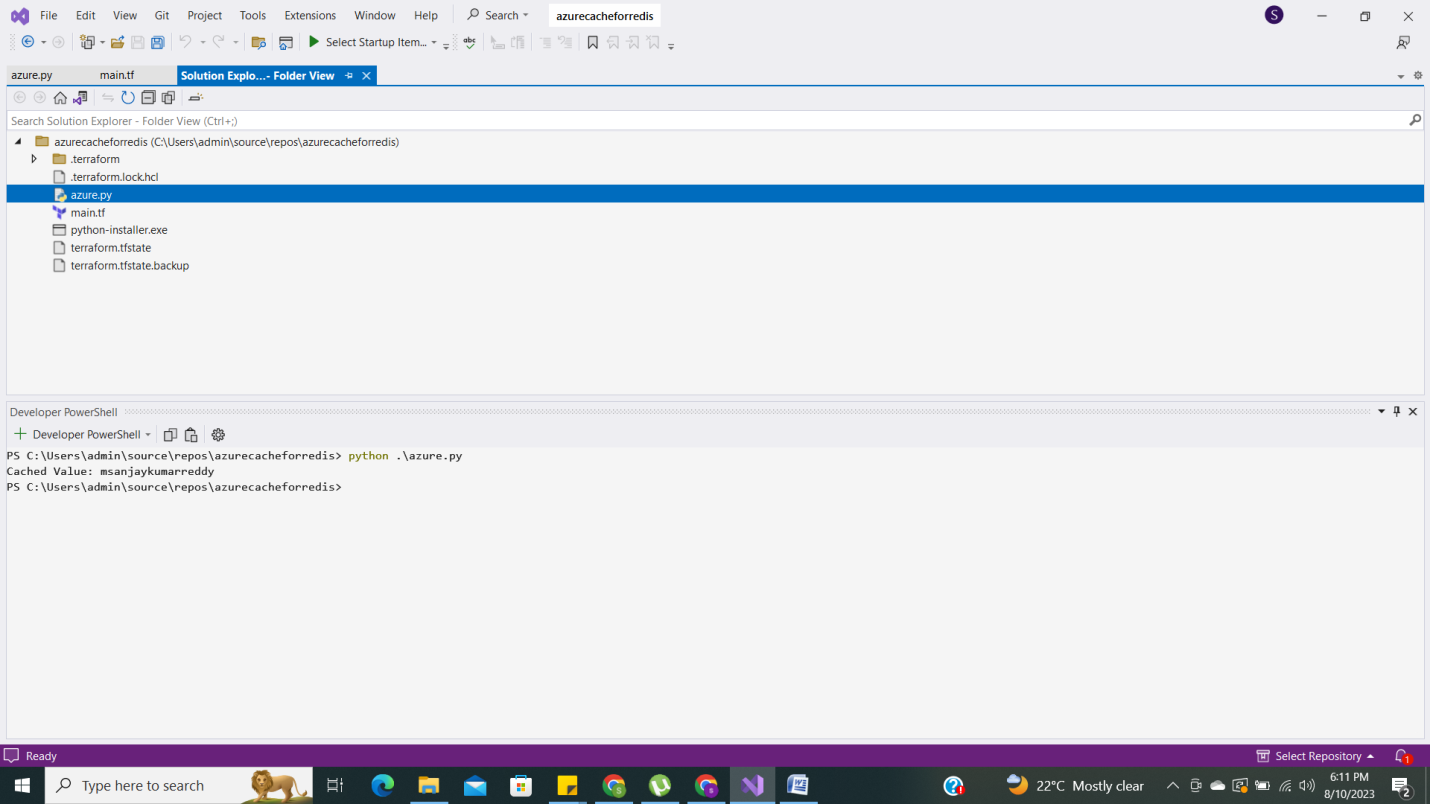
output "container\_registry\_name" {

value = azurerm\_container\_registry.example.name

}

**------------------------------------------------------------------------------------------------------------------------------------------**

**Terraform script to created Azure Cache for redis.  
file streacture .**

****

**Code :  
main.tf**

provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "example" {

name = "redis-example-rg"

location = "East US"

}

resource "azurerm\_redis\_cache" "example" {

name = "redis-example-pip"

location = azurerm\_resource\_group.example.location

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

capacity = 1

family = "C"

sku\_name = "Basic"

}

output "redis\_connection\_string" {

value = azurerm\_redis\_cache.example.primary\_connection\_string

sensitive = true

}

**azure.py**

import redis

hostname = "redis-example-pip.redis.cache.windows.net"

port = 6380

access\_key = "IWbSlbNMzvBeigHxBMtjHmczBTYou5jnZAzCaGmt2Jg="

# Create a Redis client

redis\_client = redis.StrictRedis(

host=hostname,

port=port,

password=access\_key,

ssl=True,

decode\_responses=True

)

# Set a key-value pair in the cache

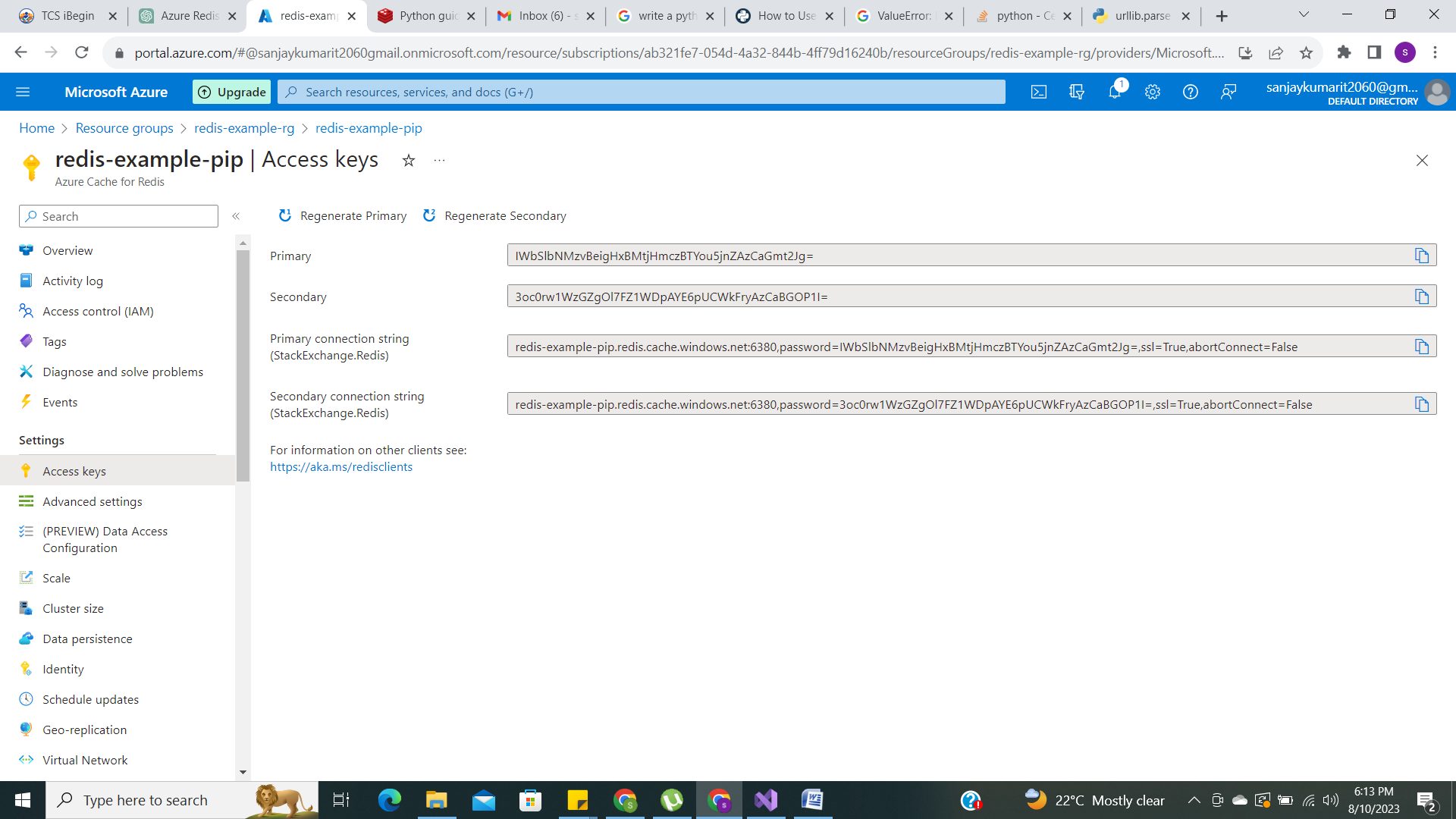
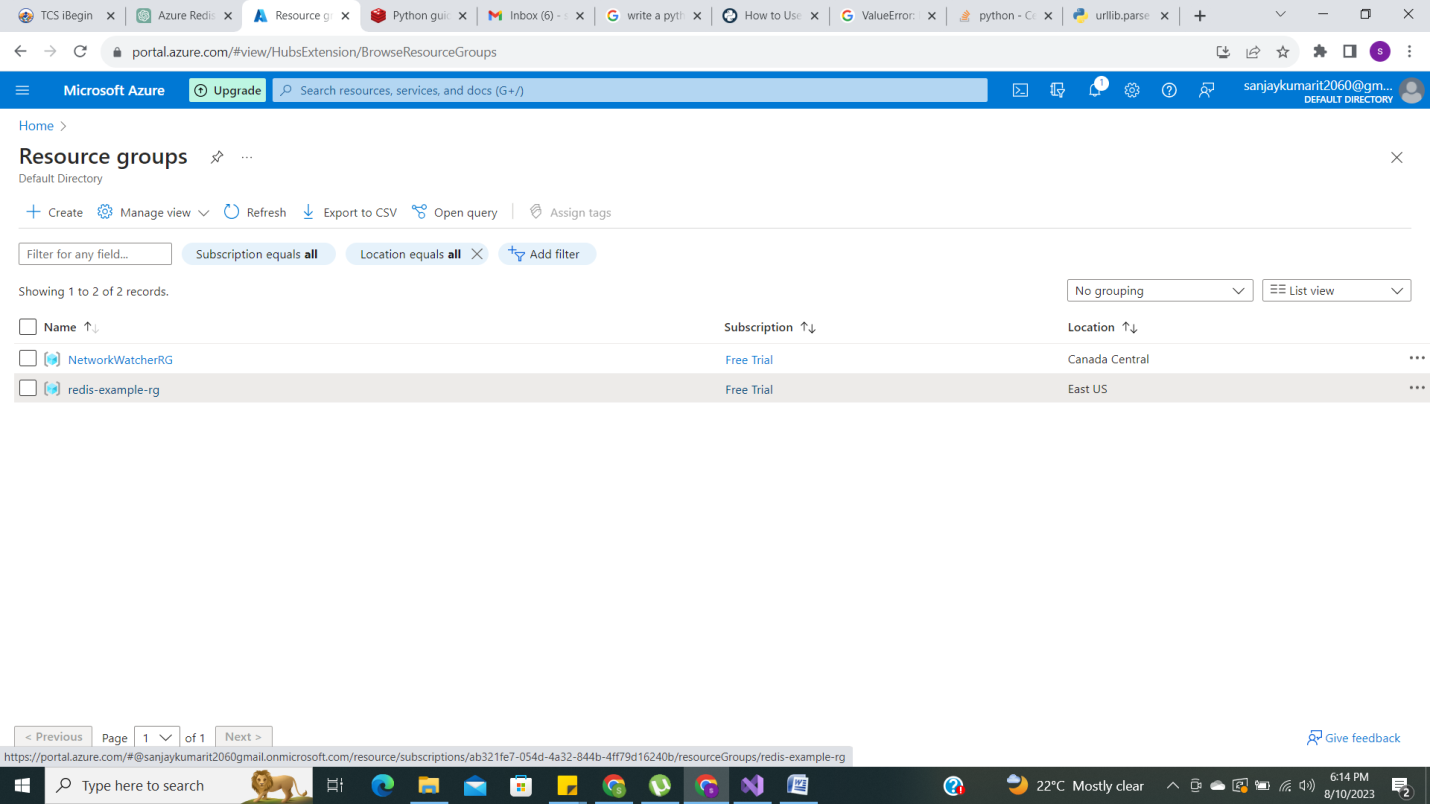
redis\_client.set("my\_name", "msanjaykumarreddy")

# Get the value from the cache

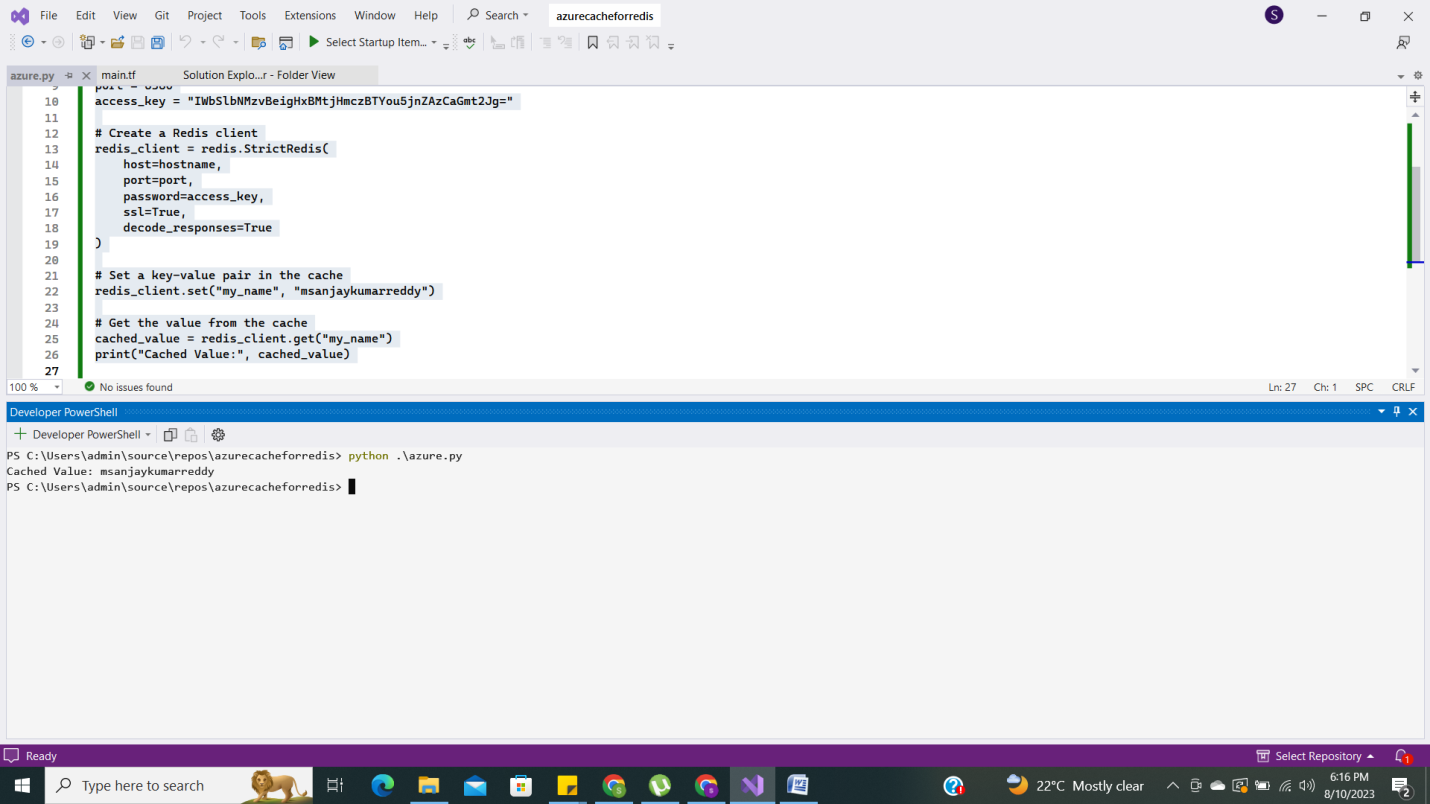
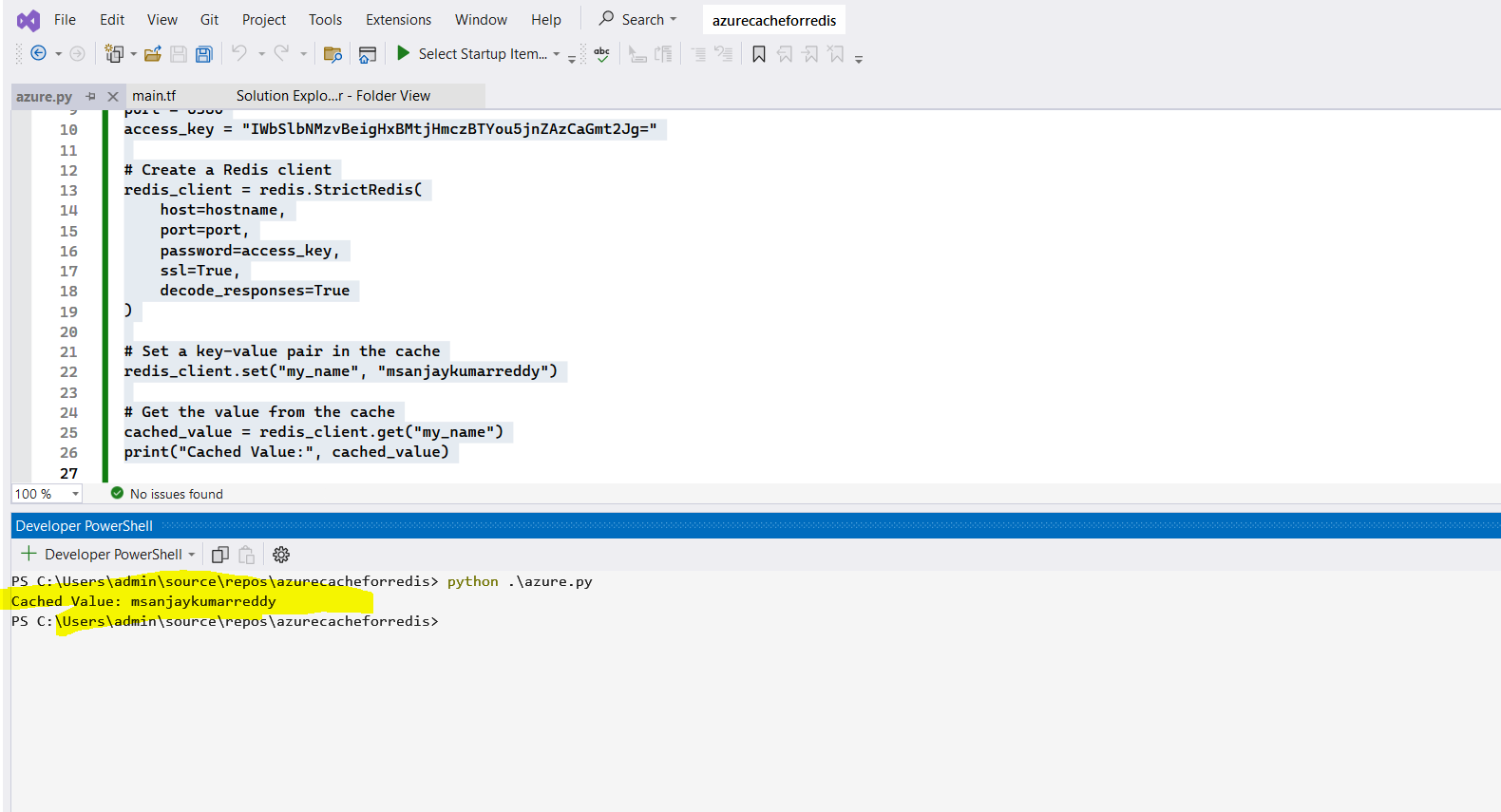
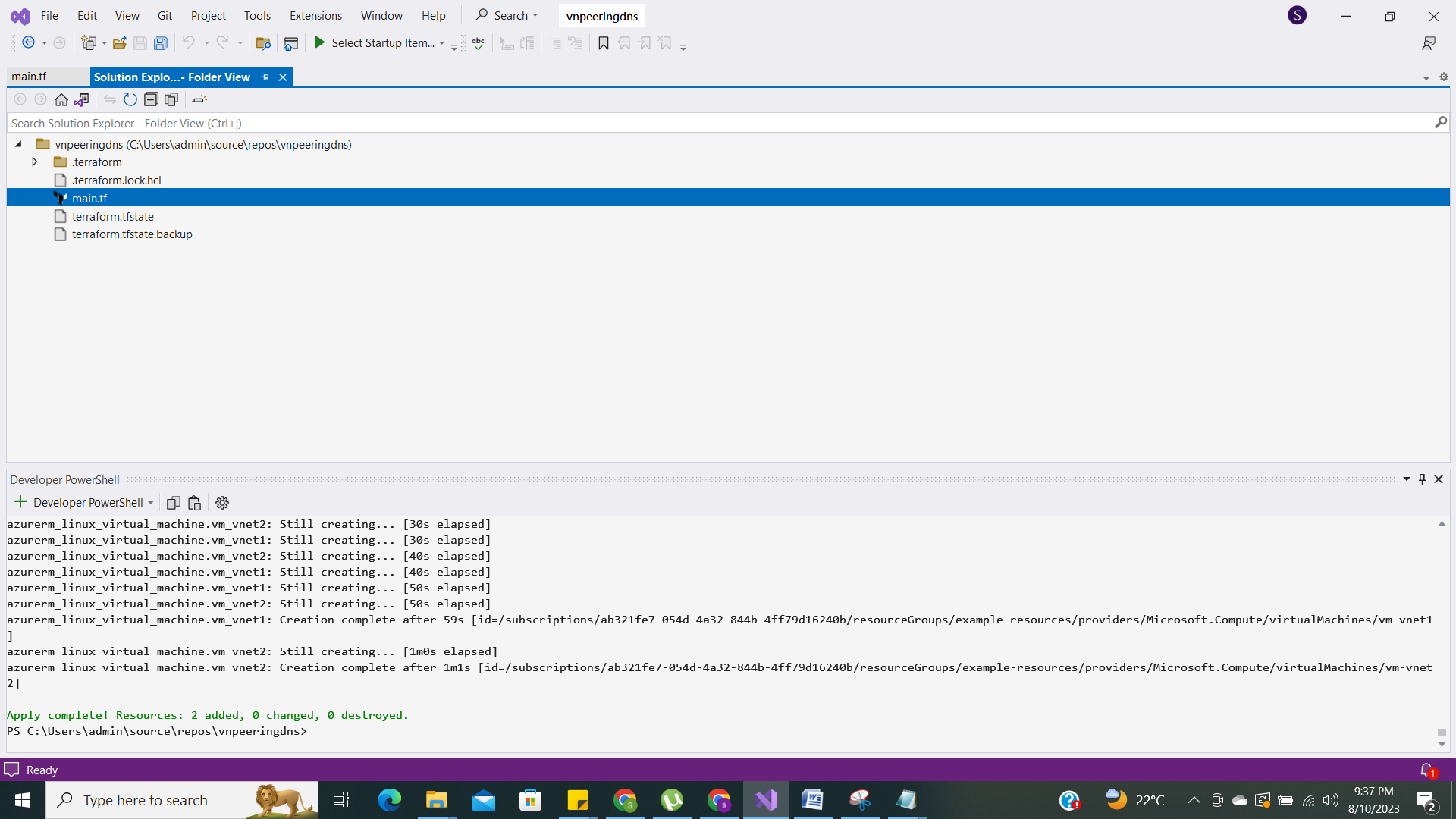
cached\_value = redis\_client.get("my\_name")

print("Cached Value:", cached\_value)

**Resource created in azure portal.**

**  
**

**Connection from python program. Storing temporary value and fetch from cache.**

**  
  
  
Terraform script to create VNET,VNET PEERING,PRIVATE DNS.  
  
File Stricture.  
  
  
terraform code .  
  
main.tf.**provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

resource "azurerm\_resource\_group" "example" {

name = "example-resources"

location = "East US"

}

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

name = "vnet1"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.example.location

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

}

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

name = "vnet2"

address\_space = ["10.1.0.0/16"]

location = azurerm\_resource\_group.example.location

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

}

resource "azurerm\_virtual\_network\_peering" "peering\_vnet1\_to\_vnet2" {

name = "vnet1-to-vnet2"

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

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

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

allow\_virtual\_network\_access = true

allow\_forwarded\_traffic = true

}

resource "azurerm\_virtual\_network\_peering" "peering\_vnet2\_to\_vnet1" {

name = "vnet2-to-vnet1"

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

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

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

allow\_virtual\_network\_access = true

allow\_forwarded\_traffic = true

}

resource "azurerm\_private\_dns\_zone" "private\_dns\_zone\_vnet1" {

name = "privatelink.vnet1.internal"

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

}

resource "azurerm\_private\_dns\_zone" "private\_dns\_zone\_vnet2" {

name = "privatelink.vnet2.internal"

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

}

resource "azurerm\_linux\_virtual\_machine" "vm\_vnet1" {

name = "vm-vnet1"

location = azurerm\_resource\_group.example.location

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

size = "Standard\_D2\_v4"

admin\_username = "adminuser"

admin\_password = "Admin@123456"

admin\_ssh\_key {

username = "adminuser"

public\_key = file("~/.ssh/id\_rsa.pub") # Replace with the path to your public SSH key

}

network\_interface\_ids = [azurerm\_network\_interface.nic\_vnet1.id]

os\_disk {

name = "vmvnet1OSDisk"

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

}

resource "azurerm\_linux\_virtual\_machine" "vm\_vnet2" {

name = "vm-vnet2"

location = azurerm\_resource\_group.example.location

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

size = "Standard\_D2\_v4"

admin\_username = "adminuser"

admin\_password = "Admin@123456"

admin\_ssh\_key {

username = "adminuser"

public\_key = file("~/.ssh/id\_rsa.pub") # Replace with the path to your public SSH key

}

network\_interface\_ids = [azurerm\_network\_interface.nic\_vnet2.id]

os\_disk {

name = "vmvnet2OSDisk"

caching = "ReadWrite"

storage\_account\_type = "Standard\_LRS"

}

source\_image\_reference {

publisher = "Canonical"

offer = "UbuntuServer"

sku = "18.04-LTS"

version = "latest"

}

}

resource "azurerm\_network\_interface" "nic\_vnet1" {

name = "nic-vnet1"

location = azurerm\_resource\_group.example.location

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

enable\_accelerated\_networking = true

ip\_configuration {

name = "ipconfig1"

subnet\_id = azurerm\_subnet.subnet\_vnet1.id

private\_ip\_address\_allocation = "Dynamic"

}

}

resource "azurerm\_network\_interface" "nic\_vnet2" {

name = "nic-vnet2"

location = azurerm\_resource\_group.example.location

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

enable\_accelerated\_networking = true

ip\_configuration {

name = "ipconfig1"

subnet\_id = azurerm\_subnet.subnet\_vnet2.id

private\_ip\_address\_allocation = "Dynamic"

}

}

resource "azurerm\_subnet" "subnet\_vnet1" {

name = "subnet1"

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

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

address\_prefixes = ["10.0.1.0/24"]

}

resource "azurerm\_subnet" "subnet\_vnet2" {

name = "subnet2"

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

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

address\_prefixes = ["10.1.1.0/24"]

}

resource "azurerm\_private\_dns\_zone\_virtual\_network\_link" "vnet1\_link" {

name = "vnet1-link"

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

private\_dns\_zone\_name = azurerm\_private\_dns\_zone.private\_dns\_zone\_vnet1.name

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

}

resource "azurerm\_private\_dns\_zone\_virtual\_network\_link" "vnet2\_link" {

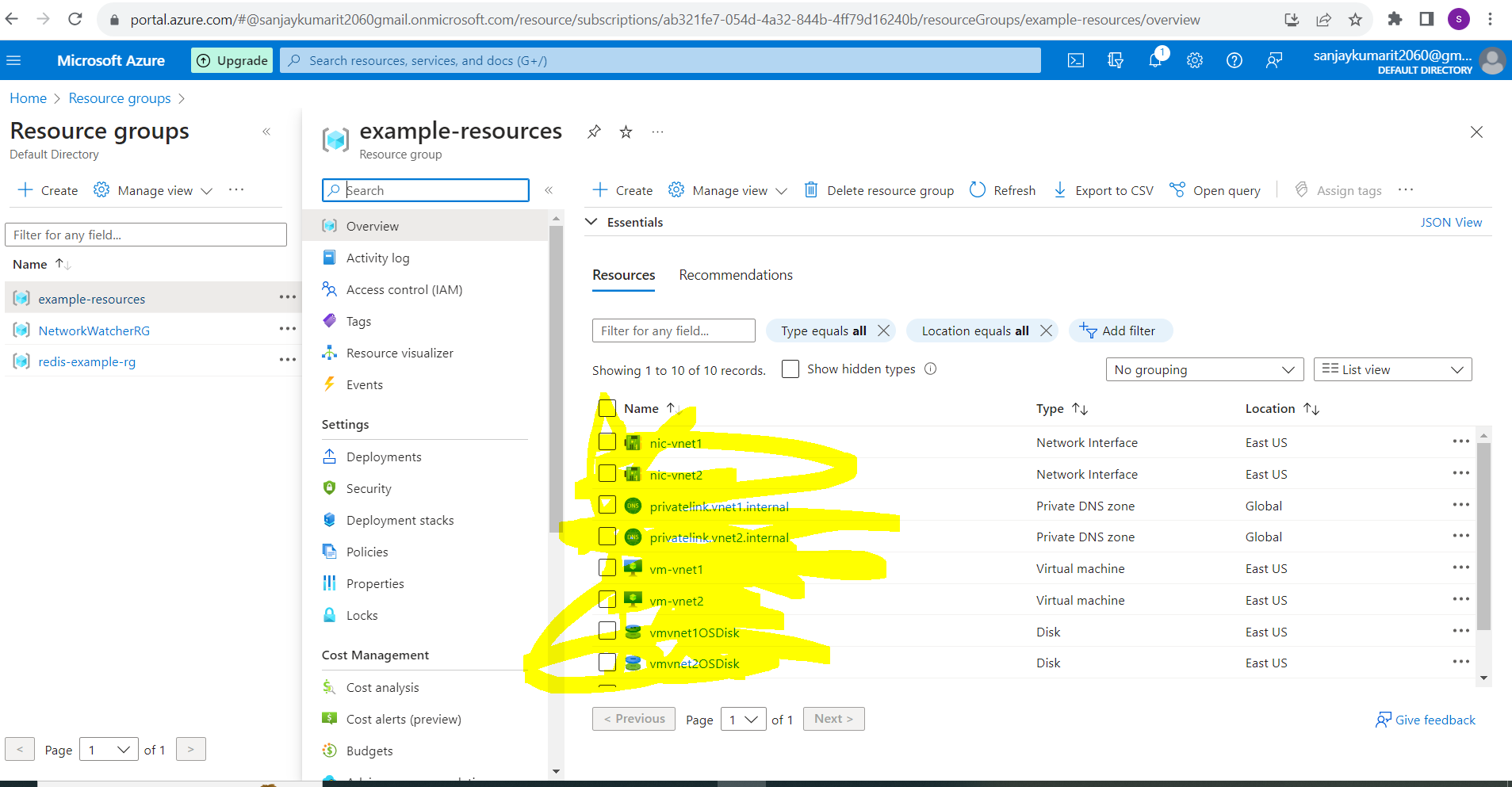
name = "vnet2-link"

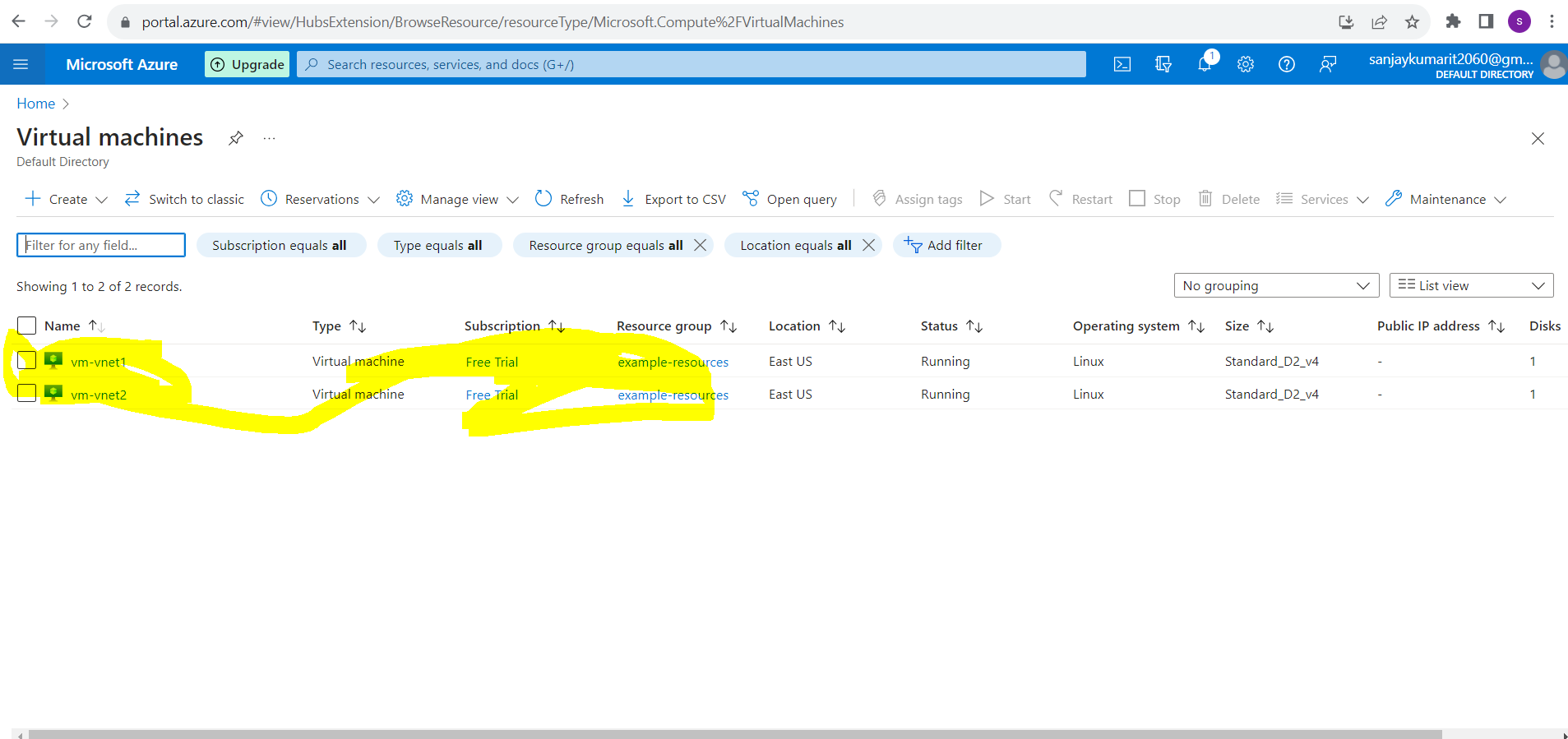
resource\_group\_name = azurerm\_resource\_group.example.name

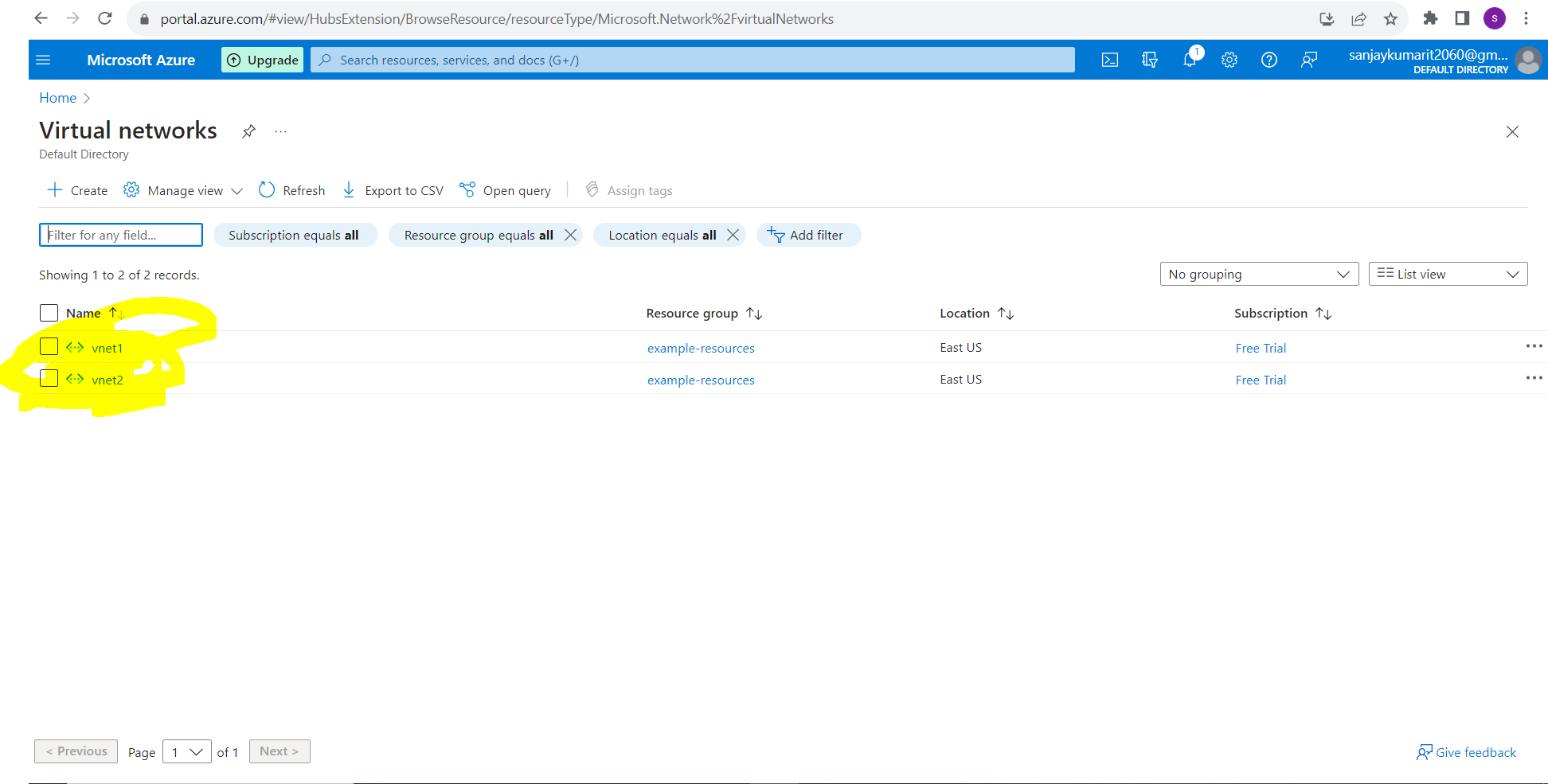
private\_dns\_zone\_name = azurerm\_private\_dns\_zone.private\_dns\_zone\_vnet2.name

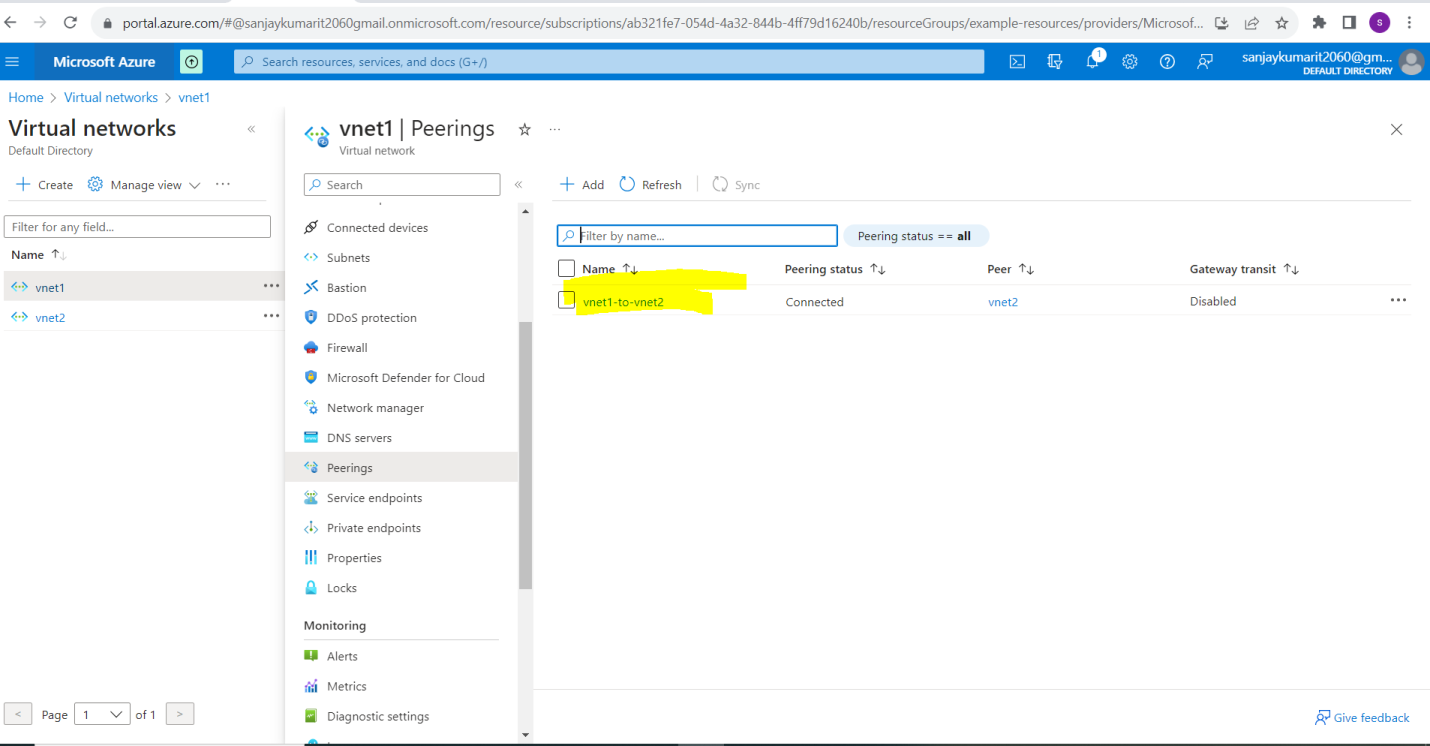
virtual\_network\_id = azurerm\_virtual\_network.vnet2.id

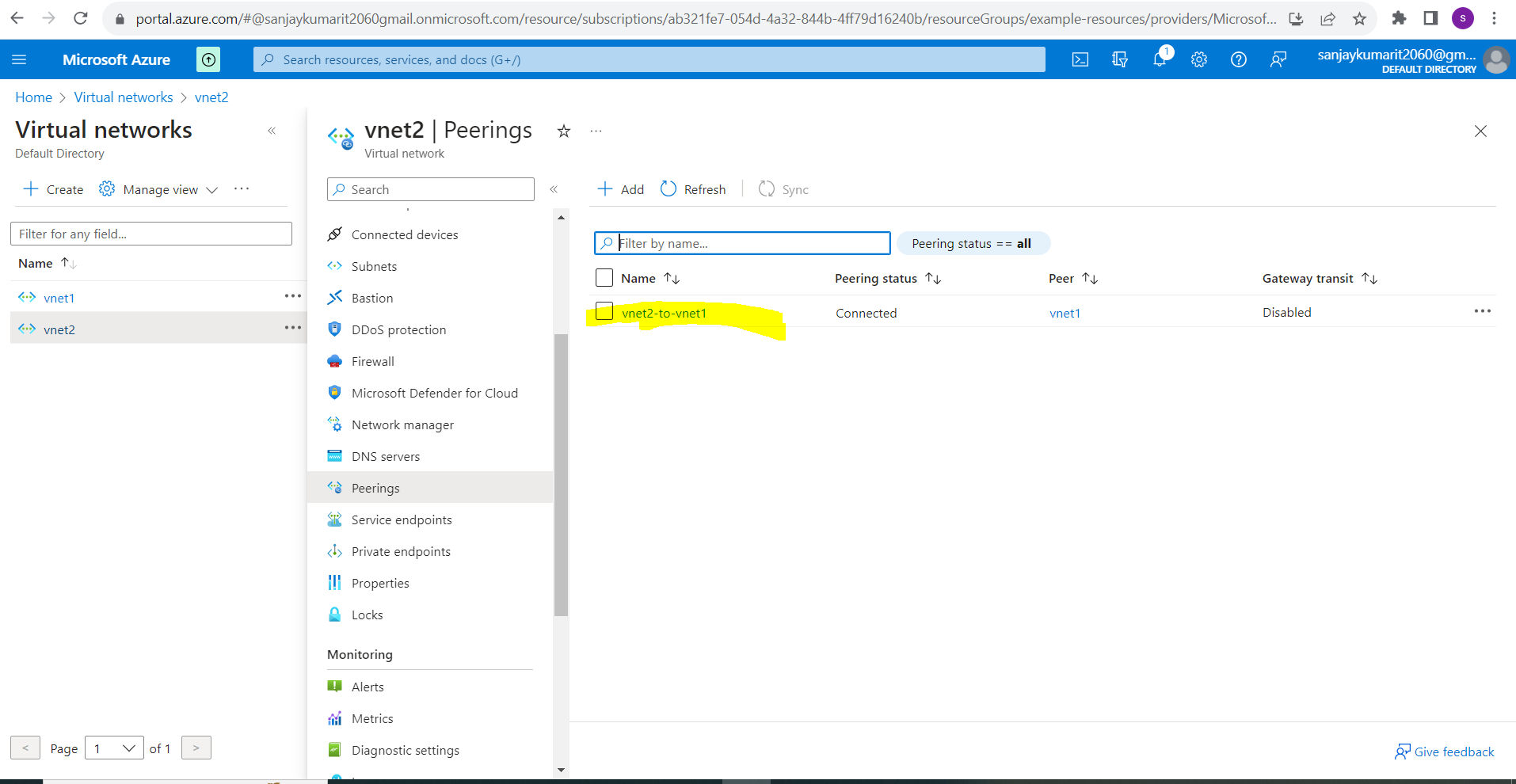
}

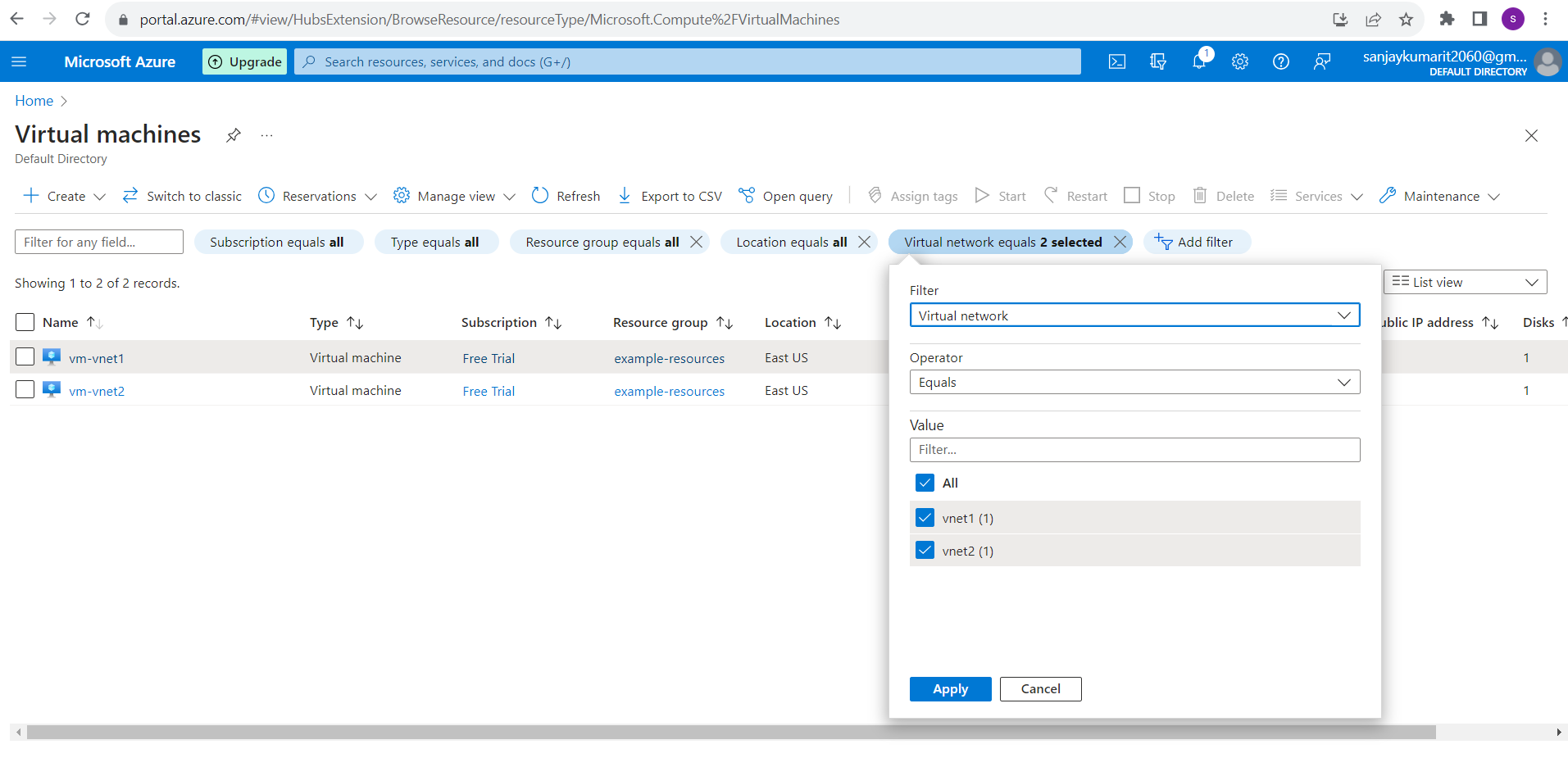
****

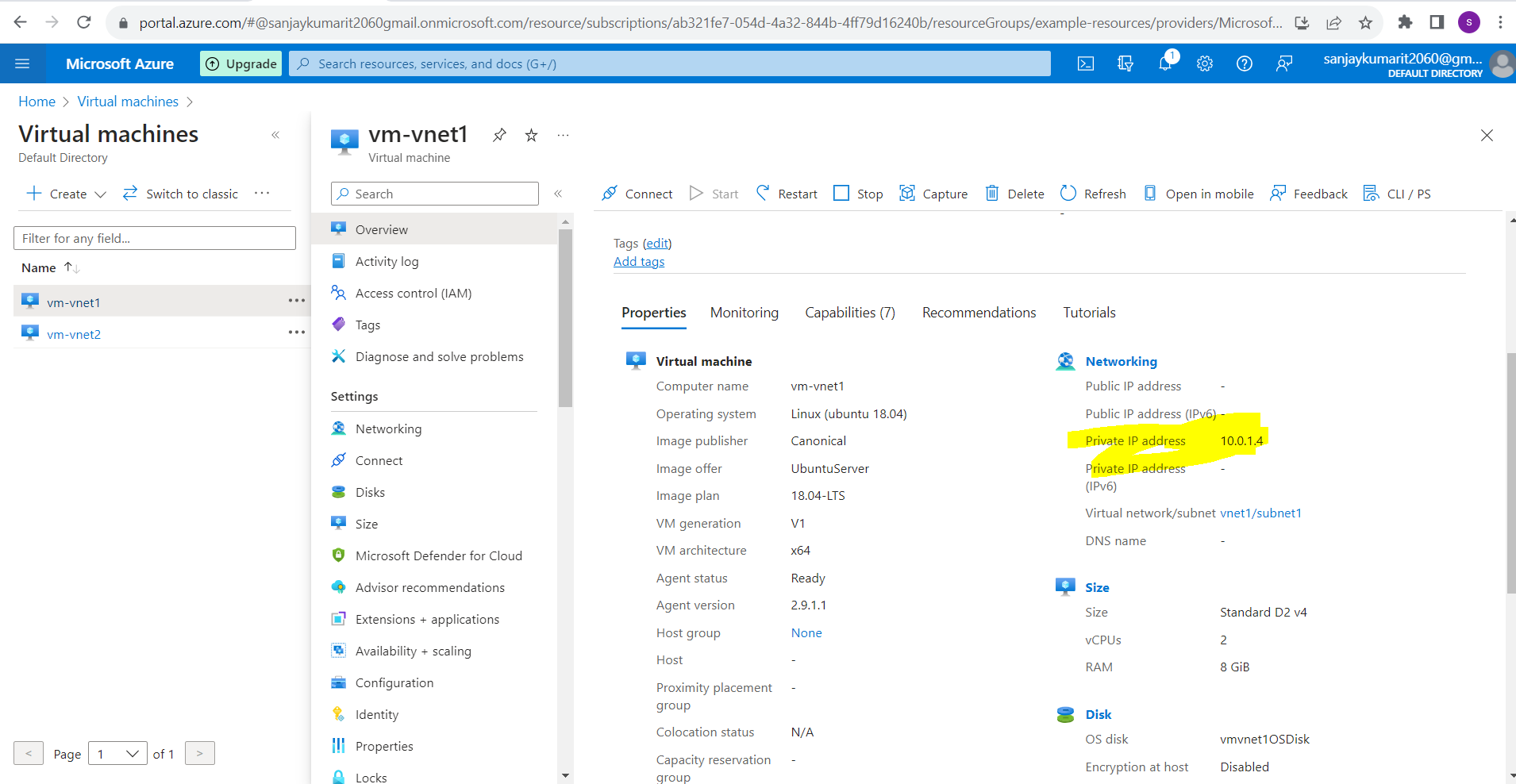
**Vnets :  
  
  
**

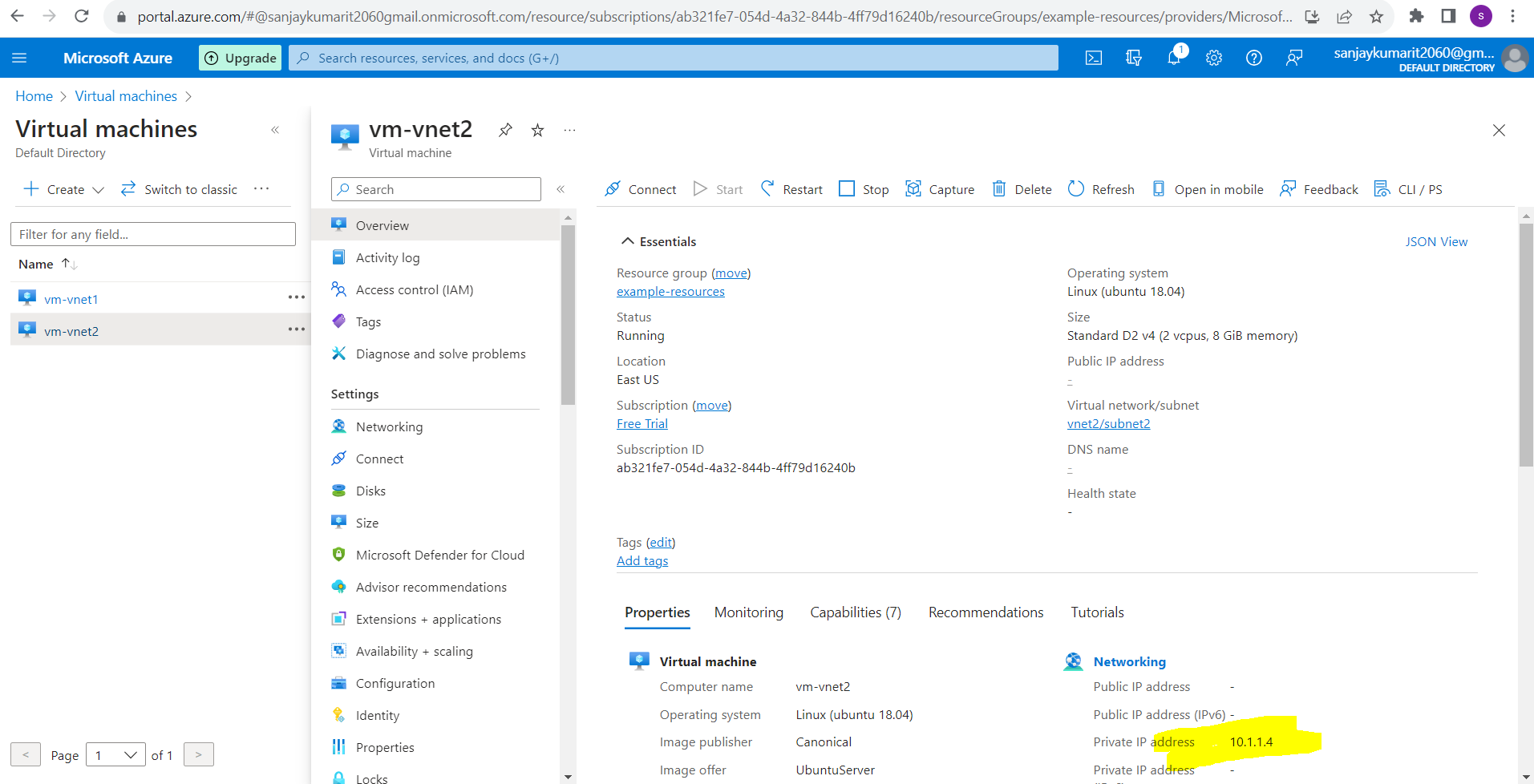
****

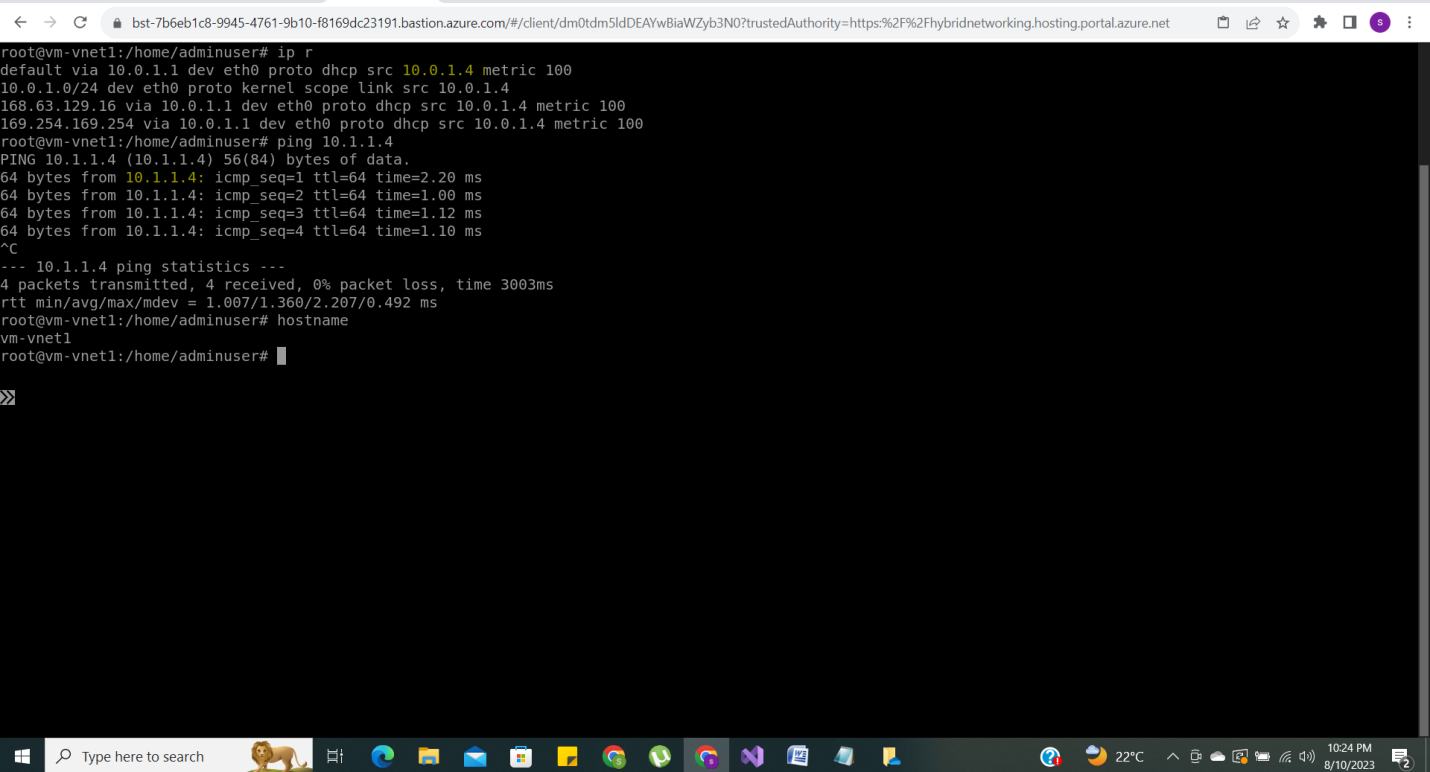
****

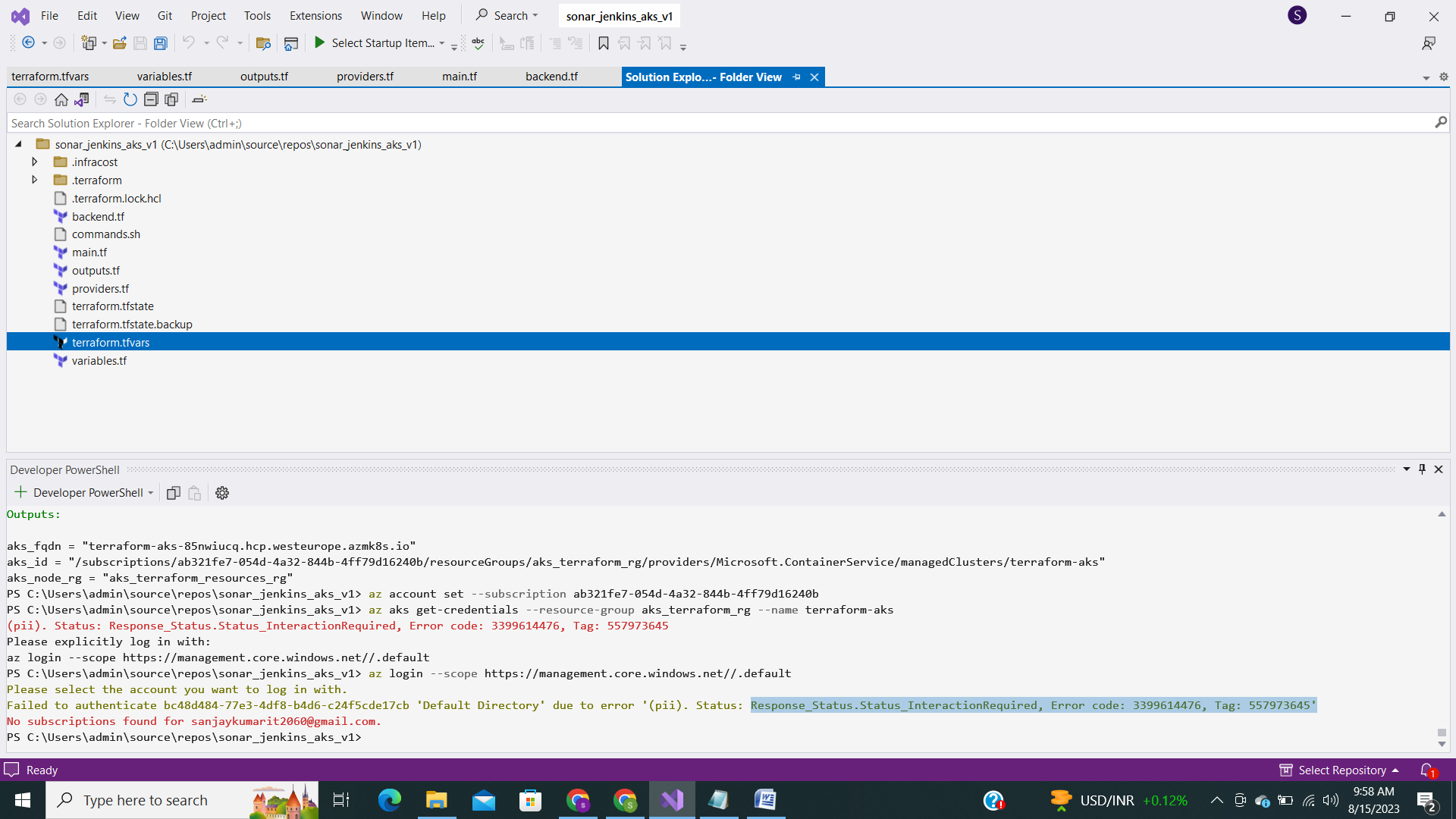
****

****

****

****

****

**Terraform with AKS cluster deploying Jenkins and sonarqube .  
  
Code stricture:  
  
**

**CODE :  
  
main.tf**

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

name = var.resource\_group\_name

location = var.location

}

resource "azurerm\_kubernetes\_cluster" "aks" {

name = var.cluster\_name

kubernetes\_version = var.kubernetes\_version

location = var.location

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

dns\_prefix = var.cluster\_name

node\_resource\_group = var.node\_resource\_group

default\_node\_pool {

name = "system"

node\_count = var.system\_node\_count

vm\_size = "Standard\_DS2\_v2"

type = "VirtualMachineScaleSets"

# availability\_zones = [1, 2, 3]

enable\_auto\_scaling = false

}

identity {

type = "SystemAssigned"

}

network\_profile {

load\_balancer\_sku = "standard"

network\_plugin = "kubenet" # azure (CNI)

}

}

**Output.tf**

output "aks\_id" {

value = azurerm\_kubernetes\_cluster.aks.id

}

output "aks\_fqdn" {

value = azurerm\_kubernetes\_cluster.aks.fqdn

}

output "aks\_node\_rg" {

value = azurerm\_kubernetes\_cluster.aks.node\_resource\_group

}  
  
  
provider.tf  
  
provider "azurerm" {

features {}

client\_id = "ff574c2d-8d9d-403c-8b8d-b33c2a68fcda"

client\_secret ="BIf8Q~Y6q0ELbHDL0jTjz8pmJhlKySkMPbvVGdbR"

tenant\_id = "bc48d484-77e3-4df8-b4d6-c24f5cde17cb"

subscription\_id = "ab321fe7-054d-4a32-844b-4ff79d16240b"

}

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "3.54.0"

}

}

}  
  
  
k8 yml files .  
  
**jenkins-service.yml**  
  
apiVersion: v1

kind: Service

metadata:

name: jenkins-service

spec:

selector:

app: jenkins

ports:

- protocol: TCP

port: 80

targetPort: 8080

type: LoadBalancer

**Jenkins.yml**apiVersion: apps/v1

kind: Deployment

metadata:

name: jenkins

spec:

replicas: 1

selector:

matchLabels:

app: jenkins

template:

metadata:

labels:

app: jenkins

spec:

containers:

- name: jenkins

image: jenkins/jenkins:lts

ports:

- containerPort: 8080

volumeMounts:

- name: jenkins-data

mountPath: /var/jenkins\_home

volumes:

- name: jenkins-data

emptyDir: {}

**sonarqube-deployment.yml**apiVersion: apps/v1

kind: Deployment

metadata:

name: sonarqube

spec:

replicas: 1

selector:

matchLabels:

app: sonarqube

template:

metadata:

labels:

app: sonarqube

spec:

containers:

- name: sonarqube

image: sonarqube:latest

ports:

- containerPort: 9000

**sonarqube-service.yml**

apiVersion: v1

kind: Service

metadata:

name: sonarqube-service

spec:

selector:

app: sonarqube

ports:

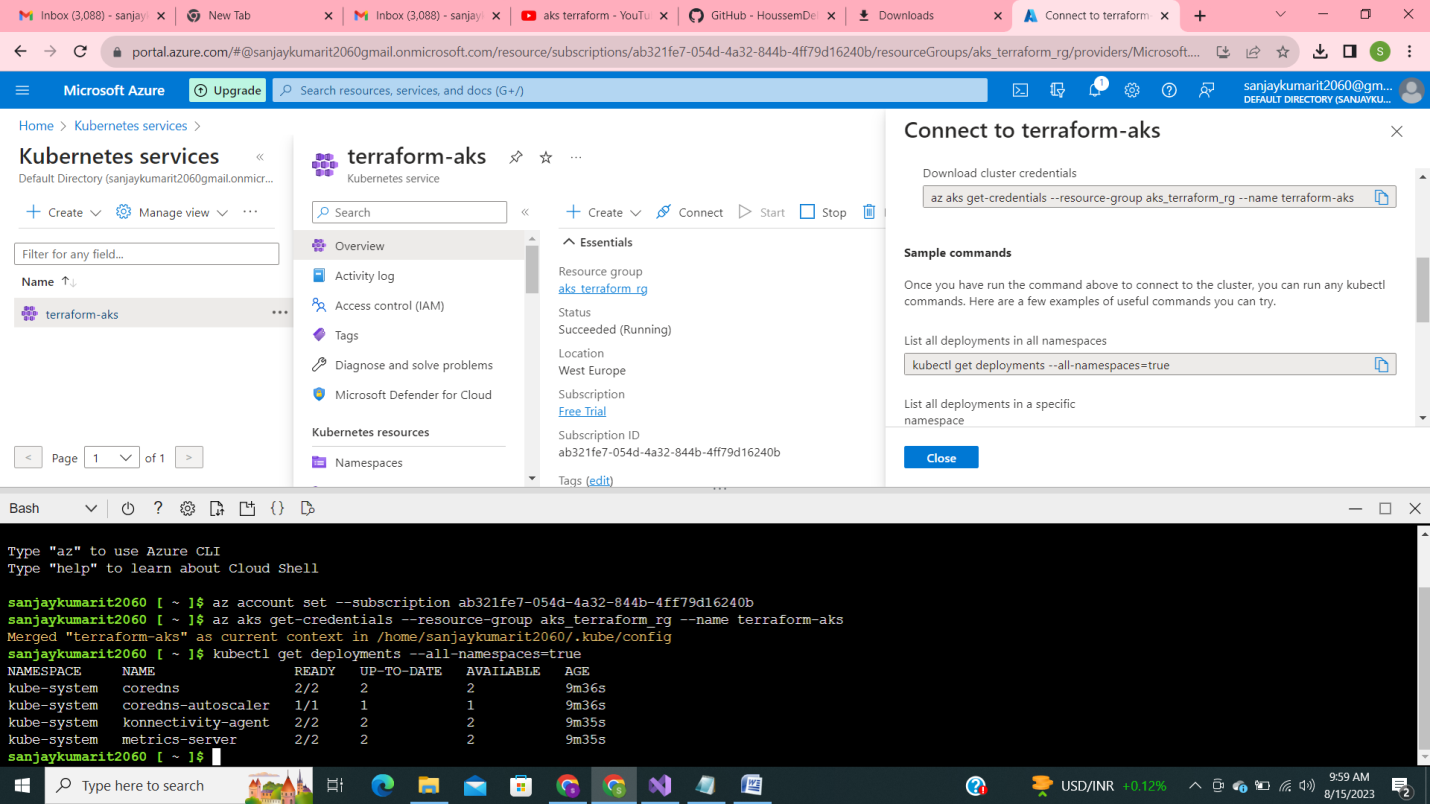
- protocol: TCP

port: 80

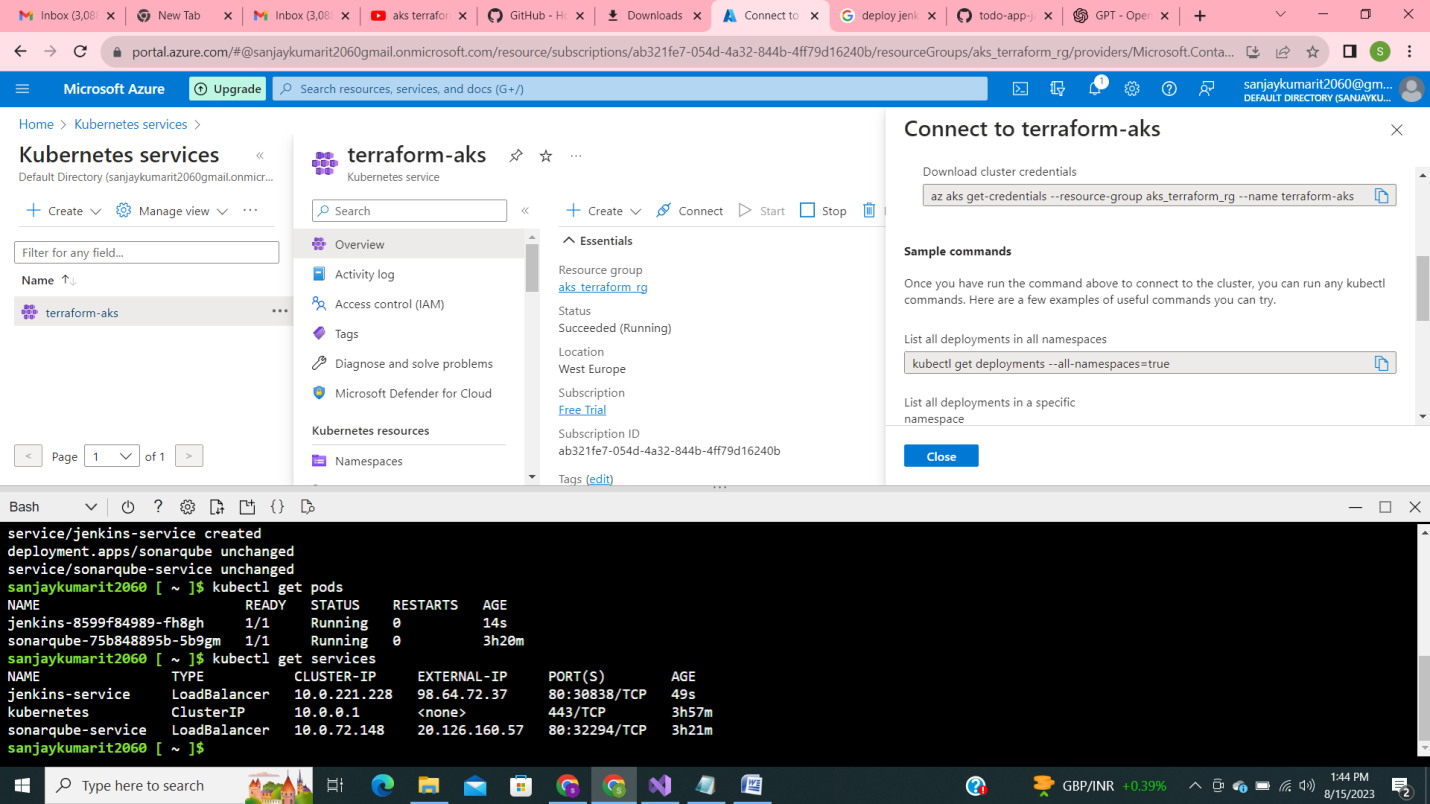
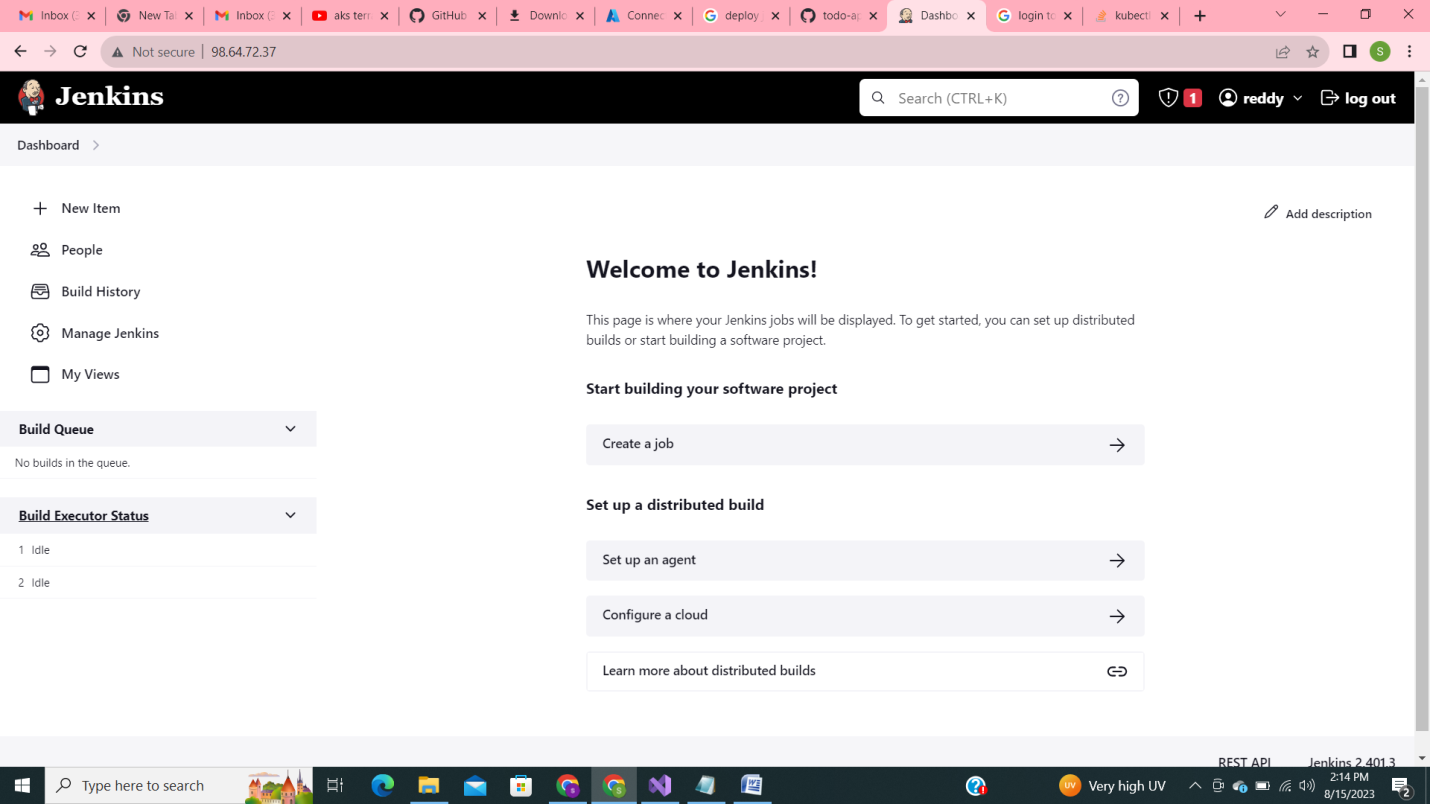
targetPort: 9000

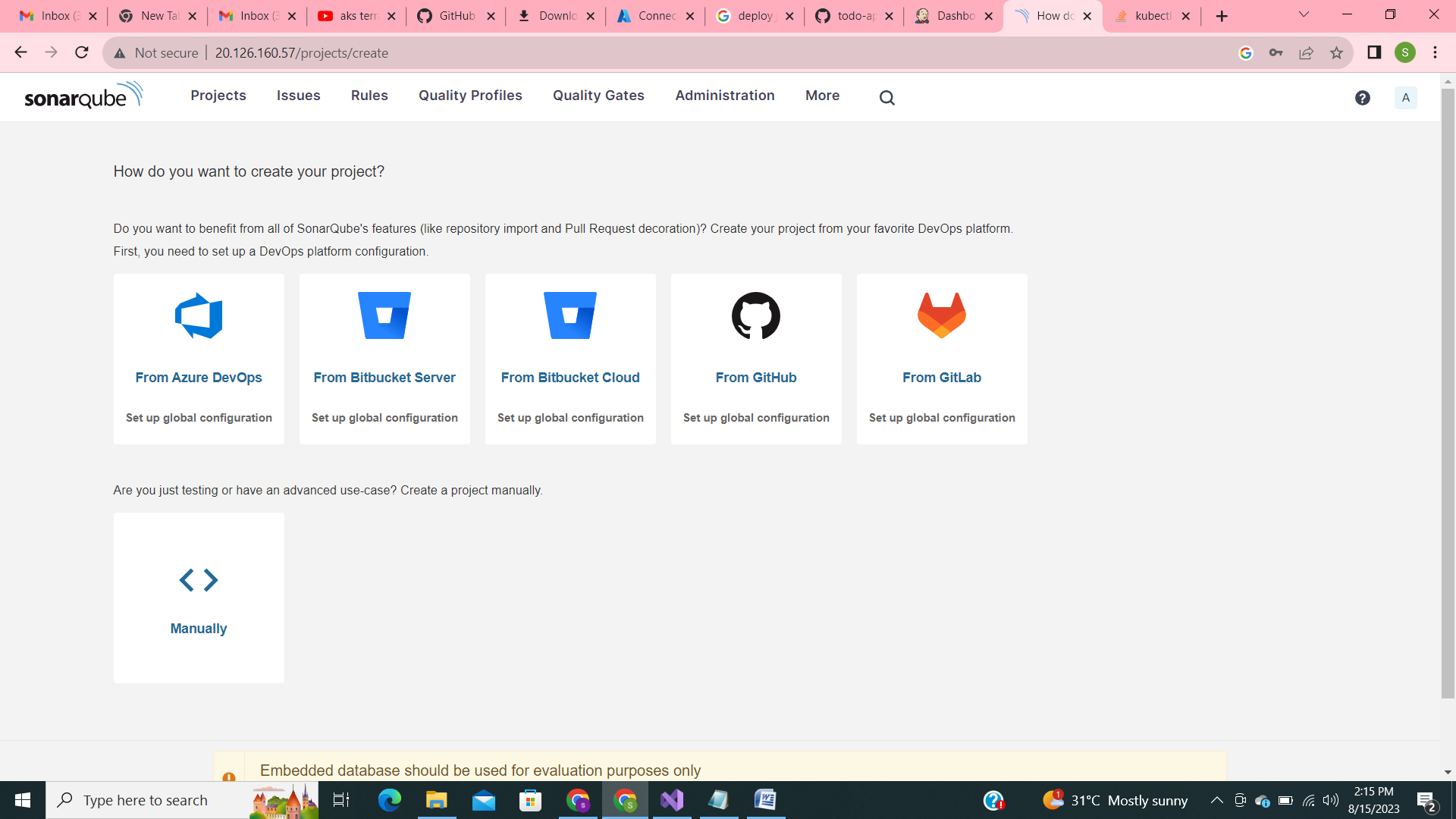
type: LoadBalancer

**------------------------------------------------------------------------------------------------------------------------------------------**

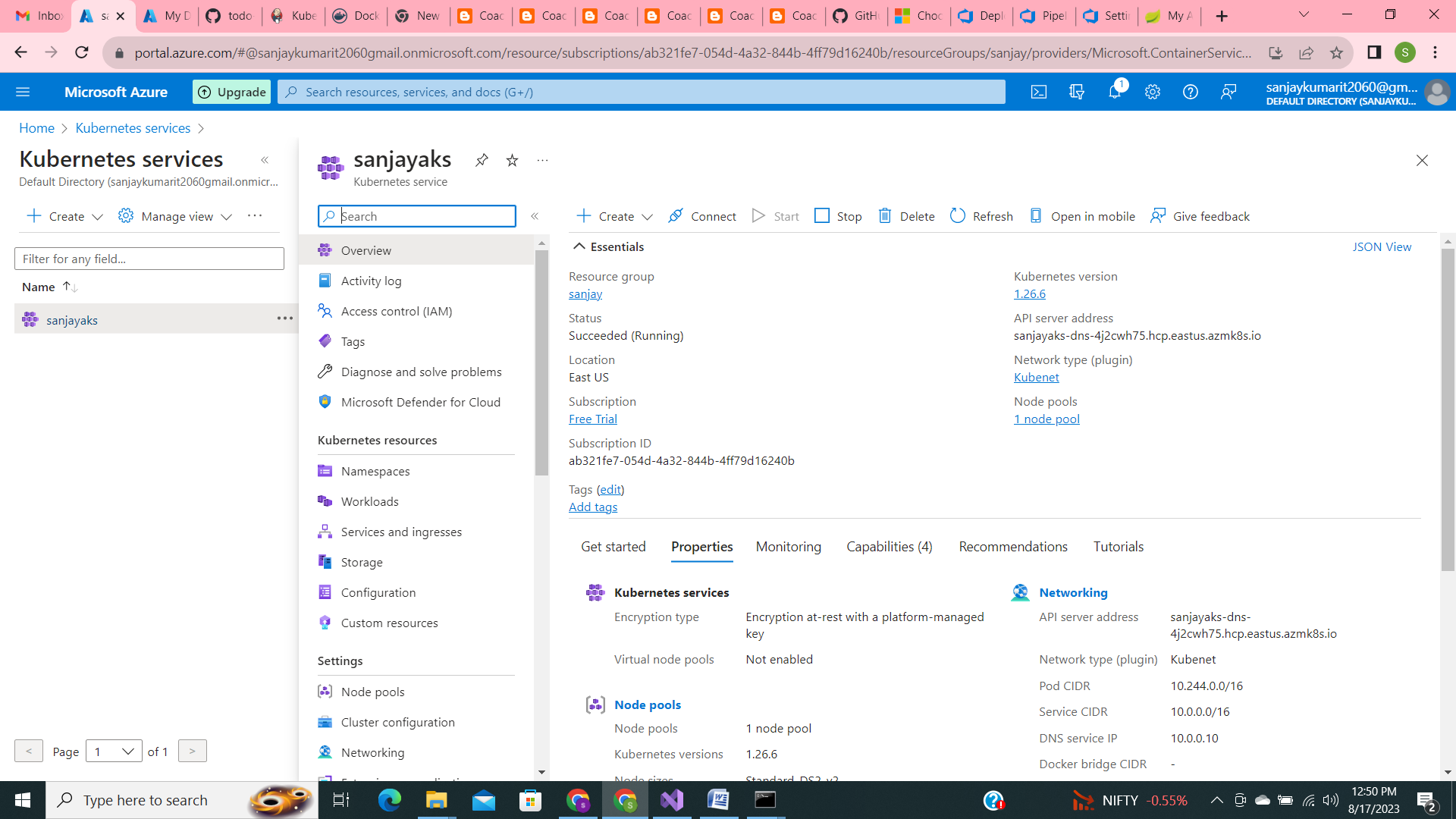
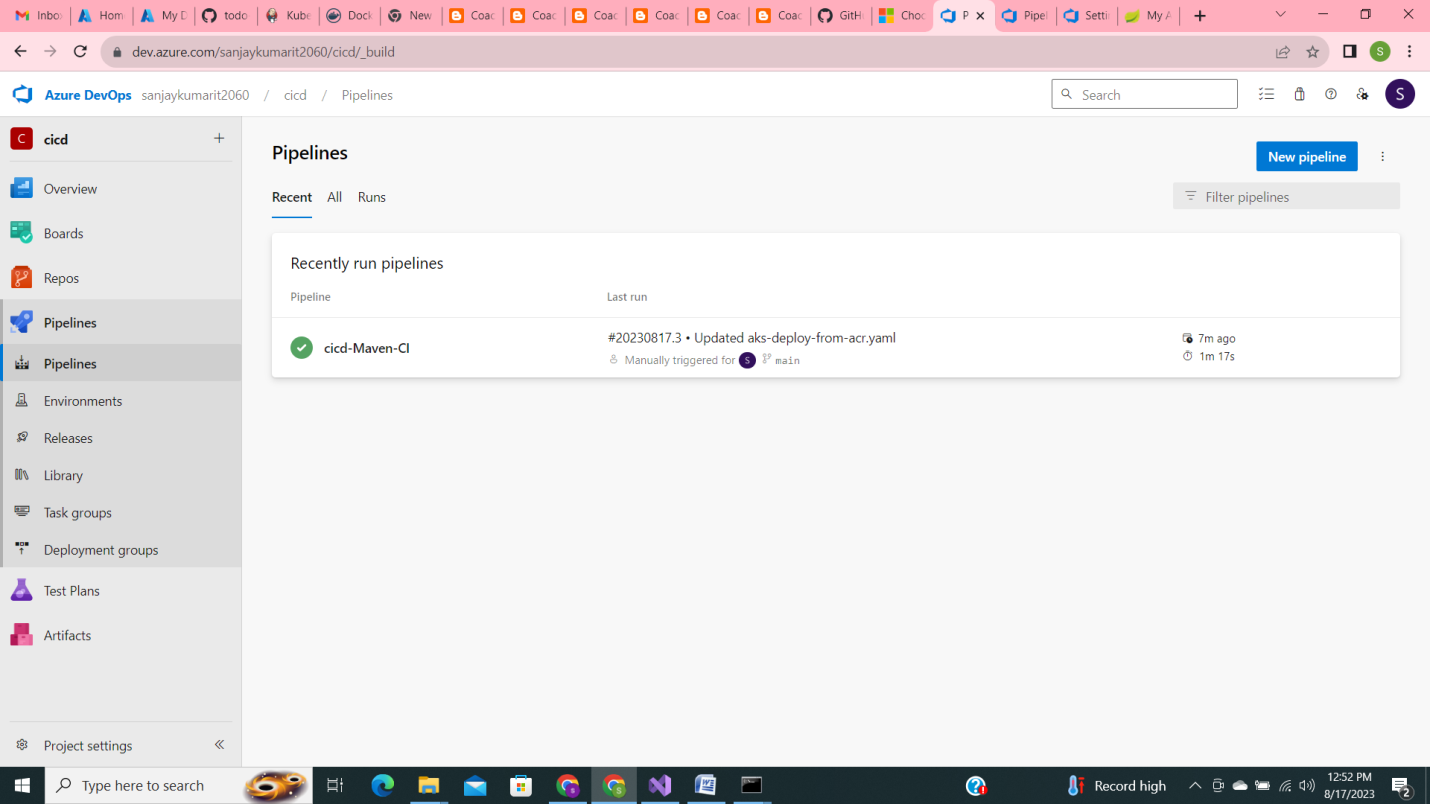
**k8 cluster login   
   
  
pods Jenkins,sonarqube**

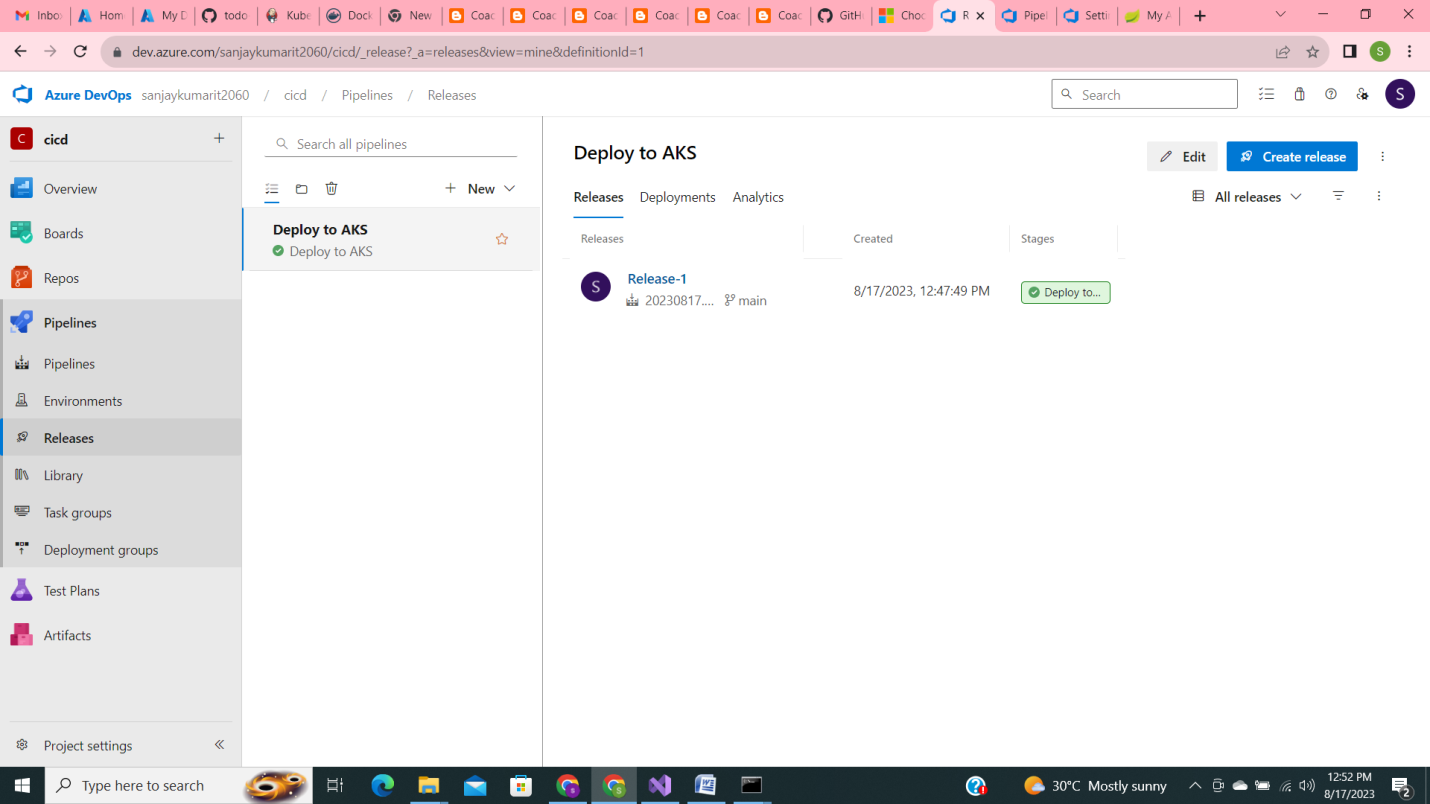
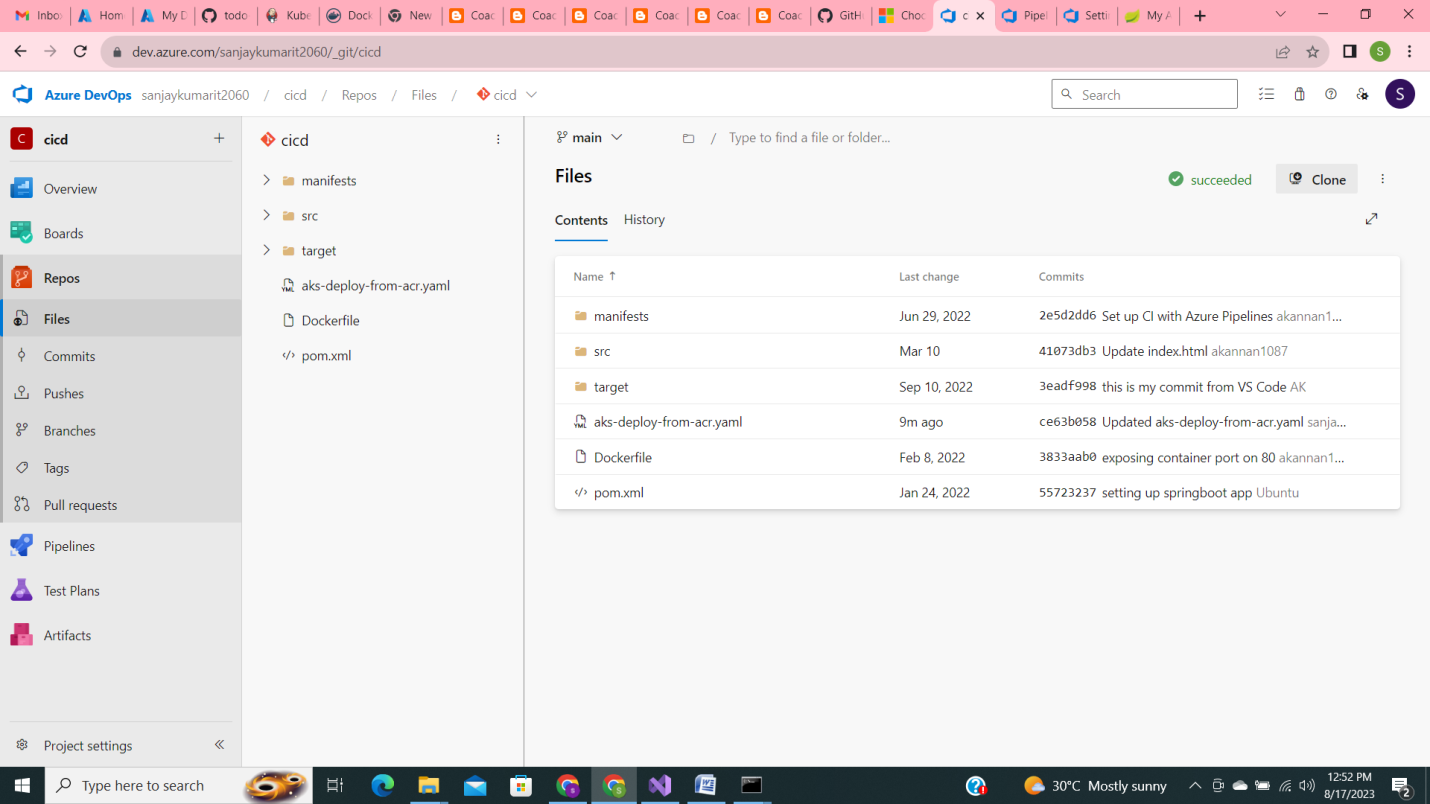
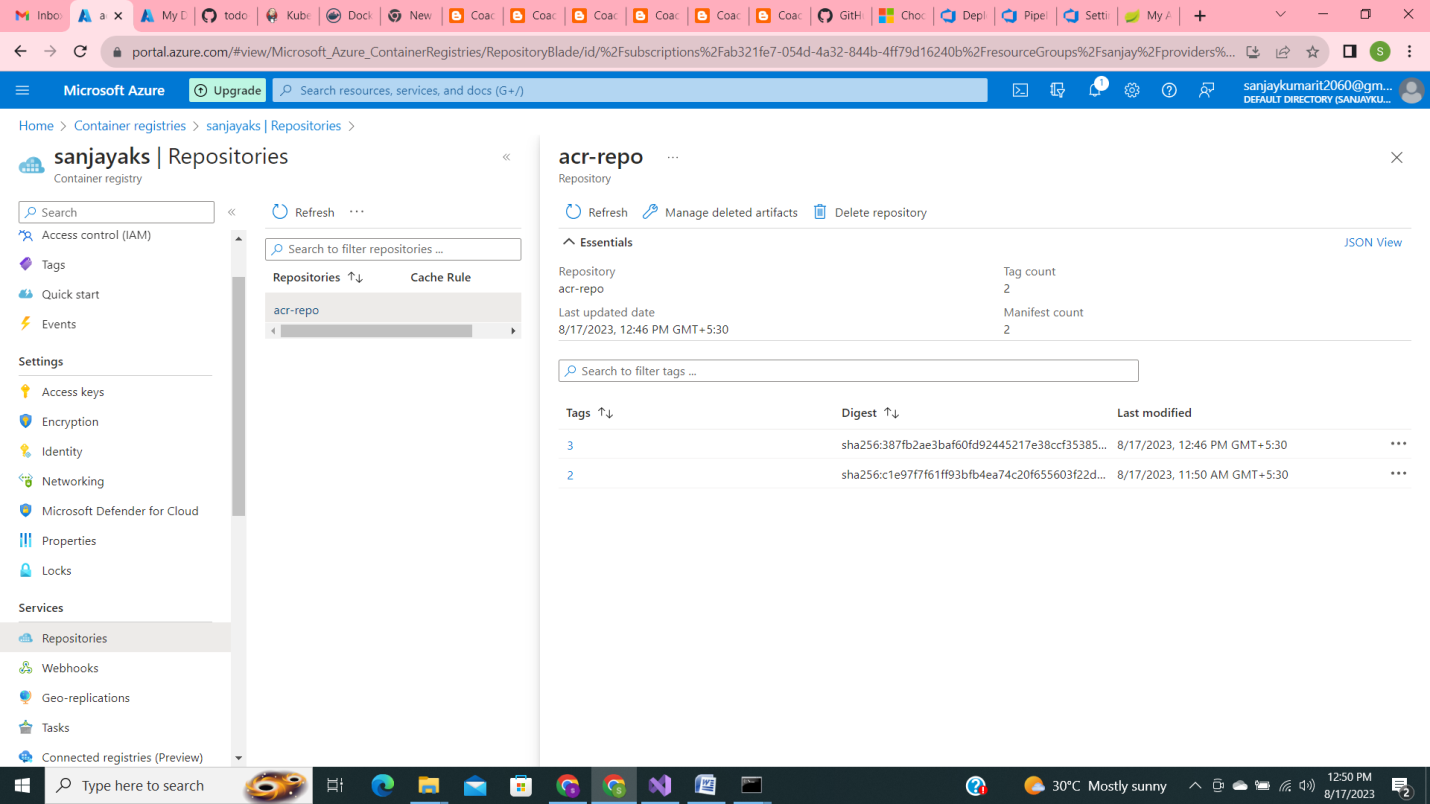
**Jenkins services ,sonarqube services**

****kubectl exec -n <name space here> <pod-name> -it -- /bin/sh  
  
**kubectl exec -n default jenkins-8599f84989-fh8gh -it -- /bin/sh  
  
**

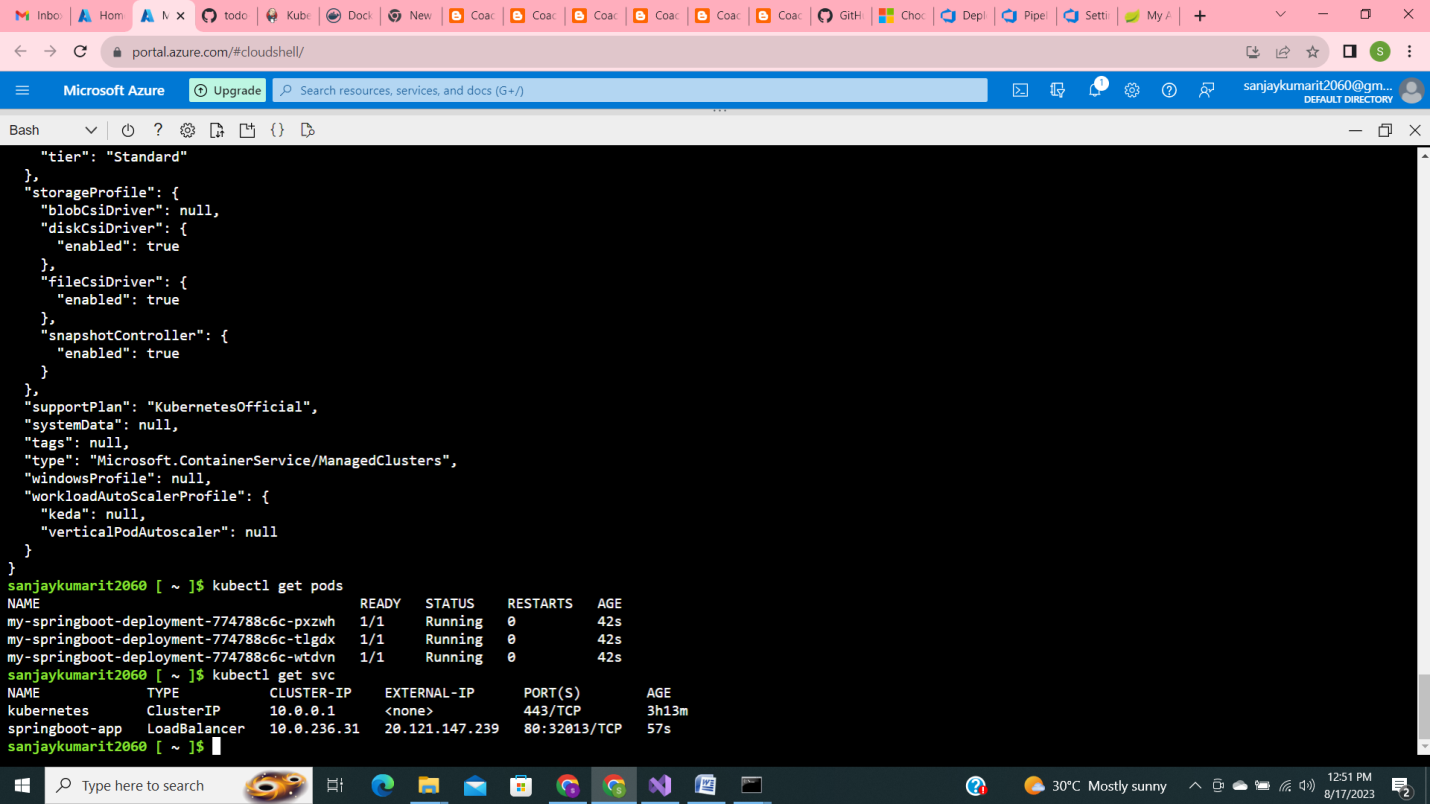
**SONAR QUBE :  
  
  
Write a terraform code for creating CICD pipeline and deploy micro service in AKS cluster .  
  
 az aks update -n sanjayaks -g sanjay --attach-acr sanjayaks**

**---------------Complete cicd pipeline with micro services deployments ----------------------**

**  
  
**

**  
  
  
**

**pods**

****

**application   
  
**