# Assignment 1 Create a new directory and write a terraform code which

# would create Ec2 instance and use Existing Security Group and Key.

#

provider "aws" {

region = "ap-south-1"

}

resource "aws\_instance" "web1" {

ami = "ami-013e83f579886baeb"

instance\_type = "t2.micro"

key\_name = "web1keypair"

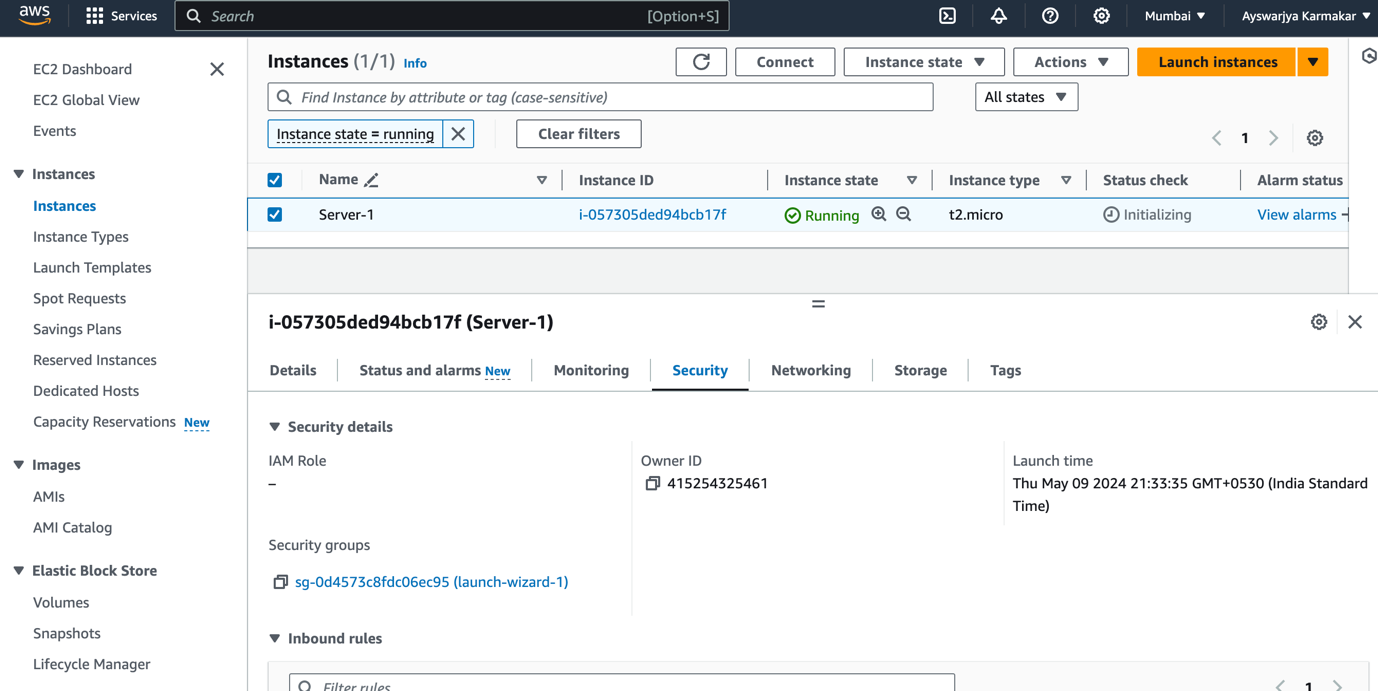
vpc\_security\_group\_ids = ["sg-0d4573c8fdc06ec95"]

tags = {

Name = "Server-1"

}

}



# Assignment 2 Create a new directory and write a terraform code

# which would create Security Group and new Key and create Ec2

# instance and use newly created Security Group and Key.The security

# group should allow incoming traffic on port 22 (SSH) and port 80 (HTTP).

provider "aws" {

region = "ap-south-1"

}

resource "aws\_instance" "web2" {

ami = "ami-013e83f579886baeb"

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.key\_pair.key\_name

vpc\_security\_group\_ids = [aws\_security\_group.server2\_sg.id]

tags = {

Name = "Server-2"

}

}

#Create security group

resource "aws\_security\_group" "server2\_sg" {

name = "server2\_sg"

description = "Allow inbound ports 22, 80"

vpc\_id = "vpc-0019a55ca82b2fc1d"

#Allow incoming TCP requests on port 22 from any IP

ingress {

description = "Allow SSH Traffic"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

#Allow incoming TCP requests on port 80 from any IP

ingress {

description = "Allow HTTP Traffic"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

}

# Generates a secure private key and encodes it as PEM

resource "tls\_private\_key" "key\_pair" {

algorithm = "RSA"

rsa\_bits = 4096

}

# Create the Key Pair

resource "aws\_key\_pair" "key\_pair" {

key\_name = "ec2-key-pair"

public\_key = tls\_private\_key.key\_pair.public\_key\_openssh

}

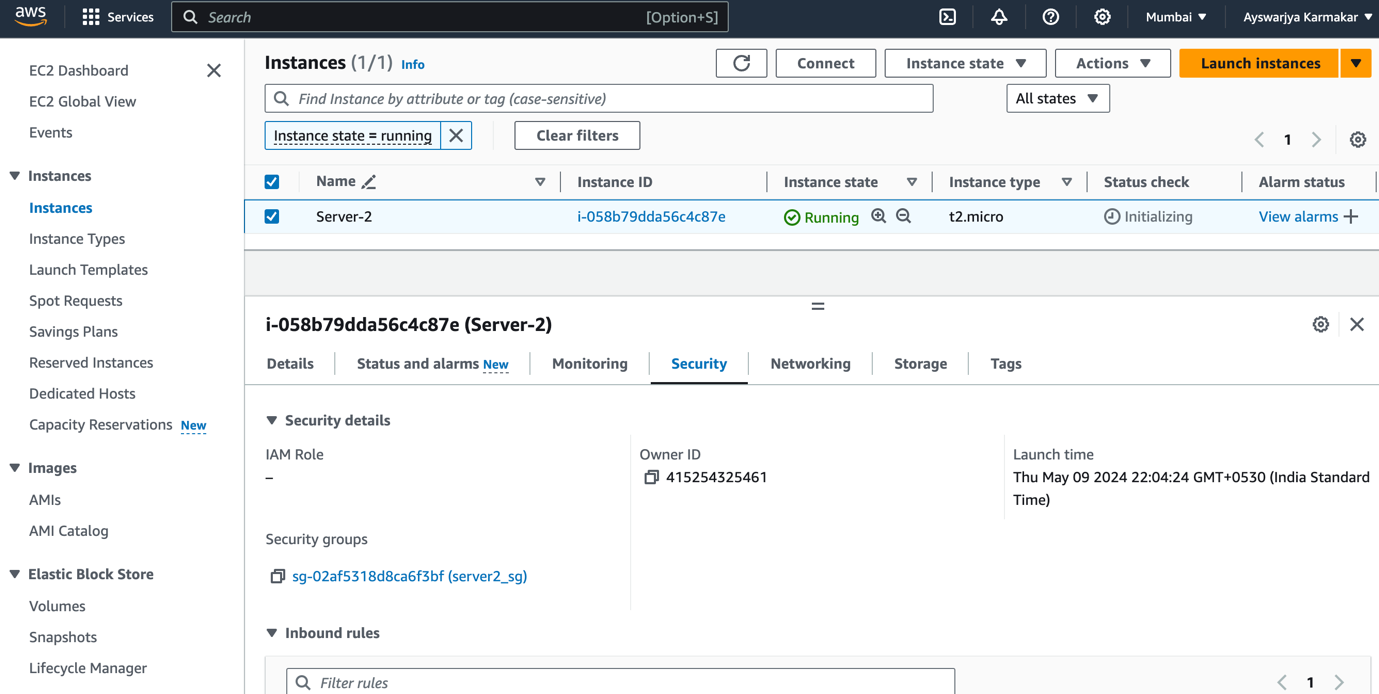
# Save file

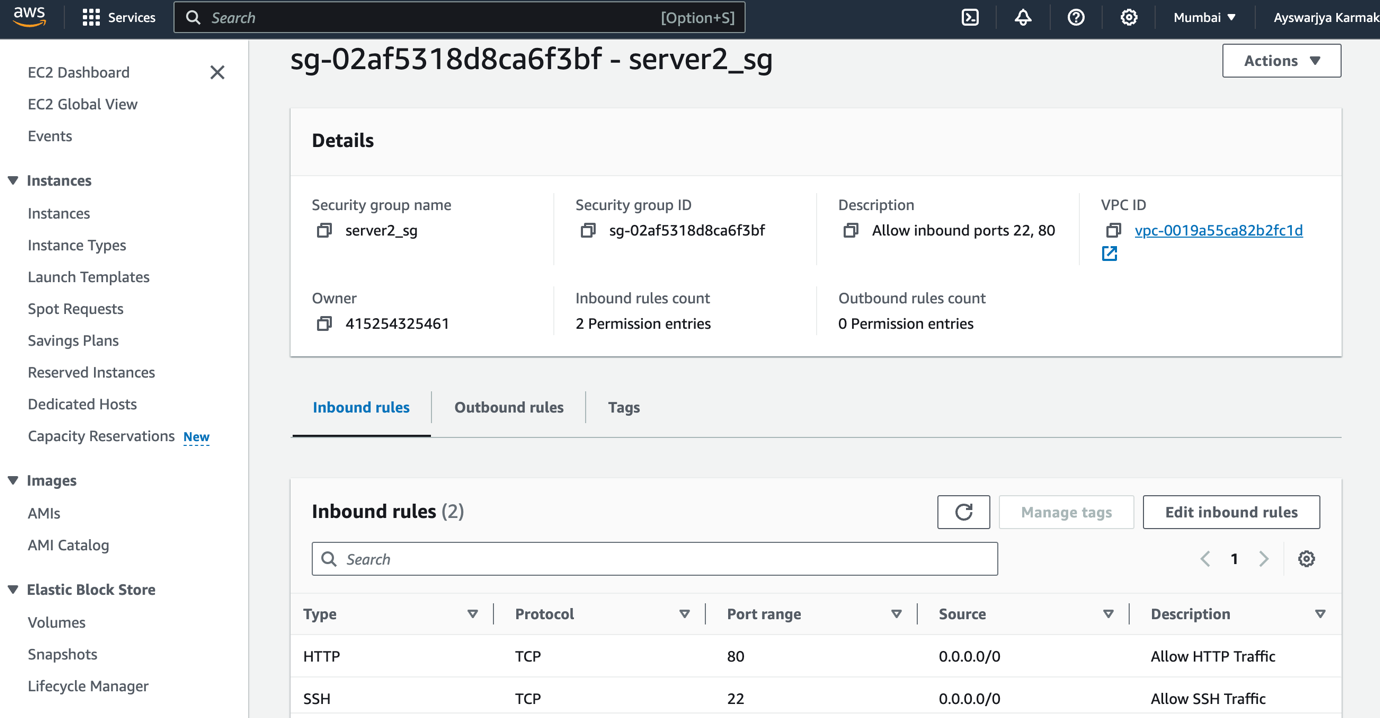
resource "local\_file" "ssh\_key" {

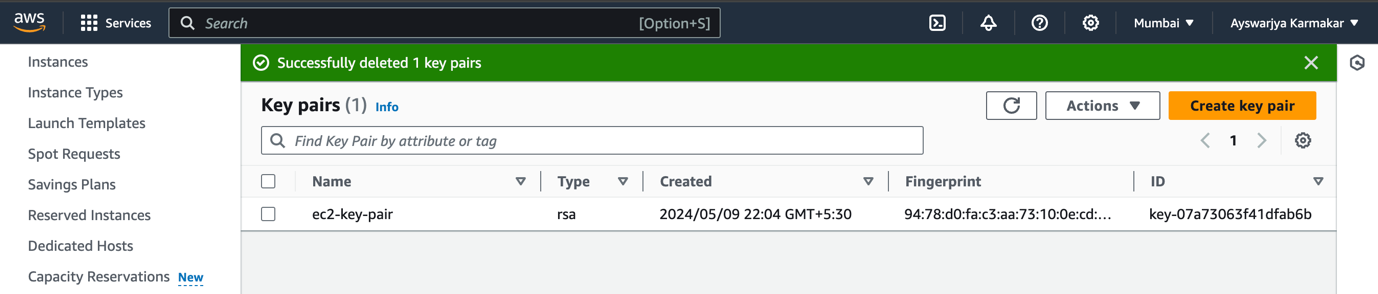
filename = "${aws\_key\_pair.key\_pair.key\_name}.pem"

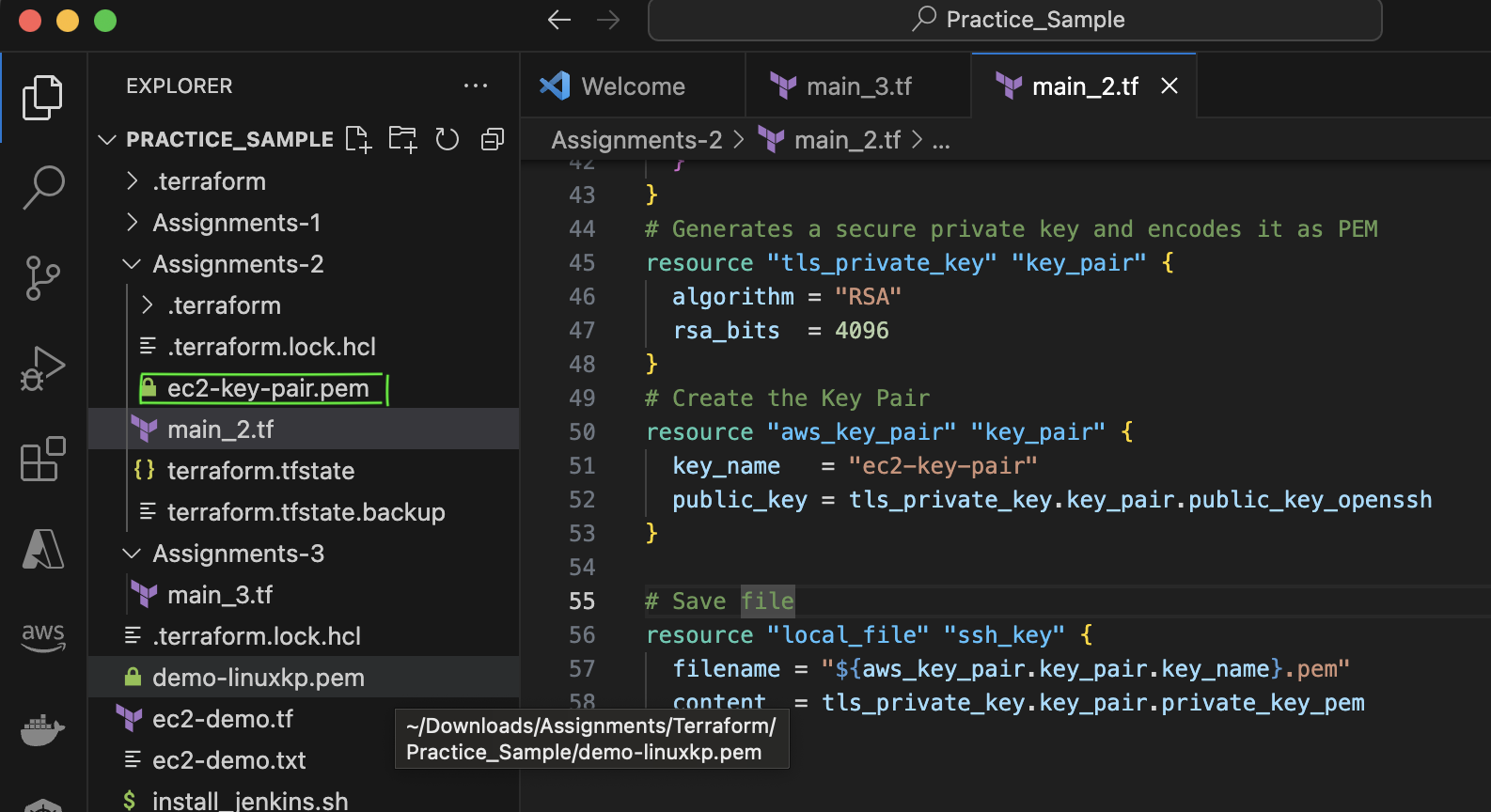
content = tls\_private\_key.key\_pair.private\_key\_pem

}









# Assignment 3 Write a terraform code would create following resources

# keypair (Public key– ON AWS – Private -key – Local) Display a pvt key

# on Console Create a Security group must open 80 Port Create a Ec2 instance

# Ubuntu using the same Key & SecurityGroup which you created above.

# You must install/Start apache2 in EC2 instance :

# $ apt-get update

# $ apt-get install apache2 -y

# $ systemctl start apache2

provider "aws" {

region = "ap-south-1"

}

# Generates a secure private key and encodes it as PEM

resource "tls\_private\_key" "key\_pair" {

algorithm = "RSA"

rsa\_bits = 4096

}

# Create the Key Pair

resource "aws\_key\_pair" "key\_pair" {

key\_name = "ubuntu-key-pair"

public\_key = tls\_private\_key.key\_pair.public\_key\_openssh

}

# Save file

resource "local\_file" "ssh\_key" {

filename = "${aws\_key\_pair.key\_pair.key\_name}.pem"

content = tls\_private\_key.key\_pair.private\_key\_pem

}

output "keypair" {

value = file(tls\_private\_key.key\_pair.private\_key\_pem)

#sensitive = true

}

output "keypair1" {

value = "${aws\_key\_pair.key\_pair.key\_name}.pem"

}

#Create security group

resource "aws\_security\_group" "server3\_sg" {

name = "server3\_sg"

description = "Allow inbound ports 80"

vpc\_id = "vpc-0019a55ca82b2fc1d"

#Allow incoming TCP requests on port 80 from any IP

ingress {

description = "Allow HTTP Traffic"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

#Allow incoming TCP requests on port 22 from any IP

ingress {

description = "Allow SSH Traffic"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

resource "aws\_instance" "web2" {

ami = "ami-0f58b397bc5c1f2e8"

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.key\_pair.key\_name

vpc\_security\_group\_ids = [aws\_security\_group.server3\_sg.id]

tags = {

Name = "Server-3"

}

connection {

type = "ssh"

user = "ubuntu"

private\_key = file("./${aws\_key\_pair.key\_pair.key\_name}.pem")

host = self.public\_ip

}

provisioner "file" {

source = "./index.html"

destination = "/tmp/index.html"

}

provisioner "remote-exec" {

inline = [

"sudo apt-get update",

"sudo apt-get install apache2 -y",

"sudo cp -r /tmp/index.html /var/www/html/",

"sudo systemctl start apache2",

]

}

}

<html>

<body style="background-color: rgb(0, 255, 8);">

<h1> Welcome to UBUNTU Server Ayswarjya </h1>

<p>{tdevs}-Learning terraform is easy</p>

</body>

</html>

