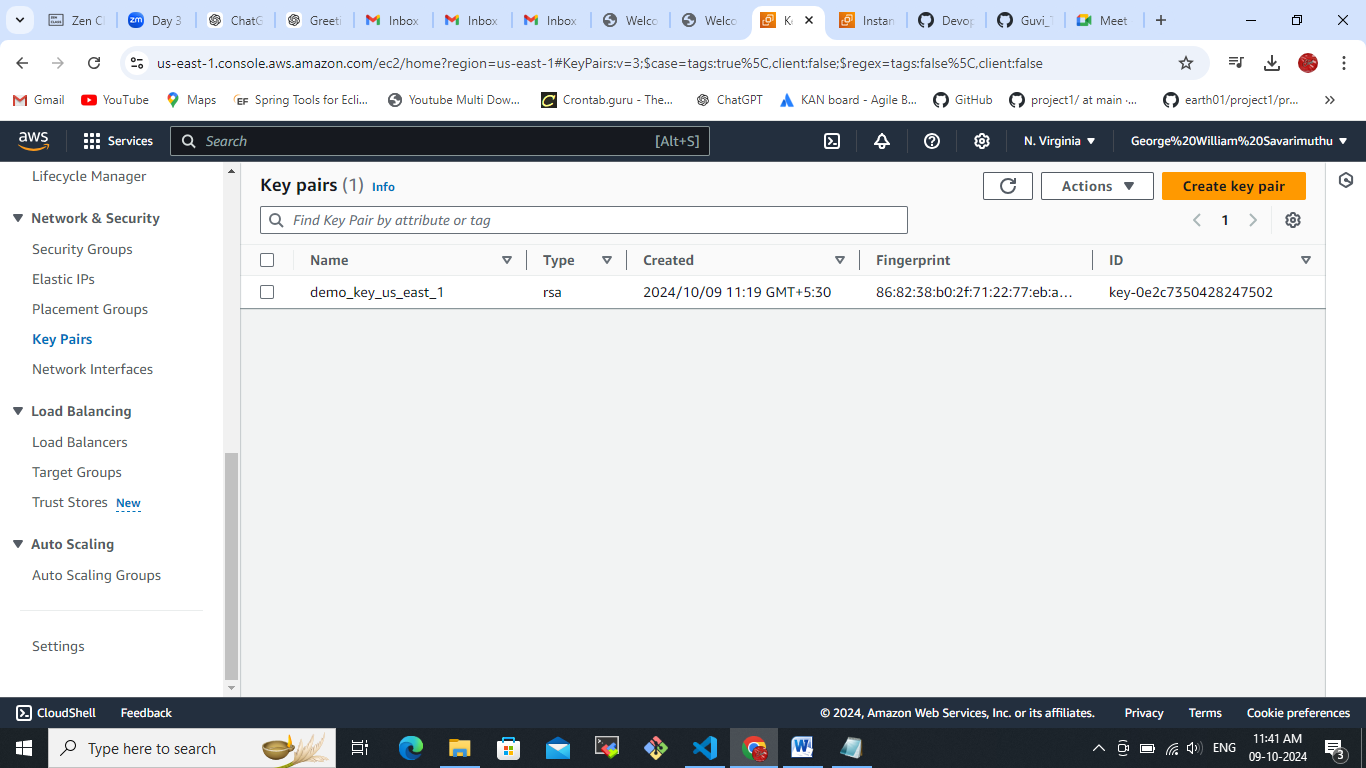
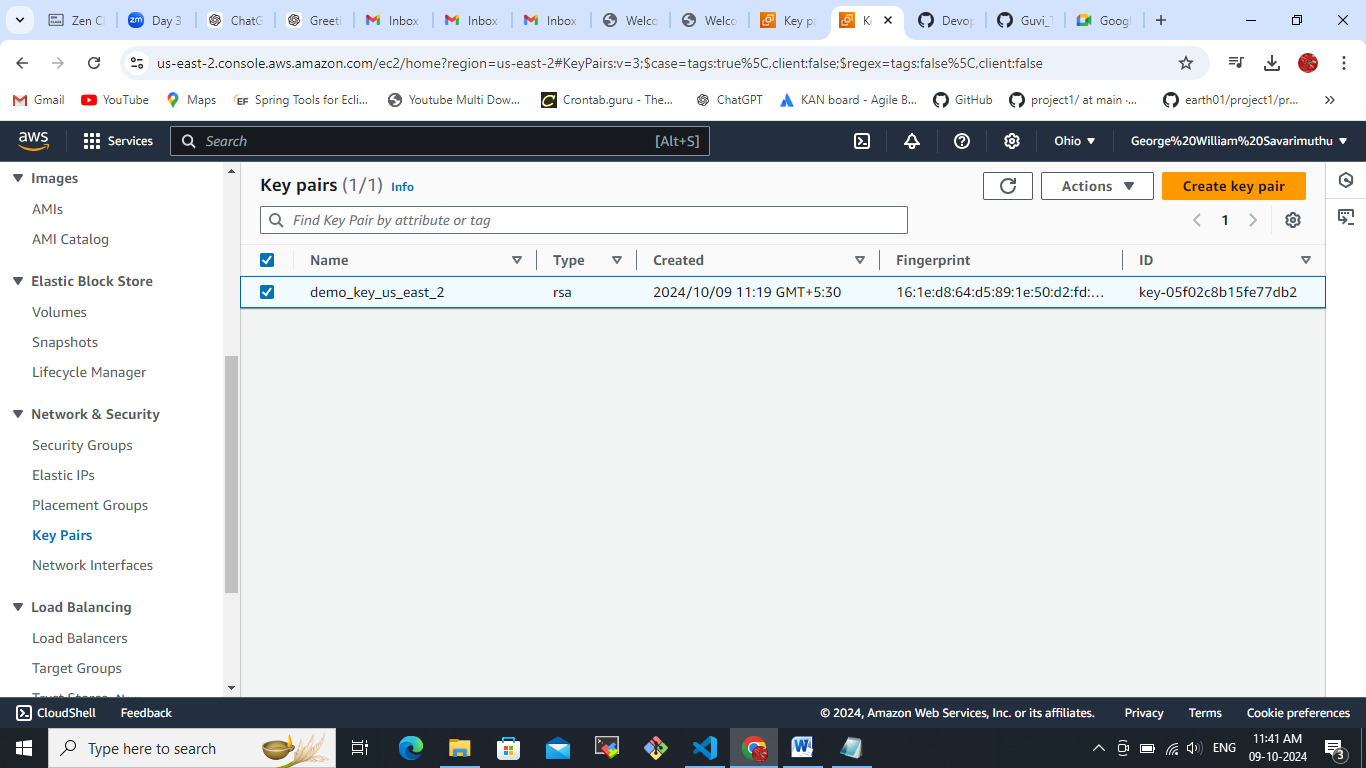
Day-25  
create 2 EC2 instance on 2 different regions and install nginx using terraform script

First we have create two key pairs from the two regions  
for us-east-1  


For us-east-2  


We have to change the terraform file as key name and subnet and others changes.  
Main..tf

# Provider Configuration for first region (us-east-1)

provider "aws" {

  alias  = "us-east-1"

  region = "us-east-1"  # First AWS Region (N. Virginia)

}

# Provider Configuration for second region (us-east-2)

provider "aws" {

  alias  = "us-east-2"

  region = "us-east-2"  # Second AWS Region (Ohio)

}

# VPC in us-east-1

resource "aws\_vpc" "vpc\_us\_east\_1" {

  provider   = aws.us-east-1

  cidr\_block = "10.0.0.0/16"

  tags = {

    Name = "us-east-1-vpc"

  }

}

# VPC in us-east-2

resource "aws\_vpc" "vpc\_us\_east\_2" {

  provider   = aws.us-east-2

  cidr\_block = "10.1.0.0/16"

  tags = {

    Name = "us-east-2-vpc"

  }

}

# Subnet in us-east-1

resource "aws\_subnet" "subnet\_us\_east\_1" {

  provider            = aws.us-east-1

  vpc\_id              = aws\_vpc.vpc\_us\_east\_1.id

  cidr\_block          = "10.0.1.0/24"

  availability\_zone   = "us-east-1a"

  map\_public\_ip\_on\_launch = true

  tags = {

    Name = "us-east-1-subnet"

  }

}

# Subnet in us-east-2

resource "aws\_subnet" "subnet\_us\_east\_2" {

  provider            = aws.us-east-2

  vpc\_id              = aws\_vpc.vpc\_us\_east\_2.id

  cidr\_block          = "10.1.1.0/24"

  availability\_zone   = "us-east-2a"

  map\_public\_ip\_on\_launch = true

  tags = {

    Name = "us-east-2-subnet"

  }

}

# Internet Gateway for us-east-1

resource "aws\_internet\_gateway" "igw\_us\_east\_1" {

  provider = aws.us-east-1

  vpc\_id   = aws\_vpc.vpc\_us\_east\_1.id

  tags = {

    Name = "us-east-1-igw"

  }

}

# Internet Gateway for us-east-2

resource "aws\_internet\_gateway" "igw\_us\_east\_2" {

  provider = aws.us-east-2

  vpc\_id   = aws\_vpc.vpc\_us\_east\_2.id

  tags = {

    Name = "us-east-2-igw"

  }

}

# Route table and route for us-east-1

resource "aws\_route\_table" "route\_table\_us\_east\_1" {

  provider = aws.us-east-1

  vpc\_id   = aws\_vpc.vpc\_us\_east\_1.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.igw\_us\_east\_1.id

  }

}

resource "aws\_route\_table\_association" "rt\_assoc\_us\_east\_1" {

  provider      = aws.us-east-1

  subnet\_id     = aws\_subnet.subnet\_us\_east\_1.id

  route\_table\_id = aws\_route\_table.route\_table\_us\_east\_1.id

}

# Route table and route for us-east-2

resource "aws\_route\_table" "route\_table\_us\_east\_2" {

  provider = aws.us-east-2

  vpc\_id   = aws\_vpc.vpc\_us\_east\_2.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.igw\_us\_east\_2.id

  }

}

resource "aws\_route\_table\_association" "rt\_assoc\_us\_east\_2" {

  provider      = aws.us-east-2

  subnet\_id     = aws\_subnet.subnet\_us\_east\_2.id

  route\_table\_id = aws\_route\_table.route\_table\_us\_east\_2.id

}

# VPC Security Group for EC2 in us-east-1 (N. Virginia)

resource "aws\_security\_group" "sg\_us\_east\_1" {

  provider = aws.us-east-1

  vpc\_id   = aws\_vpc.vpc\_us\_east\_1.id

  ingress {

    from\_port   = 22

    to\_port     = 22

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]  # SSH from anywhere

  }

  ingress {

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]  # HTTP from anywhere

  }

  egress {

    from\_port   = 0

    to\_port     = 0

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]  # Allow all outbound traffic

  }

  tags = {

    Name = "us-east-1-sg"

  }

}

# VPC Security Group for EC2 in us-east-2 (Ohio)

resource "aws\_security\_group" "sg\_us\_east\_2" {

  provider = aws.us-east-2

  vpc\_id   = aws\_vpc.vpc\_us\_east\_2.id

  ingress {

    from\_port   = 22

    to\_port     = 22

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]  # SSH from anywhere

  }

  ingress {

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]  # HTTP from anywhere

  }

  egress {

    from\_port   = 0

    to\_port     = 0

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]  # Allow all outbound traffic

  }

  tags = {

    Name = "us-east-2-sg"

  }

}

# EC2 Instance in us-east-1 (N. Virginia)

resource "aws\_instance" "ec2\_us\_east\_1" {

  provider          = aws.us-east-1

  ami               = "ami-0fff1b9a61dec8a5f"  # Updated AMI

  instance\_type     = "t2.micro"

  key\_name          =  "demo\_key\_us\_east\_1"

  subnet\_id         = aws\_subnet.subnet\_us\_east\_1.id

  vpc\_security\_group\_ids = [aws\_security\_group.sg\_us\_east\_1.id]

  associate\_public\_ip\_address = true

  tags = {

    Name = "EC2-us-east-1"

  }

  # Provisioner to install NGINX

  provisioner "remote-exec" {

    connection {

      type        = "ssh"

      user        = "ec2-user"

      private\_key = file("C:\\Users\\DELL\\Downloads\\demo\_key\_us\_east\_1.pem")

      host        = self.public\_ip

    }

    inline = [

      "sudo yum update -y",

      "sudo yum install -y nginx",

      "sudo systemctl start nginx",

      "sudo systemctl enable nginx"

    ]

  }

}

# EC2 Instance in us-east-2 (Ohio)

resource "aws\_instance" "ec2\_us\_east\_2" {

  provider          = aws.us-east-2

  ami               = "ami-09da212cf18033880"  # Updated AMI

  instance\_type     = "t2.micro"

  key\_name          = "demo\_key\_us\_east\_2"

  subnet\_id         = aws\_subnet.subnet\_us\_east\_2.id

  vpc\_security\_group\_ids = [aws\_security\_group.sg\_us\_east\_2.id]

  associate\_public\_ip\_address = true

  tags = {

    Name = "EC2-us-east-1"

  }

  # Provisioner to install NGINX

  provisioner "remote-exec" {

    connection {

      type        = "ssh"

      user        = "ec2-user"

      private\_key = file("C:\\Users\\DELL\\Downloads\\demo\_key\_us\_east\_2.pem")

      host        = self.public\_ip

    }

    inline = [

      "sudo yum update -y",

      "sudo yum install -y nginx",

      "sudo systemctl start nginx",

      "sudo systemctl enable nginx"

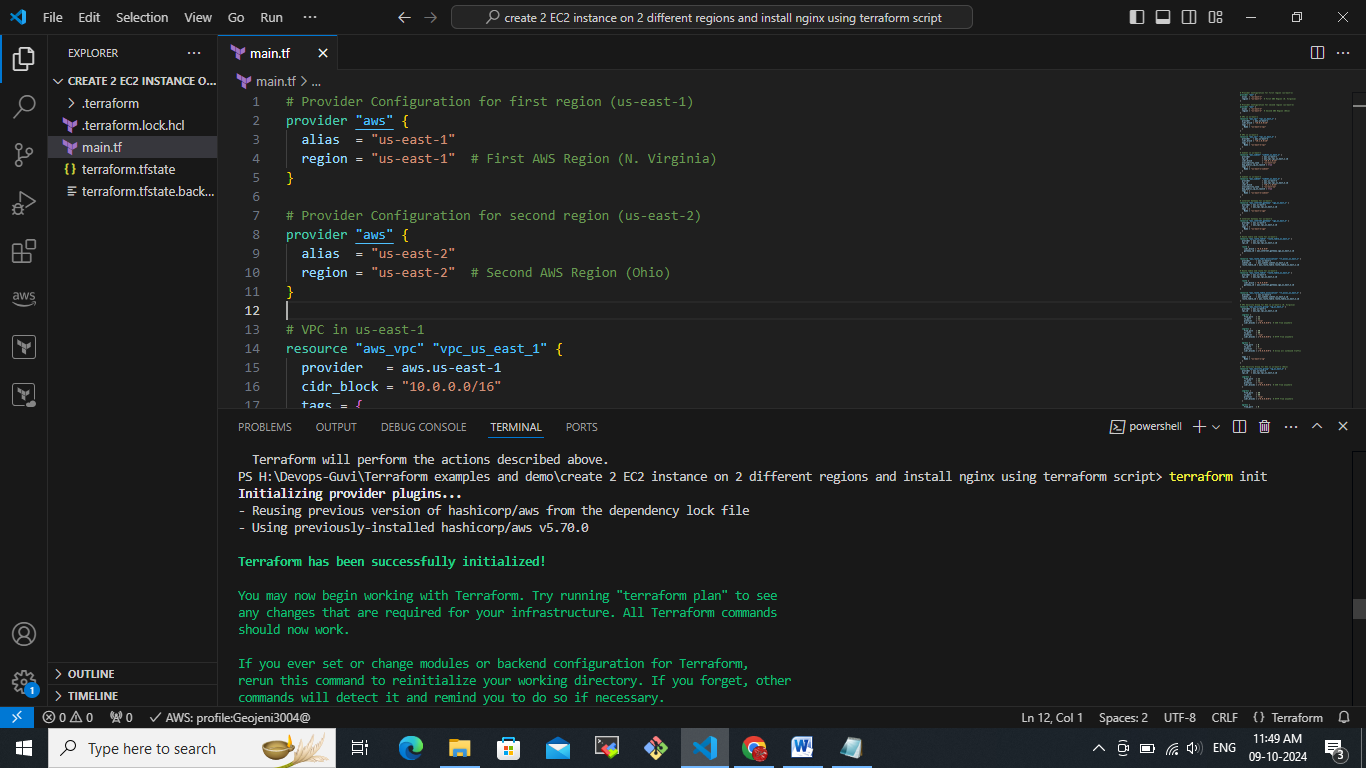
    ]

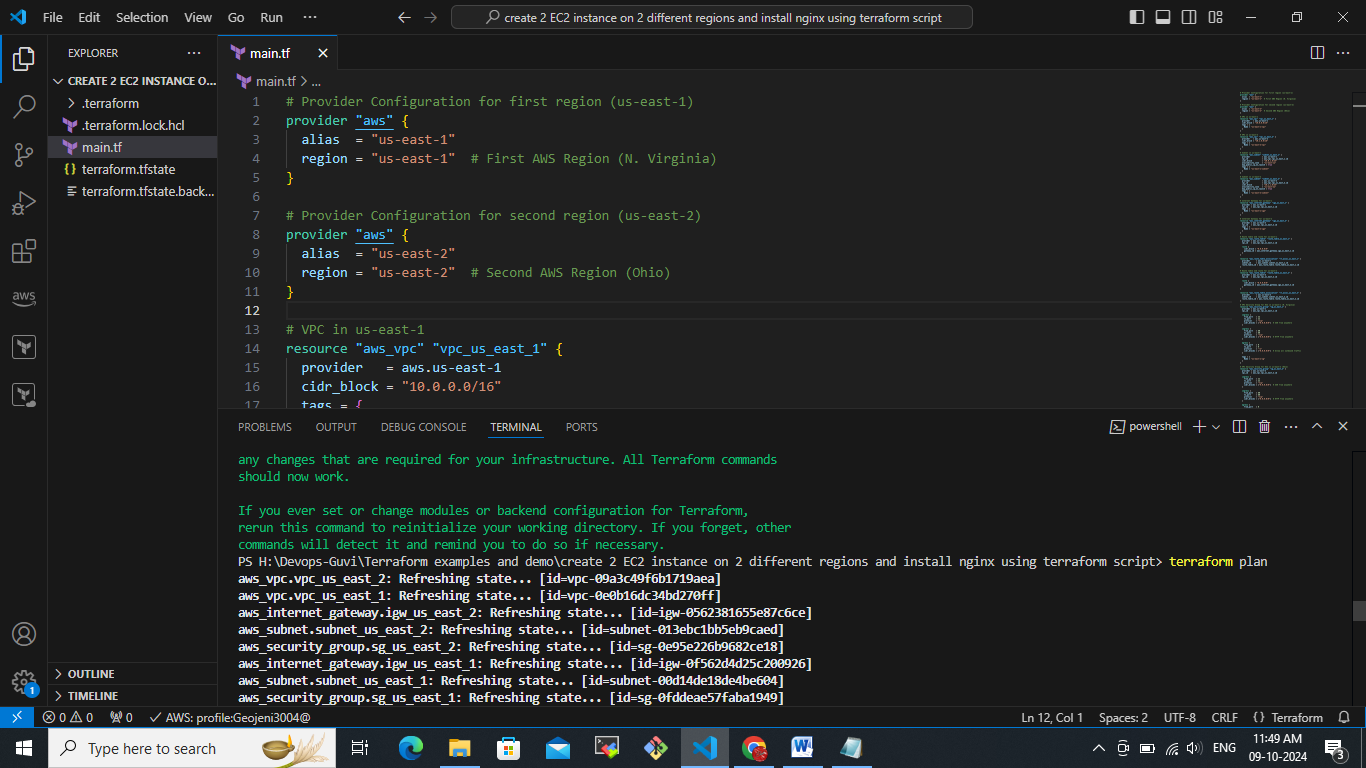
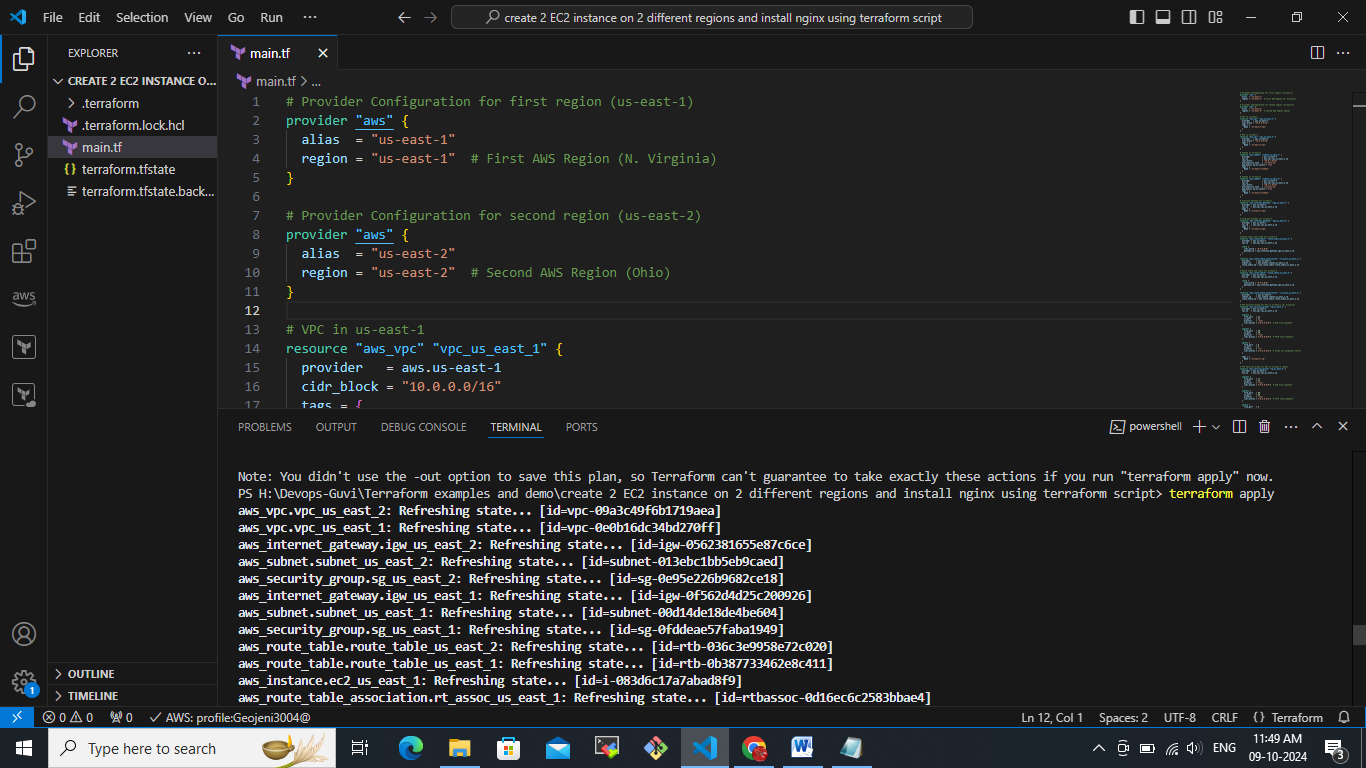
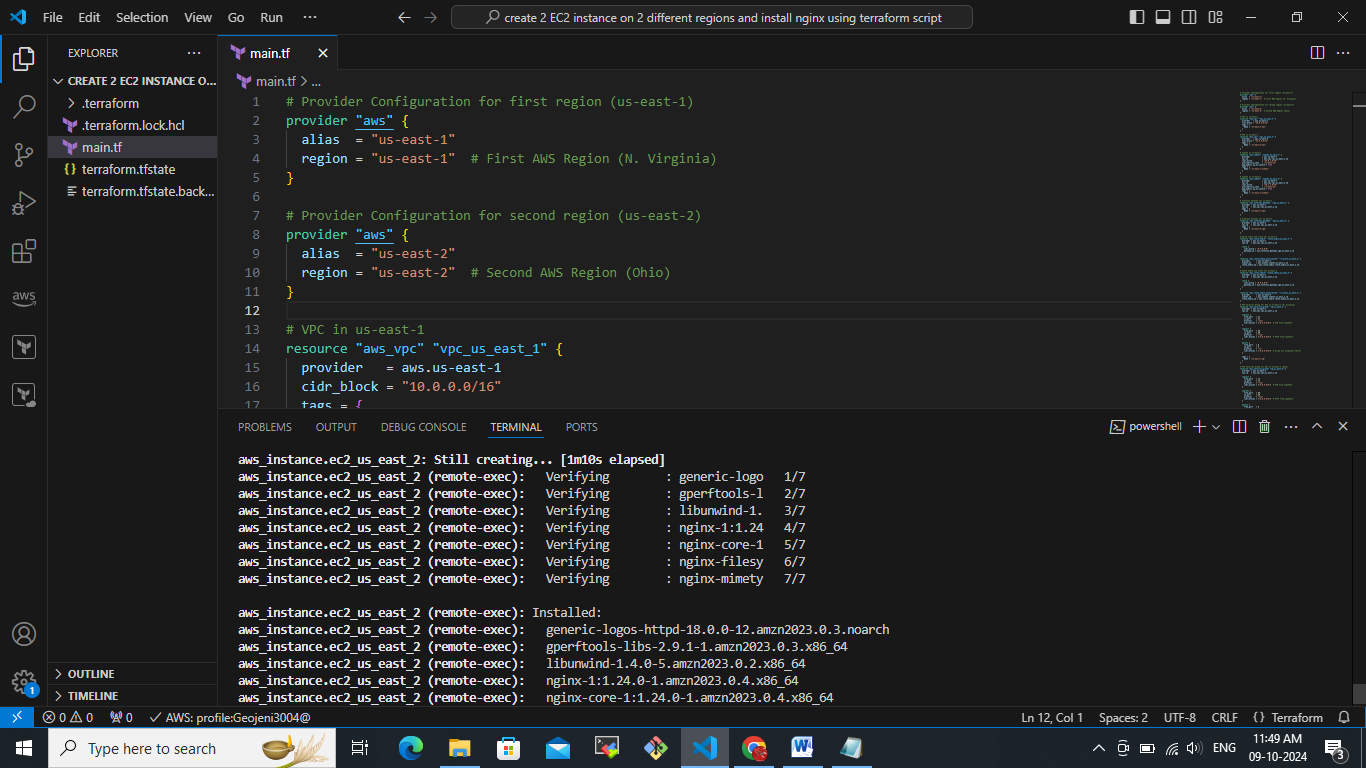
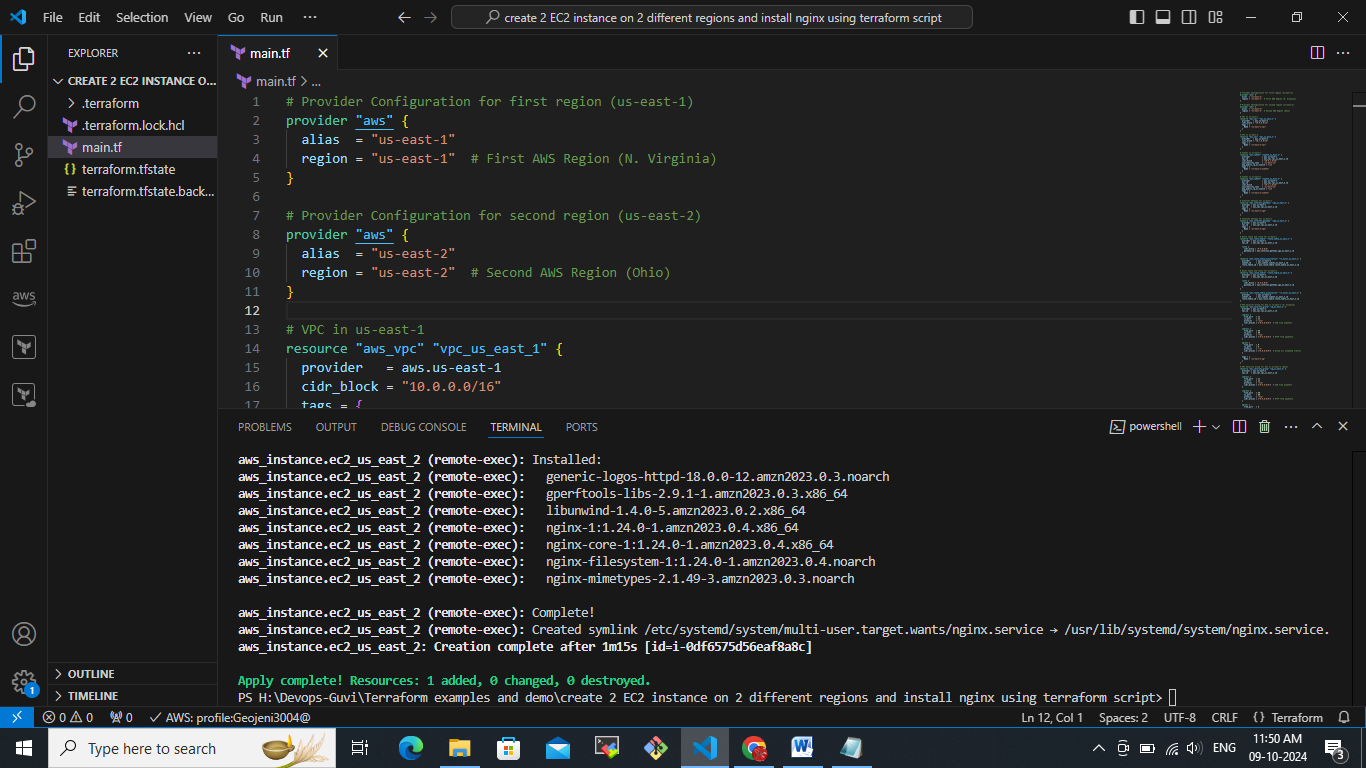
  }

}

key\_name          =  "demo\_key\_us\_east\_1"

key\_name          =  "demo\_key\_us\_east\_2"

To execute the below commands  
terraform init  


Output screen we could see two ec2 instances were created from us-east-1 and us-east-2 and we can able to access the ngix application over the public IP’s of those EC2 instances.

