**Terraform Day-06**

***Mohammed Rafiyoodin – 06-05-2024***

1) Create VPC

resource "aws\_vpc" "main" {

  cidr\_block       = "192.168.0.0/24"

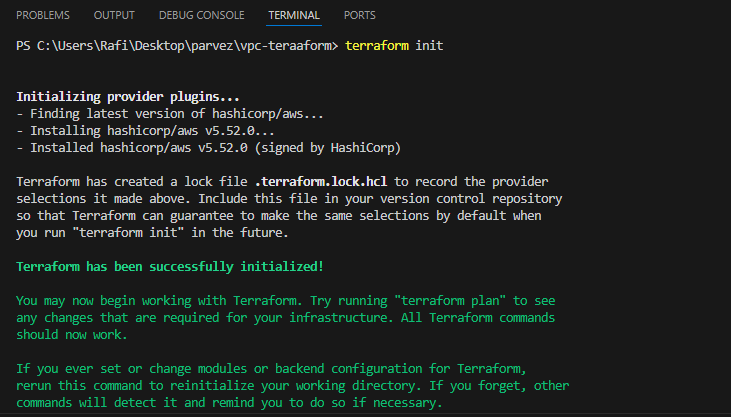
  instance\_tenancy = "default"

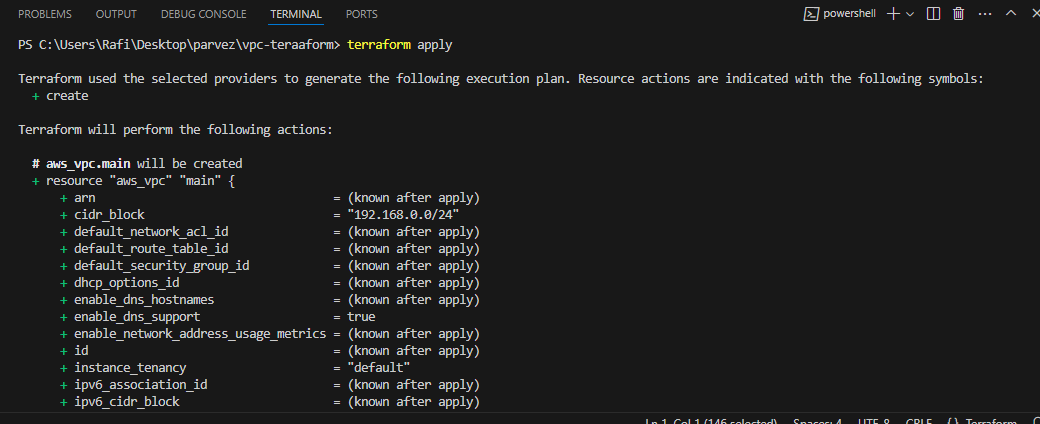
  tags = {

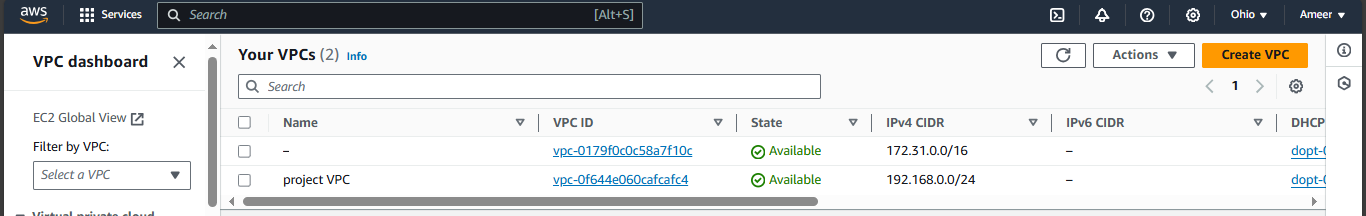
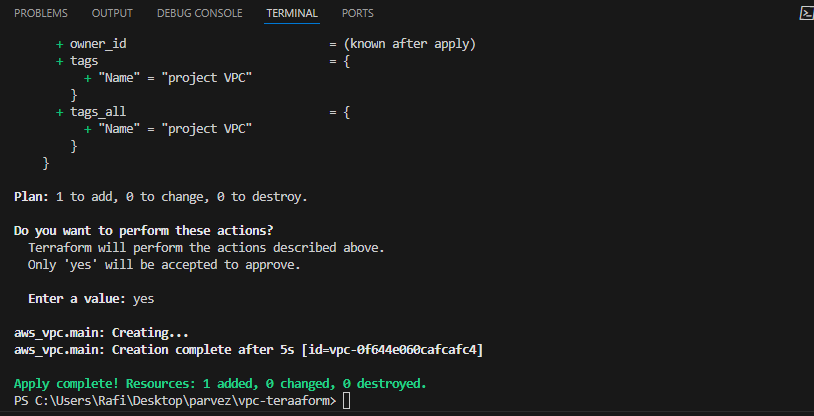
    Name = "project VPC"

  }

}







2) Create Internet gateway

# Internet Gateway

resource "aws\_internet\_gateway" "igw" {

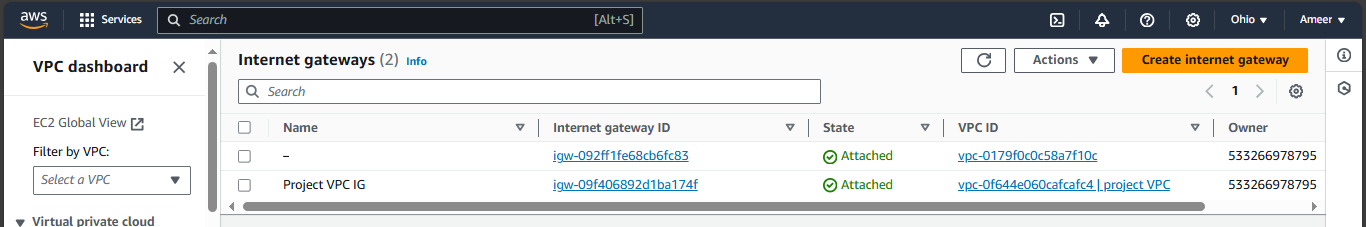
  vpc\_id = aws\_vpc.main.id

  tags = {

    Name = "Project VPC IG"

  }

}



3) Create Custom Route Table

# Creating Route Table

resource "aws\_route\_table" "route" {

  vpc\_id = aws\_vpc.main.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.igw.id

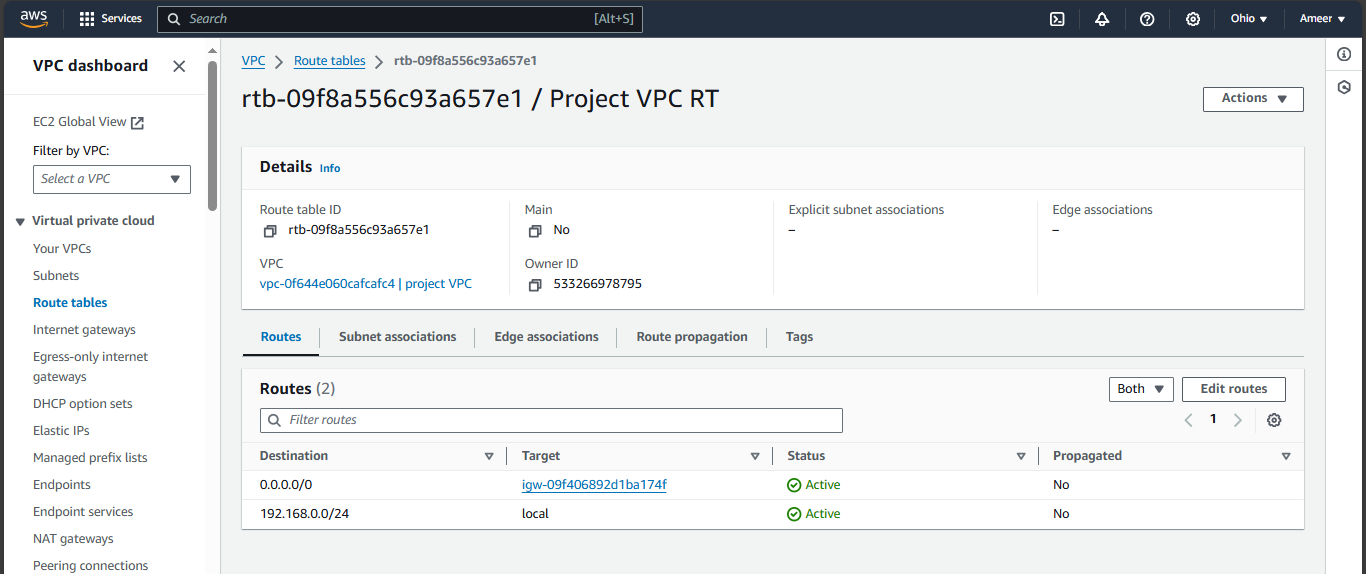
  }

  tags = {

    Name = "Project VPC RT"

  }

}



4) Create Subnet

}# Creating 1st my\_subnet

resource "aws\_subnet" "my\_subnet" {

  vpc\_id                  = aws\_vpc.main.id

  cidr\_block              = "192.168.0.0/24"

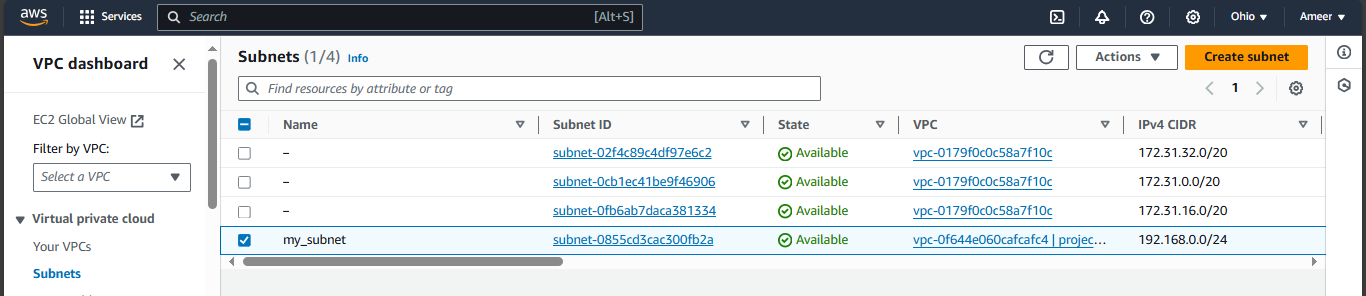
  availability\_zone       = "us-east-2b"

  tags = {

    Name = "my\_subnet"

  }

}



5) Associate subnet with Route Table

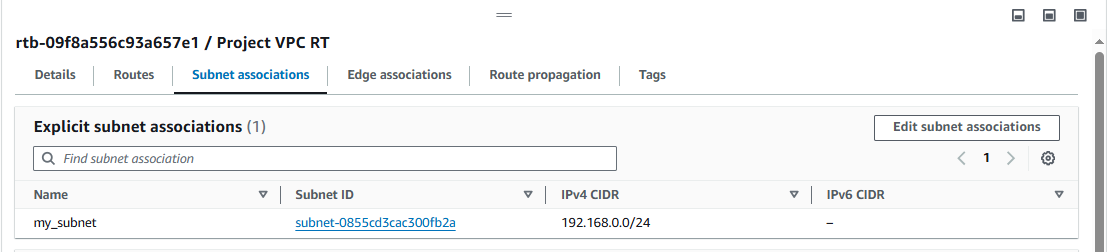
# Associating Route Table

resource "aws\_route\_table\_association" "my\_subnet\_association" {

  subnet\_id      = aws\_subnet.my\_subnet.id

  route\_table\_id = aws\_route\_table.route.id

}



6) Create Security Group to allow port 22.80,443

# Creating Security Group

resource "aws\_security\_group" "Rafisg" {

  vpc\_id = "${aws\_vpc.main.id}"

  # Inbound Rules

  # HTTP access from anywhere

  ingress {

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  # HTTPS access from anywhere

  ingress {

    from\_port   = 443

    to\_port     = 443

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  # SSH access from anywhere

  ingress {

    from\_port   = 22

    to\_port     = 22

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  # Outbound Rules

  # Internet access to anywhere

  egress {

    from\_port   = 0

    to\_port     = 0

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]

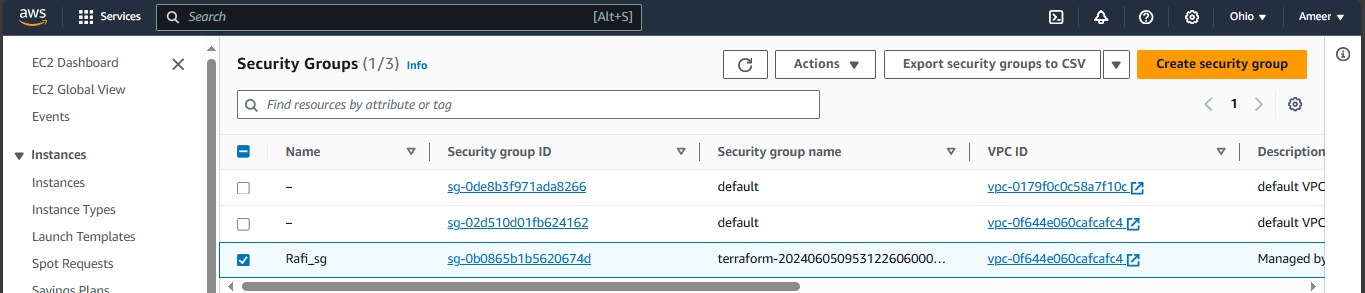
  }

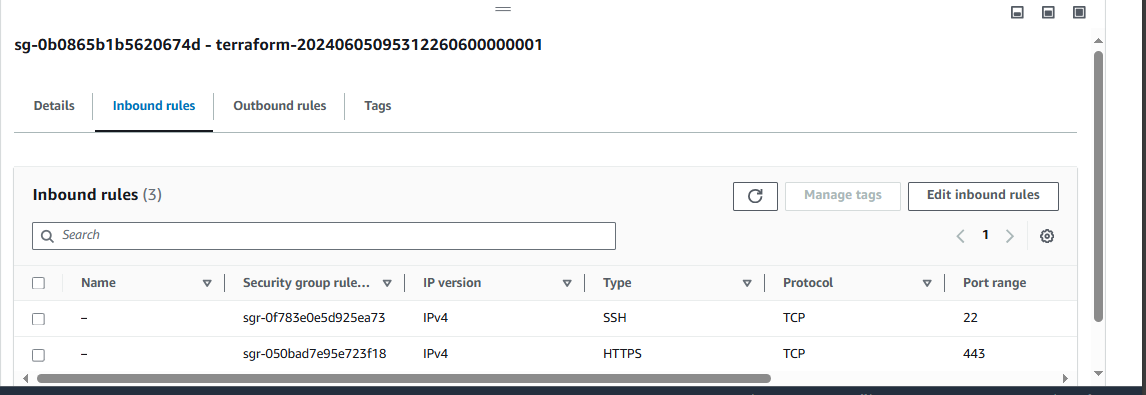
  tags = {

    Name = "Rafi\_sg"

  }

}





7) Create a network interface with an ip in the subnet that was created in step 4

resource "aws\_network\_interface" "project\_interface" {

subnet\_id = aws\_subnet.my\_subnet.id

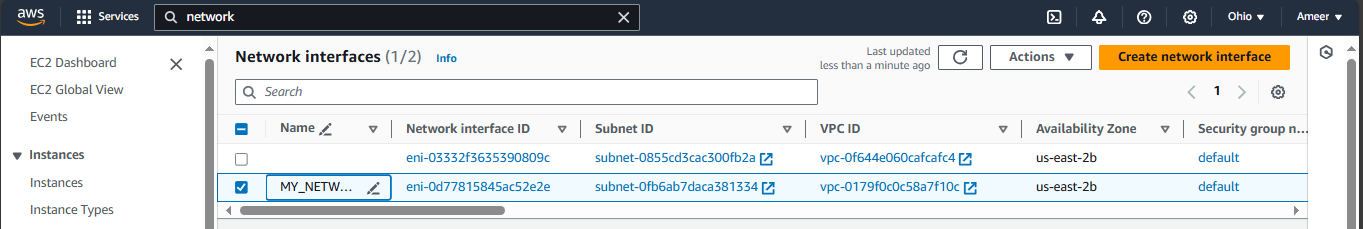
private\_ips = ["192.168.0.25"]

tags = {

name = "MY\_NETWORK\_INTERFACE"

}

}



8) Assign an elastic IP to the network interface created in step 7

resource "aws\_eip" "my\_ip" {

vpc = true

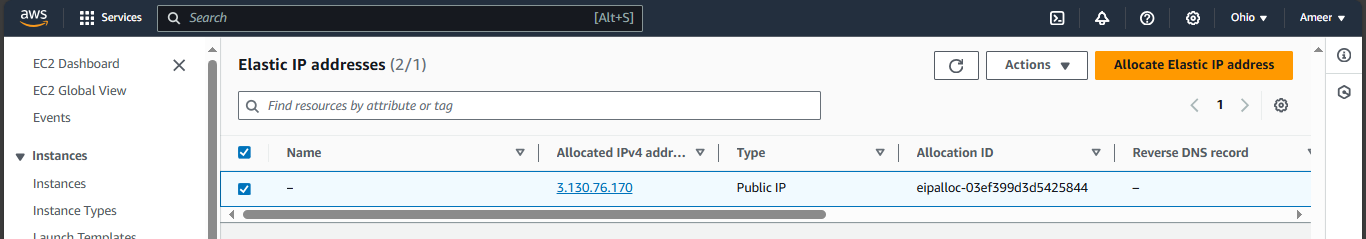
}

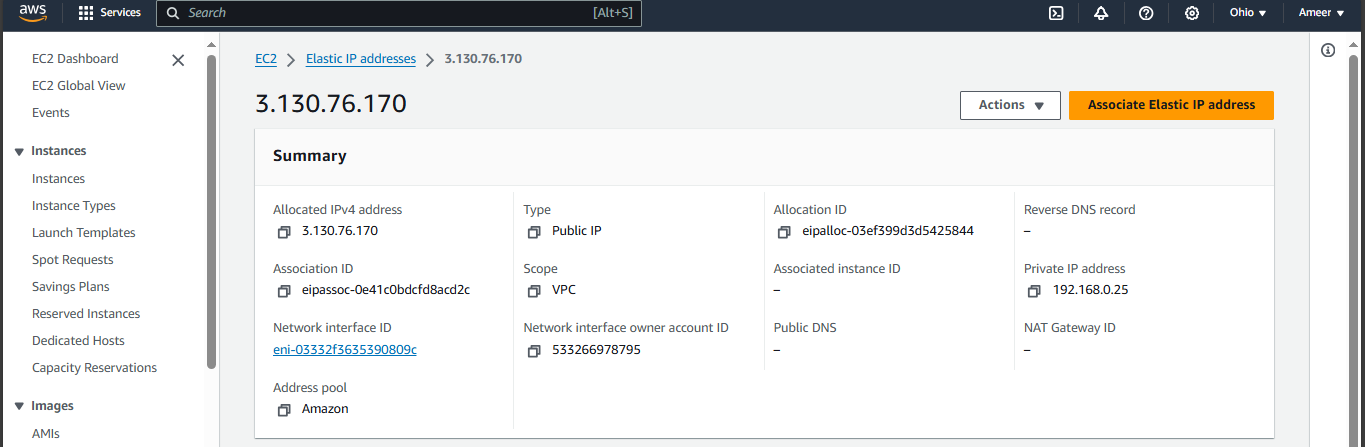
resource "aws\_eip\_association" "my\_eip\_association" {

    network\_interface\_id = aws\_network\_interface.project\_interface.id

    allocation\_id = aws\_eip.my\_ip.id

}





9) Create Ubuntu server and install/enable apache2

resource "aws\_instance" "ubuntu\_image" {

    ami = "ami-09040d770ffe2224f"

    instance\_type = "t2.micro"

    subnet\_id = aws\_subnet.my\_subnet.id

    vpc\_security\_group\_ids = [aws\_security\_group.Rafisg.id]

    user\_data = <<-EOF

                 #!/bin/bash

                 apt-get update

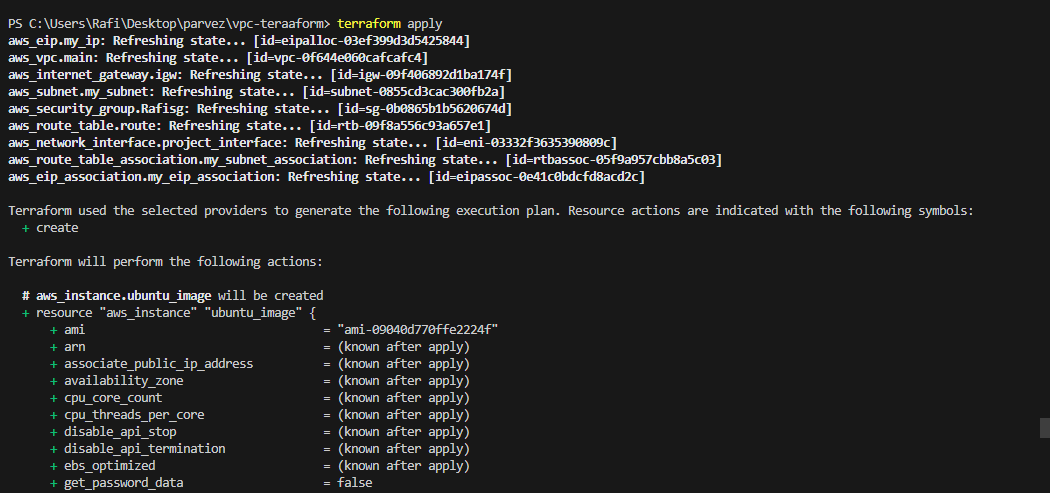
                 apt-get install -y apache2

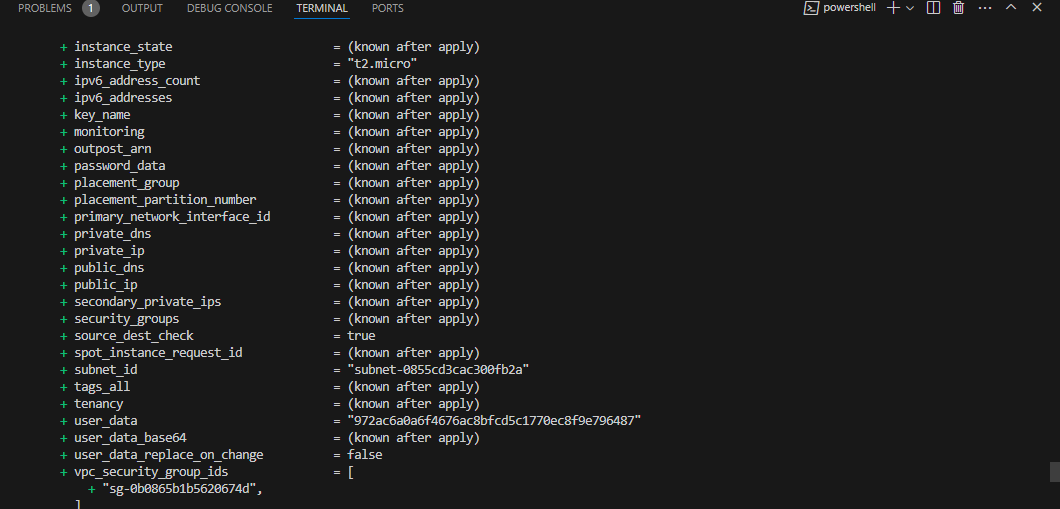
                 systemctl enable apache2

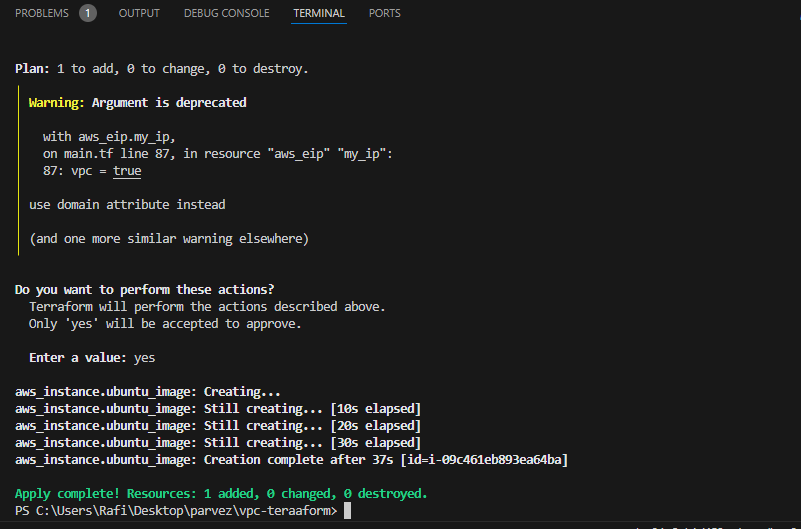
                 systemctl start apache2

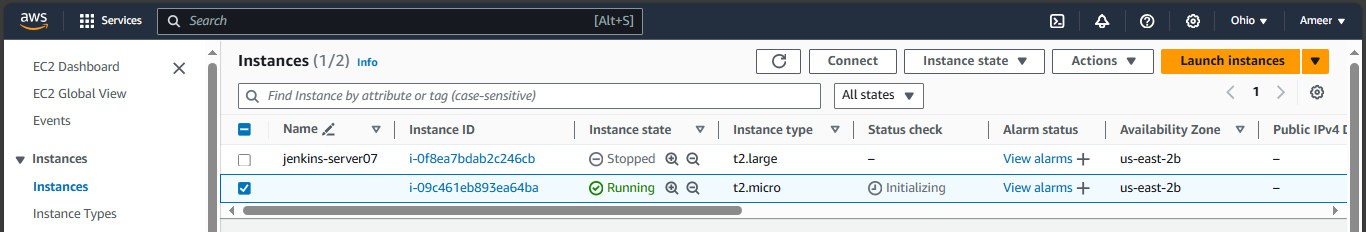
                 EOF

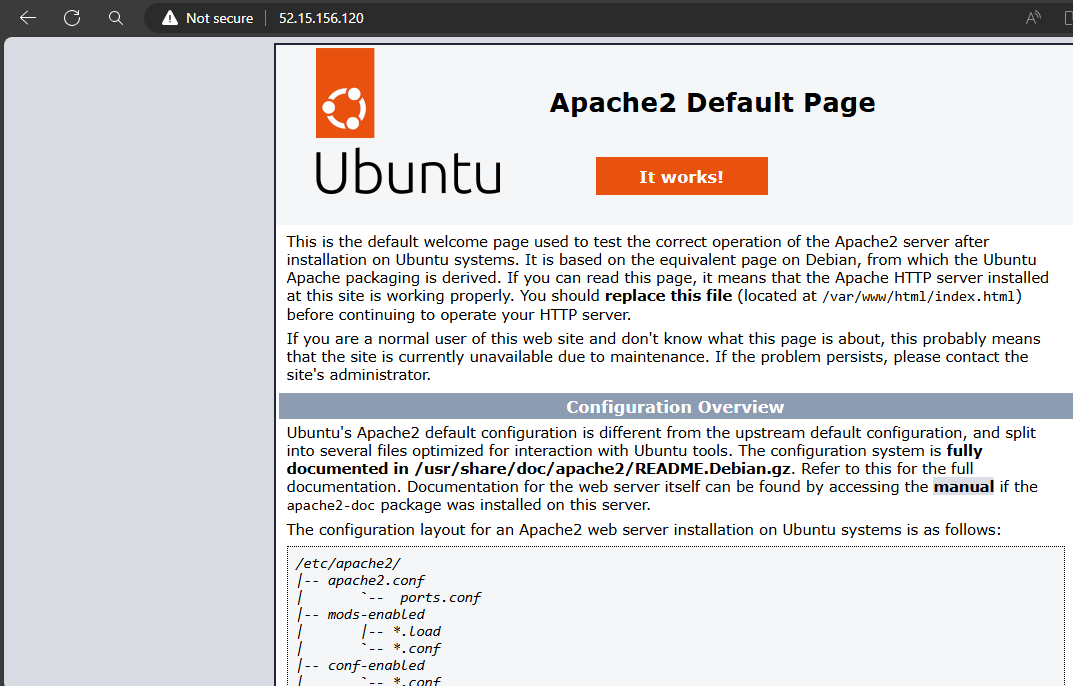
}







******

******

**Configure s3 as backend and dynamo db locking for multi user execution.**

#Create s3 using terraform

resource "aws\_s3\_bucket" "s3\_bucket" {

    bucket = "s3-backend-babji123"

    acl = "private"

}

# Create dynamo db using terraform

resource "aws\_dynamodb\_table" "dynamodb-terraform-state-lock" {

  name = "terraform-state-dynamoDB"

  hash\_key = "LockID"

  read\_capacity = 20

  write\_capacity = 20

  attribute {

    name = "LockID"

    type = "S"

  }

}

#S3 as backend for terraform.tfstate file

terraform {

  backend "s3" {

    bucket = "s3-backend-babji123"

    dynamodb\_table = "terraform-state-dynamoDB"

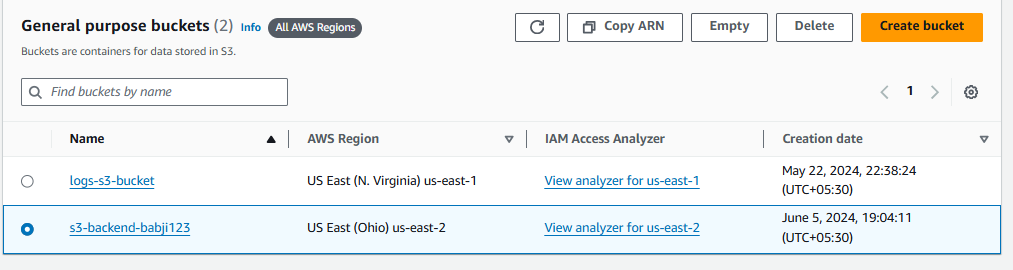
    key    = "terraform.tfstate"

    region = "us-east-2"

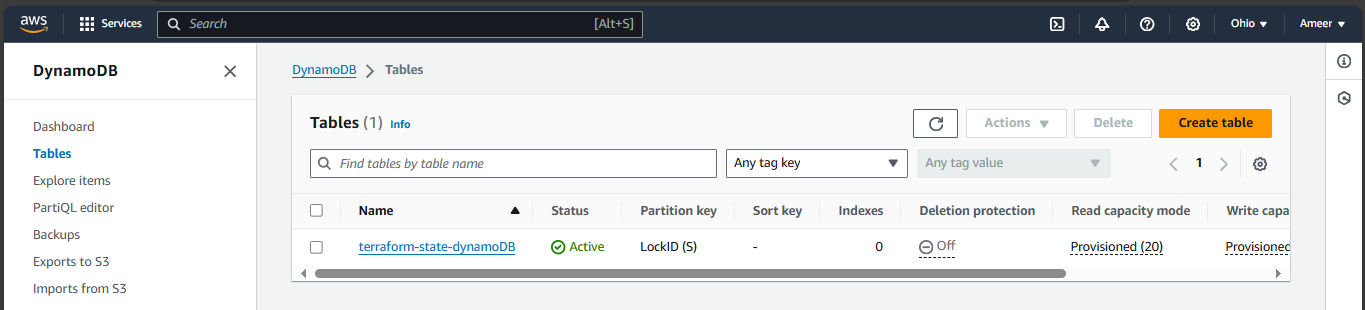
  }

}

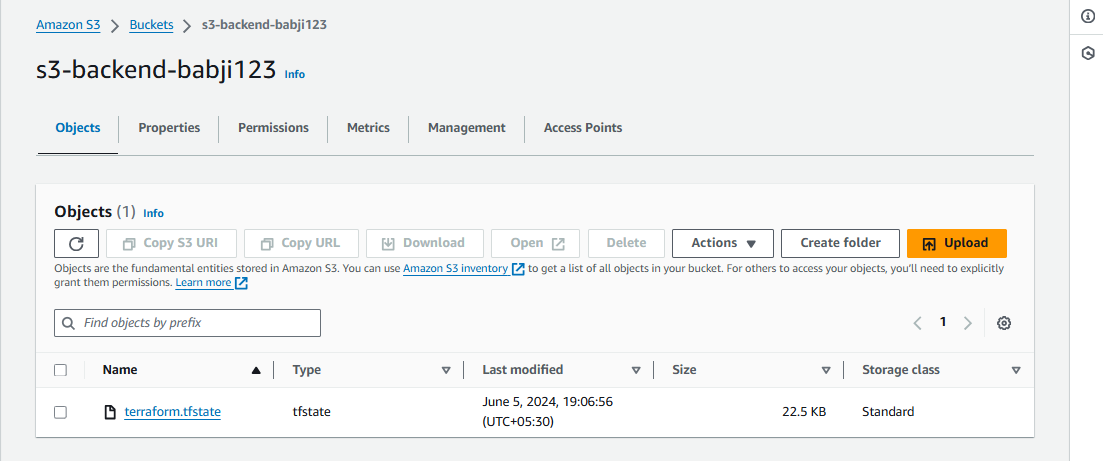
**Created S3 Bucket**

****

**Created dynamoDB**

****

**Created S3 as backend for terraform.tfstate.file**

****