1. **Create VPC**

**Here need to create the folder as terraform-aws-web in that needto create files as main.tf, var.tf, out.tf, provider.tf and need to update the terraform script to create vpc, IGW, RT, Subnet, SG, Elastic ip, ubuntu server….!**

Here we have created the vpc by using the terraform

**resource "aws\_vpc" "main" {**

**cidr\_block = var.vpc\_cidr**

**enable\_dns\_hostnames = true**

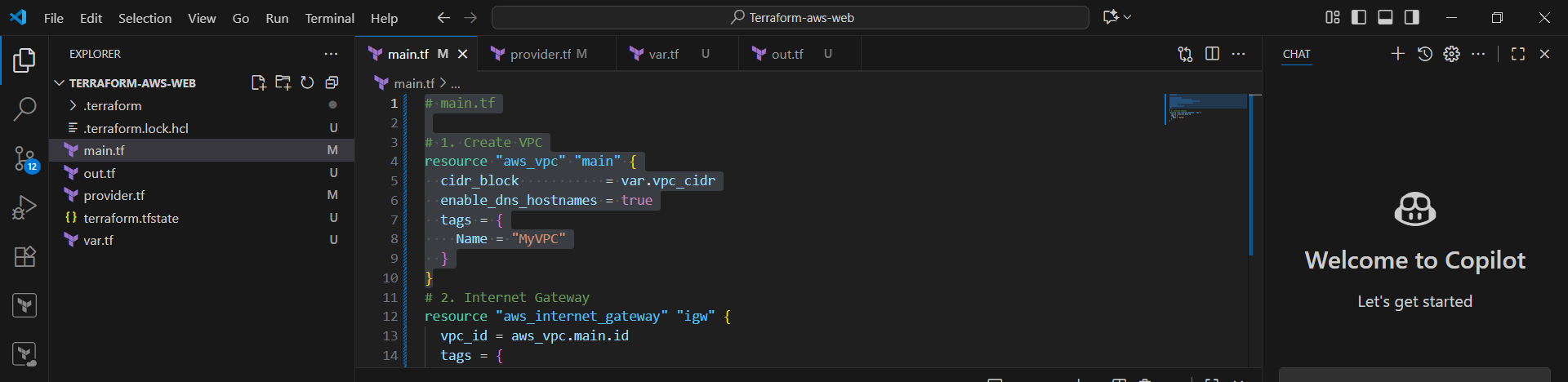
**tags = {**

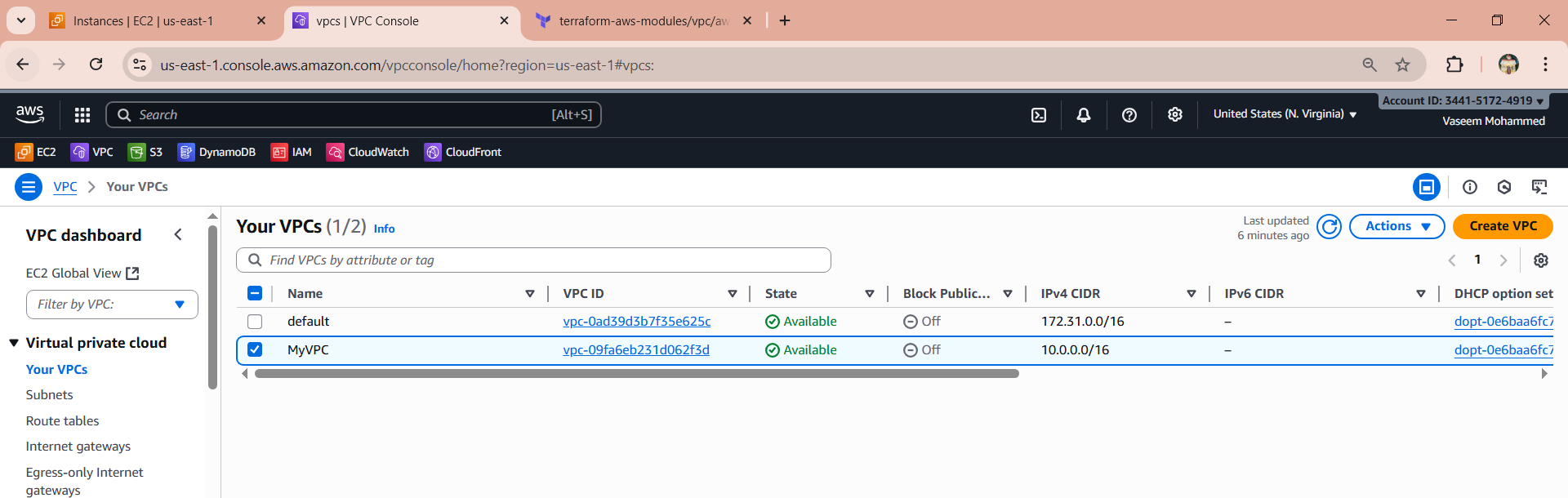
**Name = "MyVPC"**

**}**

**}**

**terraform init --> terraform apply**





1. **Create Internet gateway**

Now we have created the IGW by terraform

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

**vpc\_id = aws\_vpc.main.id**

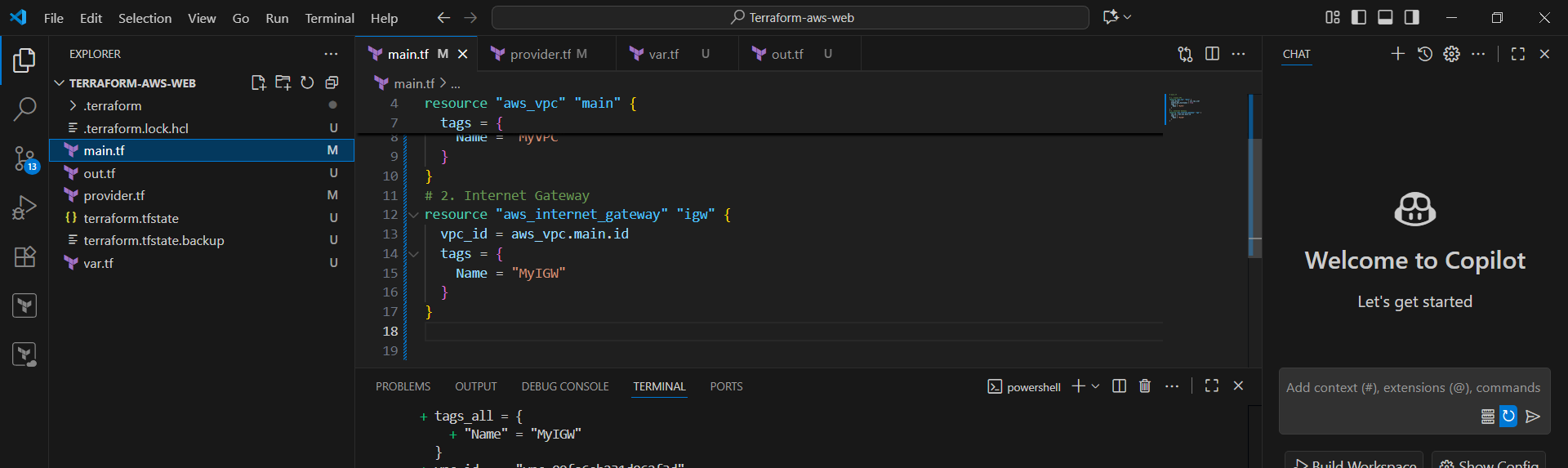
**tags = {**

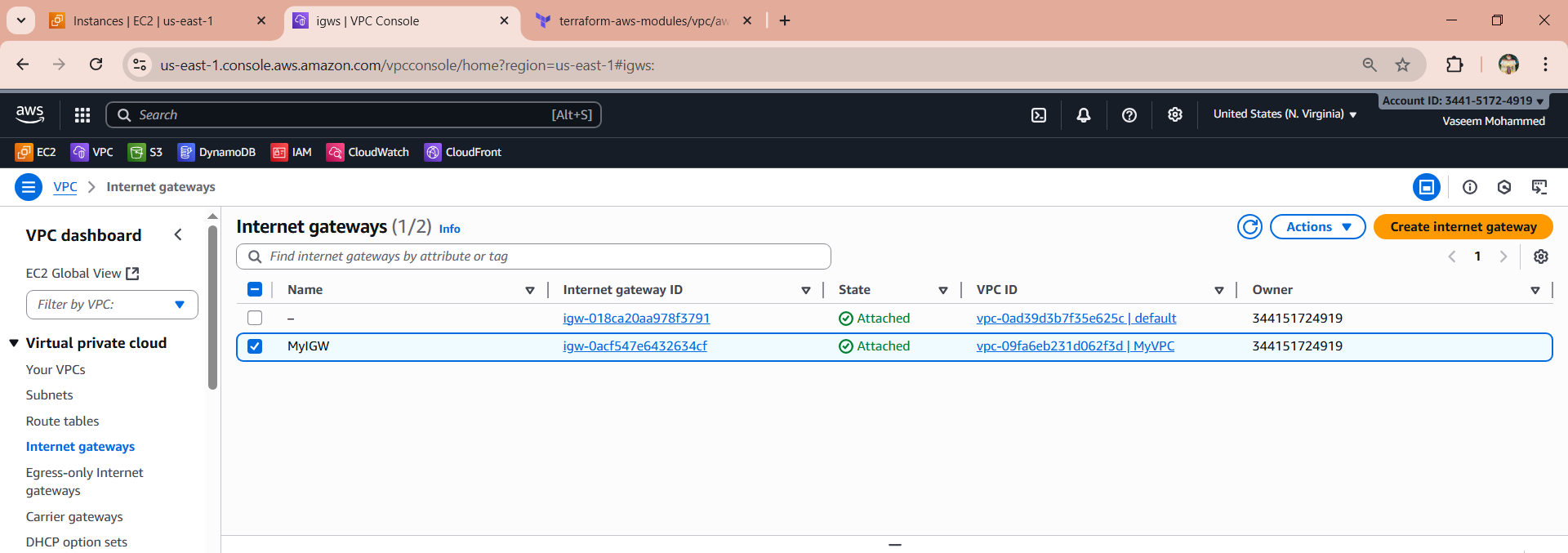
**Name = "MyIGW"**

**}**

**}**

**terraform init --> terraform apply**





1. **Create Custom Route Table**

Now we have created the RT

**resource "aws\_route\_table" "public" {**

**vpc\_id = aws\_vpc.main.id**

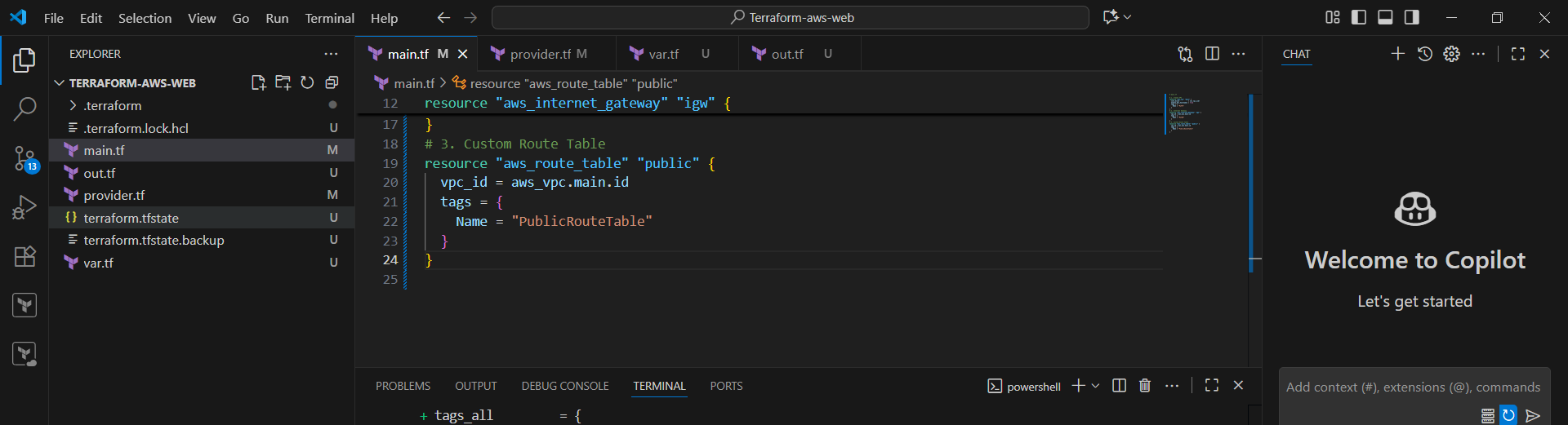
**tags = {**

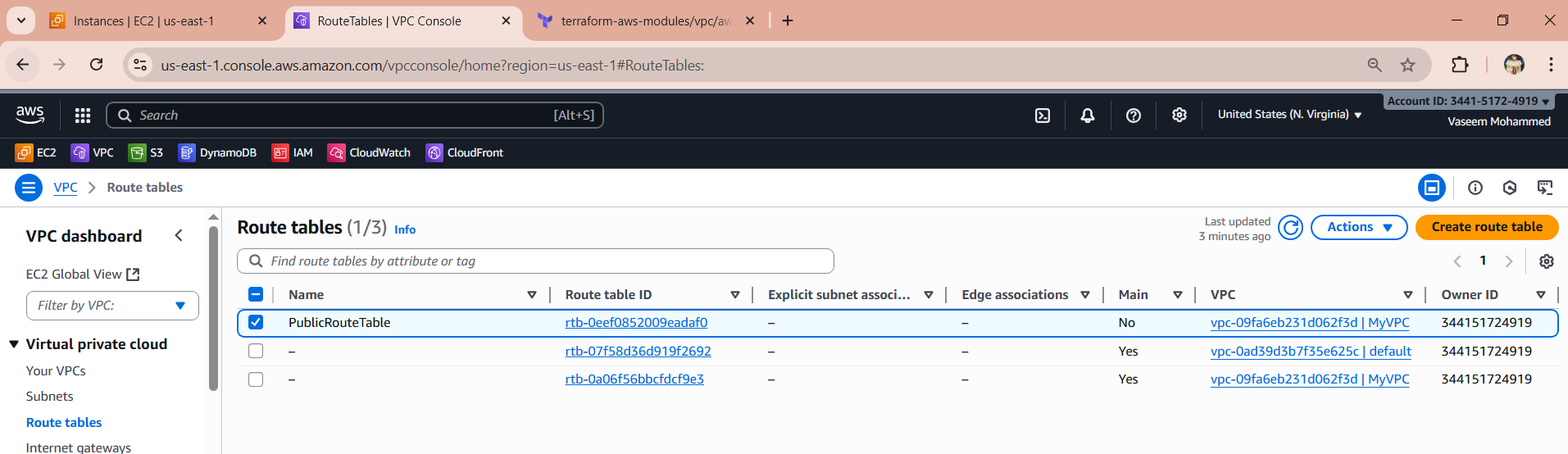
**Name = "PublicRouteTable"**

**}**

**}**

**terraform init --> terraform apply**





1. **Create Subnet**

Here we have created the subnet

**resource "aws\_subnet" "public" {**

**vpc\_id = aws\_vpc.main.id**

**cidr\_block = var.subnet\_cidr**

**availability\_zone = var.availability\_zone**

**map\_public\_ip\_on\_launch = false**

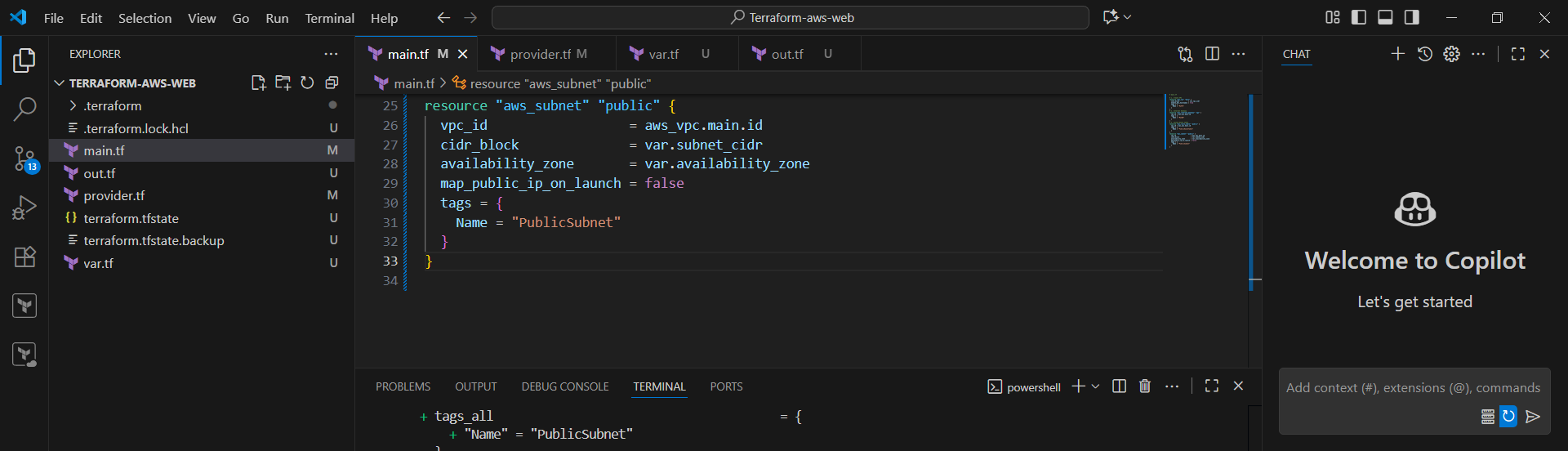
**tags = {**

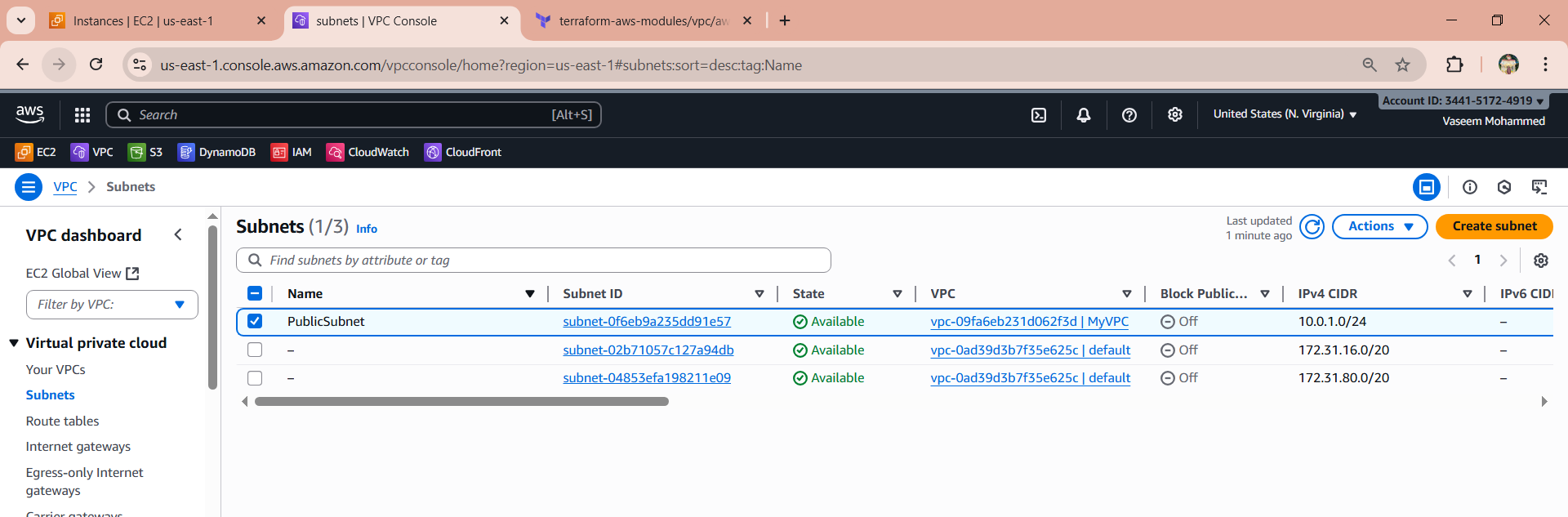
**Name = "PublicSubnet"**

**}**

**}**

**terraform init --> terraform apply**





1. **Associate subnet with Route Table**

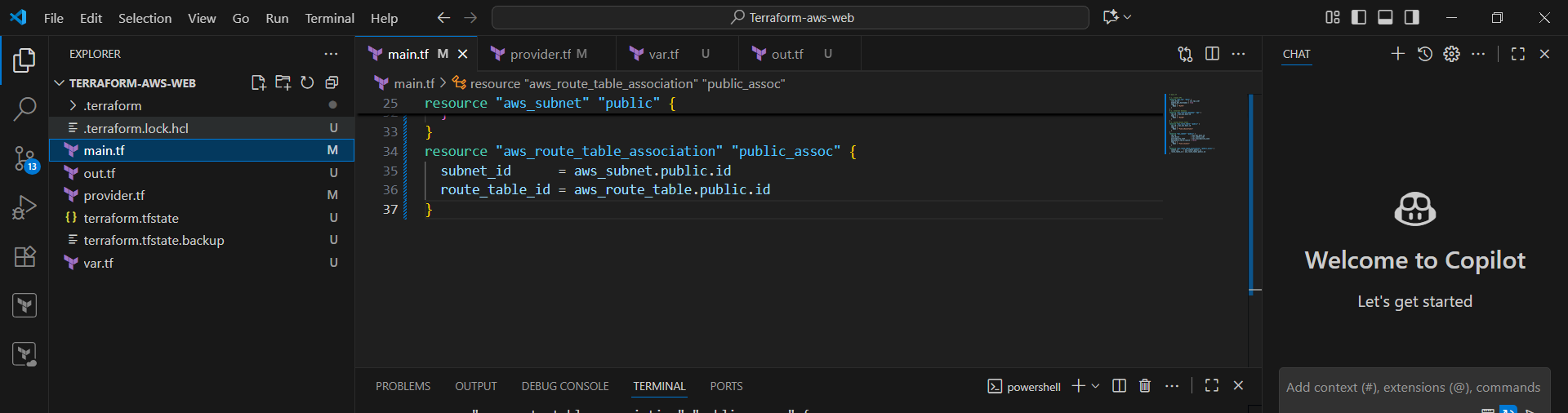
Now we have attached the subnet to route table by terraform

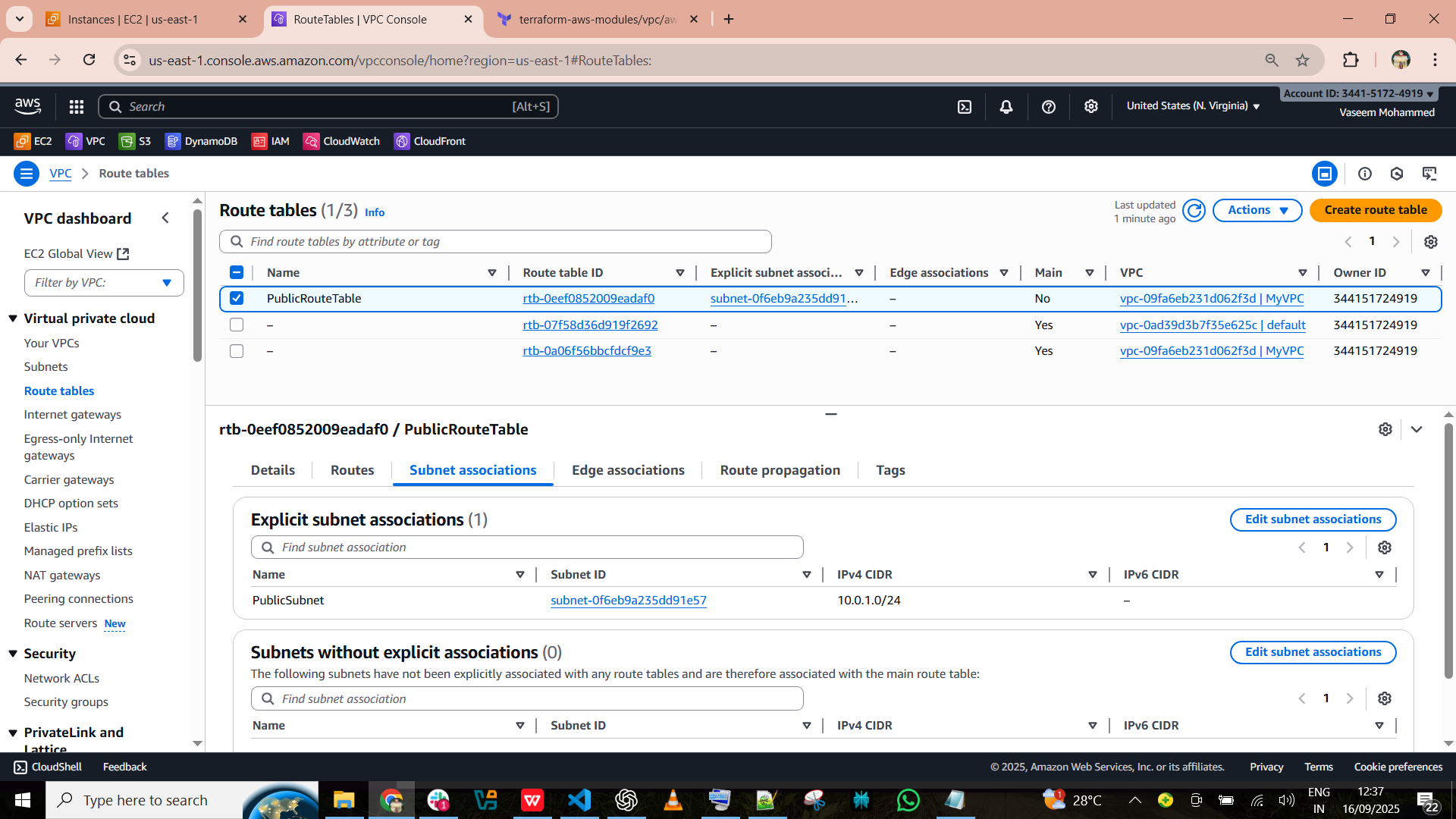
**resource "aws\_route\_table\_association" "public\_assoc" {**

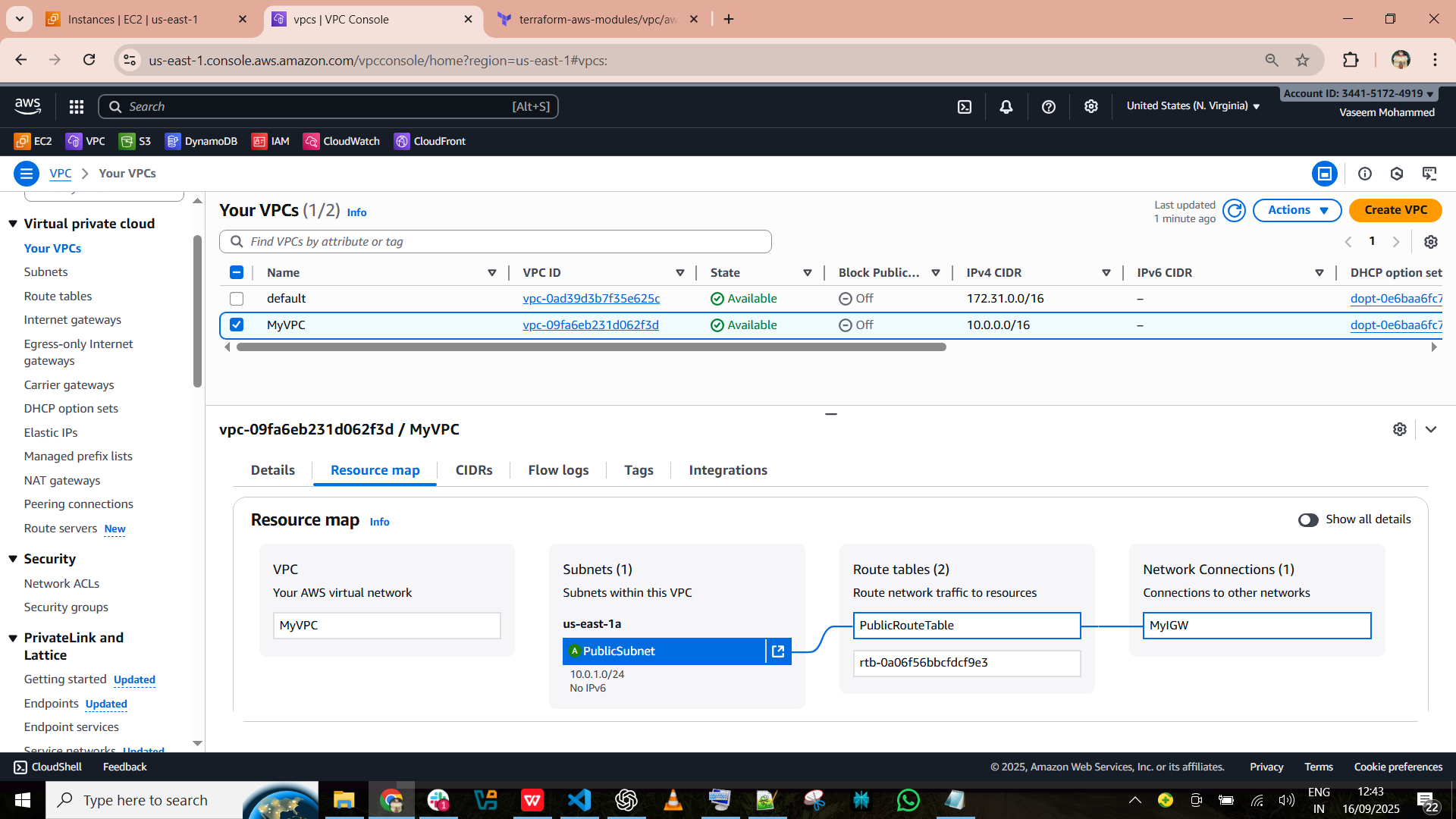
**subnet\_id = aws\_subnet.public.id**

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

**}**







1. **Create Security Group to allow port 22,80,443**

Here we have created the SG

**resource "aws\_security\_group" "web\_sg" {**

**vpc\_id = aws\_vpc.main.id**

**name = "web-sg"**

**ingress {**

**description = "Allow SSH"**

**from\_port = 22**

**to\_port = 22**

**protocol = "tcp"**

**cidr\_blocks = [var.allowed\_ssh\_cidr]**

**}**

**ingress {**

**description = "Allow HTTP"**

**from\_port = 80**

**to\_port = 80**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**ingress {**

**description = "Allow HTTPS"**

**from\_port = 443**

**to\_port = 443**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**egress {**

**description = "Allow all outbound"**

**from\_port = 0**

**to\_port = 0**

**protocol = "-1"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

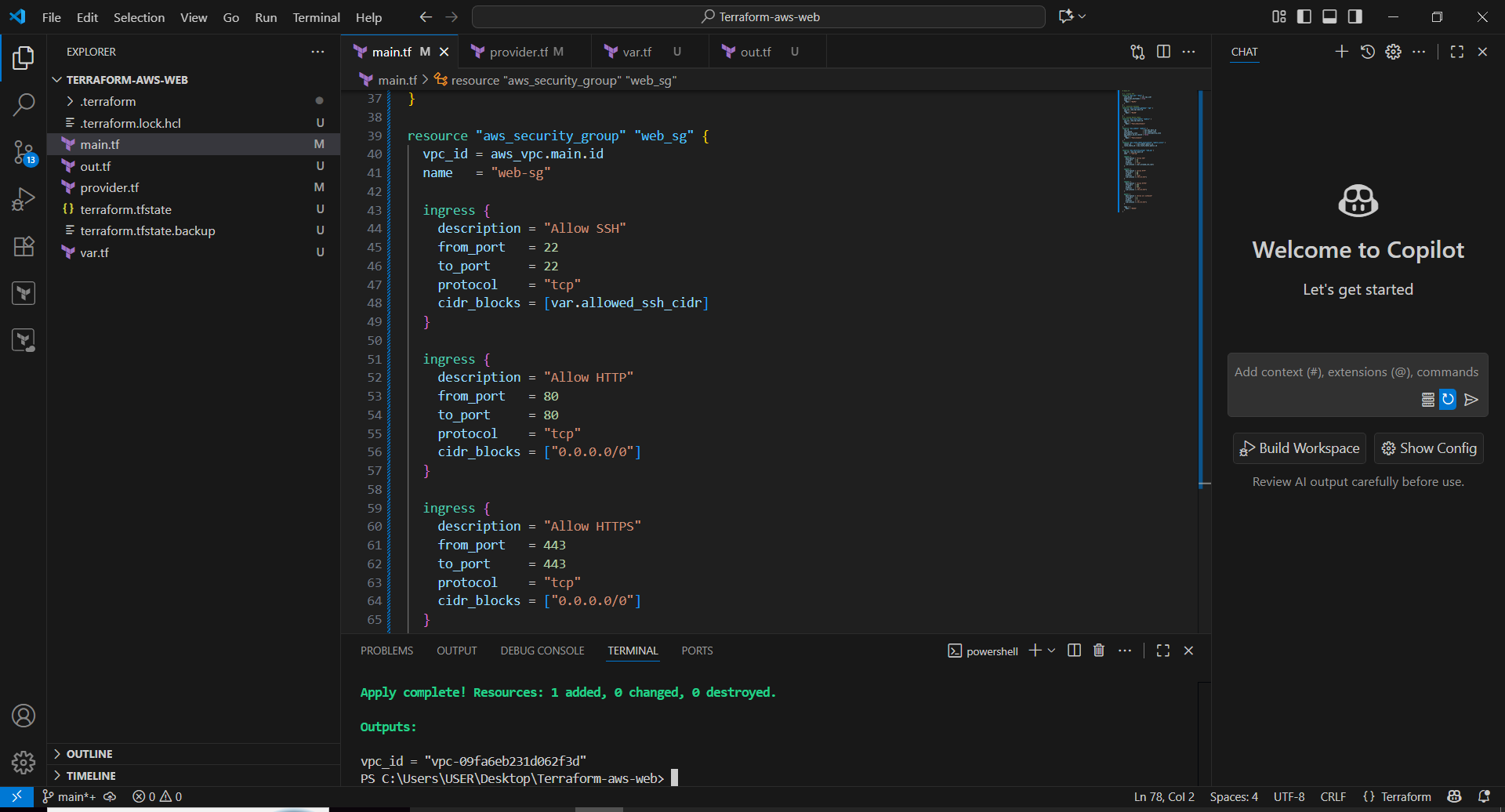
**tags = {**

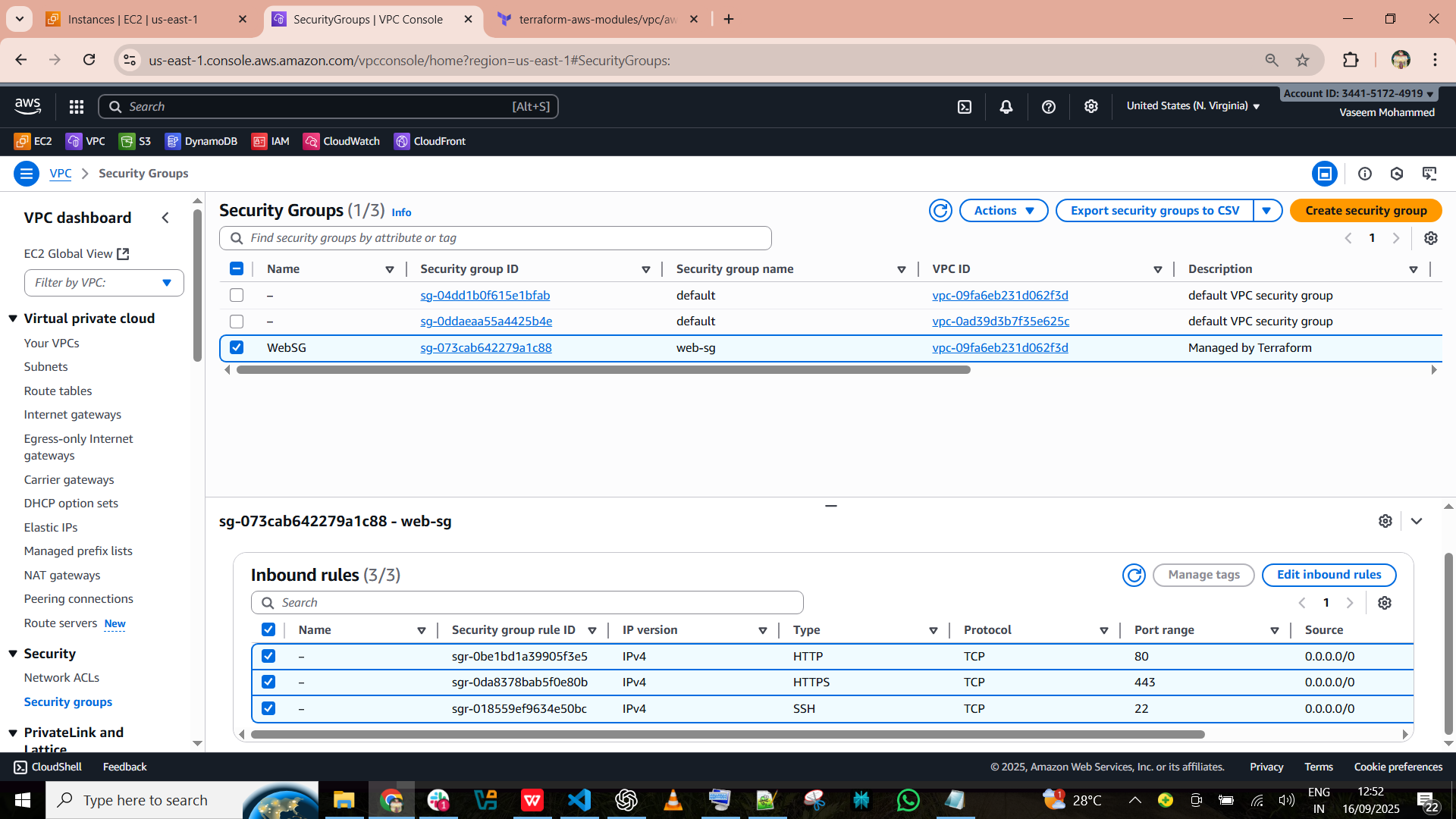
**Name = "WebSG"**

**}**

**}**

**terraform init --> terraform apply**





1. **Create a network interface with an ip in the subnet that was created in step 4**

Now we have created the NI by terraform

**resource "aws\_network\_interface" "web\_eni" {**

**subnet\_id = aws\_subnet.public.id**

**private\_ips = [var.instance\_private\_ip]**

**security\_groups = [aws\_security\_group.web\_sg.id]**

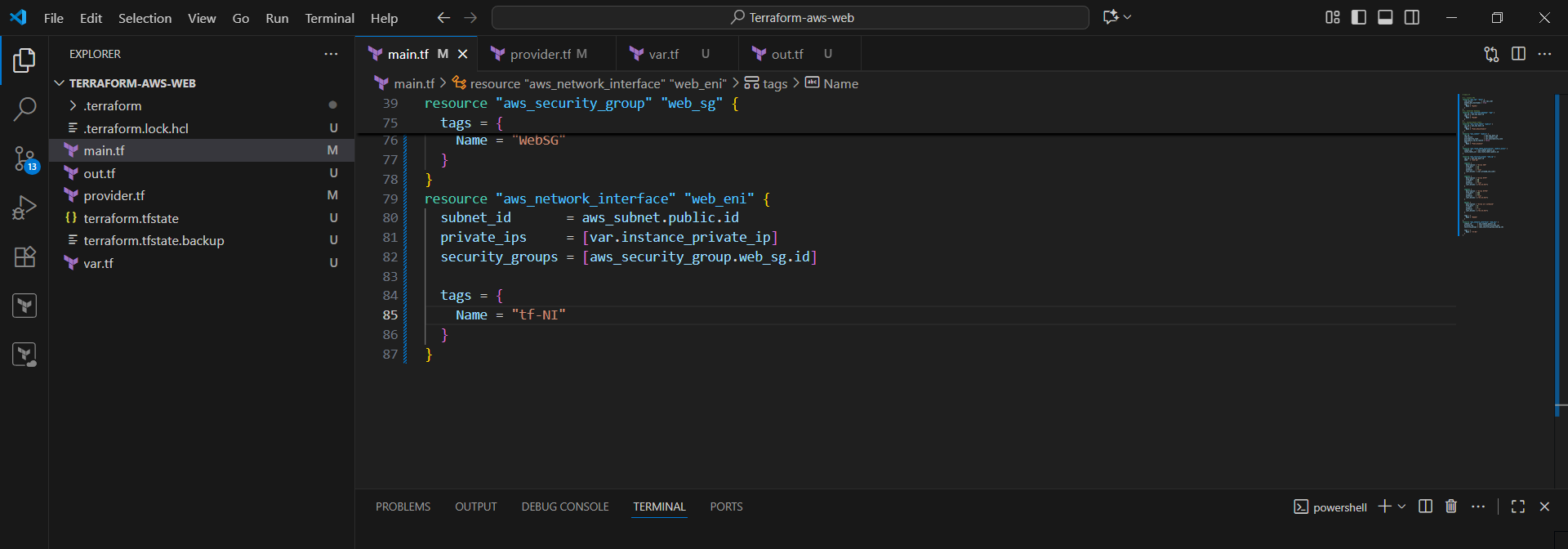
**tags = {**

**Name = "tf-NI"**

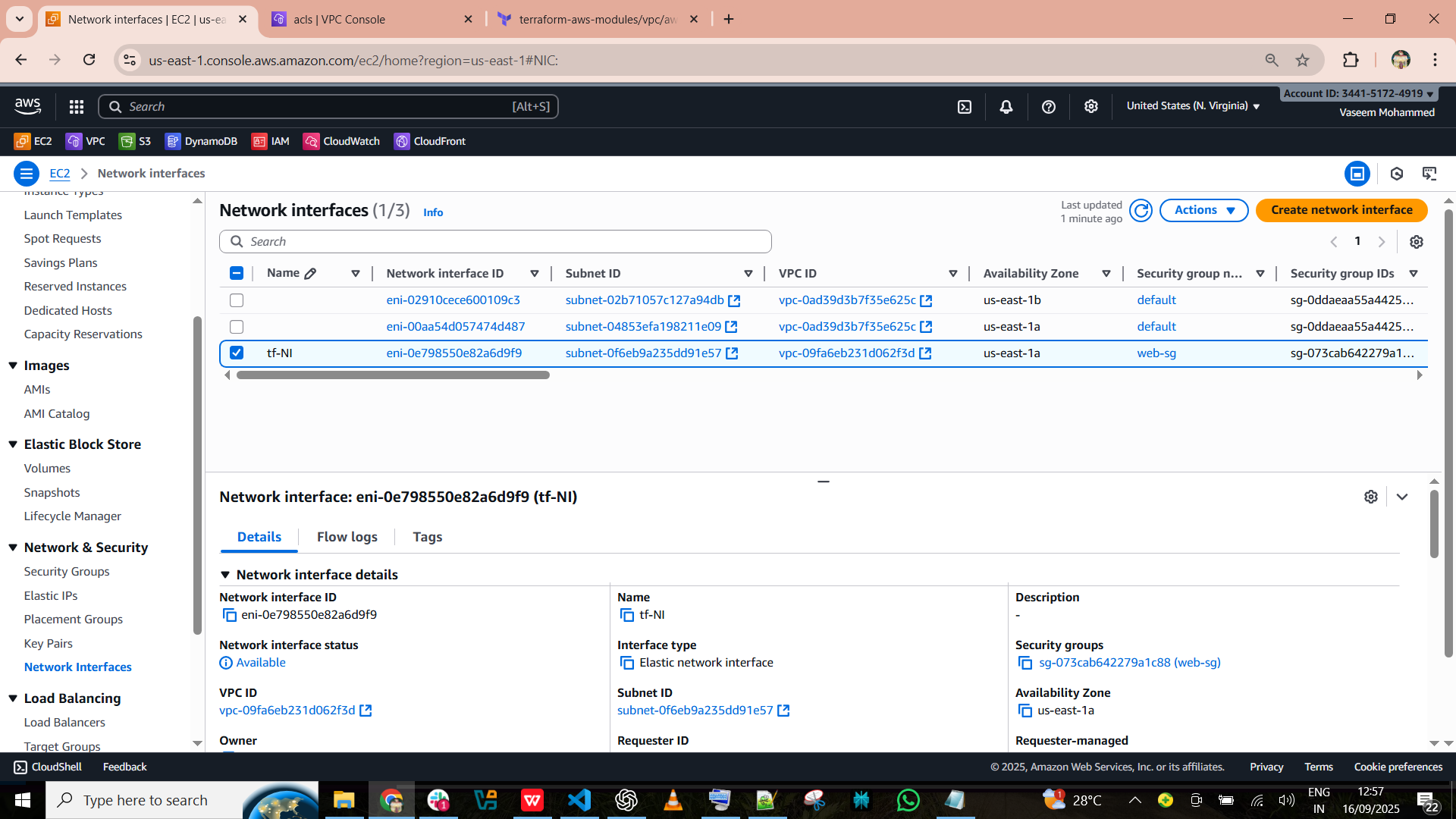
**}**

**}**

**terraform init --> terraform apply**



Go to EC2 -->scroll down --> Network interface



1. **Assign an elastic IP to the network interface created in step 7**

Here we have created the Elastic ip for step 7

**resource "aws\_eip" "web\_eip" {**

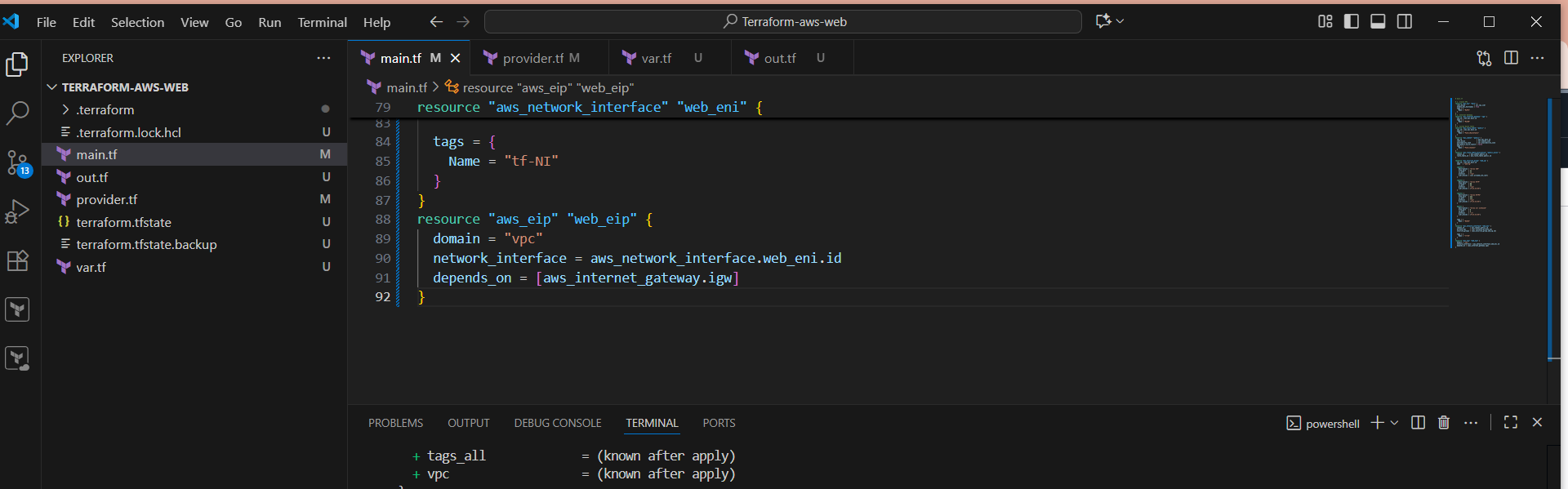
**domain = "vpc"**

**network\_interface = aws\_network\_interface.web\_eni.id**

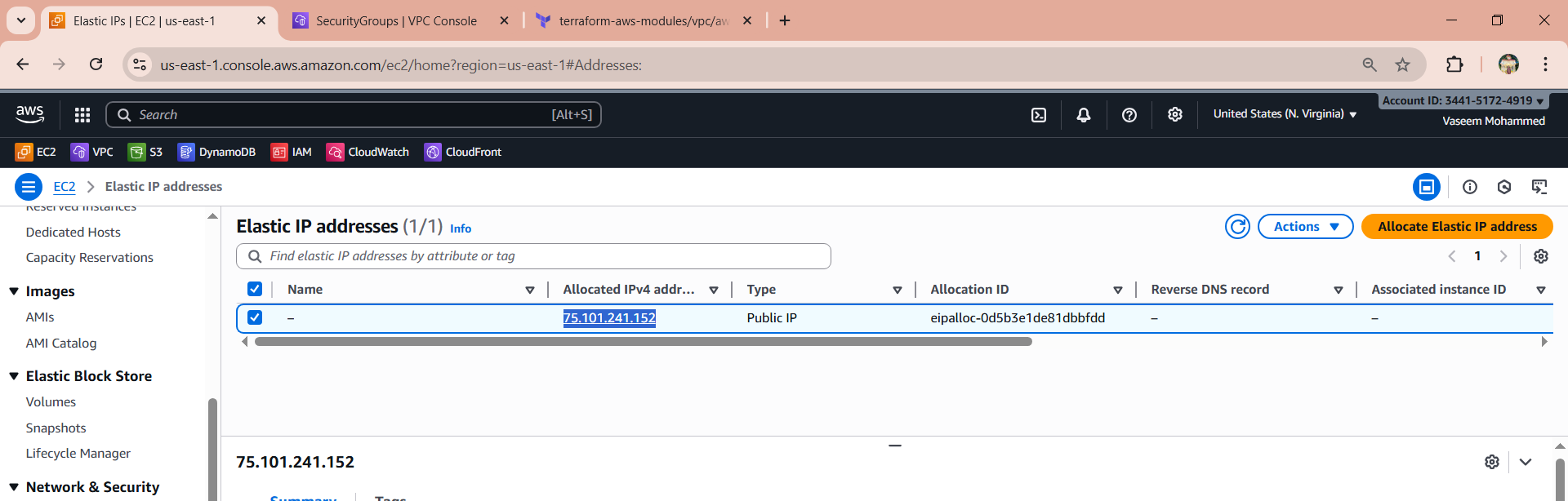
**depends\_on = [aws\_internet\_gateway.igw]**

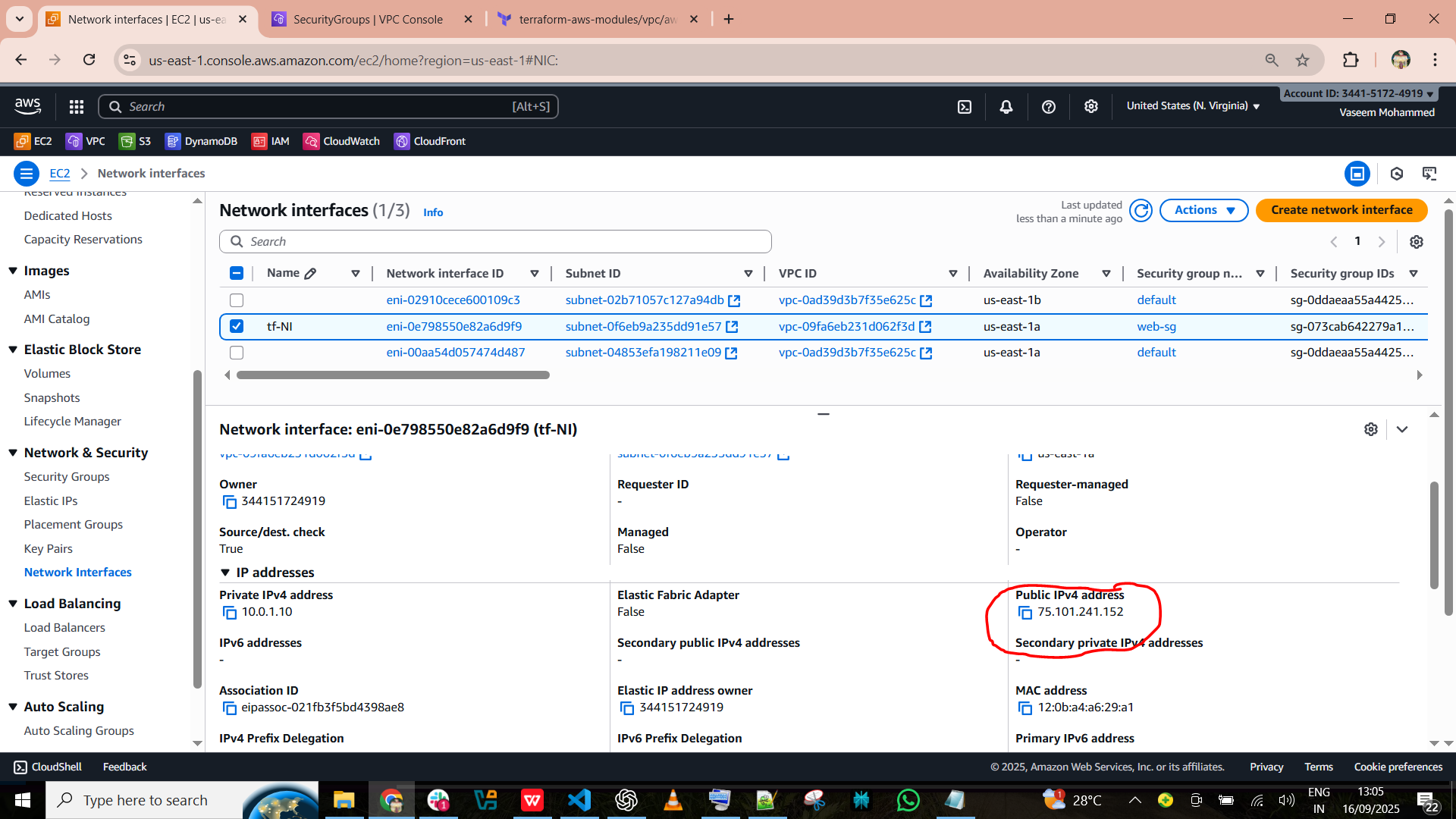
**}**

**terraform init --> terraform apply**



Go to EC2 -->scroll down --> elastic IP’s





1. **Create Ubuntu server and install/enable apache2**

Here we have launch the instance by AMI ubuntu by terraform

**data "aws\_ami" "ubuntu" {**

**most\_recent = true**

**owners = ["099720109477"] # Canonical**

**filter {**

**name = "name"**

**values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-\*"]**

**}**

**}**

**resource "aws\_instance" "web" {**

**ami = data.aws\_ami.ubuntu.id**

**instance\_type = "t2.micro"**

**key\_name = var.key\_name**

**network\_interface {**

**device\_index = 0**

**network\_interface\_id = aws\_network\_interface.web\_eni.id**

**}**

**user\_data = <<-EOF**

**#!/bin/bash**

**apt-get update -y**

**apt-get install -y apache2**

**systemctl enable apache2**

**systemctl start apache2**

**EOF**

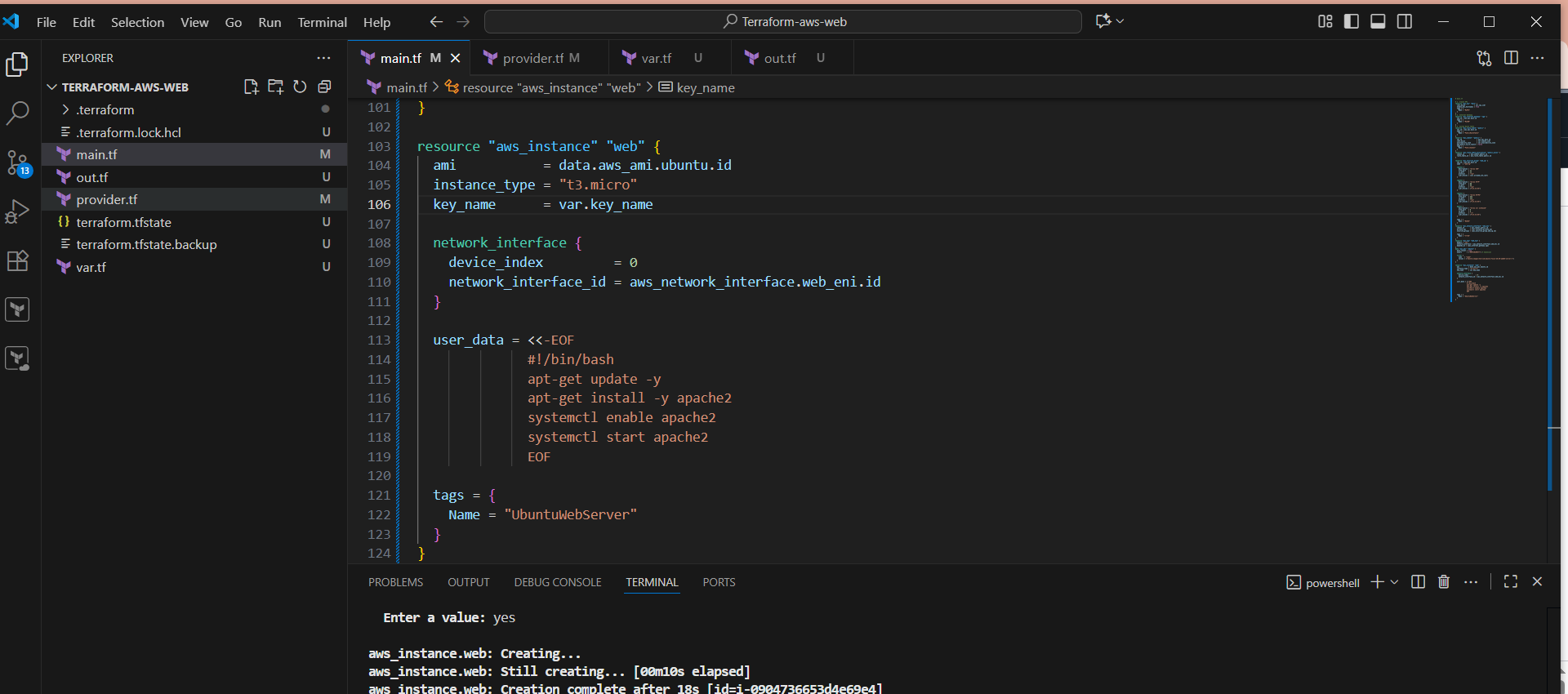
**tags = {**

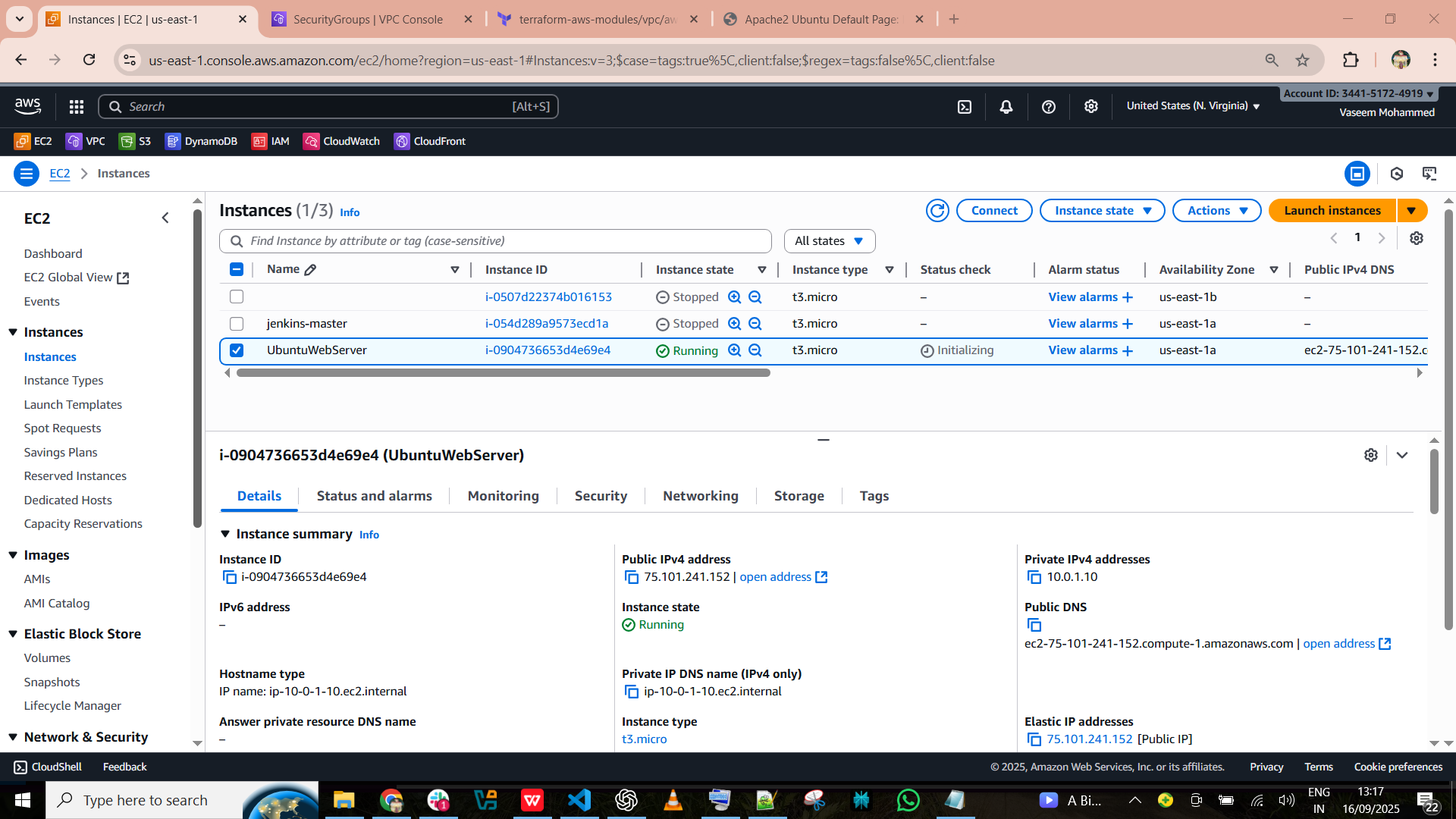
**Name = "UbuntuWebServer"**

**}**

**}**

**terraform init --> terraform apply**





Now copy the public IP and check the outcome with that ip ddress on browser



