**Terraform Final Task**

1. **Create VPC**
2. **Create Internet gateway**
3. **Create Custom Route Table**
4. **Create Subnet**
5. **Associate subnet with Route Table**
6. **Create Security Group to allow port 22.80,443**
7. **Create a network interface with an ip in the subnet that was created in step 4**
8. **Assign an elastic IP to the network interface created in step 7**
9. **Create Ubuntu server and install/enable apache2**

**This configuration builds a basic networking setup in AWS:**

* **VPC**
* **Subnet**
* **Internet Gateway**
* **Route Table**
* **Route Table Association**

**Step 1: Create VPC**

**In terraform .main**

**resource "aws\_vpc" "my\_vpc" {**

**cidr\_block = "10.0.0.0/16"**

**enable\_dns\_support = true**

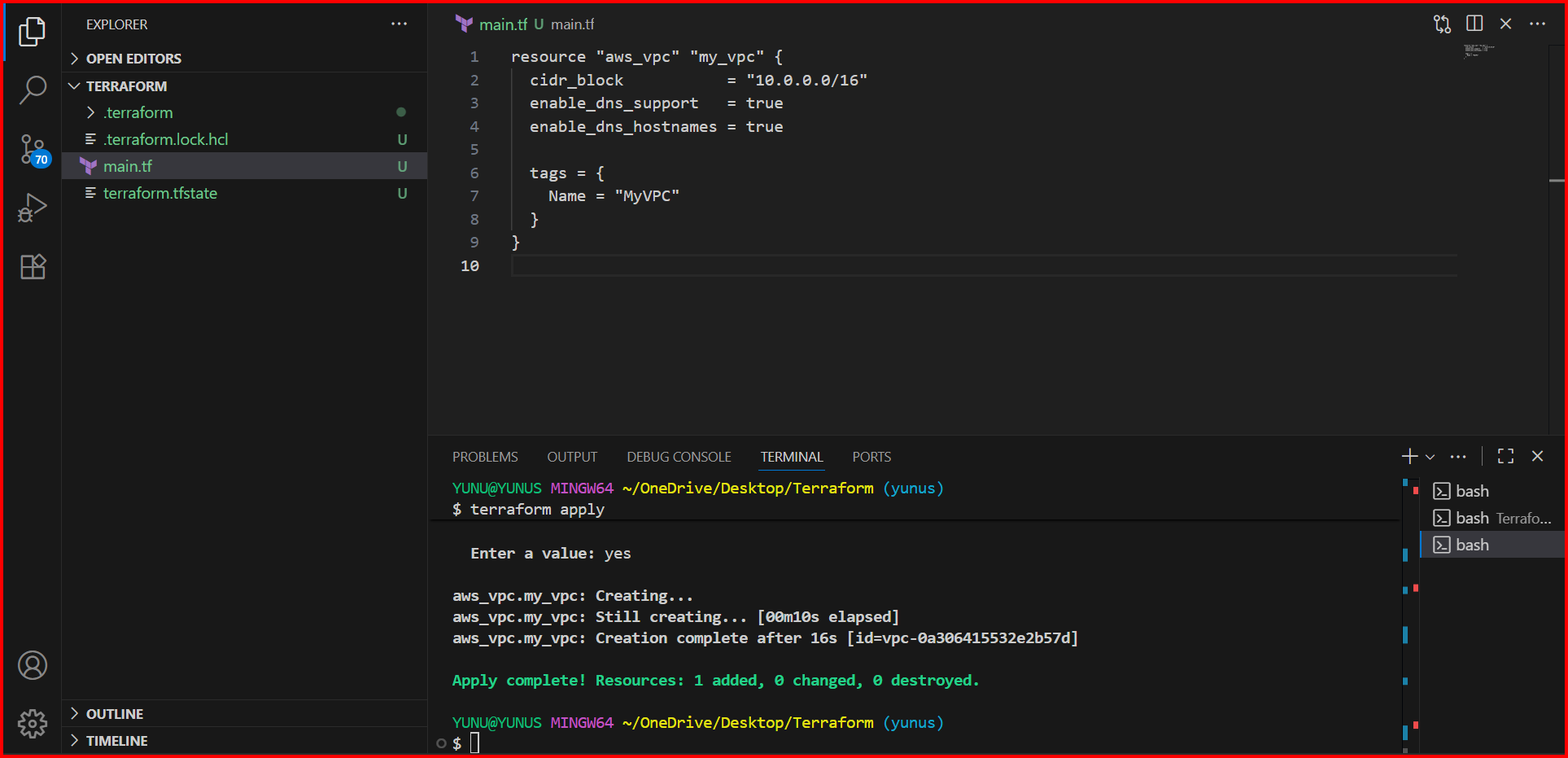
**enable\_dns\_hostnames = true**

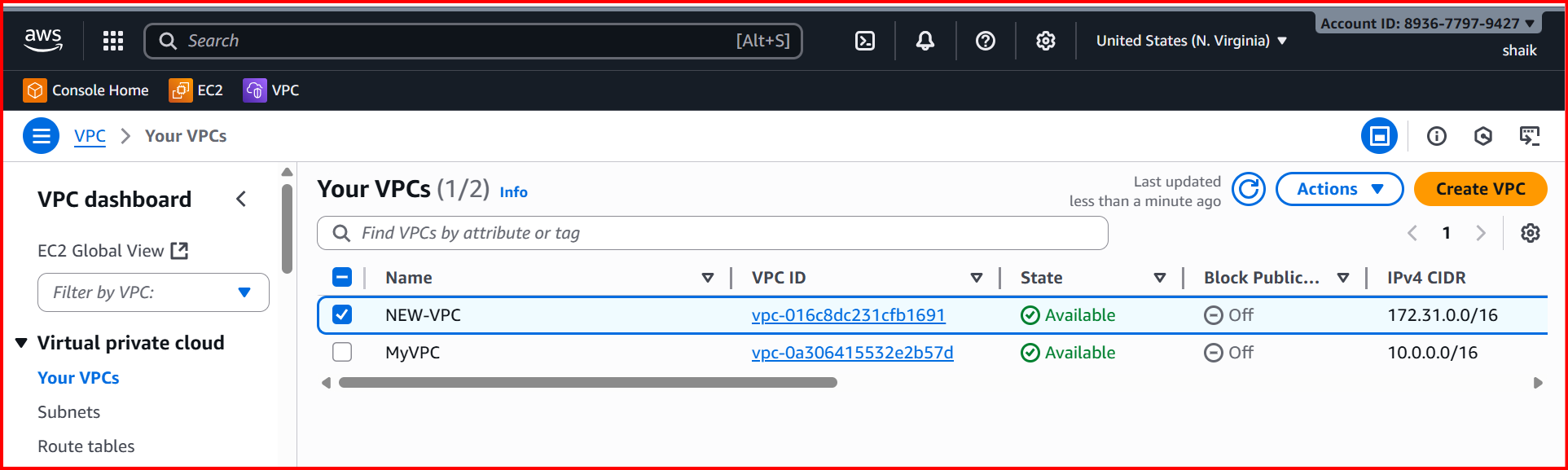
**tags = {**

**Name = "MyVPC"**

**}**

**}**

****

****

**Step 2: Create Subnet**

**Step 1 & Step 2**

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

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

**cidr\_block = "10.0.1.0/24"**

**availability\_zone = "us-east-1a"**

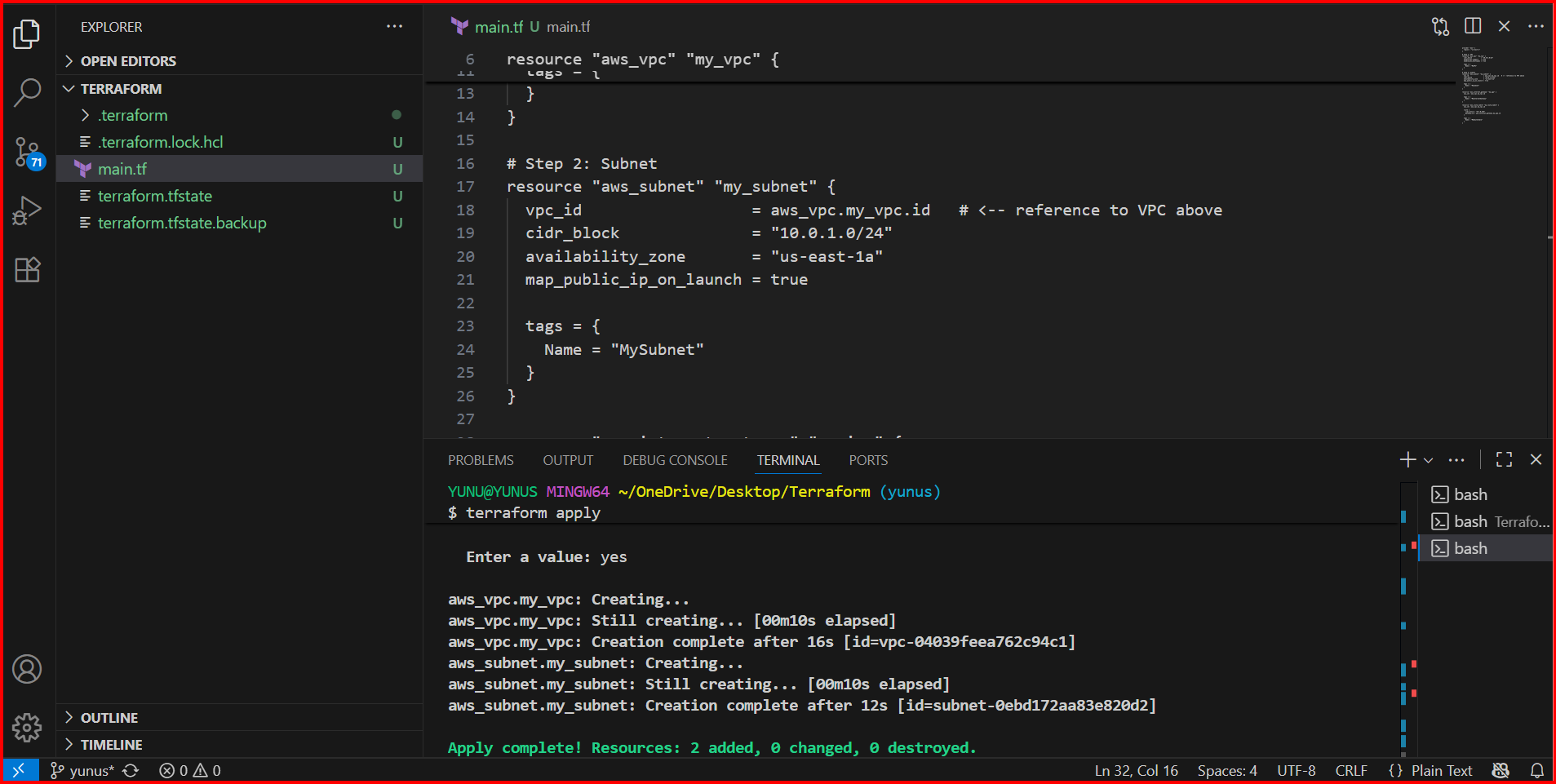
**map\_public\_ip\_on\_launch = true**

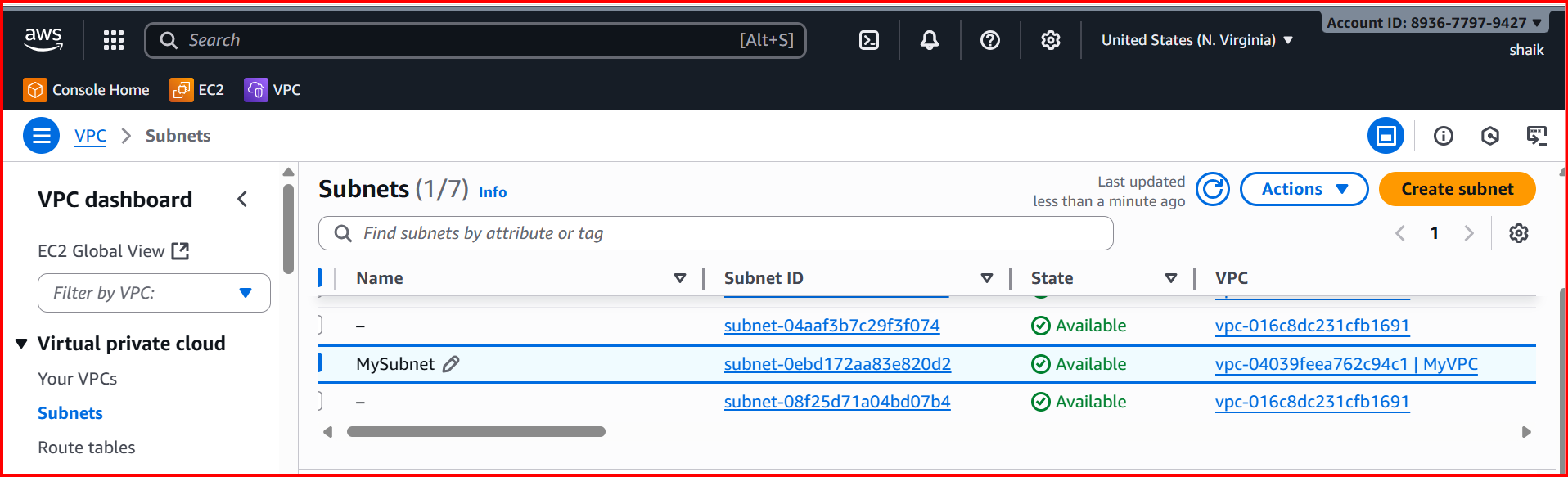
**tags = {**

**Name = "MySubnet"**

**}**

**}**

****

****

**Step 3: Create Internet Gateway**

**Step1 / Step2 /Step3**

**resource "aws\_internet\_gateway" "my\_igw" {**

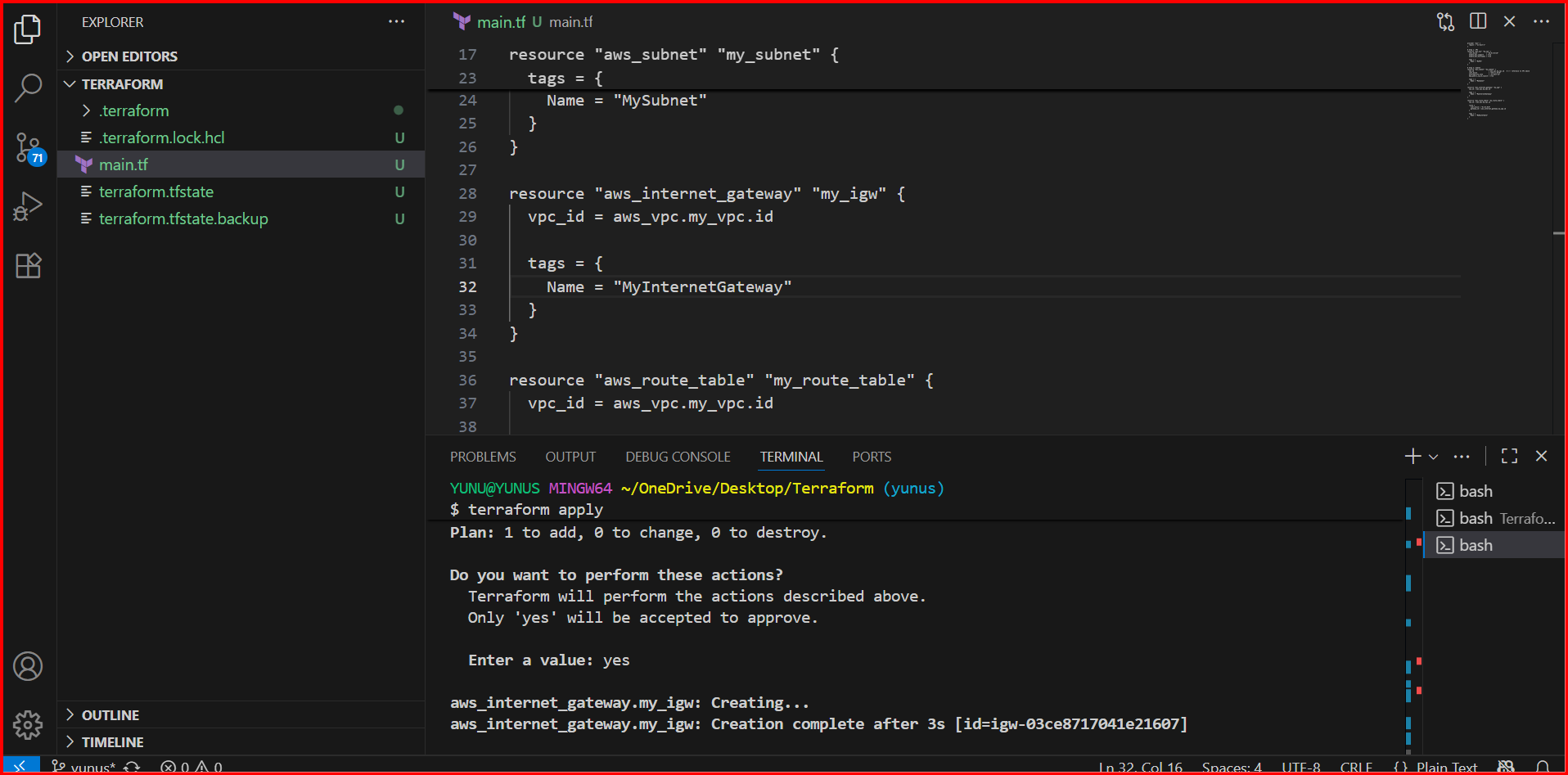
**vpc\_id = aws\_vpc.my\_vpc.id**

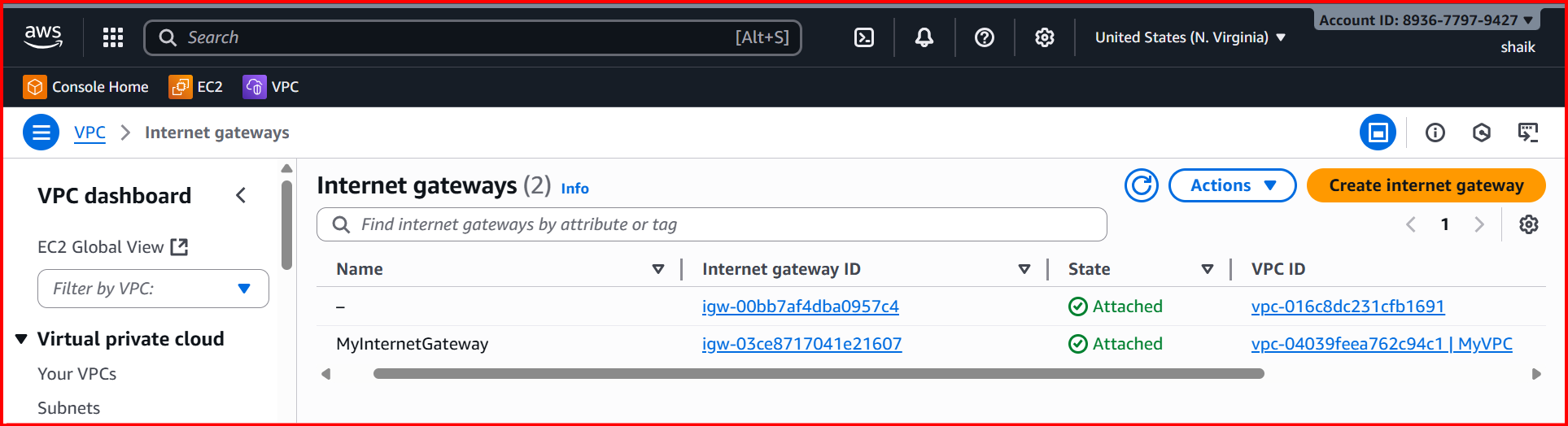
**tags = {**

**Name = "MyInternetGateway"**

**}**

**}**

****

****

**Step 4: Create Route Table**

**Step1 / Step2 /Step3/Step4**

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

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

**route {**

**cidr\_block = "0.0.0.0/0"**

**gateway\_id = aws\_internet\_gateway.my\_igw.id**

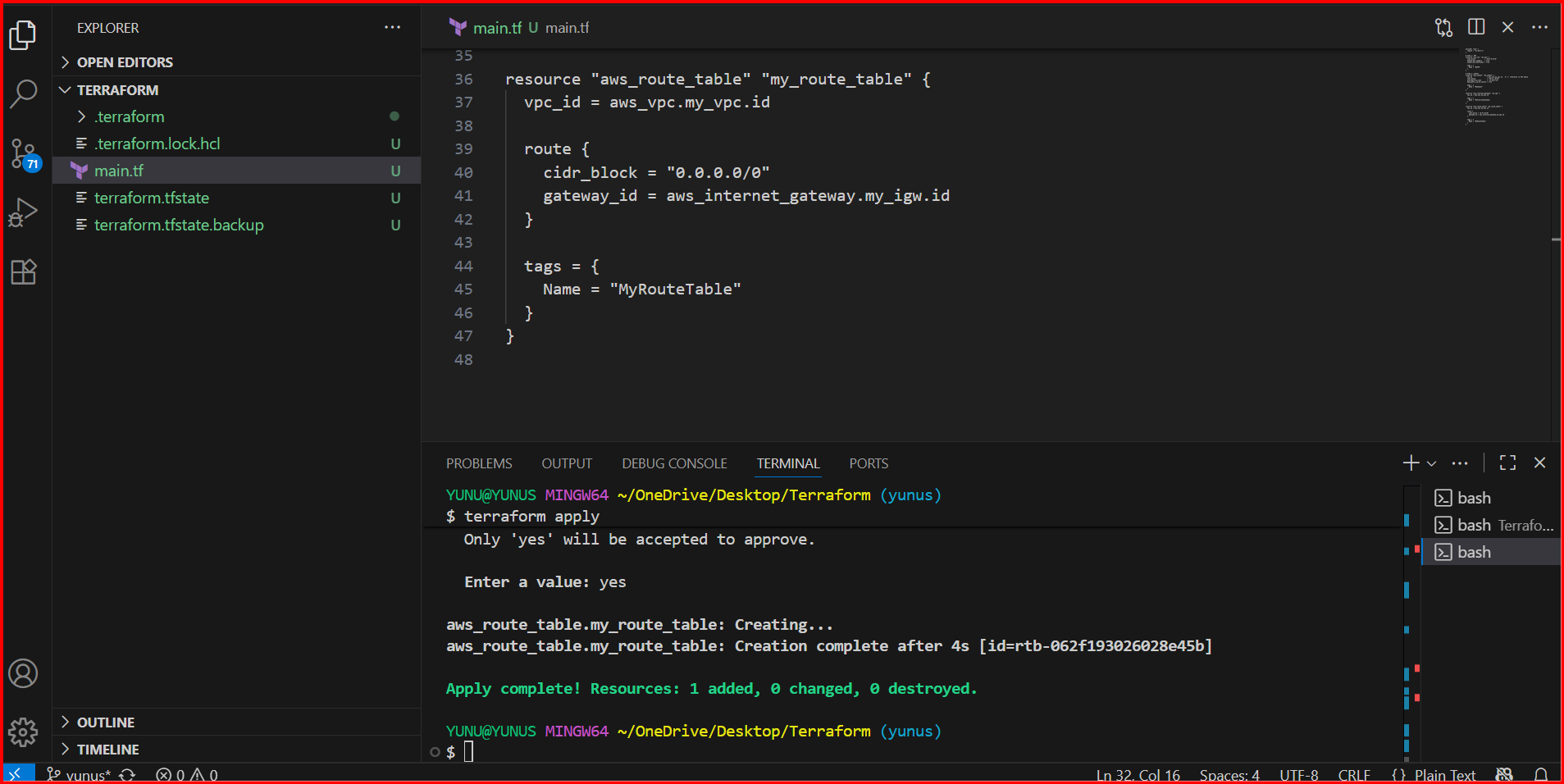
**}**

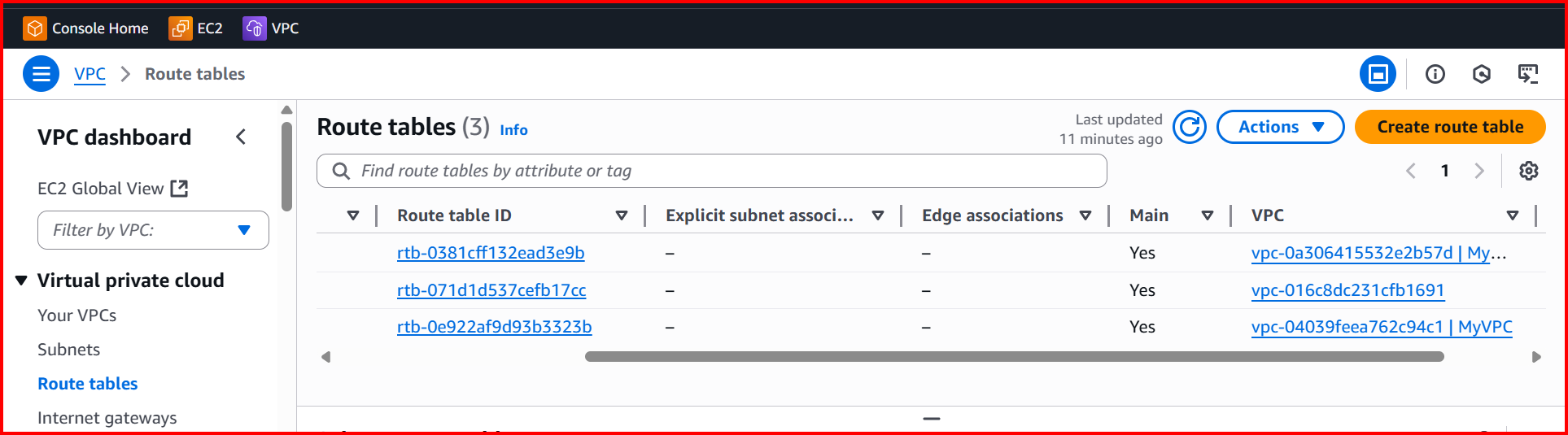
**tags = {**

**Name = "MyRouteTable"**

**}**

**}**

****

****

**Step 5: Associate Route Table with Subnet**

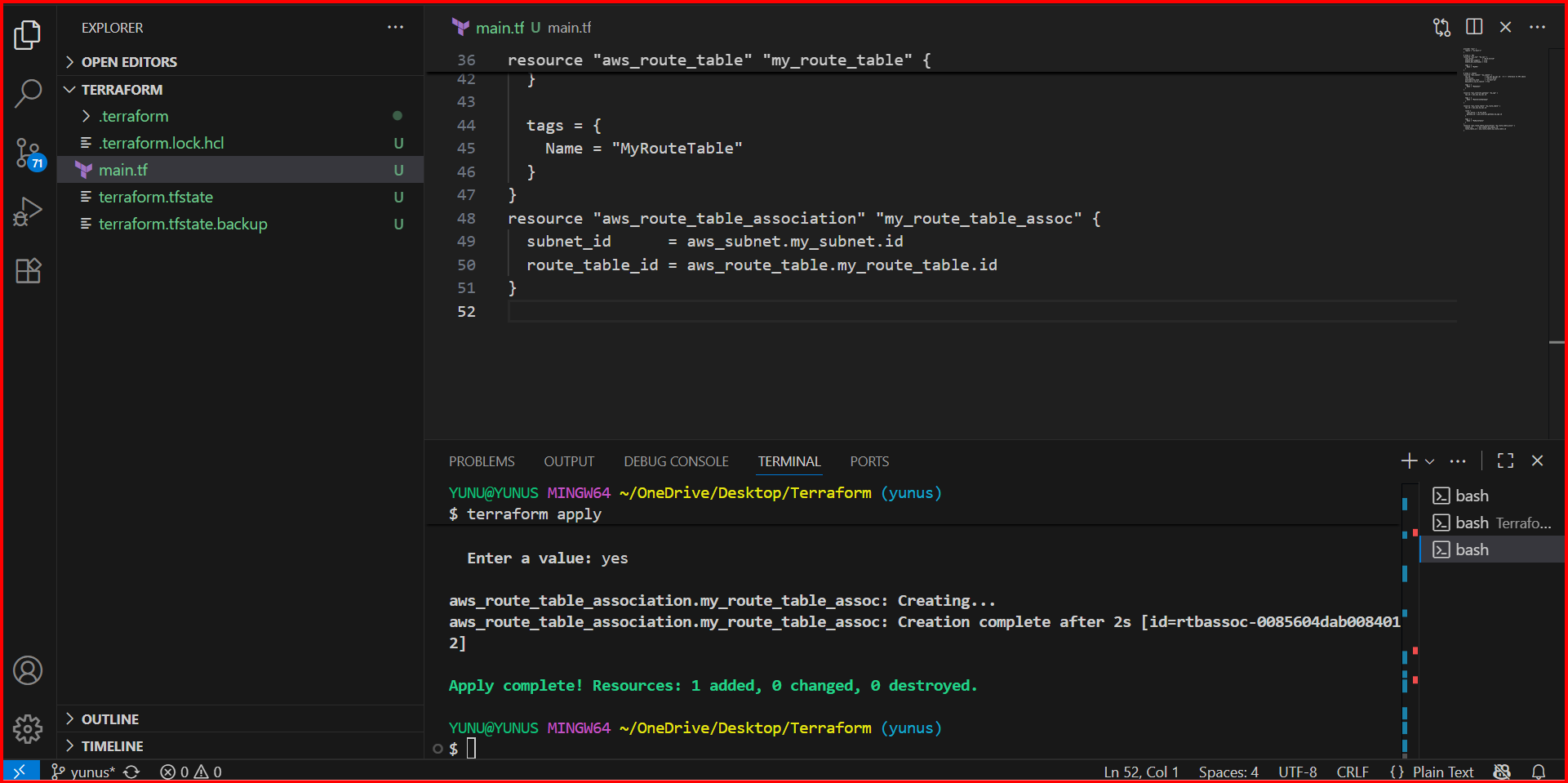
**Step1 / Step2 / Step3 /Step4 / Step5**

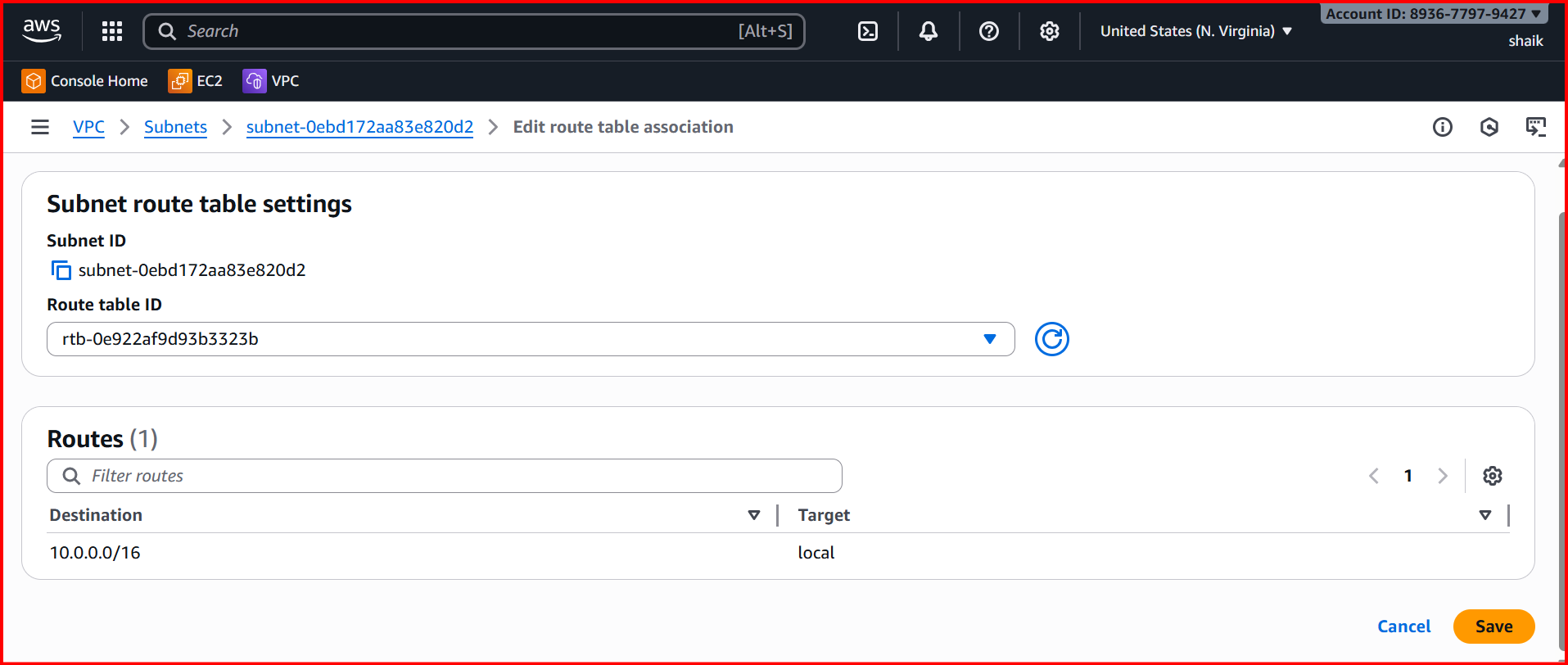
**resource "aws\_route\_table\_association" "my\_route\_table\_assoc" {**

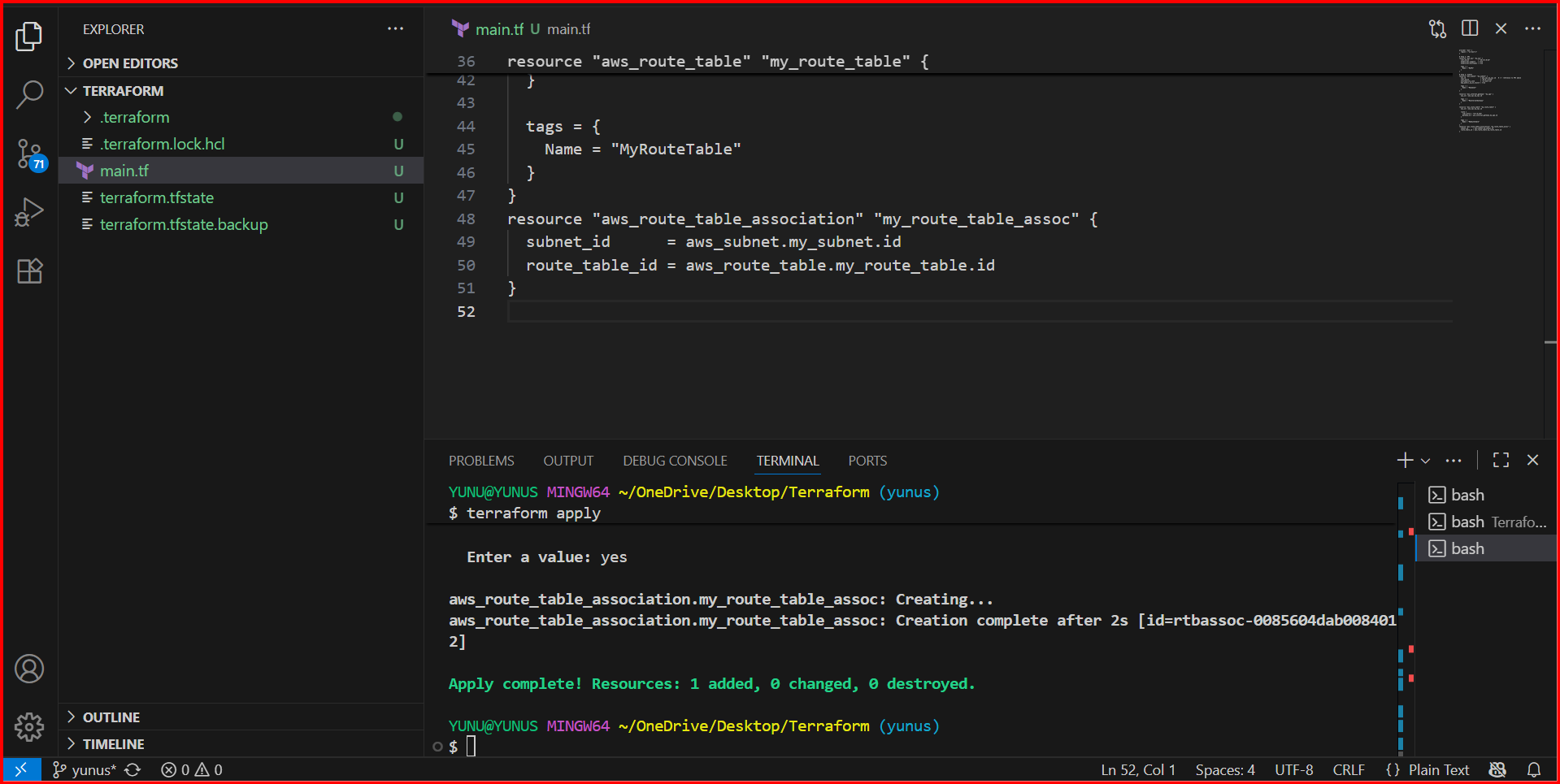
**subnet\_id = aws\_subnet.my\_subnet.id**

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

**}**

****

****

****

**Step 6** **Create Security Group to allow port 22.80,443**

**Create main.tf and terraform.tfvarf**

**You should have default VPC**

**provider "aws" {**

**region = "us-east-1"**

**}**

**variable "vpc\_id" {**

**description = "The ID of the existing VPC"**

**type = string**

**}**

**resource "aws\_security\_group" "shai-SG" {**

**name = "shai-SG"**

**description = "Allow SSH, HTTP, HTTPS"**

**vpc\_id = var.vpc\_id**

**ingress {**

**from\_port = 22**

**to\_port = 22**

**protocol = "tcp"**

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

**}**

**ingress {**

**from\_port = 80**

**to\_port = 80**

**protocol = "tcp"**

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

**}**

**ingress {**

**from\_port = 443**

**to\_port = 443**

**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"]**

**}**

**tags = {**

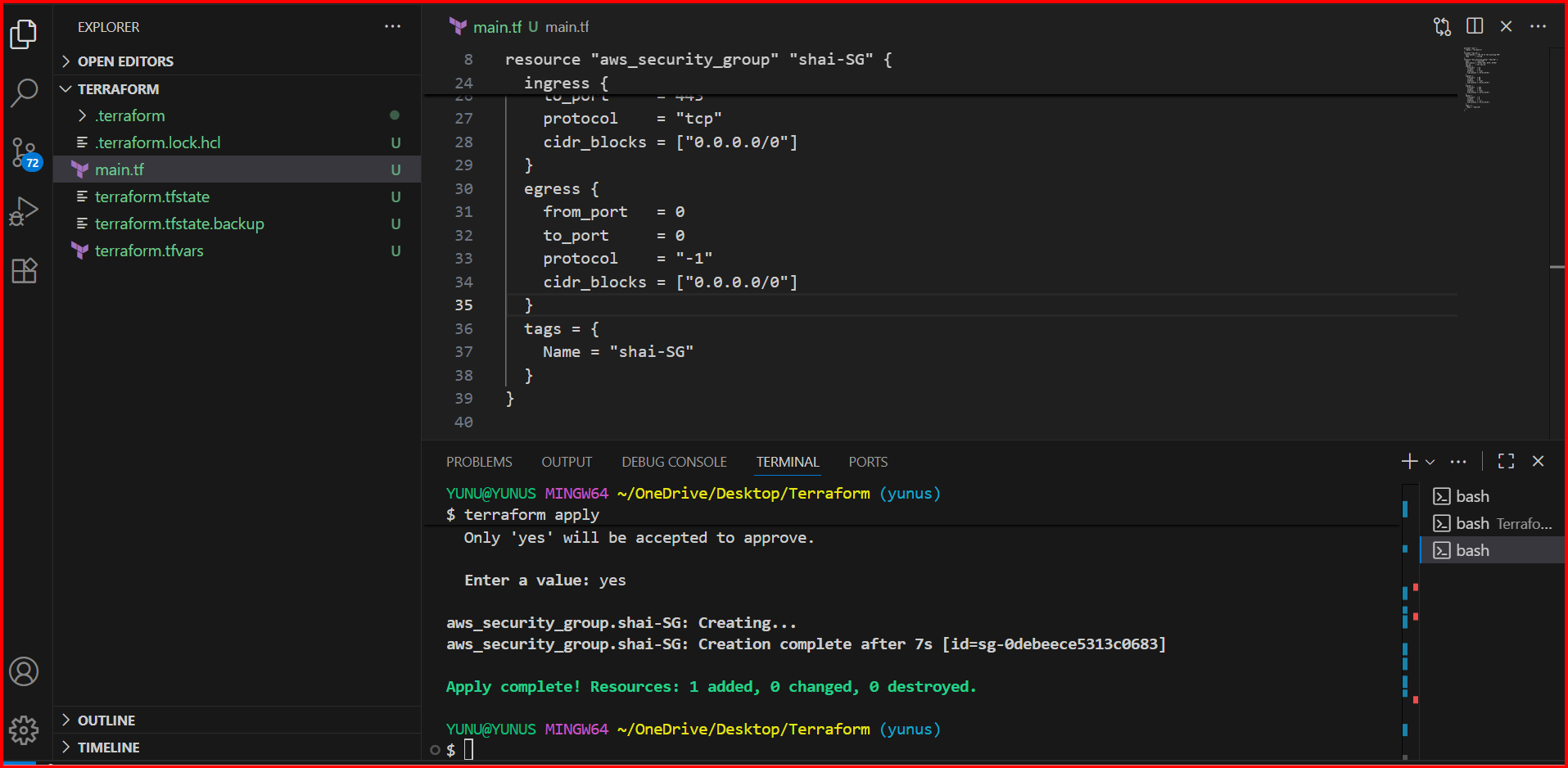
**Name = "shai-SG"**

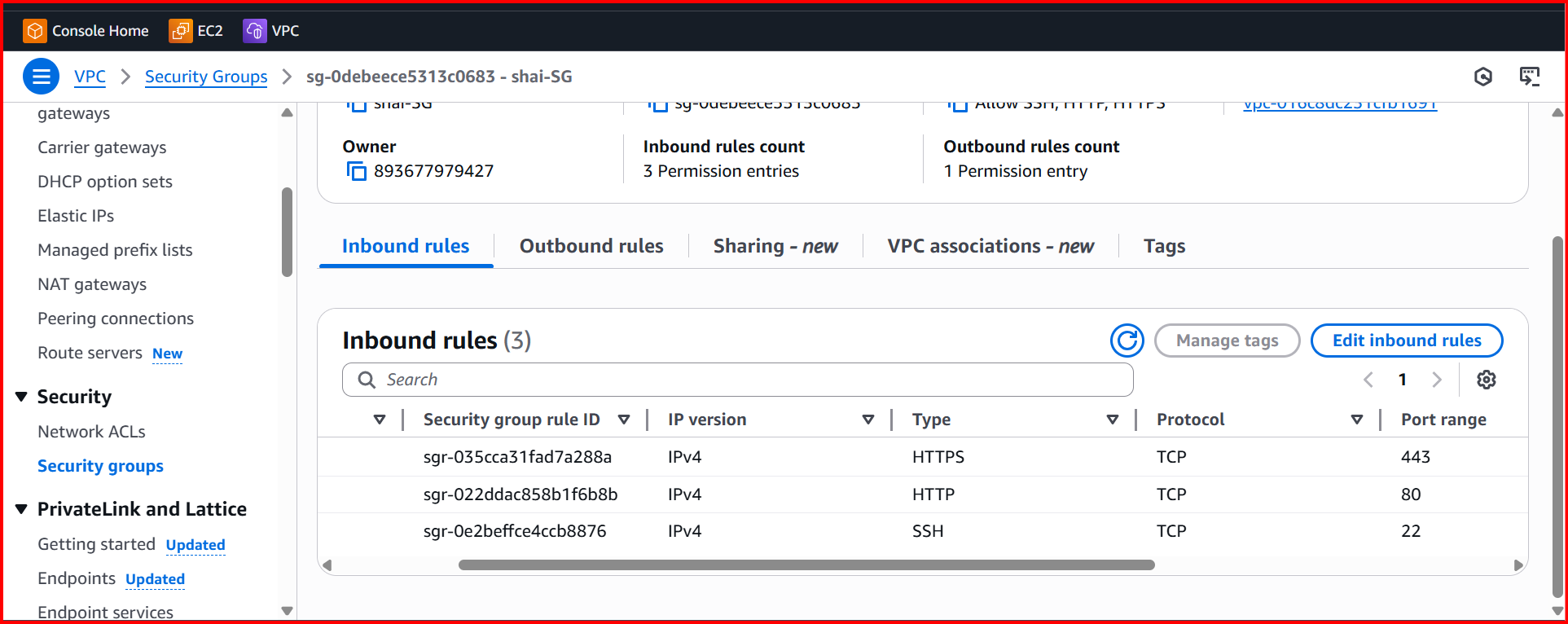
**}**

**}**

**terraform.tfvarf**

**vpc\_id = "vpc-016c8dc231cfb1691"**

****

****

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

**provider "aws" {**

**region = "us-east-1" # Change if needed**

**}**

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

**subnet\_id = "subnet-0e525850ac9ddc494"**

**description = "Example network interface created with Terraform"**

**tags = {**

**Name = "example-eni"**

**}**

**}**

**output "eni\_id" {**

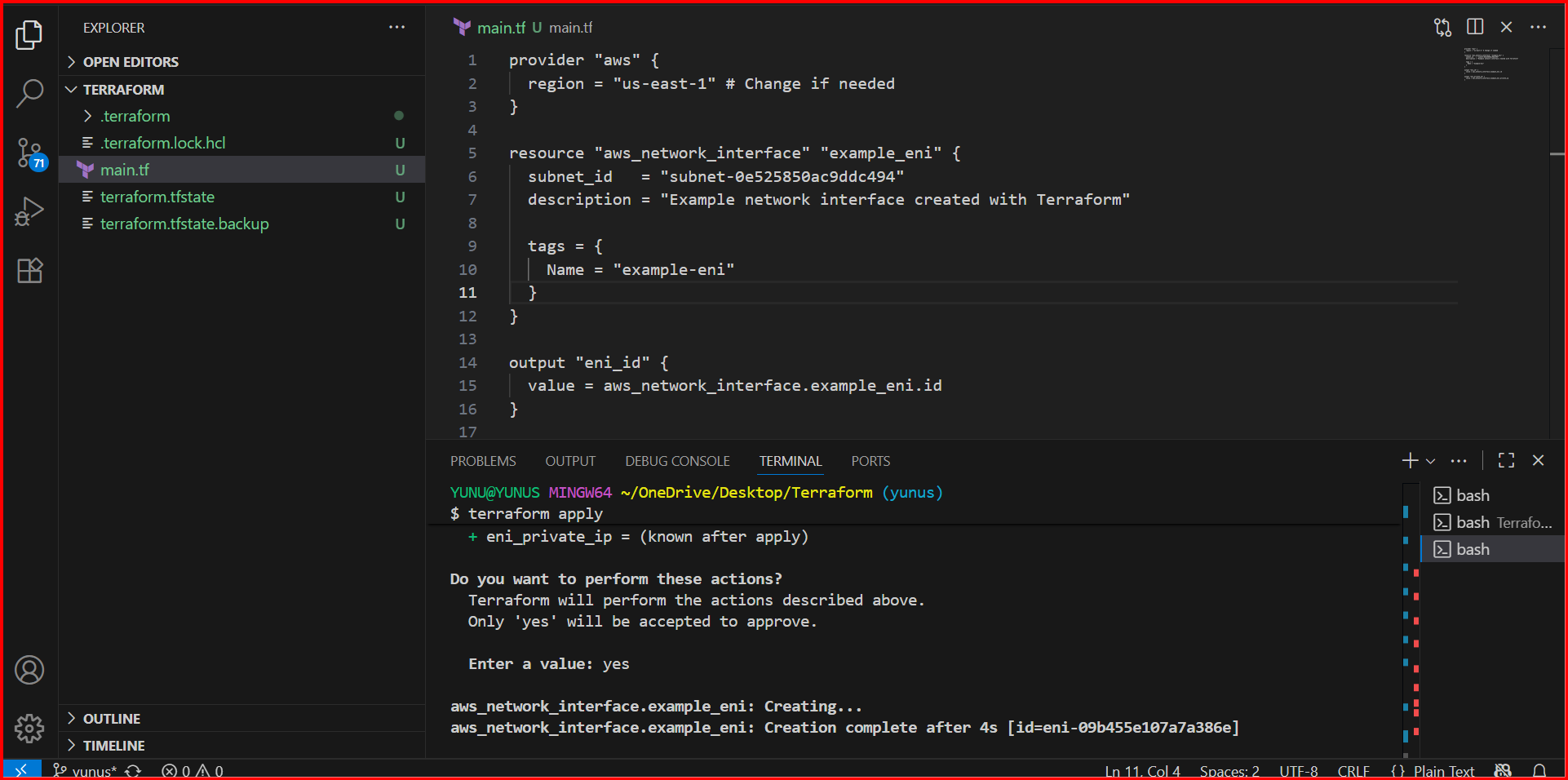
**value = aws\_network\_interface.example\_eni.id**

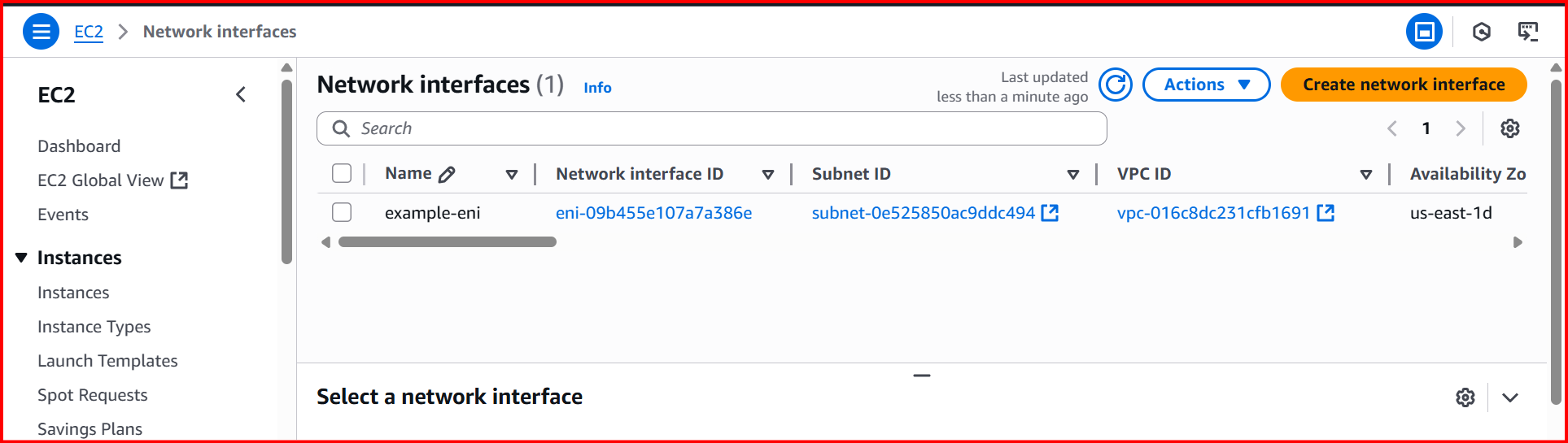
**}**

**output "eni\_private\_ip" {**

**value = aws\_network\_interface.example\_eni.private\_ip**

**}**

****

****

**Step8 Assign an elastic IP to the network interface created in step 7**

**provider "aws" {**

**region = "us-east-1" # Change if needed**

**}**

**# Create the network interface**

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

**subnet\_id = "subnet-0e525850ac9ddc494"**

**description = "Example network interface created with Terraform"**

**tags = {**

**Name = "example-eni"**

**}**

**}**

**# Allocate a new Elastic IP**

**resource "aws\_eip" "example\_eip" {**

**domain = "vpc" # Required for VPC-based EIPs**

**tags = {**

**Name = "example-eip"**

**}**

**}**

**# Associate the Elastic IP with the network interface**

**resource "aws\_eip\_association" "example\_assoc" {**

**allocation\_id = aws\_eip.example\_eip.id**

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

**}**

**# Outputs**

**output "eni\_id" {**

**value = aws\_network\_interface.example\_eni.id**

**}**

**output "eni\_private\_ip" {**

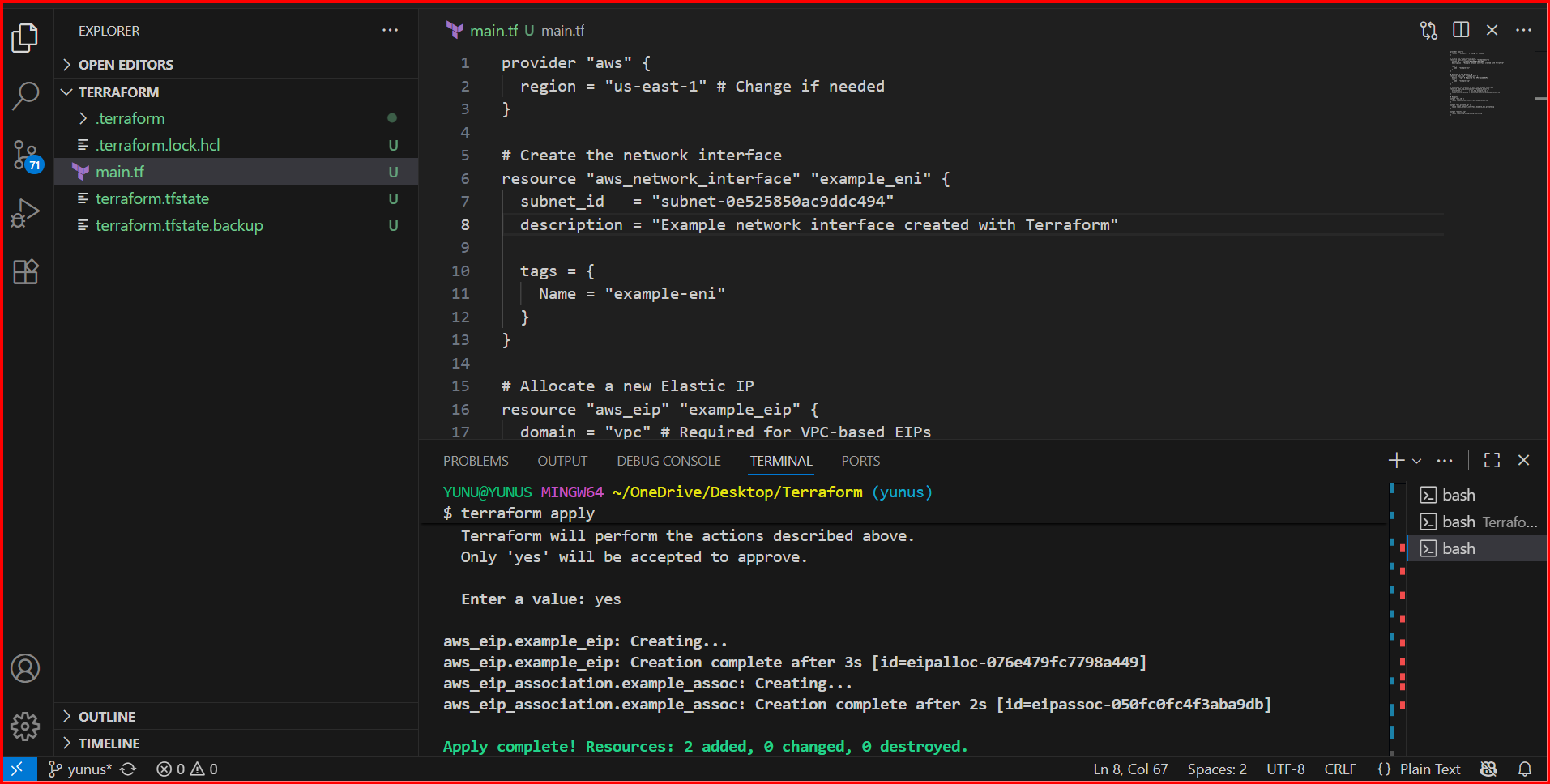
**value = aws\_network\_interface.example\_eni.private\_ip**

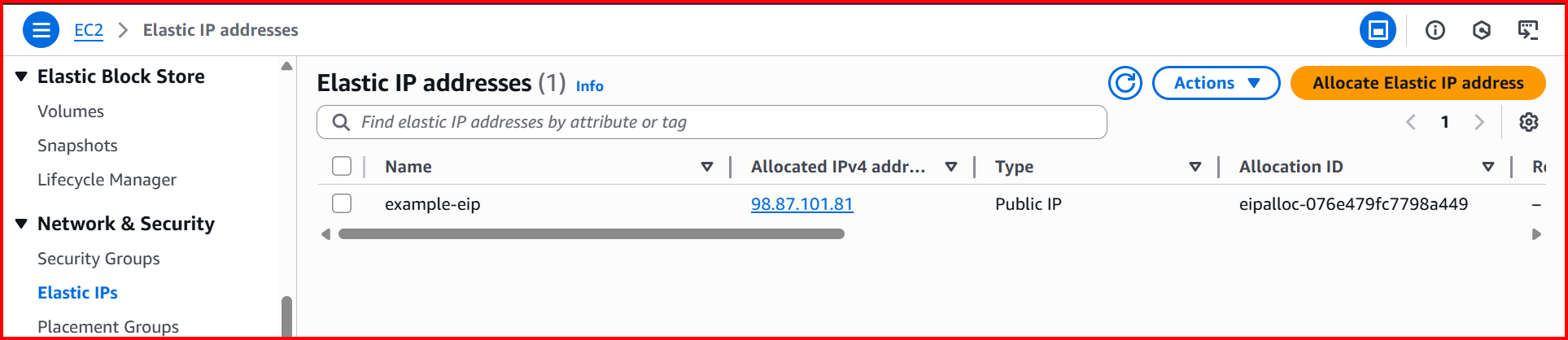
**}**

**output "elastic\_ip" {**

**value = aws\_eip.example\_eip.public\_ip**

**}**

****

****

**Step9 Create Ubuntu server and install/enable apache2**

**resource "aws\_instance" "ubuntu\_server" {**

**ami = "ami-0360c520857e3138f" # Ubuntu AMI in us-east-1 (verify if still valid)**

**instance\_type = "t3.micro"**

**subnet\_id = "subnet-04aaf3b7c29f3f074" # must be in quotes**

**vpc\_security\_group\_ids = ["sg-0b257af1cca95a062"] # must be in list with quotes**

**associate\_public\_ip\_address = true**

**key\_name = "" # add your existing key pair name here (for SSH login)**

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

**#!/bin/bash**

**apt update -y**

**apt install -y apache2**

**systemctl enable apache2**

**systemctl start apache2**

**echo "<h1>Hello from Terraform Apache2 Server Shaik Yunus</h1>" > /var/www/html/index.html**

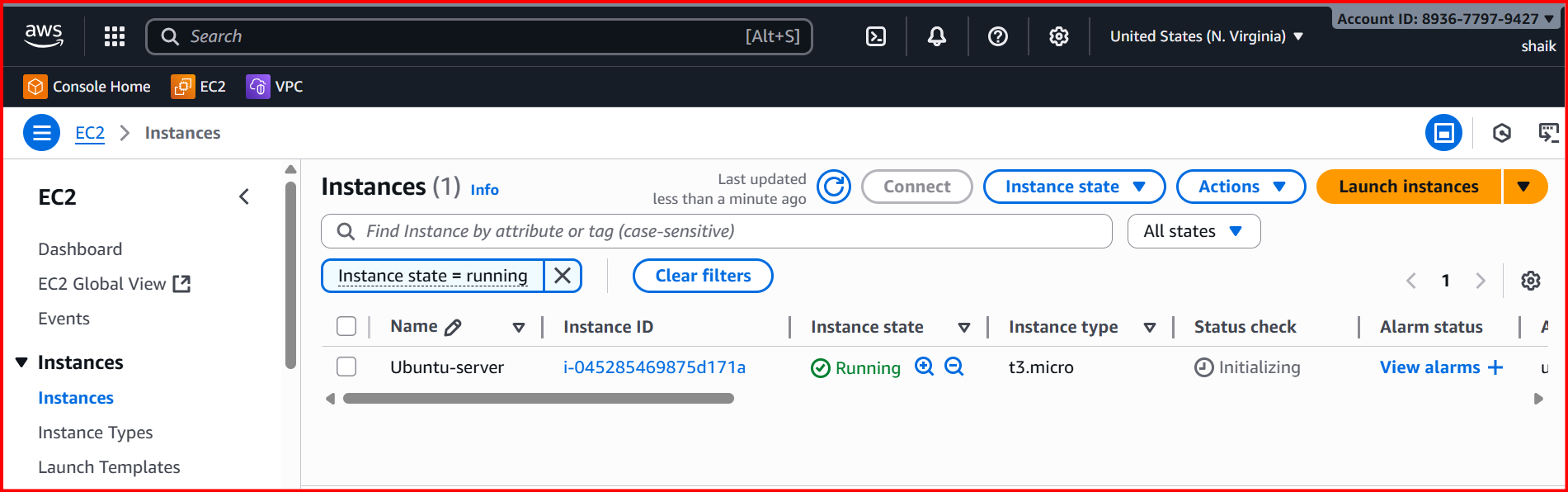
**EOF**

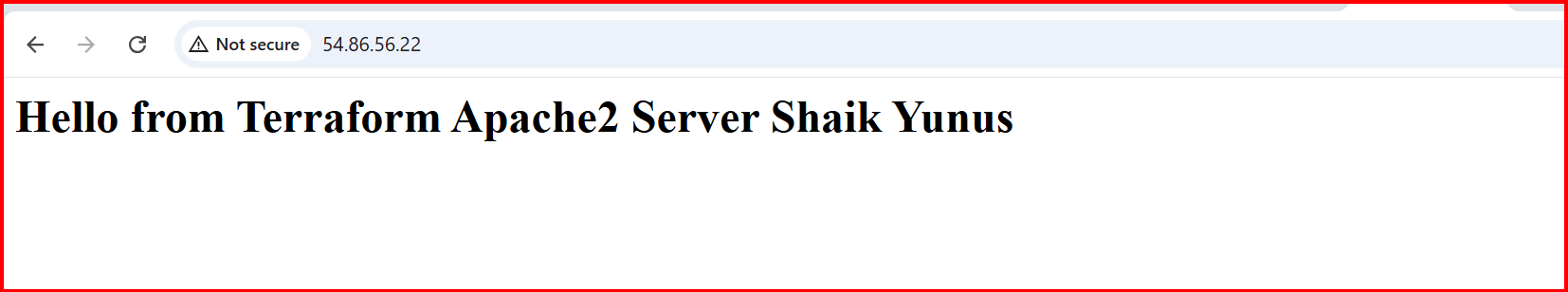
**tags = {**

**Name = "Ubuntu-server"**

**}**

**}**

****

****