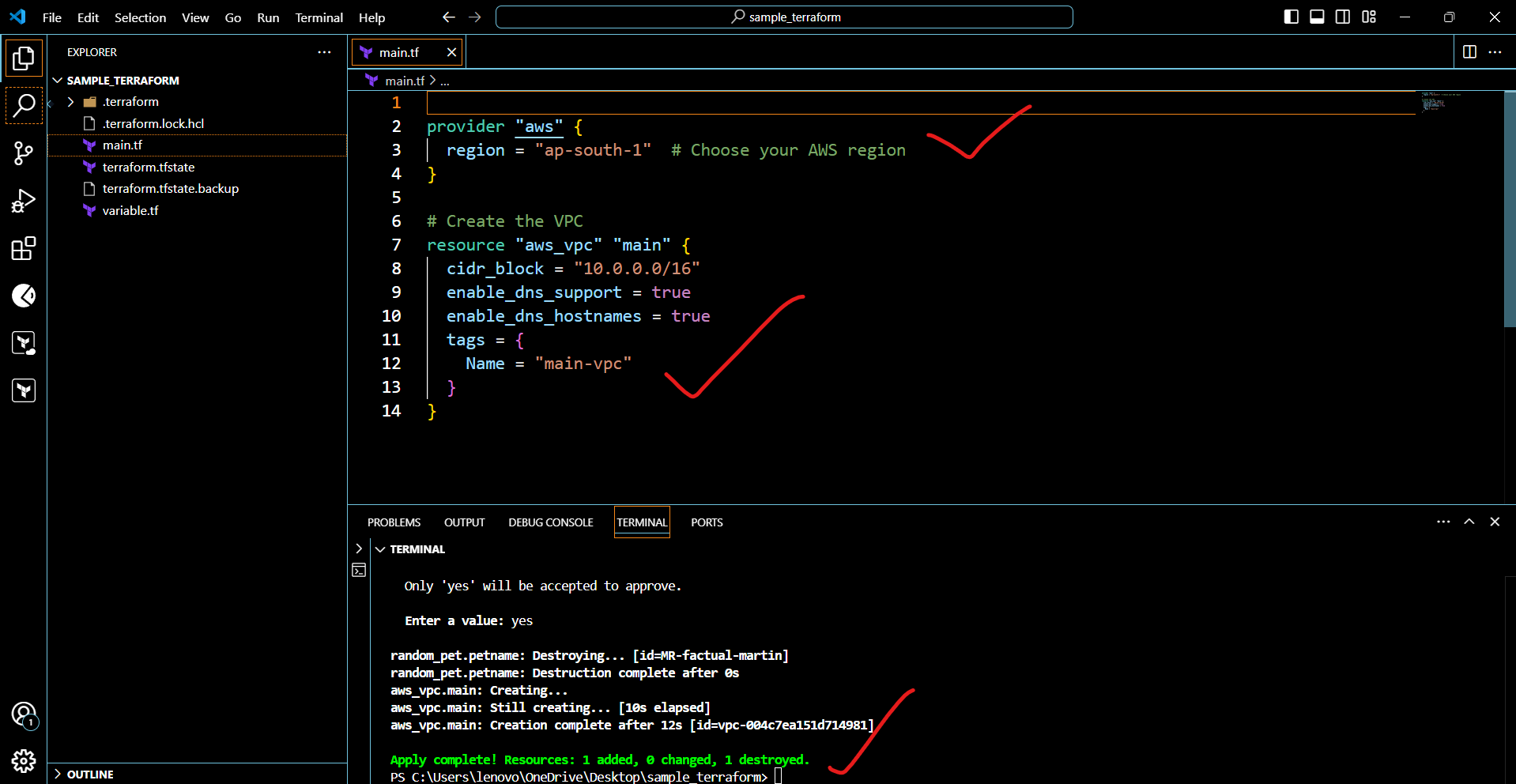
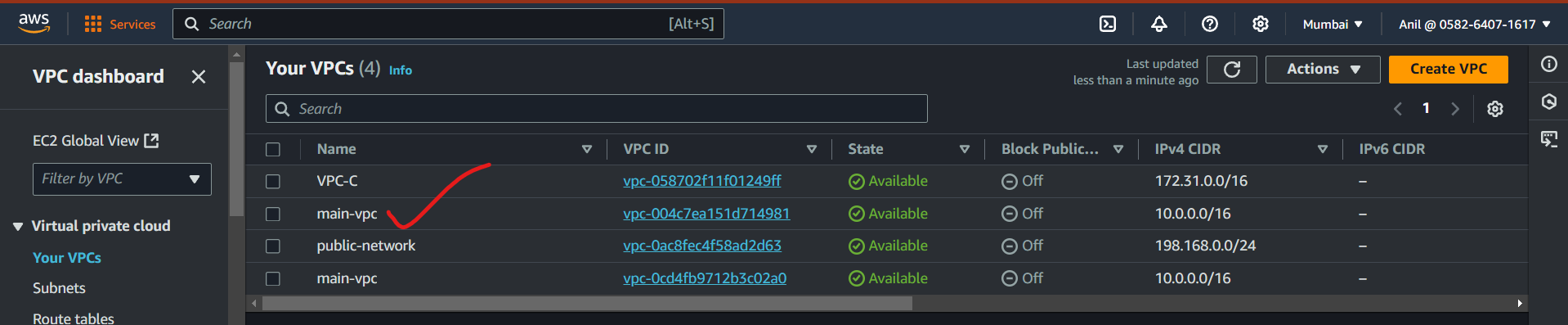
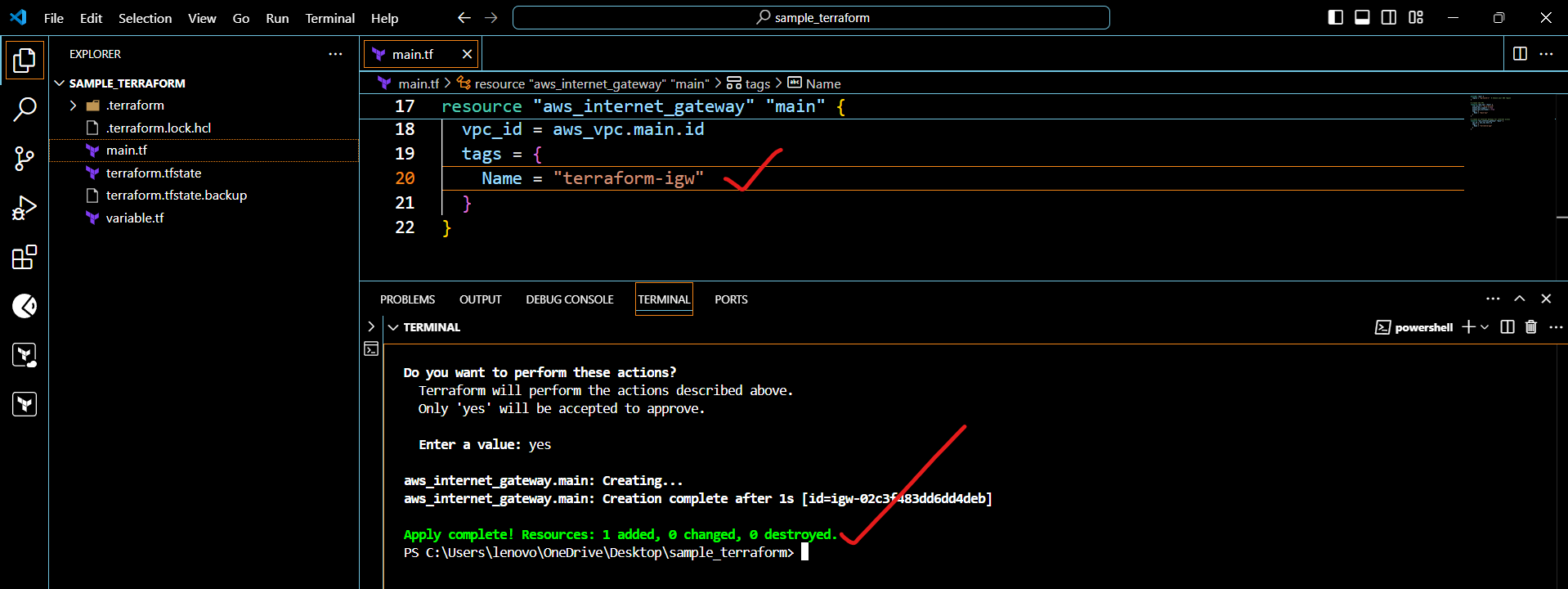
**Create VPC using Terraform  
========================**

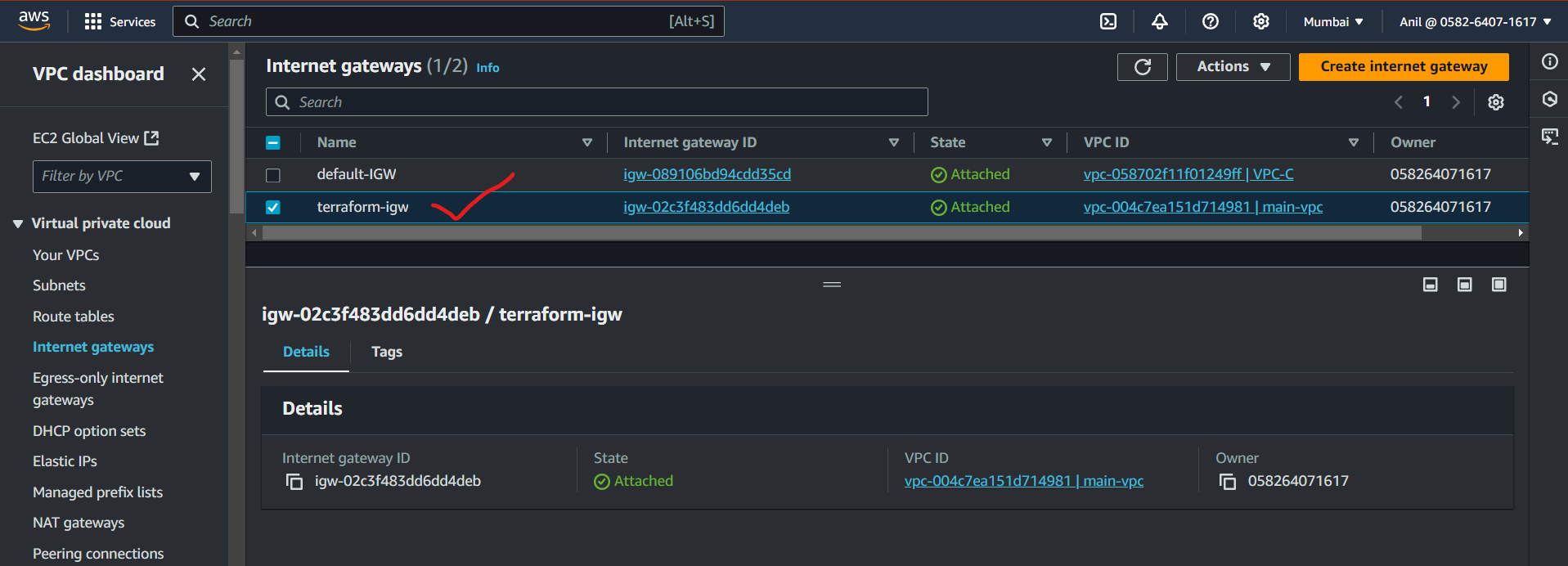
**1) Create VPC**

****

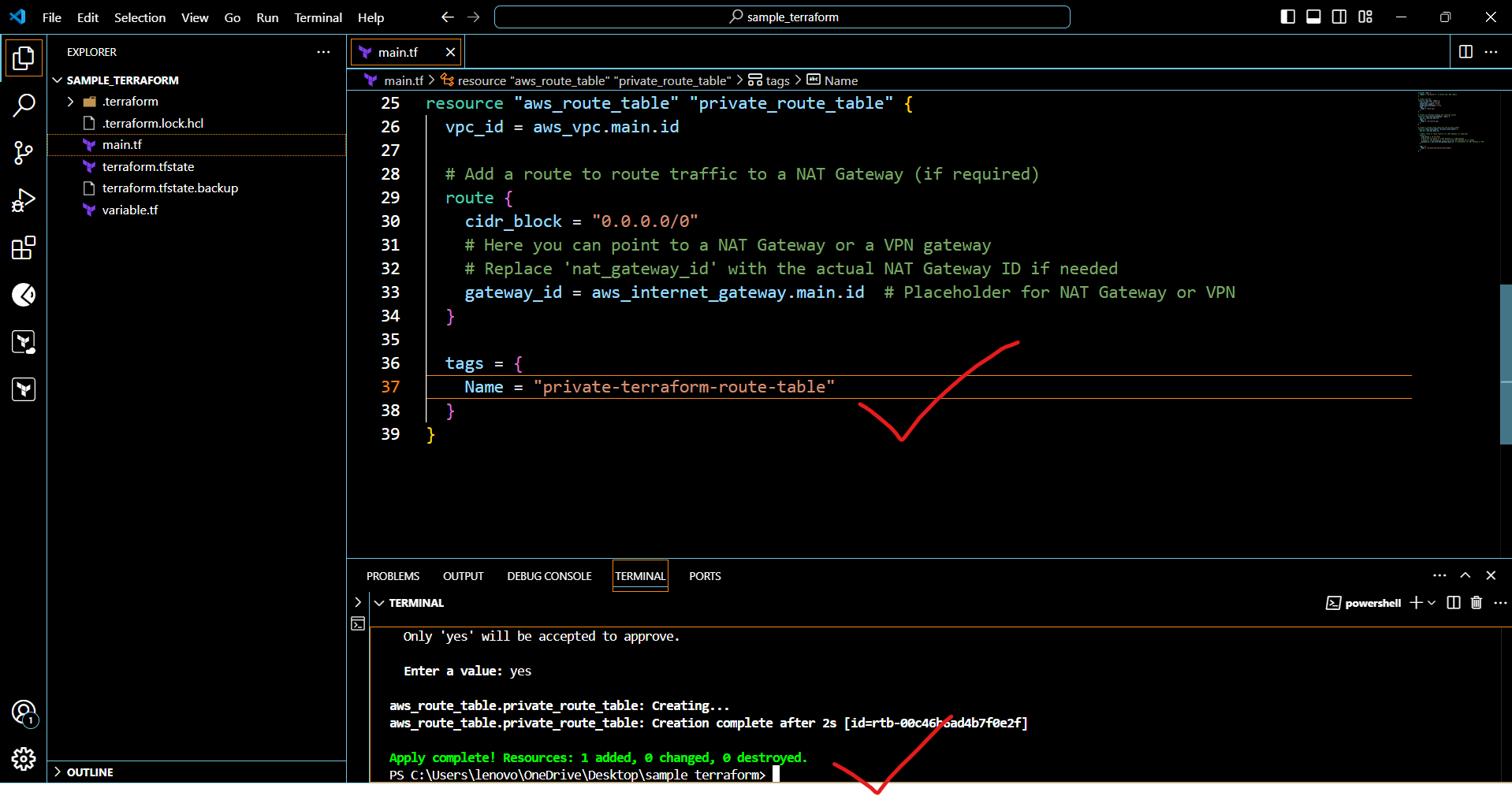
****

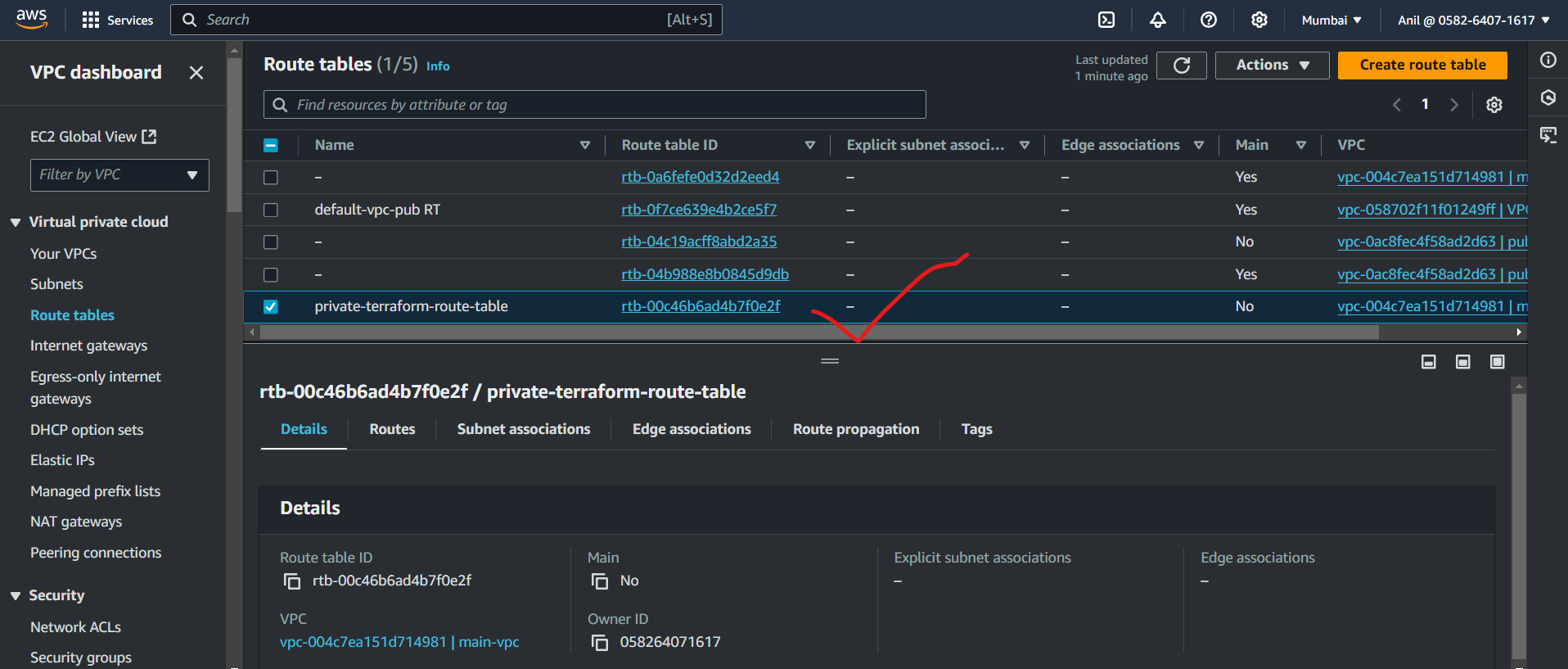
**2) Create Internet gateway**

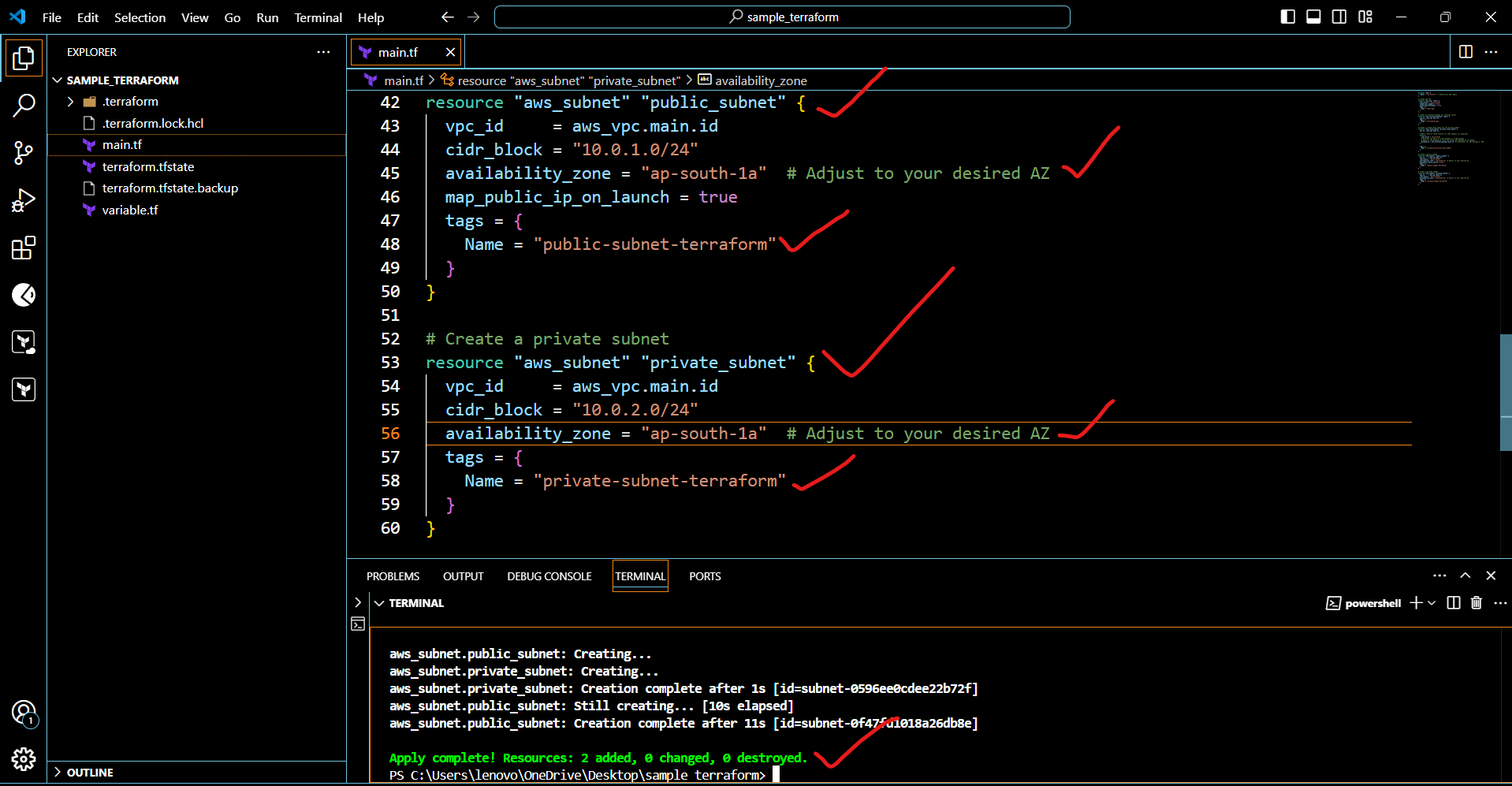
****

****

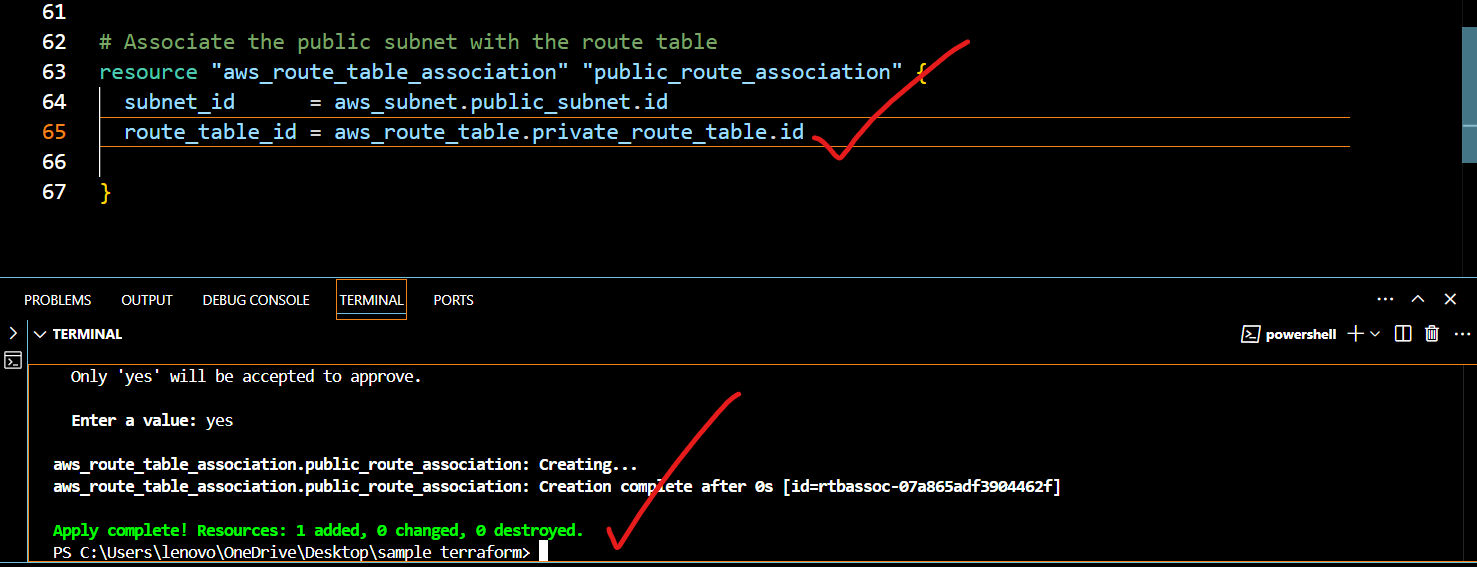
**3) Create Custom Route Table**

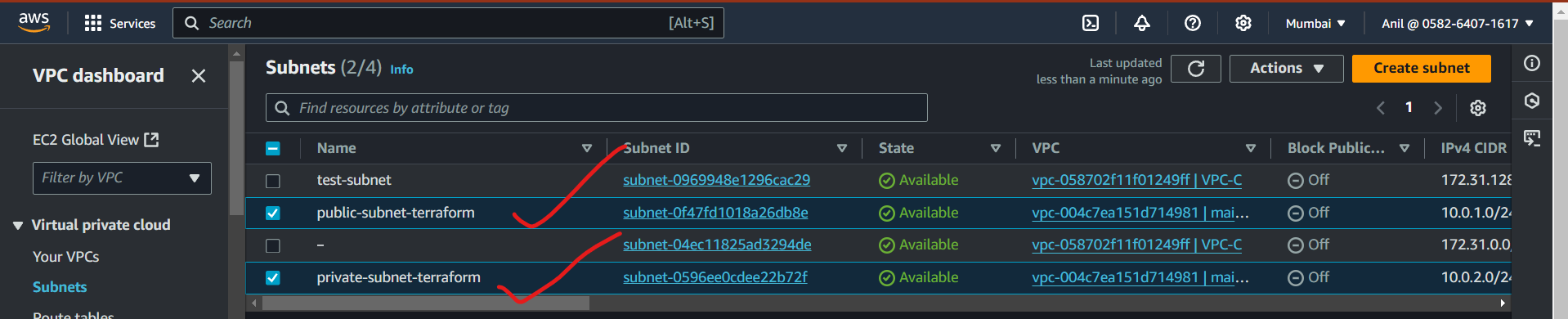
****

****

****

**4) Create Subnet**

****

****

**5) Associate subnet with Route Table**

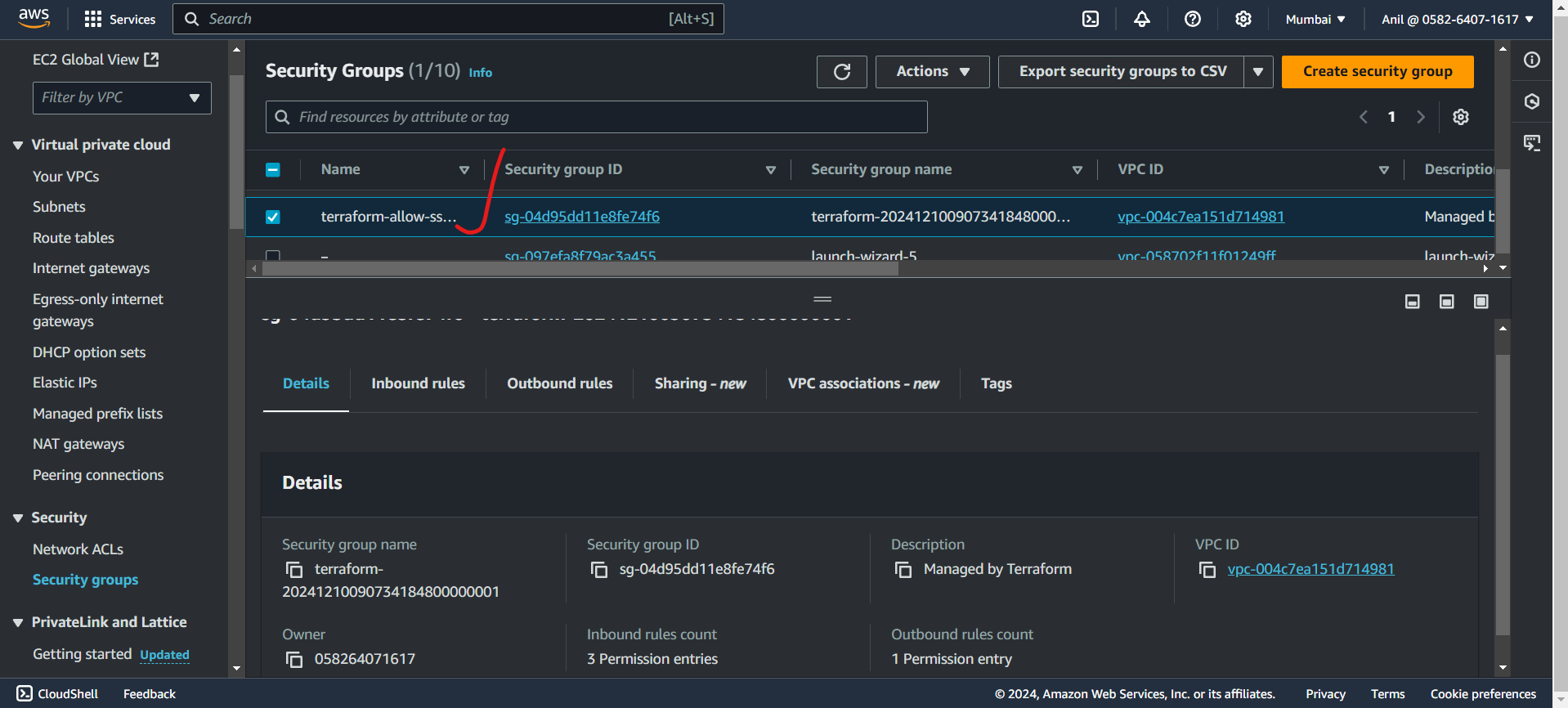
# Associate the public subnet with the route table

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

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

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

}

****

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

# Create a Security Group for allowing inbound traffic on ports 22, 80, and 443

resource "aws\_security\_group" "allow\_http\_https\_ssh" {

  vpc\_id = aws\_vpc.main.id  # Attach this SG to the VPC

  # Inbound rules: Allow traffic on ports 22 (SSH), 80 (HTTP), and 443 (HTTPS)

  ingress {

    from\_port   = 22

    to\_port     = 22

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]  # Allow SSH from anywhere (can be restricted)

  }

  ingress {

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

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

  }

  ingress {

    from\_port   = 443

    to\_port     = 443

    protocol    = "tcp"

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

  }

  # Optional: Outbound rules (default is allow all traffic)

  egress {

    from\_port   = 0

    to\_port     = 0

    protocol    = "-1"  # Allow all outbound traffic

    cidr\_blocks = ["0.0.0.0/0"]

  }

  tags = {

    Name = "terraform-allow-ssh-http-https"

  }

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

resource "aws\_instance" "tf\_instance" {

  ami                    = "ami-053b12d3152c0cc71"  # Update this with the correct AMI ID for your region

  instance\_type          = "t2.micro"

  key\_name               = "example"

  subnet\_id              = aws\_subnet.public\_subnet.id  # Use the public subnet for internet access

  vpc\_security\_group\_ids = [aws\_security\_group.allow\_http\_https\_ssh.id]  # Use the security group ID here

  associate\_public\_ip\_address = true

  user\_data = <<-EOF

        #!/bin/bash

        sudo apt update -y

        sudo apt install -y apache2

        sudo systemctl start apache2

        sudo systemctl enable apache2

        echo "created apache2 server" > /var/www/html/index.html

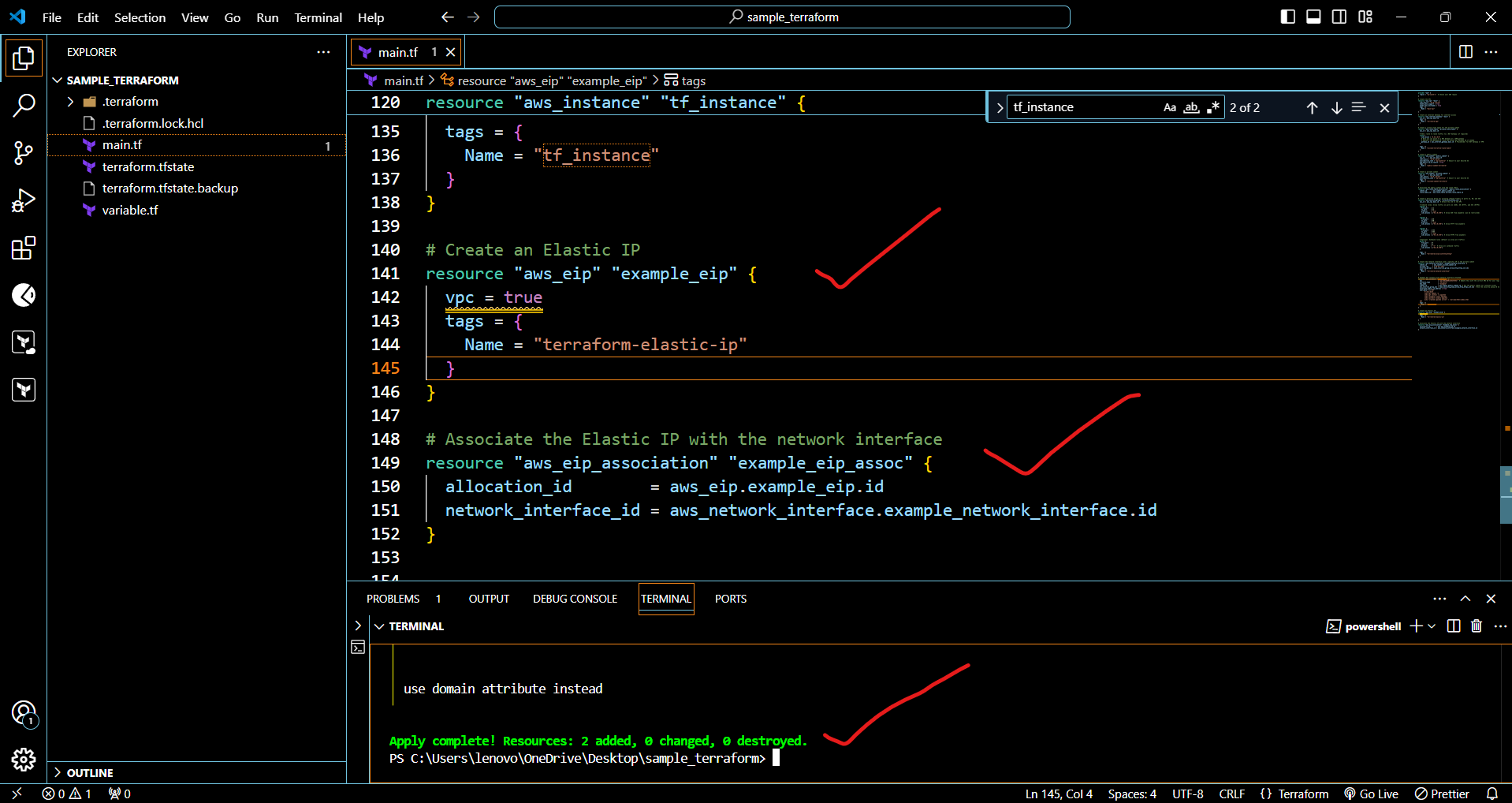
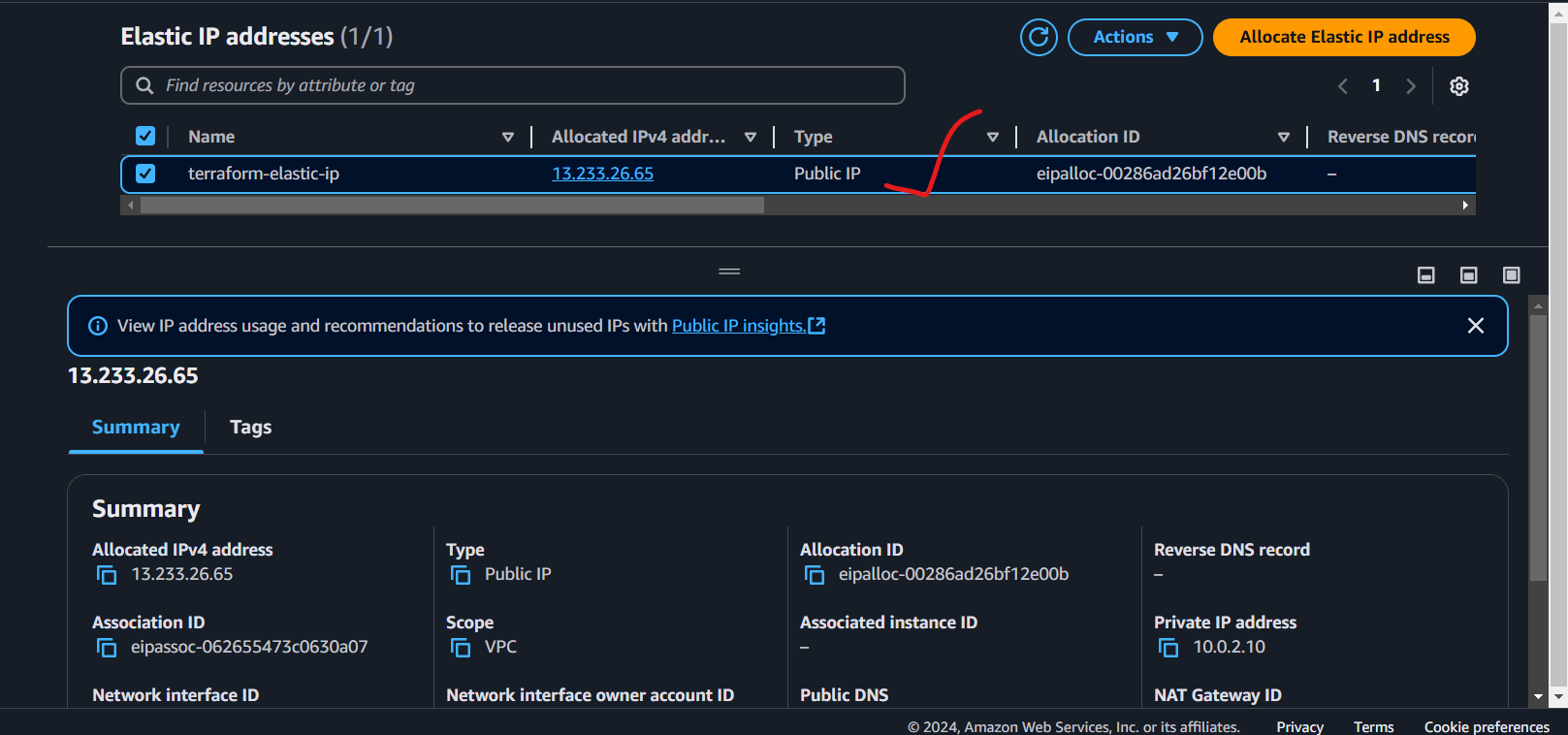
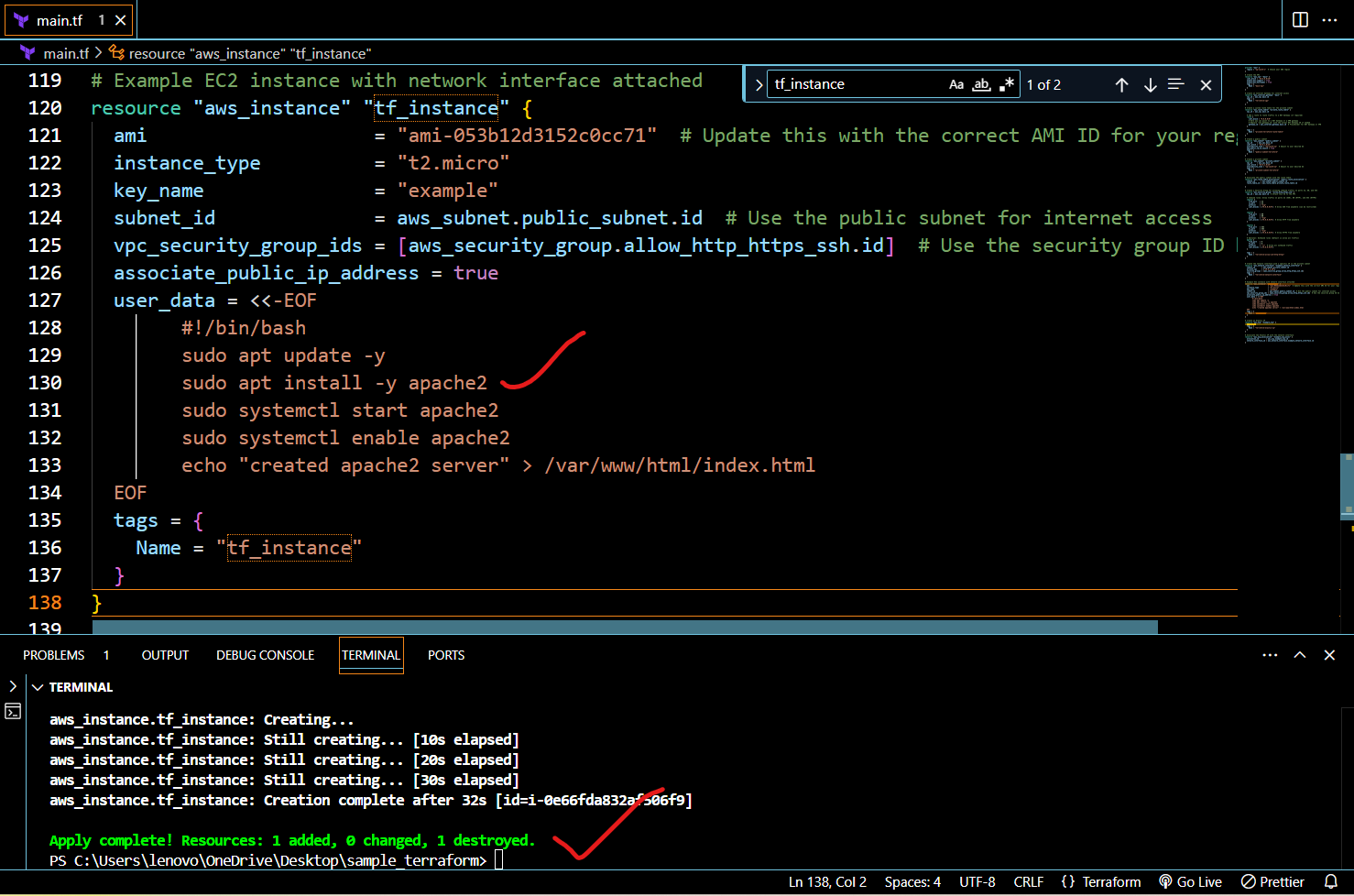
  EOF

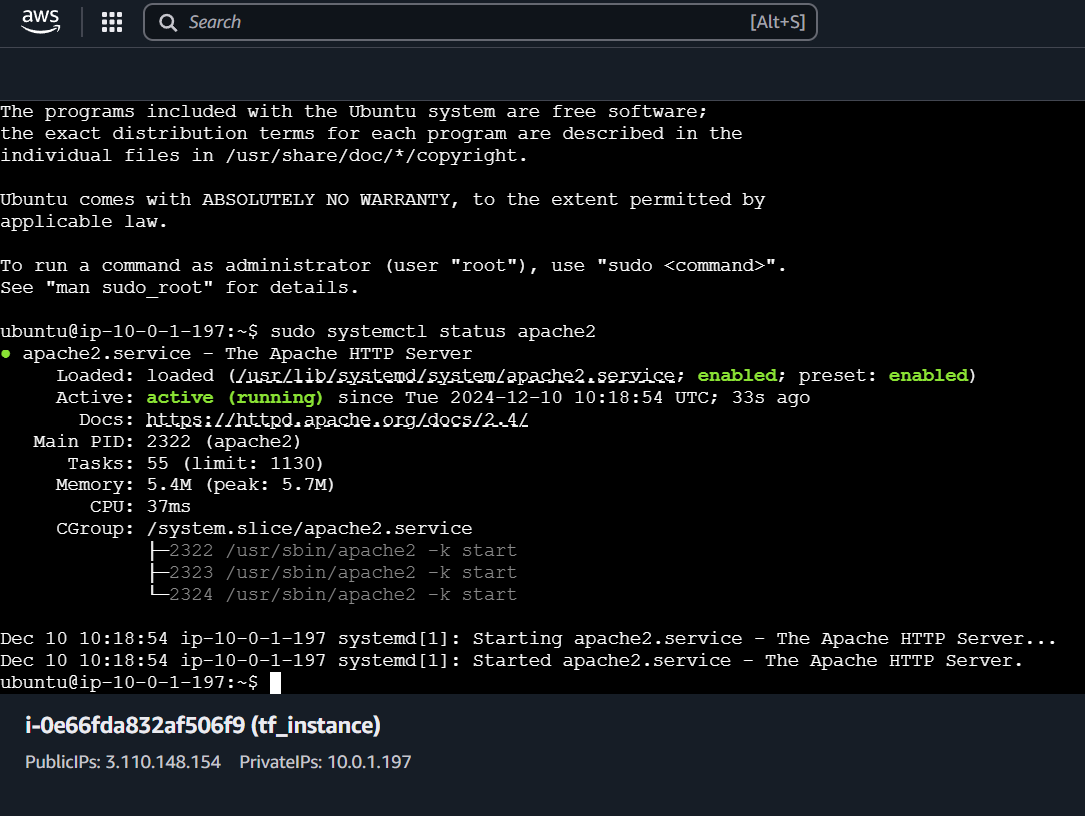
  tags = {

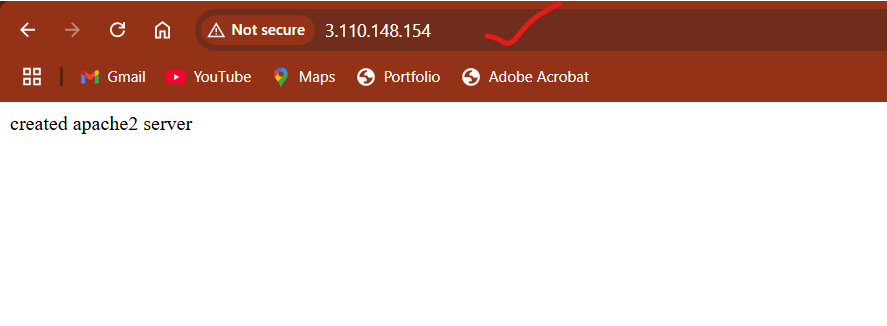
    Name = "tf\_instance"

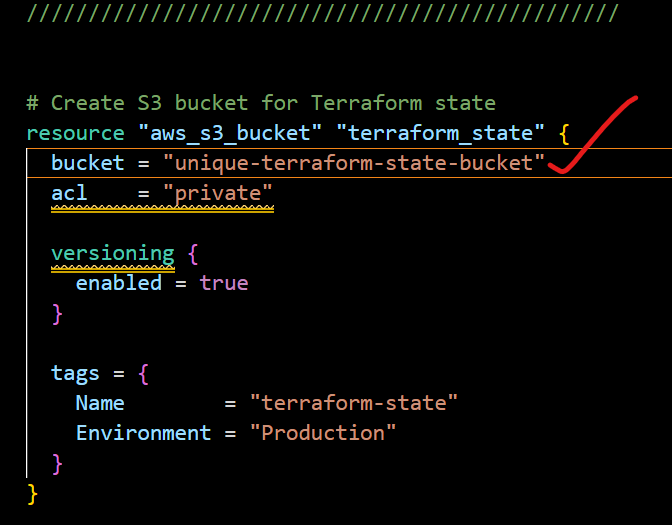
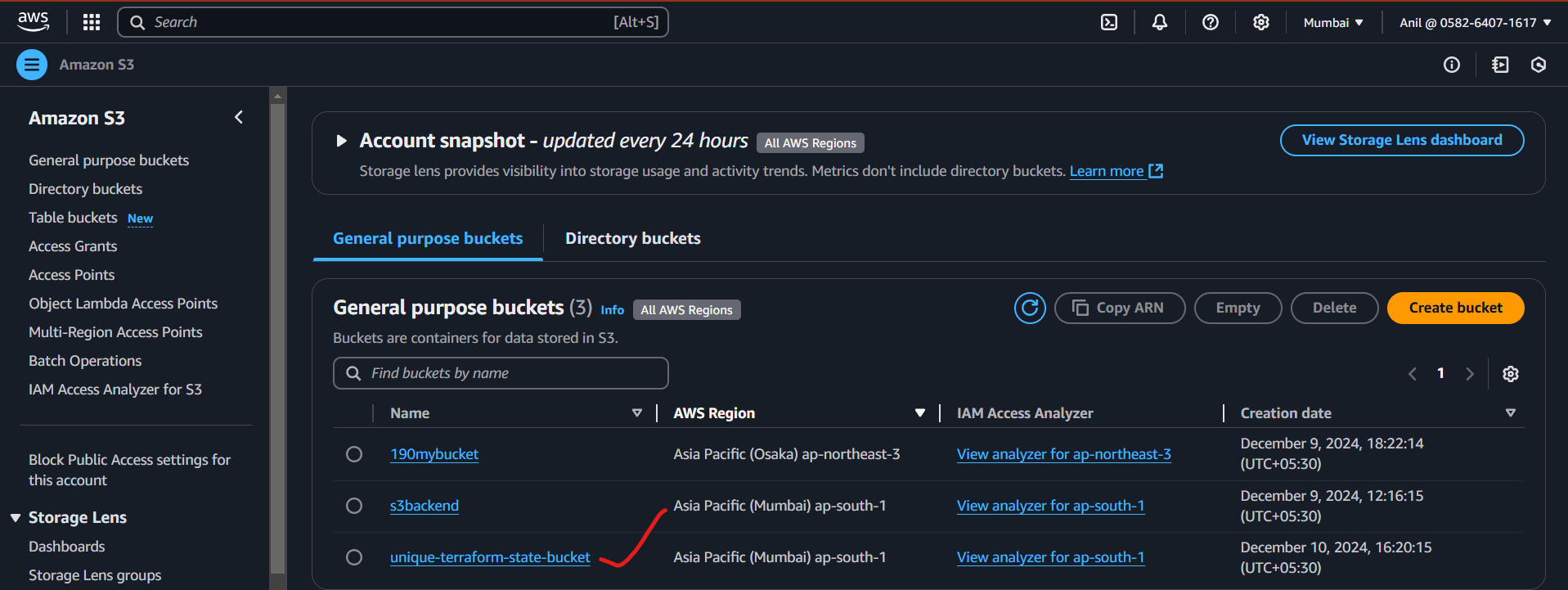
  }

}

**8) Assign an elastic IP to the network interface created in step 7   
  
  
9) Create Ubuntu server and install/enable apache2  
**

****

****

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

Dynamodb  
  
