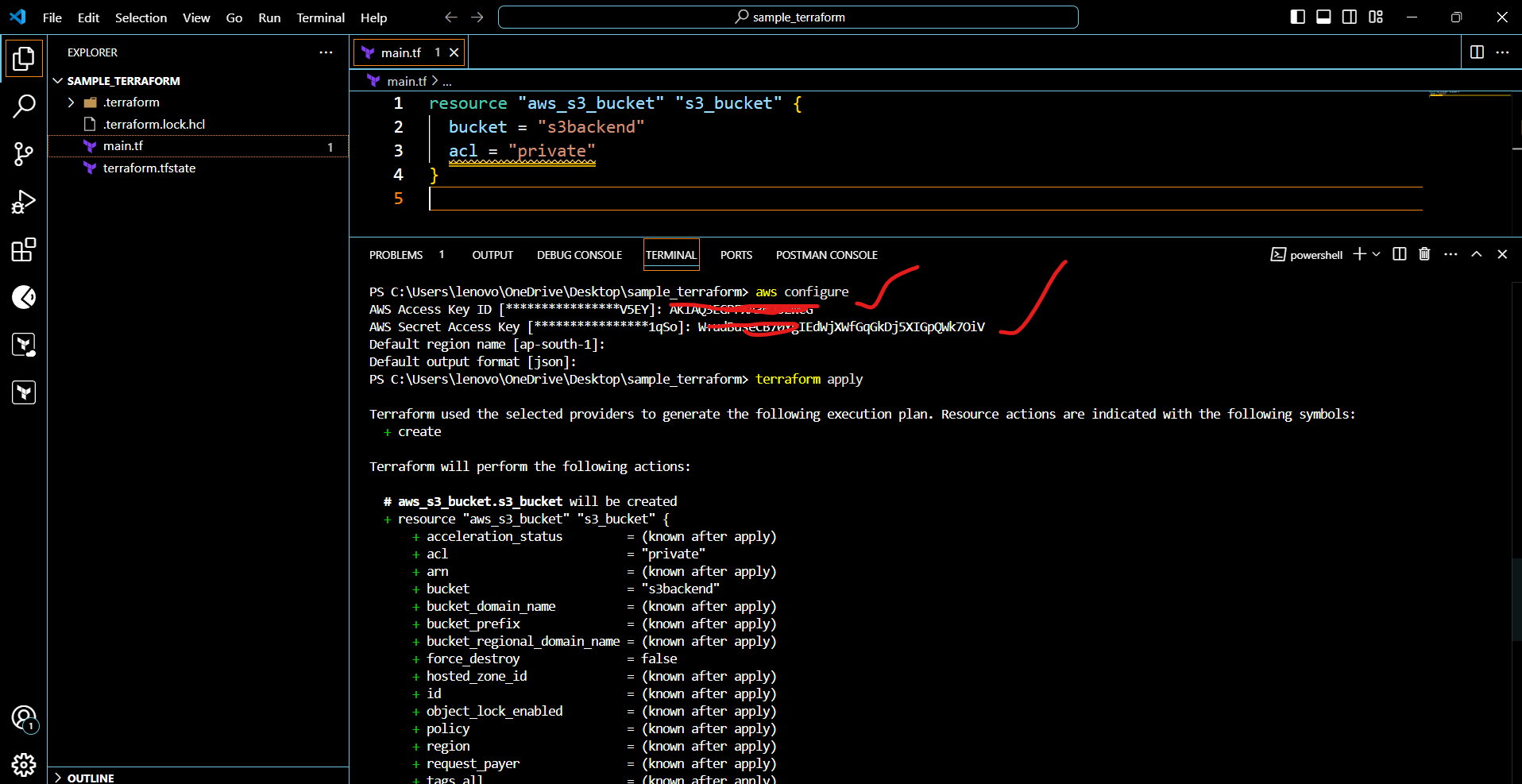
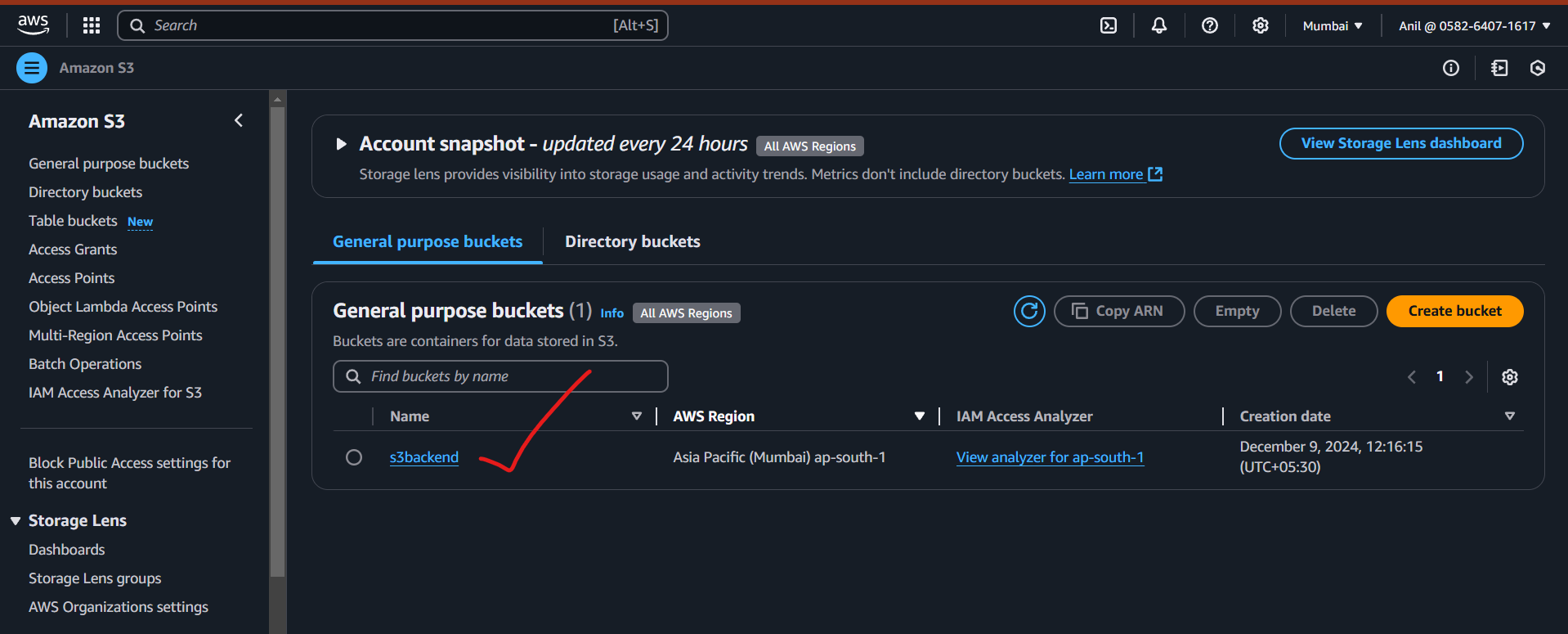
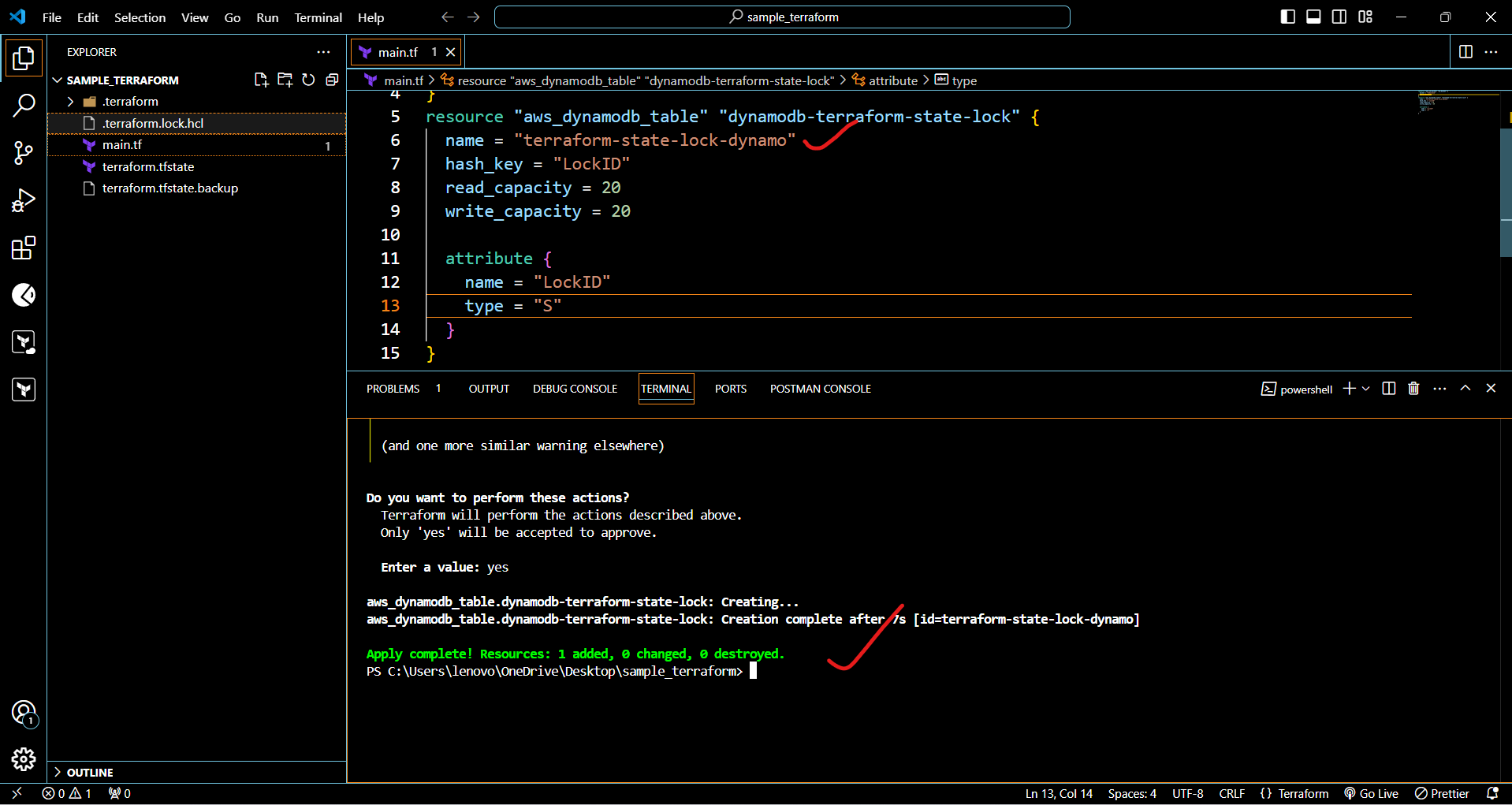
**1) Watch terraform-05 video.**

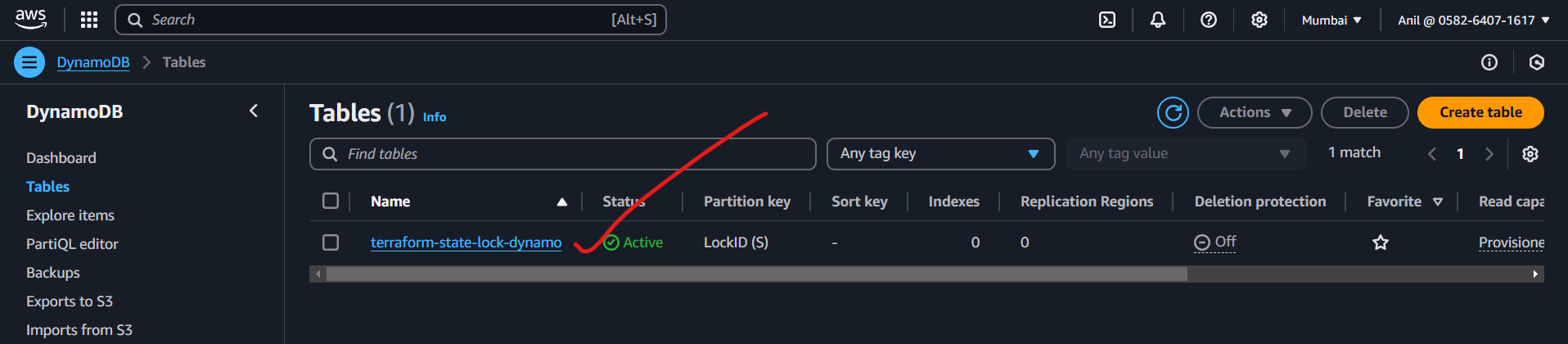
completed

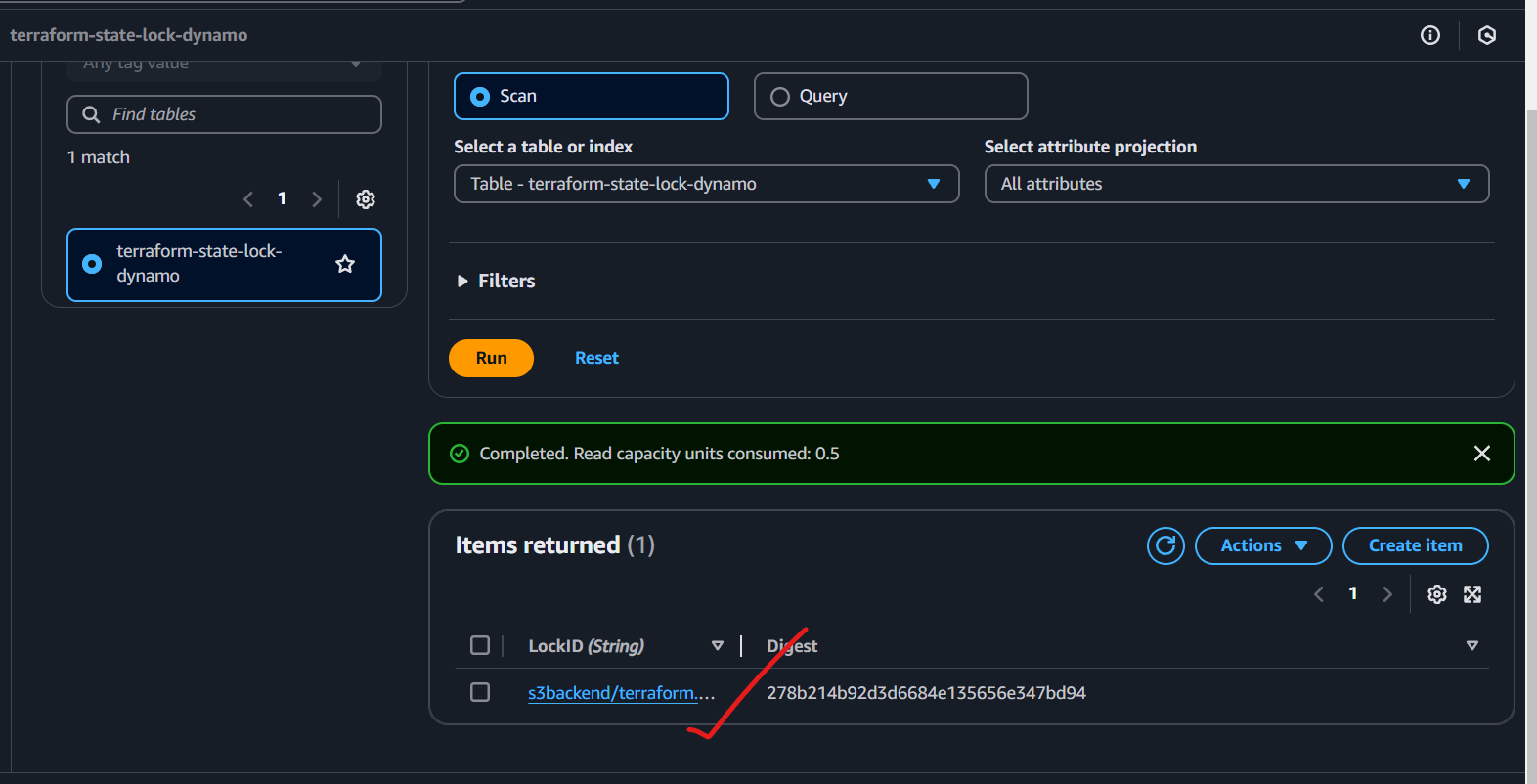
**2) Execute the script shown in video.**

Create s3 backend  




Dynamodb  






**3) Create one ec2 instance with httpd installed using terraform script.**

provider "aws" {

  region = "ap-south-1" # Replace with your desired AWS region

}

resource "aws\_instance" "httpd\_instance" {

  ami           = "ami-0614680123427b75e" # Amazon Linux 2 AMI (replace if needed)

  instance\_type = "t2.micro"

  # Configure user data to install httpd

  user\_data = <<-EOF

              #!/bin/bash

              yum update -y

              yum install -y httpd

              systemctl start httpd

              systemctl enable httpd

              echo "<h1>Welcome to Apache Server on EC2!</h1>" > /var/www/html/index.html

              EOF

  tags = {

    Name = "httpd-instance"

  }

  # Assign a key pair if needed

  key\_name = "example" # Replace with your key pair name

  # Add a security group for HTTP access

  vpc\_security\_group\_ids = [aws\_security\_group.http\_access.id]

}

resource "aws\_security\_group" "http\_access" {

  name        = "http-access"

  description = "Allow HTTP access"

  ingress {

    from\_port   = 80

    to\_port     = 80

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"] # Allow access from all IPs

  }

  egress {

    from\_port   = 0

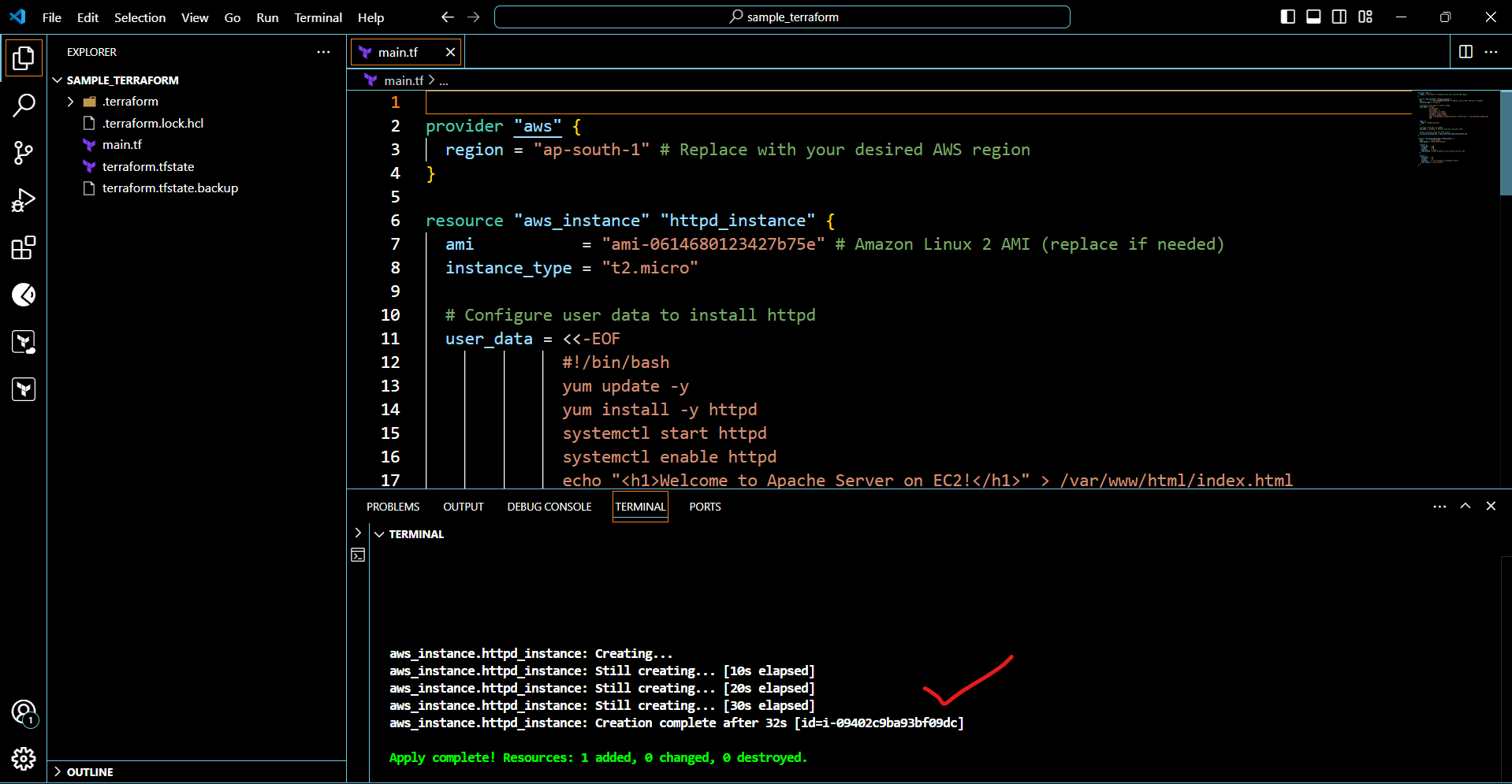
    to\_port     = 0

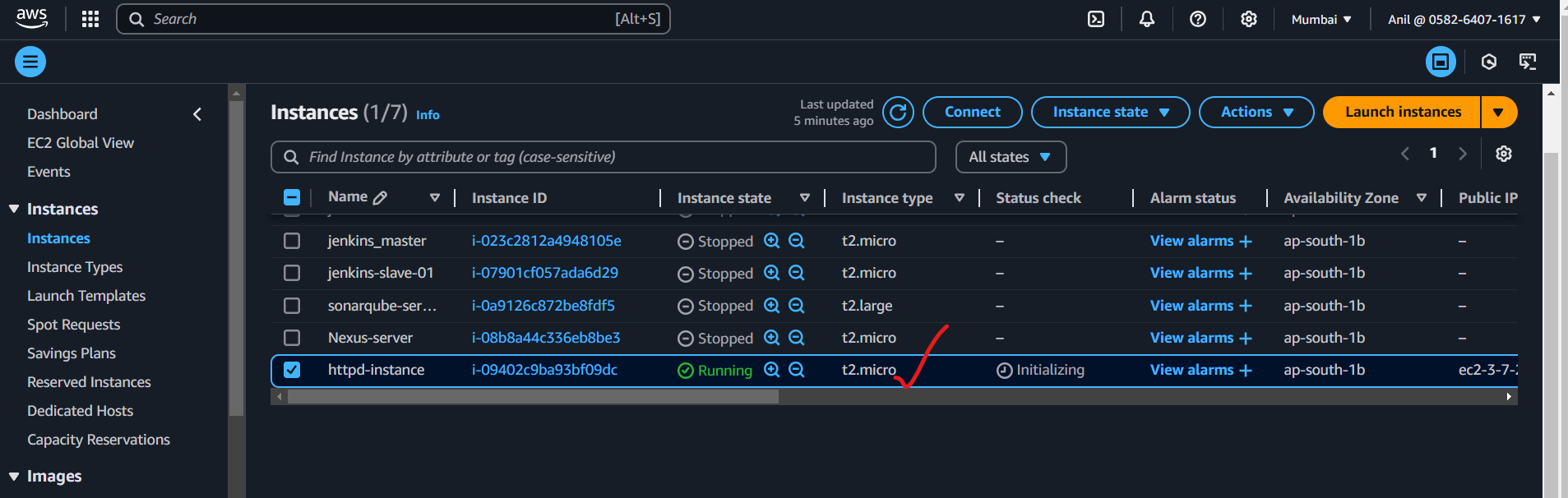
    protocol    = "-1" # Allow all outbound traffic

    cidr\_blocks = ["0.0.0.0/0"]

  }

}

****

****

****

**4) Setup s3 as backend to the task 3.**

terraform {

  backend "s3" {

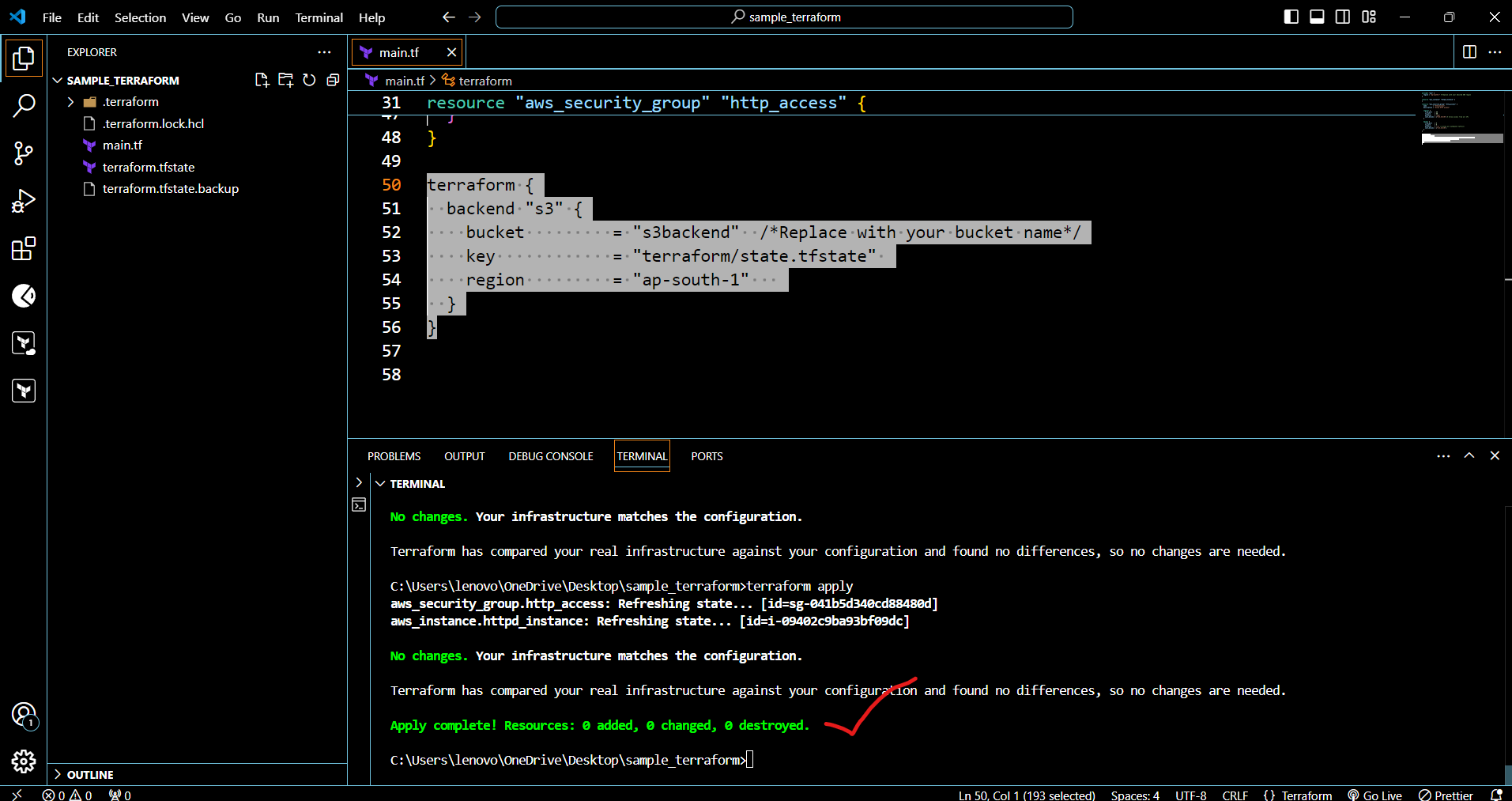
    bucket         = "s3backend"  /\*Replace with your bucket name\*/

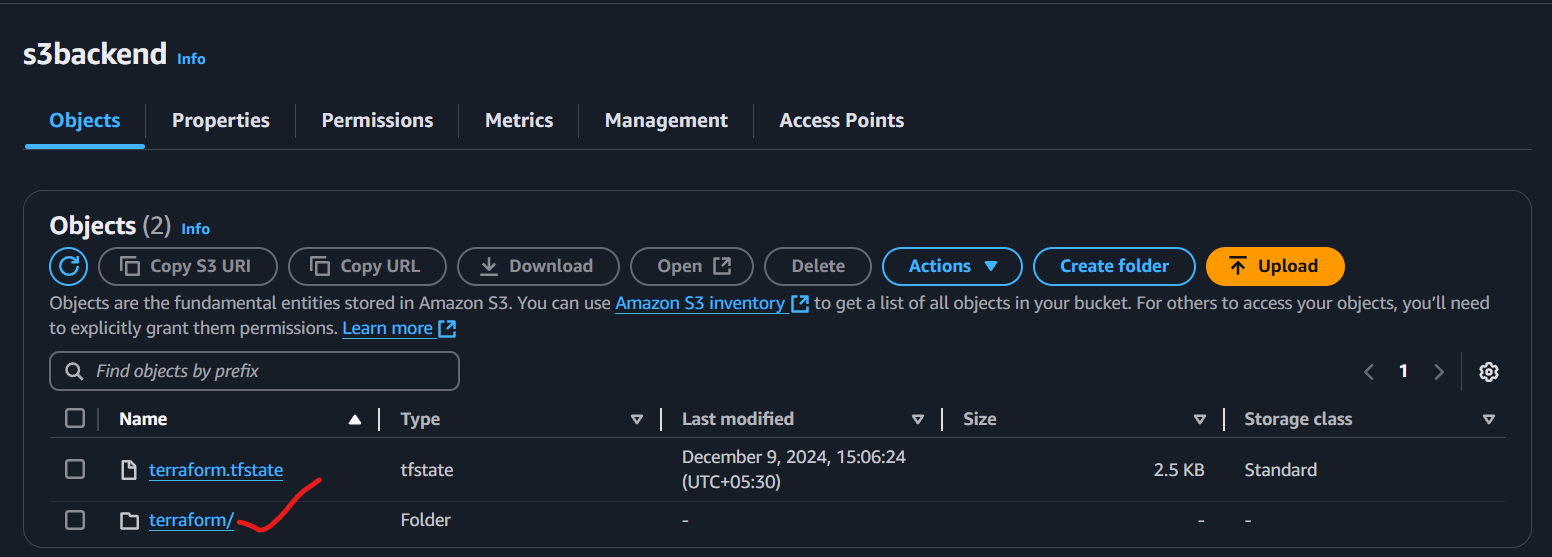
    key            = "terraform/state.tfstate"

    region         = "ap-south-1"

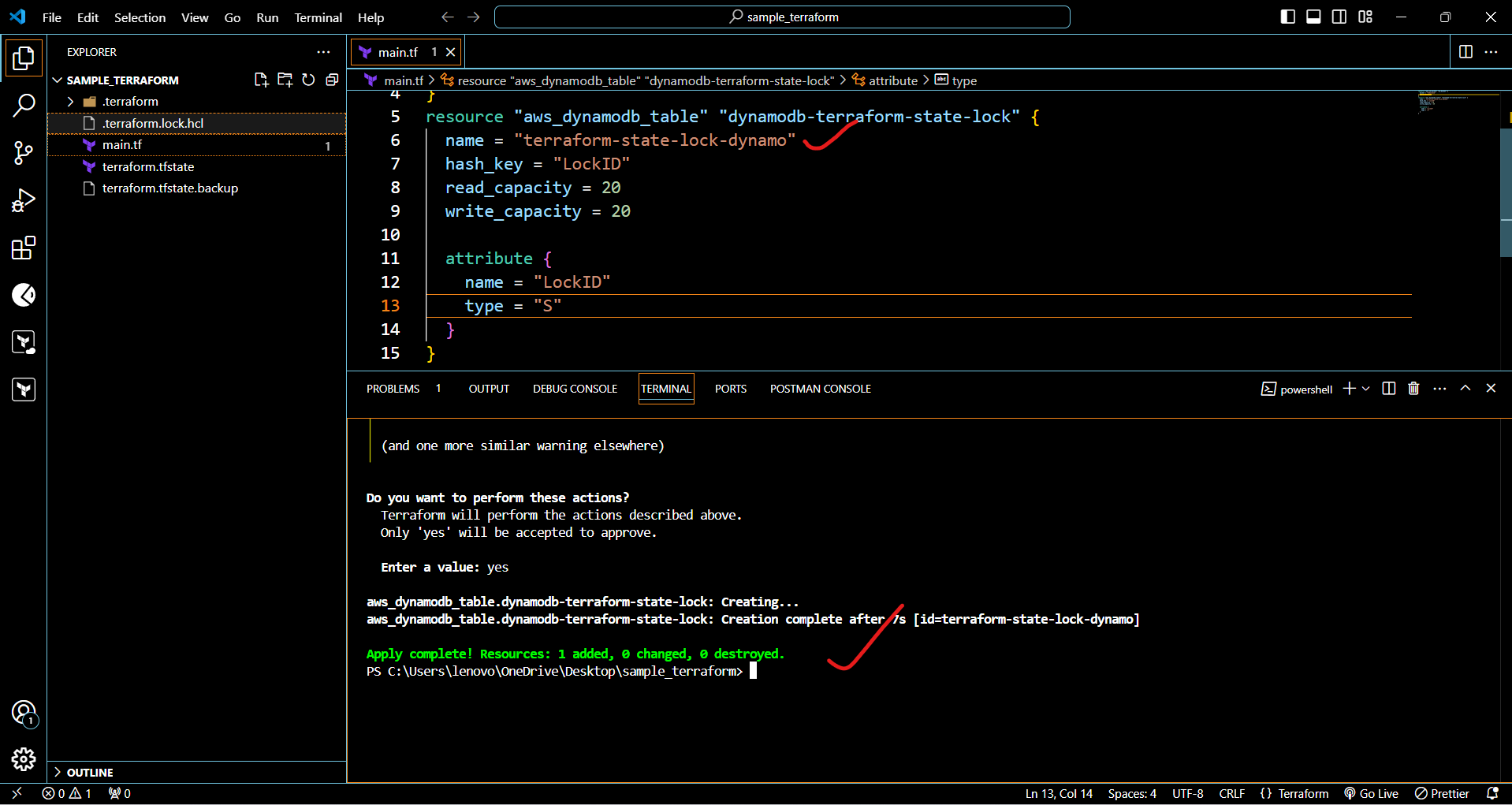
  }

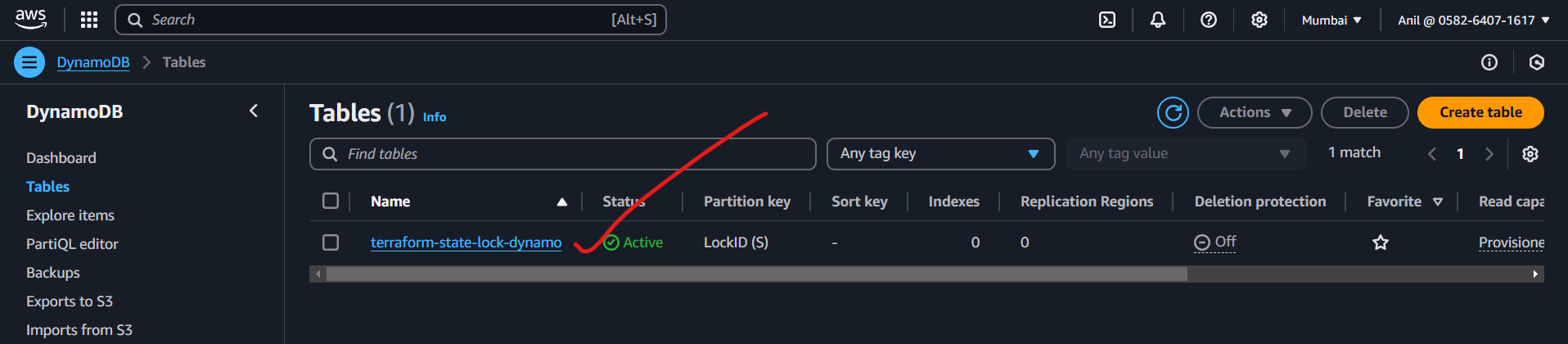
}

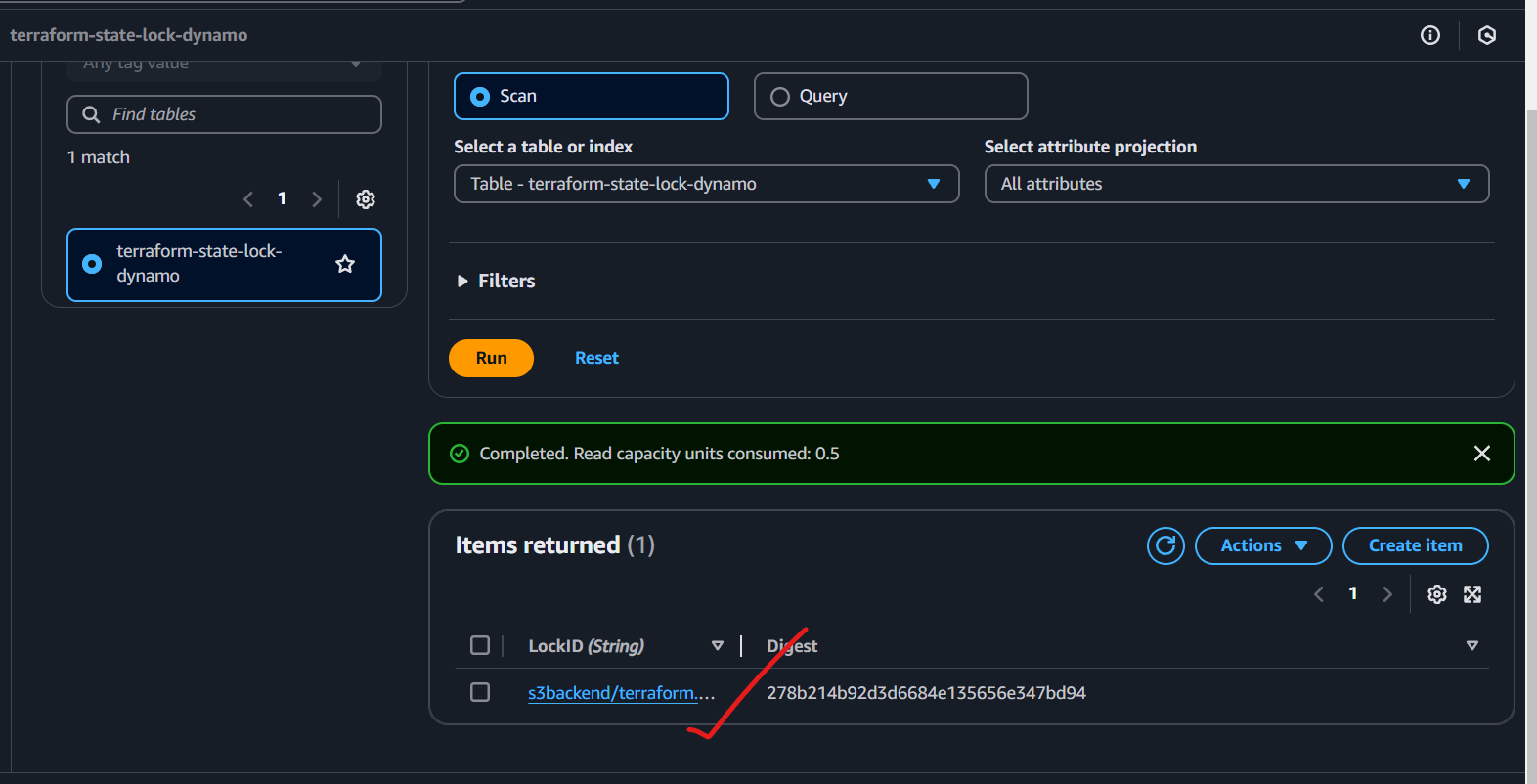
****

****

**5) Setup dynamo db locking for task3.**



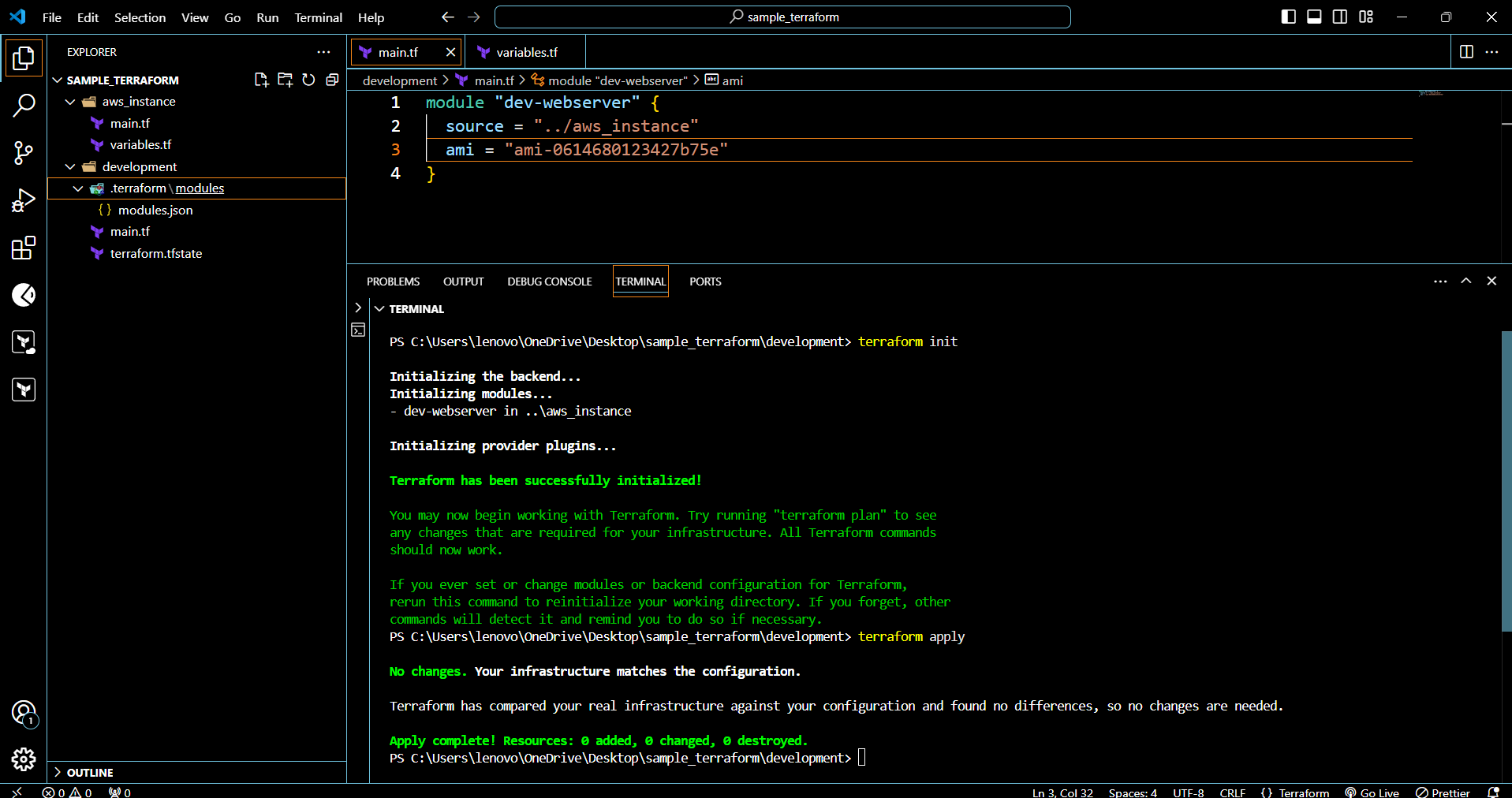




**6) Watch terraform-06 video.**

completed

**7) Execute the script shown in video.**

****

**8) Provision ec2,s3 and vpc using Terraform modules.**

provider "aws" {

region = "us-west-1"

access\_key = "AKIAW3MD7G7LBGTD56E7"

secret\_key = "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"

}

resource "aws\_instance" "Ec2" {

ami = "ami-0175bdd48fdb0973b"

instance\_type = "t2.micro"

key\_name = "Ncalifornia" vpc\_security\_group\_ids = [aws\_security\_group.main.id]

provisioner "file" {

source ="C:/Users/ramee/Desktop/Terraform/.terraform/Ncalifornia.pem" destination = "/home/ec2-user/Ncalifornia.pem"

}

connection {

type = "ssh"

host = self.public\_ip

user = "ec2-user"

private\_key = file("${path.module}/Ncalifornia.pem") timeout = "4m"

}

tags = {

Name = "My\_EC2\_Instance"

}

}

resource "aws\_security\_group" "main" { name = "allow\_ssh"

description = "Allow SSH inbound traffic"

ingress { from\_port = 22

to\_port = 22

protocol = "tcp" cidr\_blocks = ["0.0.0.0/0"]

}

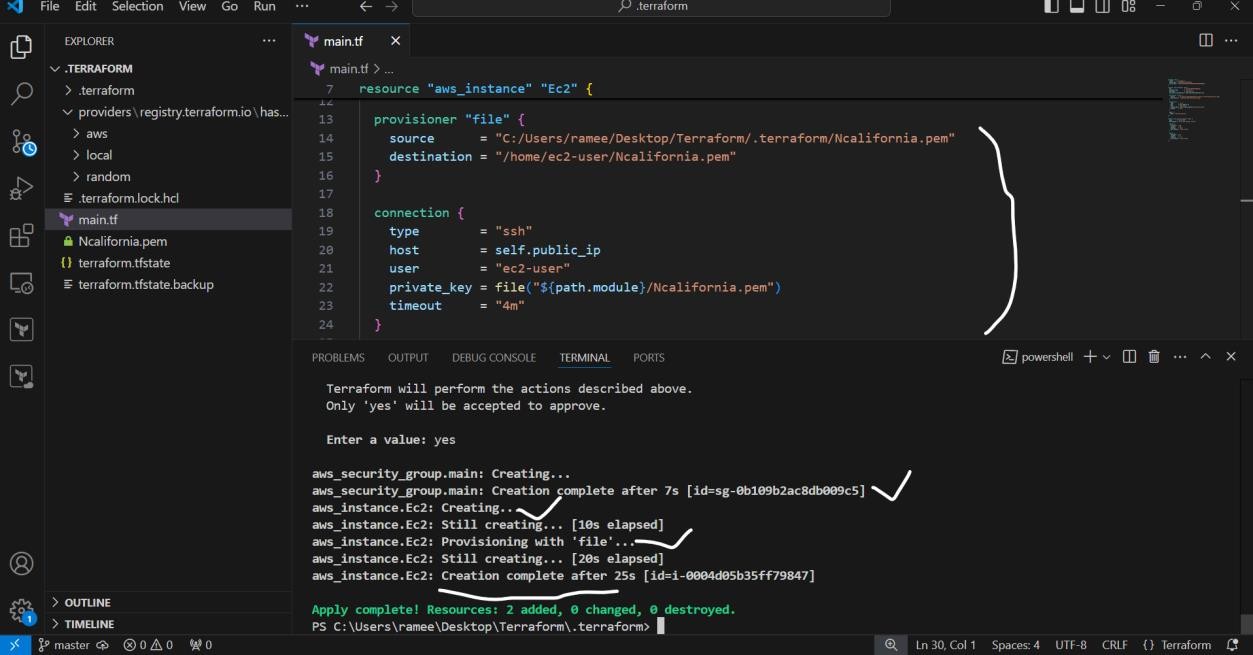
egress { from\_port = 0

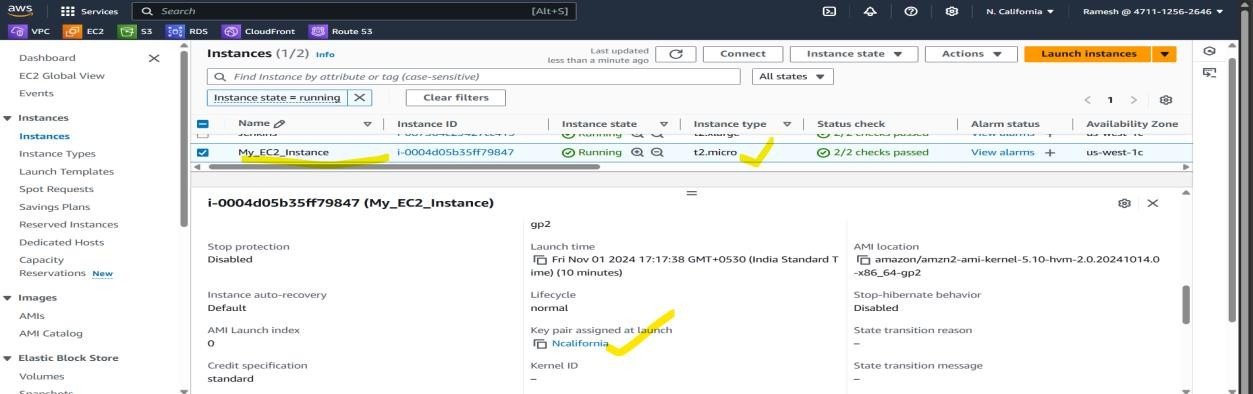
to\_port = 0

protocol = "-1" cidr\_blocks = ["0.0.0.0/0"]

}

}





**9) Provision ec2 for 3 different environments (Dev, Staging and Prod) using terraform workspaces.**