LAB ACTIVITY 1

- Shaurya Srinet (RA2111032010006)

CREATE AN EC2 INSTANCE USING TERRAFORM

Procedure:

1. Install Terraform: If you haven't already, download and install Terraform from the official website: https://www.terraform.io/downloads.html

2. Set up AWS Credentials: Ensure you have AWS access key ID and secret access key with the required permissions to create EC2 instances. You can set these up via AWS IAM.

3. Create a Terraform Configuration File: Open a text editor and create a new file named `main.tf`.

4. Write Terraform Configuration:

```
provider "aws" {
  region = "us-east-1"   # Change this to your desired AWS region
}

resource "aws_instance" "example" {
  ami           = "ami-07d9b9ddc6cd8dd30"  # Change this to the desired AMI ID
  instance_type = "t2.micro"               # Change this to your desired
instance type

  tags = {
    Name = "New_Instance"  # Specify the name for your EC2 instance
  }
}
```

Replace `"ami-07d9b9ddc6cd8dd30"` with your desired AMI ID.

5. Save the File: Save the `main.tf` file in a directory of your choice.

6. Initialize Terraform: Open a terminal or command prompt in the directory where your `main.tf` file is located and run: **terraform init**

7. Review and Apply Changes: Before applying the changes, review what Terraform plans to do by running: **terraform plan**

If everything looks good, apply the changes: **terraform apply**

Type `**yes**` when prompted to confirm.

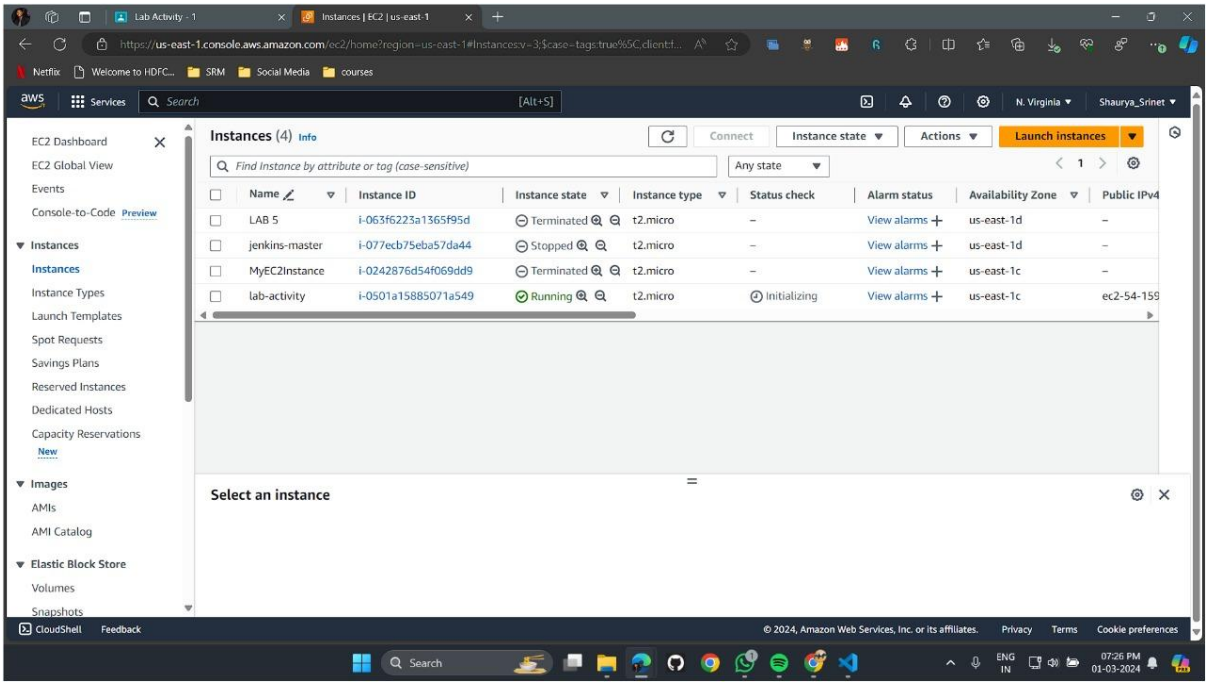
8. Verify: Once the apply command completes successfully, verify that your EC2 instance is created by logging into the AWS Management Console, navigating to the EC2 service, and checking the instances list.

9. Clean Up: To clean up resources created by Terraform, you can run: **terraform destroy**

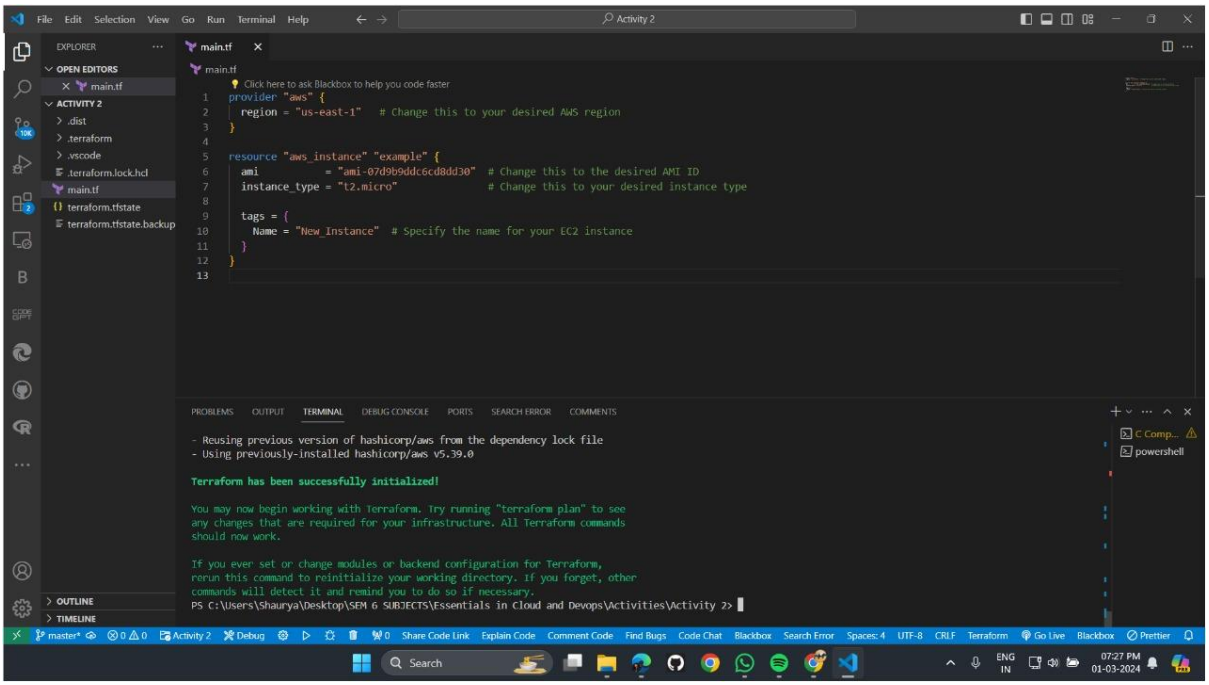
This command will remove all the resources defined in your Terraform configuration.

<u>Output Screenshots:</u>

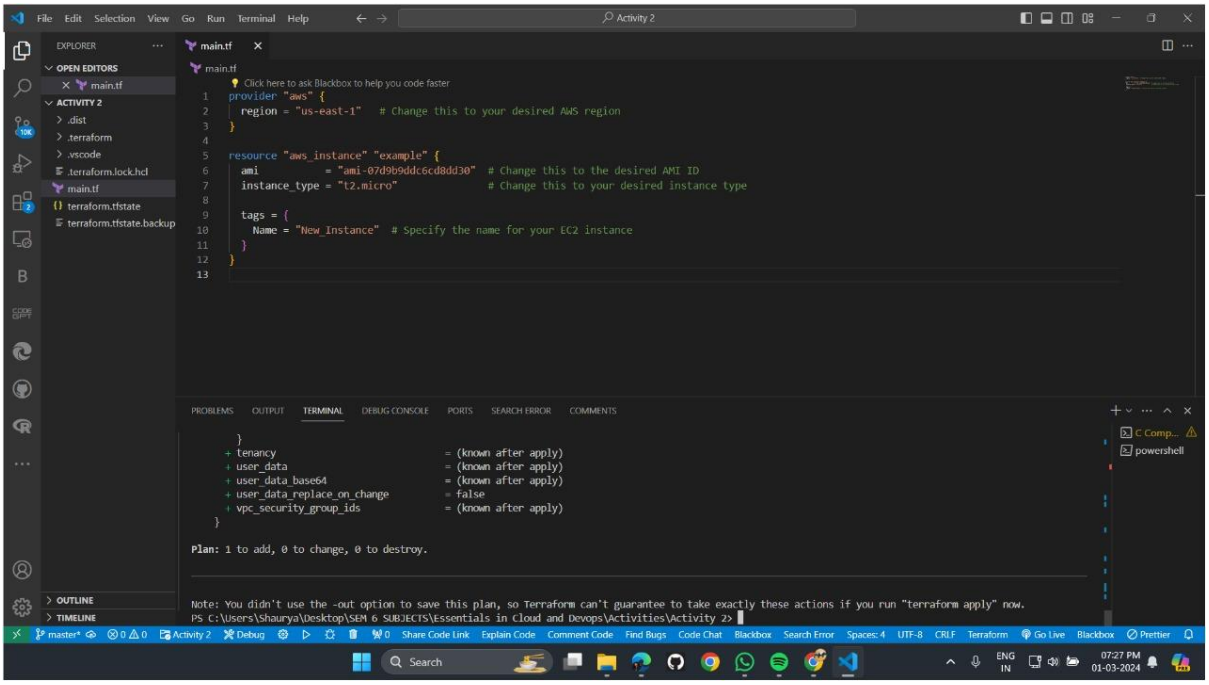
1. EC2 instance dashboard at the start:



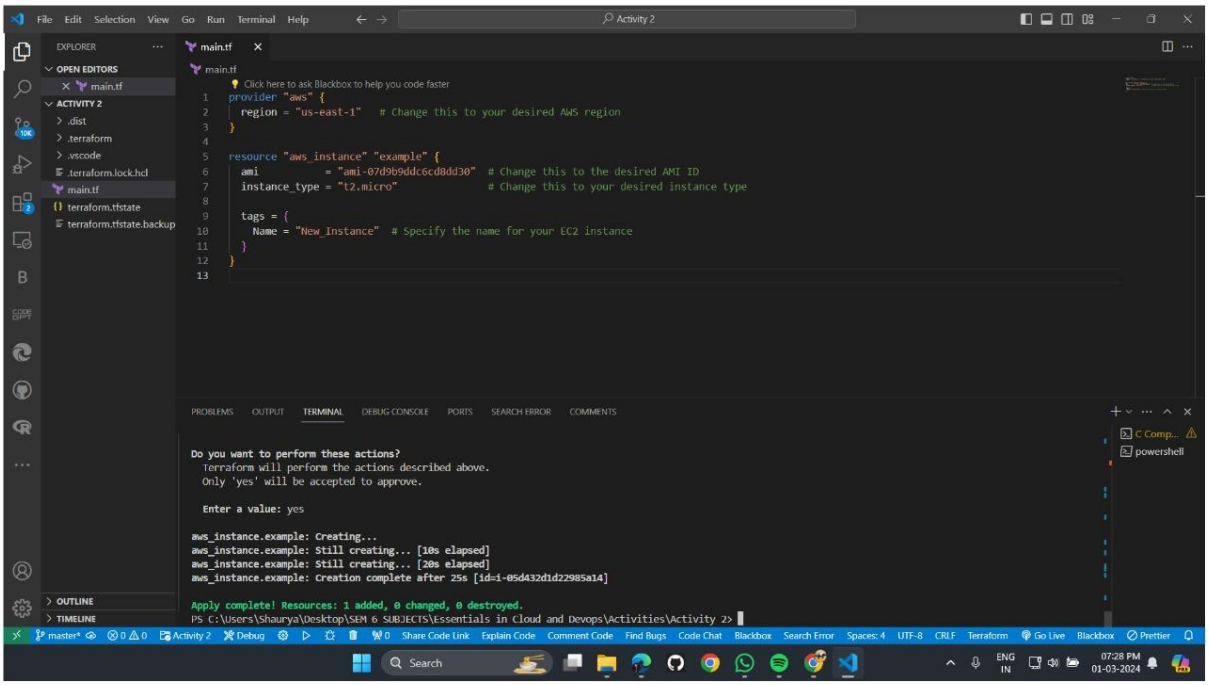
2. terraform init command:



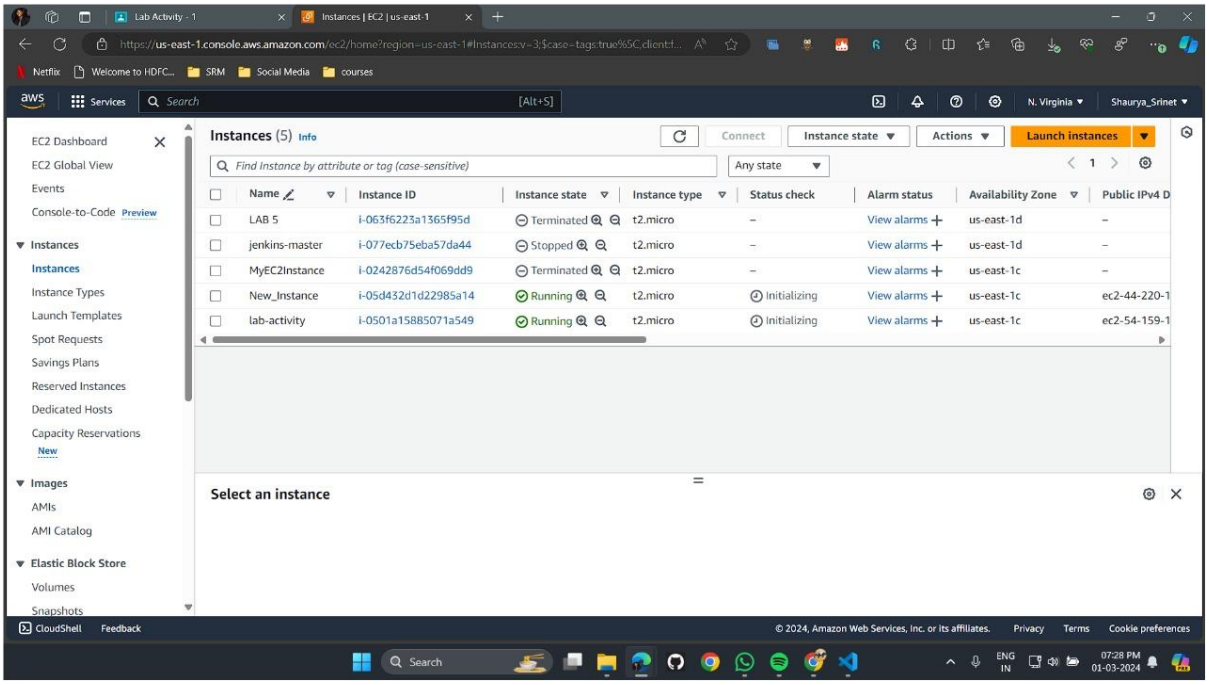
3. terraform plan command:



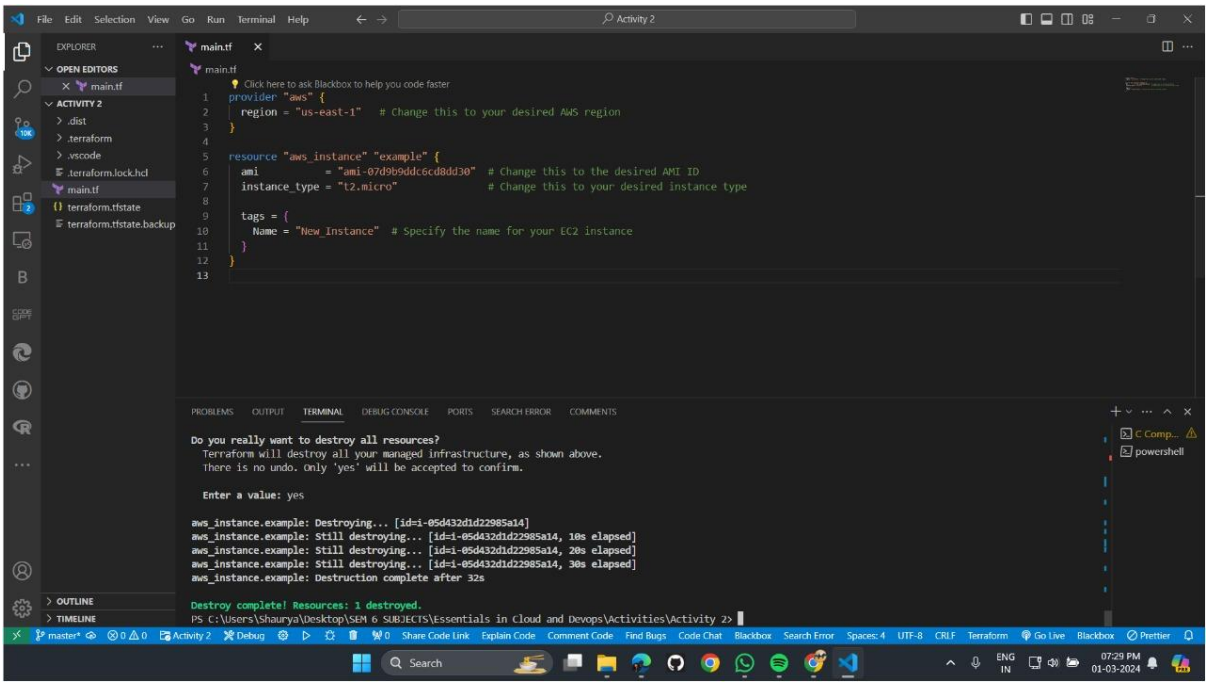
4. terraform apply command:



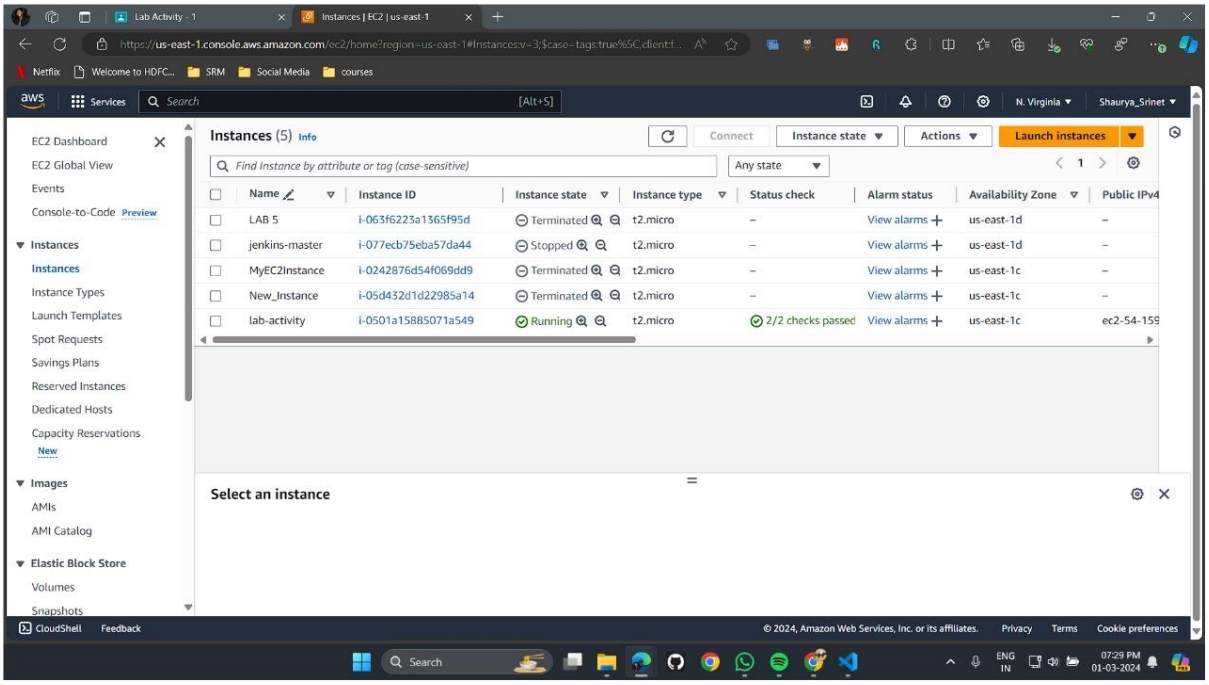
5. EC2 Dashboard after creating the instance:



6. terraform destroy command:



7. EC2 Dashboard after destroying the instance:



CREATE A VPN USING TERRAFORM

Procedure:

1. Install Terraform: If you haven't already, download and install Terraform from the official website: https://www.terraform.io/downloads.html

2. Set up AWS Credentials: Ensure you have AWS access key ID and secret access key with the required permissions to create VPN resources. You can set these up via AWS IAM.

3. Create a Terraform Configuration File: Open a text editor and create a new file named `main.tf`.

4. Write Terraform Configuration:

```
# Define AWS provider
provider "aws" {
  region = "us-east-1"   # Change this to your desired AWS region
}

# Create a Virtual Private Gateway
resource "aws_vpn_gateway" "example" {
  vpc_id = "vpc-0ba4add071b4c0bd9"  # Replace this with your actual VPC ID

  tags = {
    Name = "MyVPNGateway"  # Specify the name for your VPN gateway
  }
}

# Create a Customer Gateway
resource "aws_customer_gateway" "example" {
  bgp_asn    = 65000                 # Replace this with your network's BGP
ASN
  ip_address = "54.197.194.197"         # Replace this with your network's
public IP address
  type       = "ipsec.1"             # The type of VPN connection (ipsec.1 for
IPSec)
}

# Create a VPN Connection
resource "aws_vpn_connection" "example" {
  customer_gateway_id = aws_customer_gateway.example.id
  vpn_gateway_id      = aws_vpn_gateway.example.id
```

```
  type              = "ipsec.1"      # The type of VPN connection (ipsec.1 for
IPSec)
  static_routes_only = false          # Whether the VPN connection uses static
routes only

  tags = {
    Name = "MyVPNConnection"  # Specify the name for your VPN connection
  }
}
```

Replace `"vpc-12345678"` with your VPC ID. Modify the `bgp_asn`, `ip_address`, and other parameters in the `aws_customer_gateway` resource according to your network setup.

5. Save the File: Save the `main.tf` file in a directory of your choice.

6. Initialize Terraform: Open a terminal or command prompt in the directory where your `main.tf` file is located and run: **terraform init**

7. Review and Apply Changes: Before applying the changes, review what Terraform plans to do by running: **terraform plan**

If everything looks good, apply the changes: **terraform apply**

Type `**yes**` when prompted to confirm.

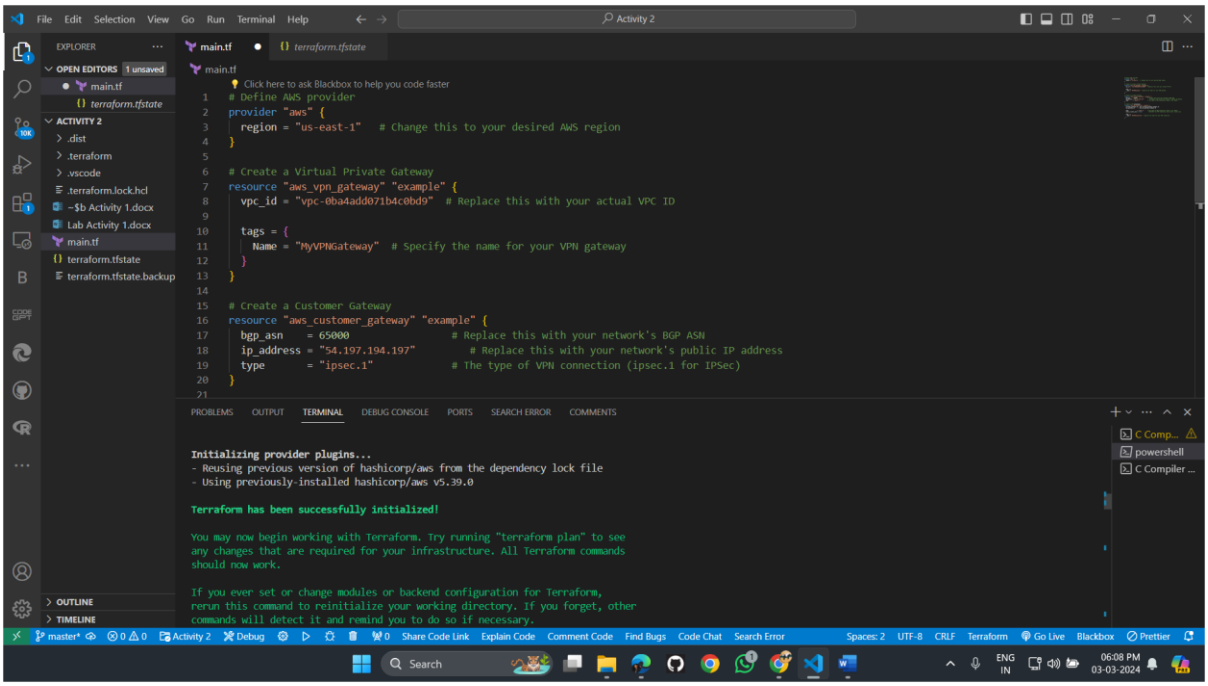
8. Verify: Once the apply command completes successfully, verify that your VPN resources are created by logging into the AWS Management Console, navigating to the VPC service, and checking the VPN connections and gateways.

9. Clean Up: To clean up resources created by Terraform, you can run: **terraform destroy**

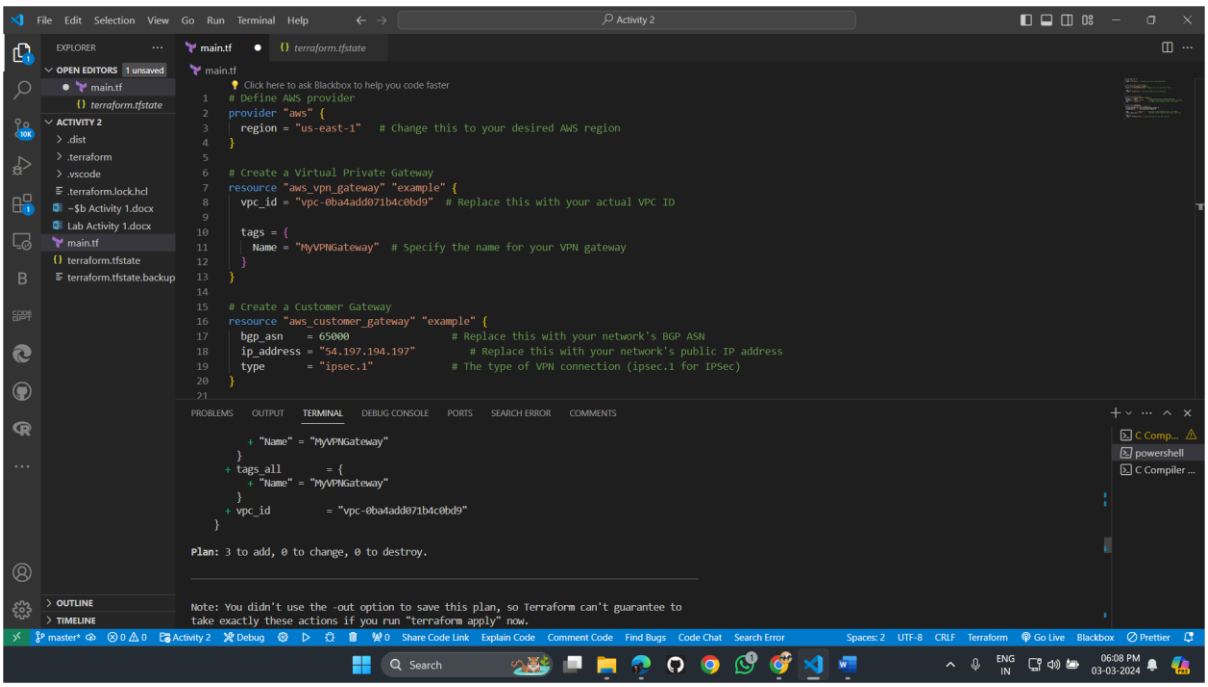
This command will remove all the resources defined in your Terraform configuration.

Output Screenshots:

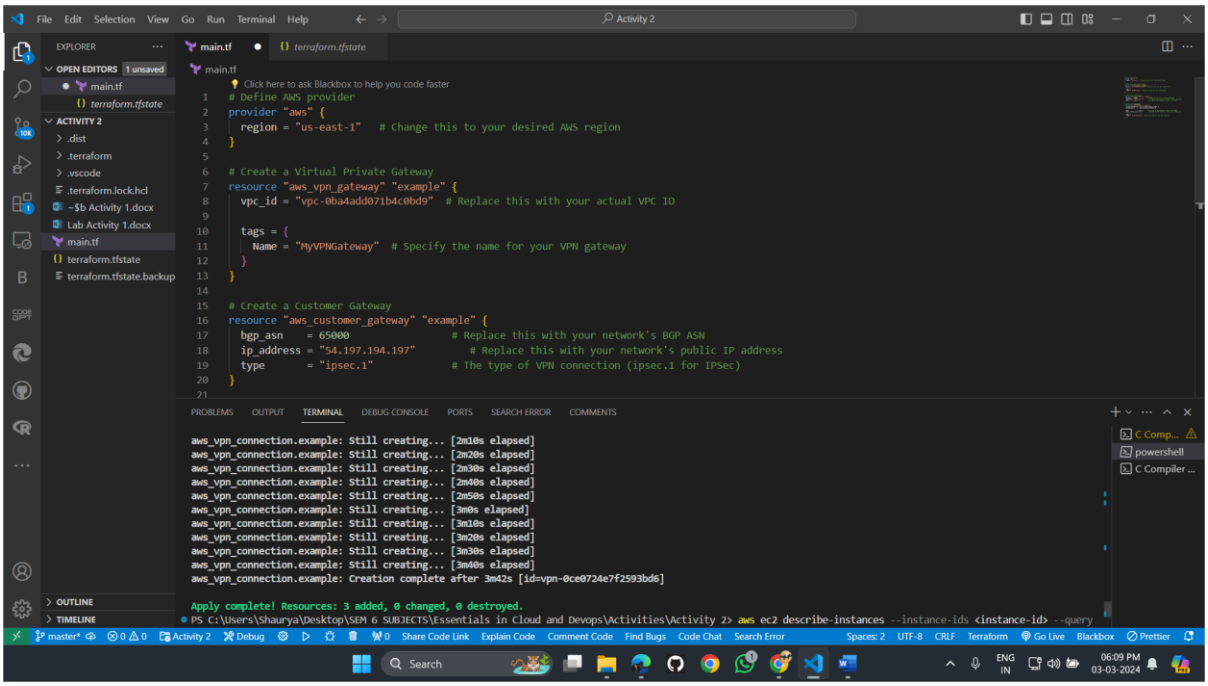
1. terraform init command:



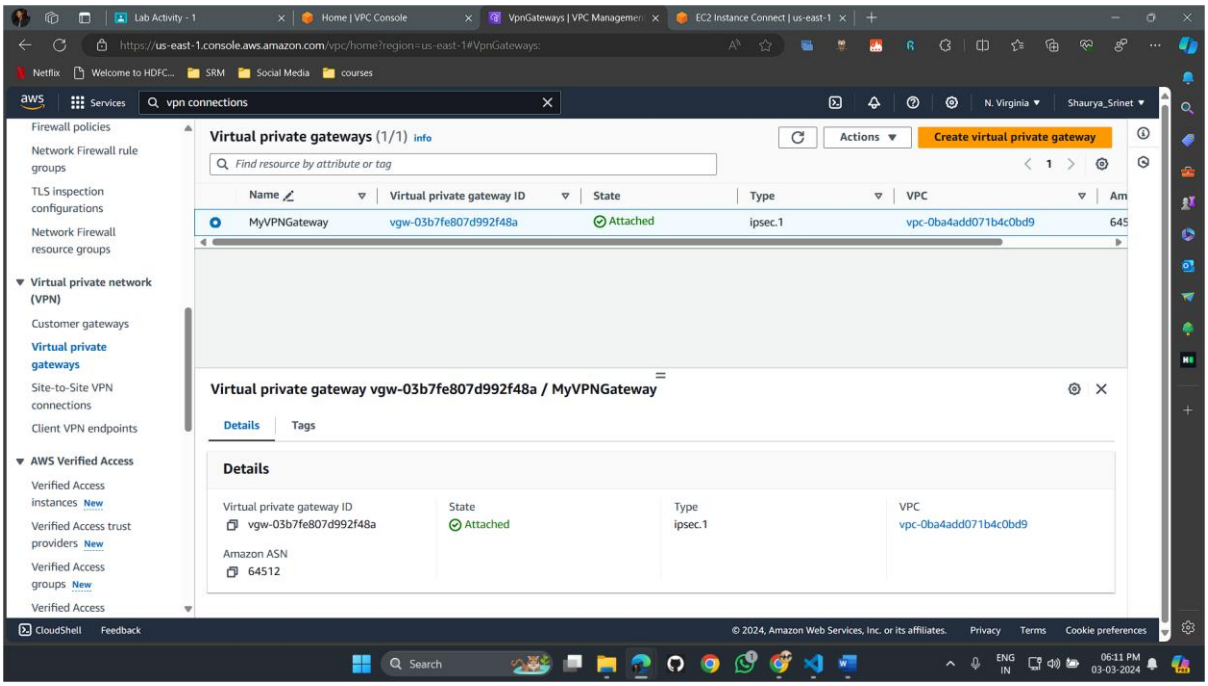
2. terraform plan command:



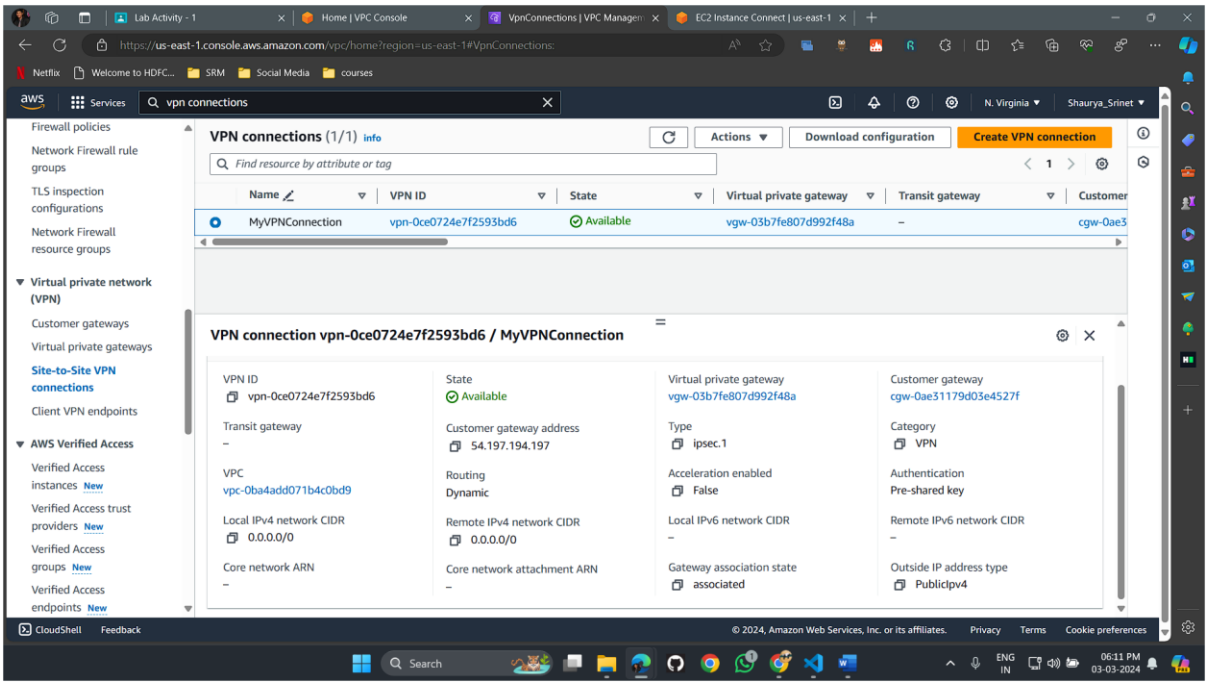
3. terraform apply command:



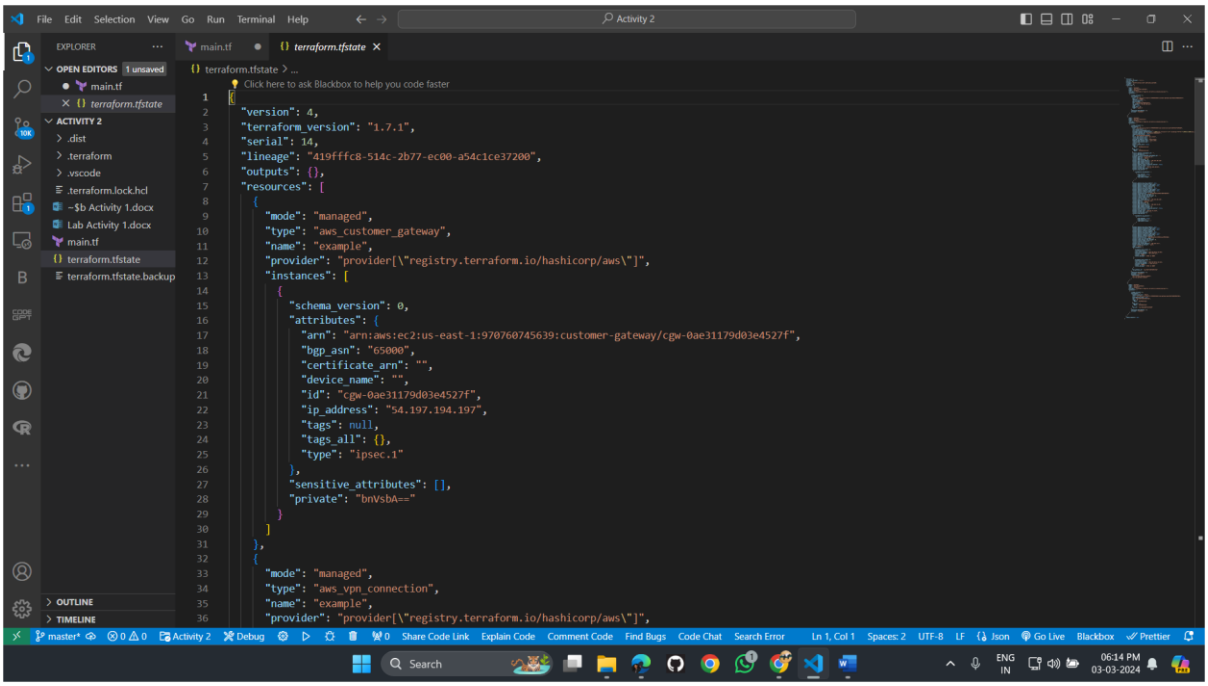
4. Virtual Private Gateways Dashboard after creating:



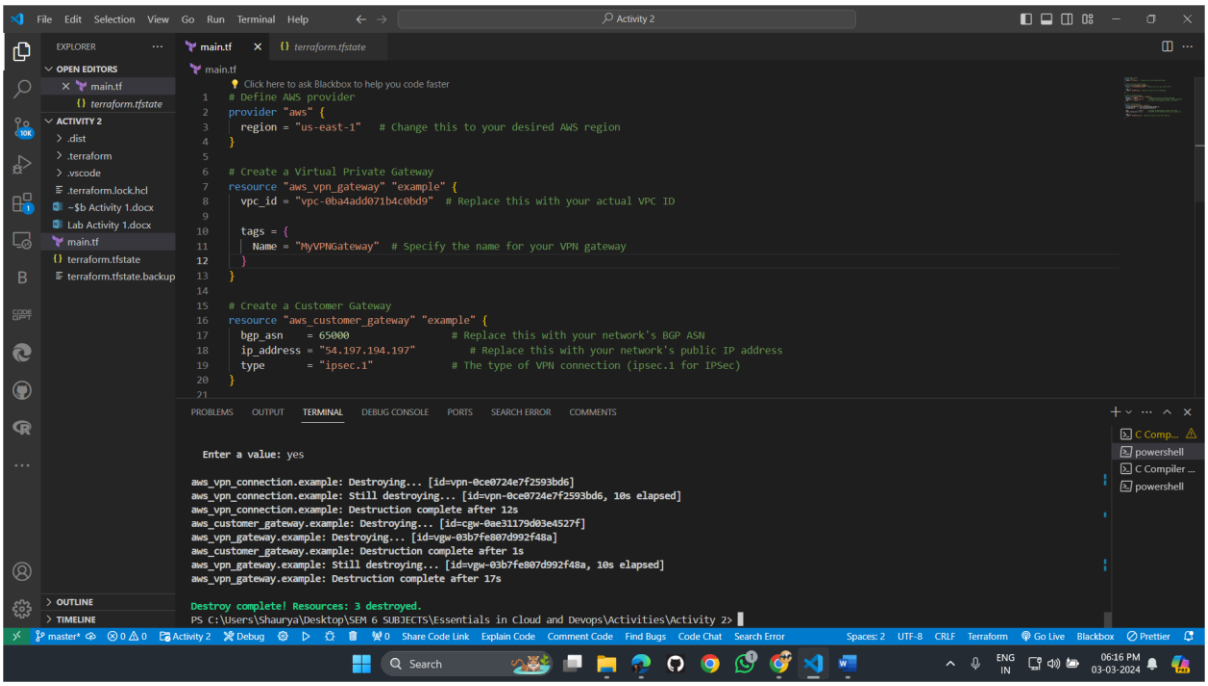
5. VPN Connections Dashboard after creating:



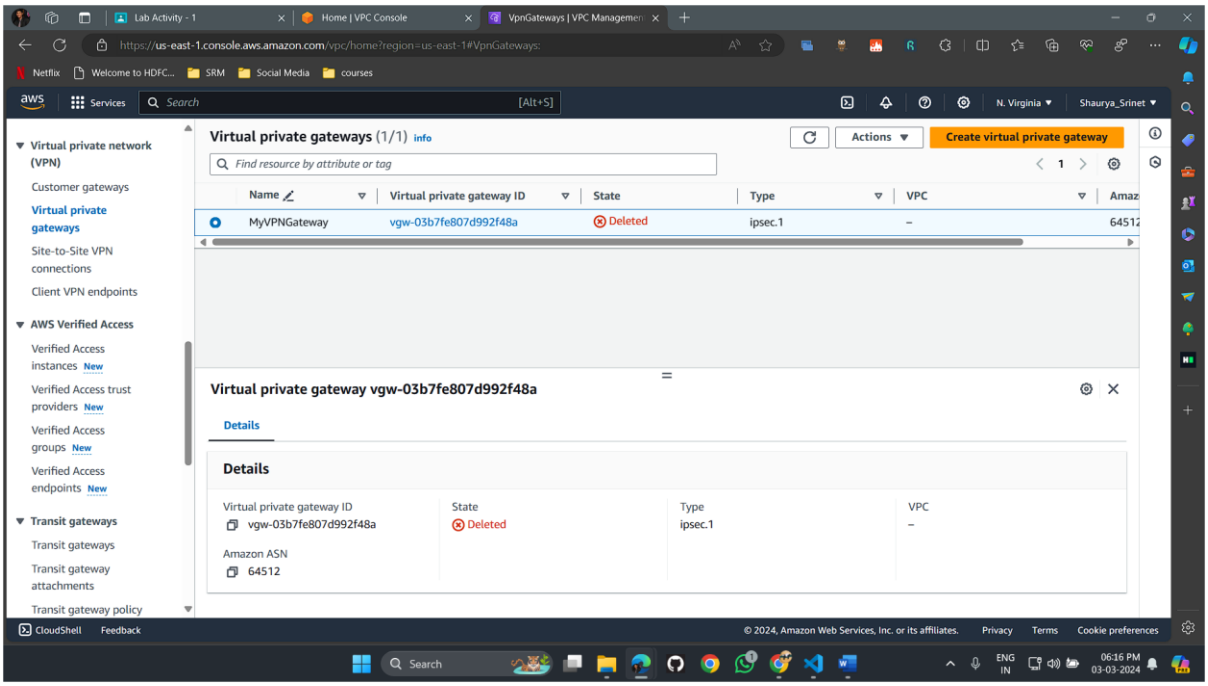
6. Terraform State File (terraform.tfstate):



7. terraform destroy command:



8. Virtual Private Gateways Dashboard after destroying:



9. VPN Connections Dashboard after destroying: