ASSIGNMENT 1

Write Terraform script to do perform following tasks on AWS cloud Platform

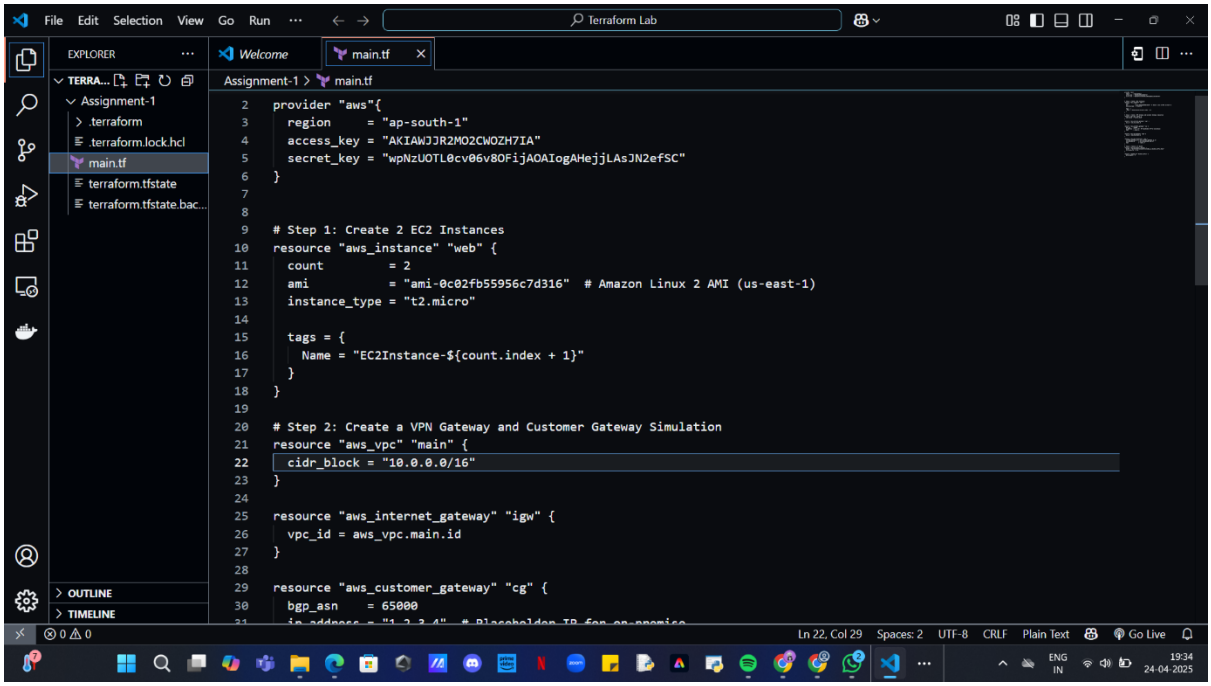
Step 1: Create two T2 Micro EC2 Instances.

Step2: Create a VPN on AWS

Step 3: Create a S3 Bucket

Step 4: Write the code for step 1,2 and 3 in a IaC terraform file and run terraform commands to execute these steps.

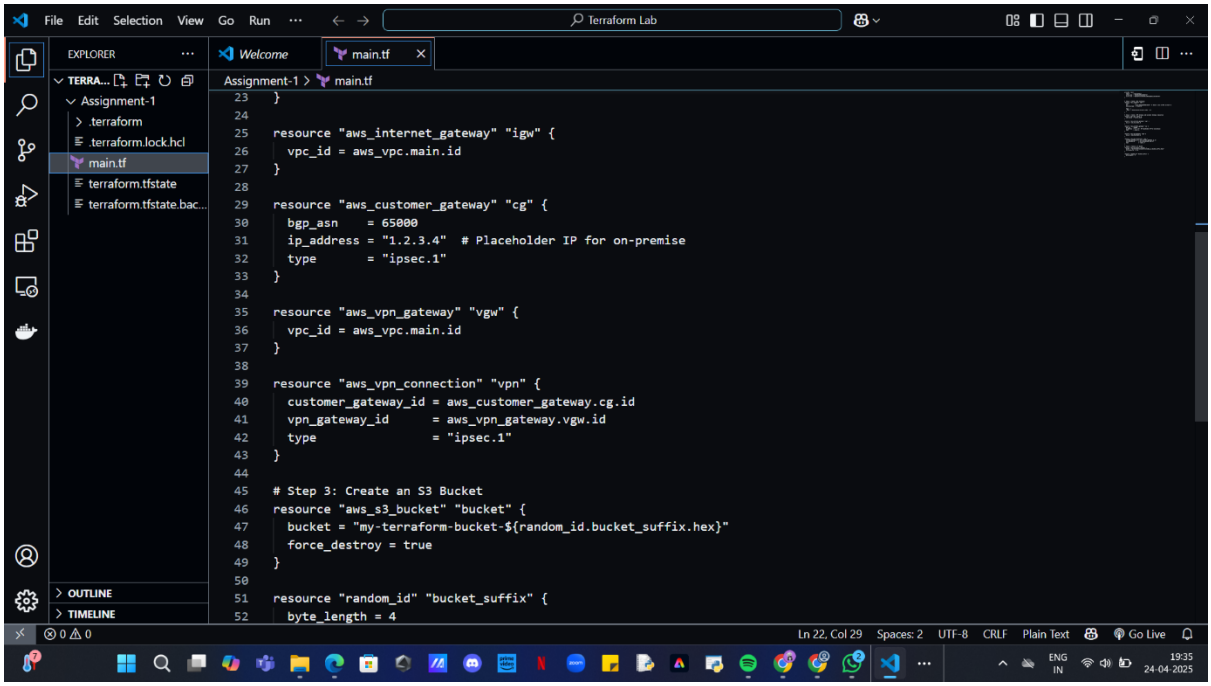
- **Codes for step 1, step 2 and step 3 :** These codes will create all the three AWS entities in one go……



Purpose: Creates **2 t2.micro EC2 instances** using a basic Amazon Linux 2 AMI.

count = 2 means it will create two instances.

Each instance gets a different name using ${count.index} (e.g., EC2Instance-1, EC2Instance-2).



EXPLAINATION OF CODE –

☁️ Step 1: EC2 Instances

hcl

CopyEdit

```
resource "aws_instance" "web" {
  count         = 2
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"


  tags = {
    Name = "EC2Instance-${count.index + 1}"
  }
}
```

- **Purpose:** Creates **2 t2.micro EC2 instances** using a basic Amazon Linux 2 AMI.
- **count = 2** means it will create two instances.
- Each instance gets a different name using ${count.index} (e.g., EC2Instance-1, EC2Instance-2).

🔐 Step 2: VPN Setup

Create a VPC (Virtual Private Cloud)

hcl

CopyEdit

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
```

- **Purpose:** A private network in AWS where your resources (like EC2, VPN) will live.

Internet Gateway

h

CopyEdit

```
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
}
```

- **Purpose:** Allows the VPC to connect to the internet.

Customer Gateway (Simulated)

hcl

CopyEdit

```
resource "aws_customer_gateway" "cg" {
  bgp_asn    = 65000
  ip_address = "1.2.3.4"  # Fake on-prem IP
  type       = "ipsec.1"
}
```

- **Purpose:** Represents your **on-premises network gateway**.
- This IP is usually from your office/home router but is simulated here.

VPN Gateway

hcl

CopyEdit

```
resource "aws_vpn_gateway" "vgw" {
  vpc_id = aws_vpc.main.id
}
```

- **Purpose:** AWS-side VPN device to connect with the customer gateway.

VPN Connection

hcl

CopyEdit

```
resource "aws_vpn_connection" "vpn" {
  customer_gateway_id = aws_customer_gateway.cg.id
  vpn_gateway_id      = aws_vpn_gateway.vgw.id
  type                = "ipsec.1"
}
```

- **Purpose:** Establishes the **VPN tunnel** between AWS and the customer gateway.

🪣 Step 3: S3 Bucket

hcl

CopyEdit

```
resource "random_id" "bucket_suffix" {
  byte_length = 4
}

resource "aws_s3_bucket" "bucket" {
  bucket = "my-terraform-bucket-${random_id.bucket_suffix.hex}"
  force_destroy = true
}
```

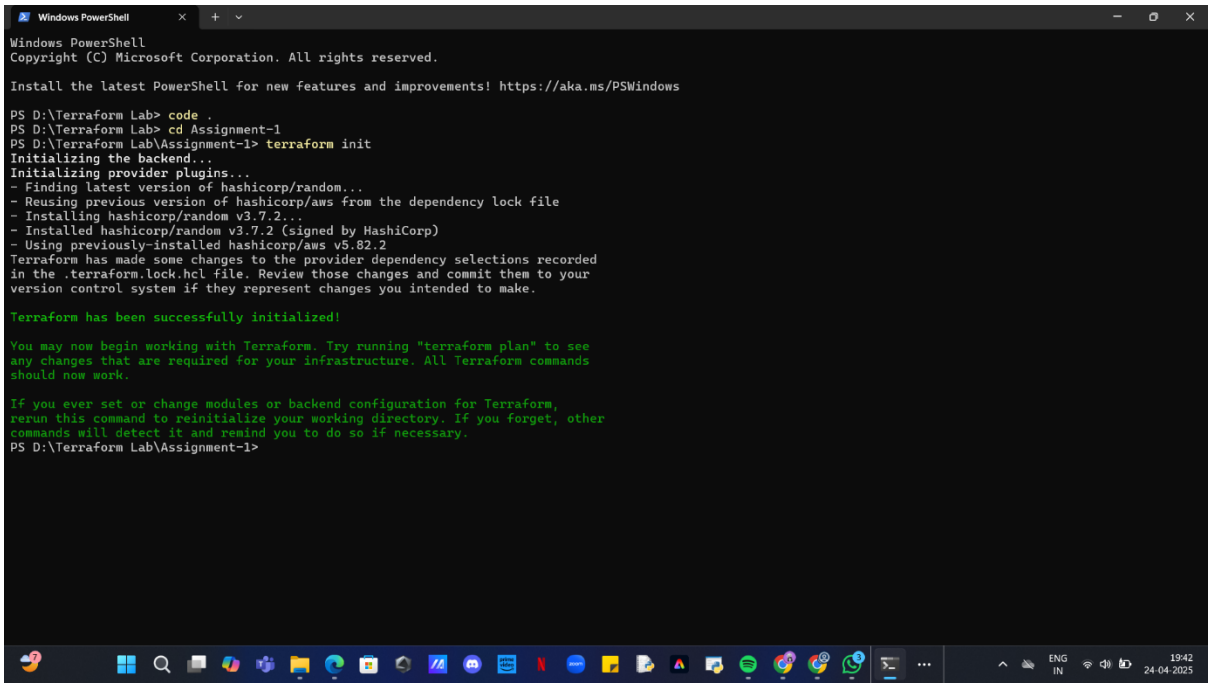
- **Purpose:** Creates a **unique S3 bucket** using a random 4-byte hex suffix.
- force_destroy = true ensures the bucket can be deleted even if it has files.

🧪 Step 4: Running Terraform

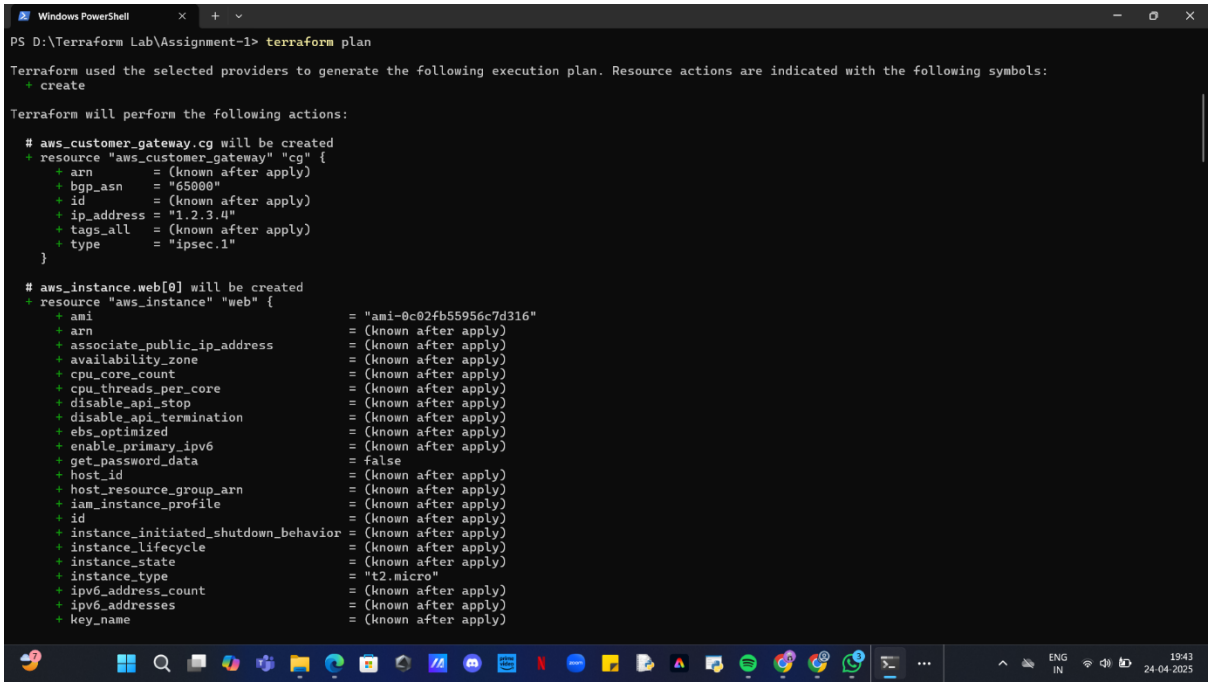
- terraform init – Sets up Terraform and downloads plugins.
- terraform plan – Previews what will be created.
- terraform apply – Actually provisions the resources on AWS.

Terraform commands to run :

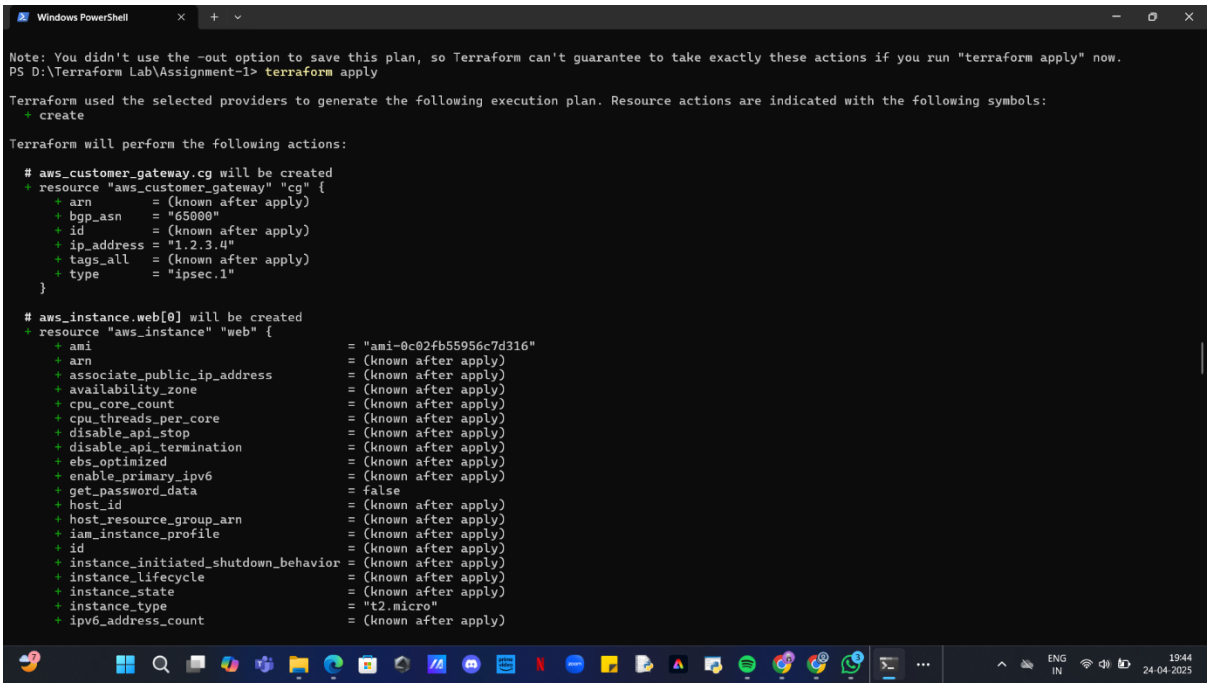
- terraform init # Initialize Terraform



- terraform plan # Preview the changes



- terraform apply # Apply and create resources



ALL THE RESOURCES SUCCESSFULLY CREATED