Assignment–9

# Module-12: Terraform Overview

Submitted by : Shaik Junaid Adil

Date of Submission: 12-07-2024

Submitted to: Vikul

L1 - Provision AWS EC2 Instance along with VPC, Subnet, Internet Gateway, Route-Table, Route☐Table Association, Security Group

Step 1: To provision AWS EC2 Instance using Terraform, install Terraform in local machine and setup the environment variable.
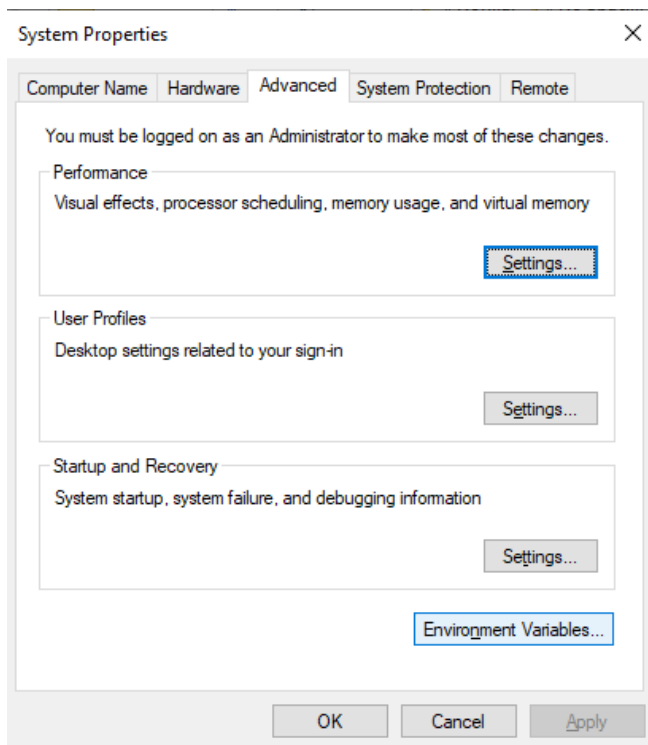
Using link to download: https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli
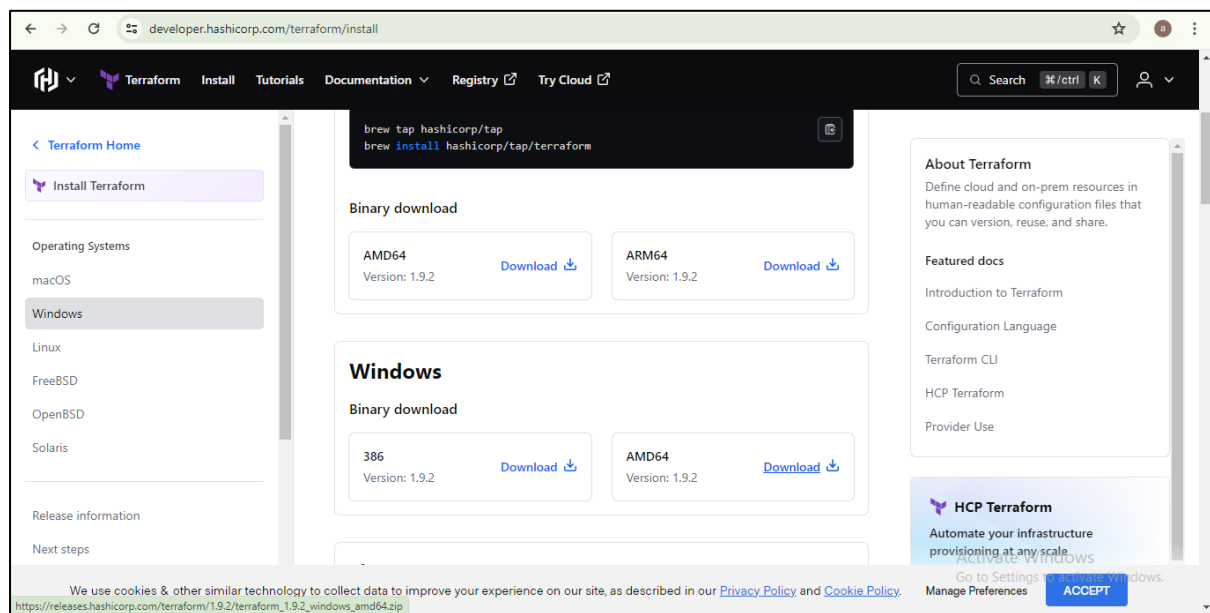
It will download a zip file. Extract the data to a separate folder.
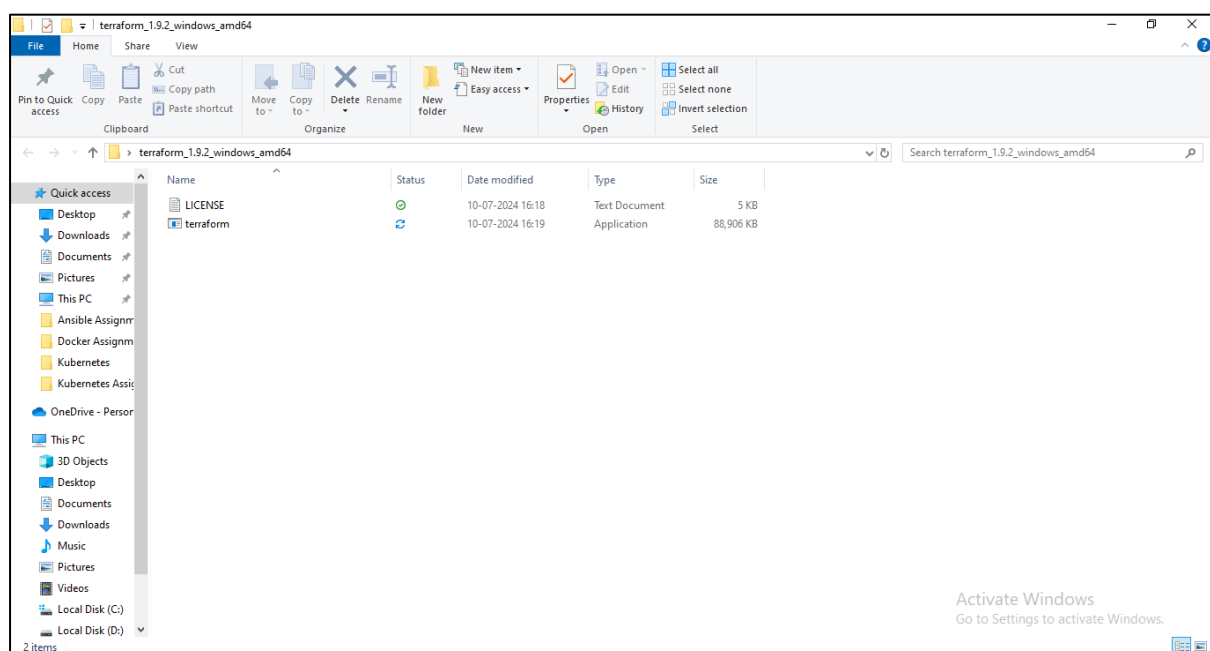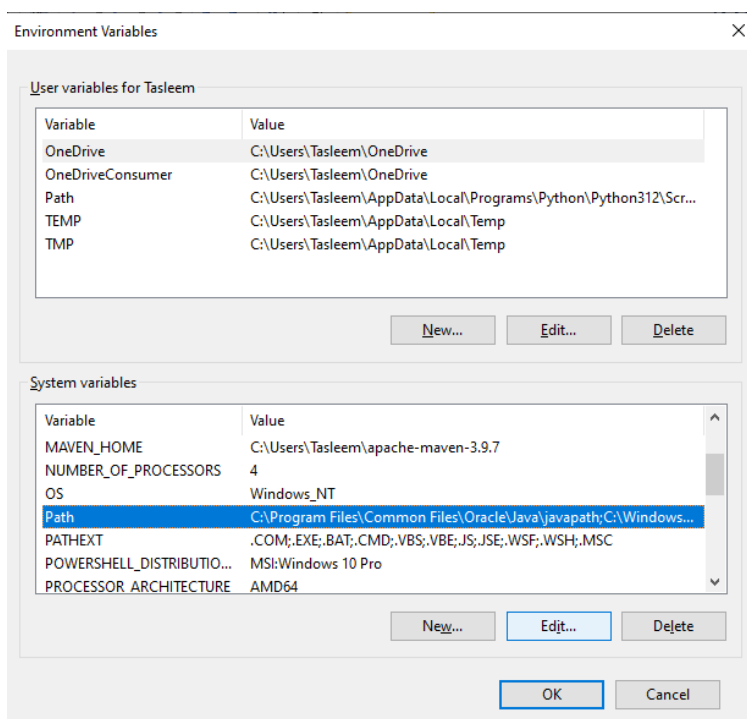
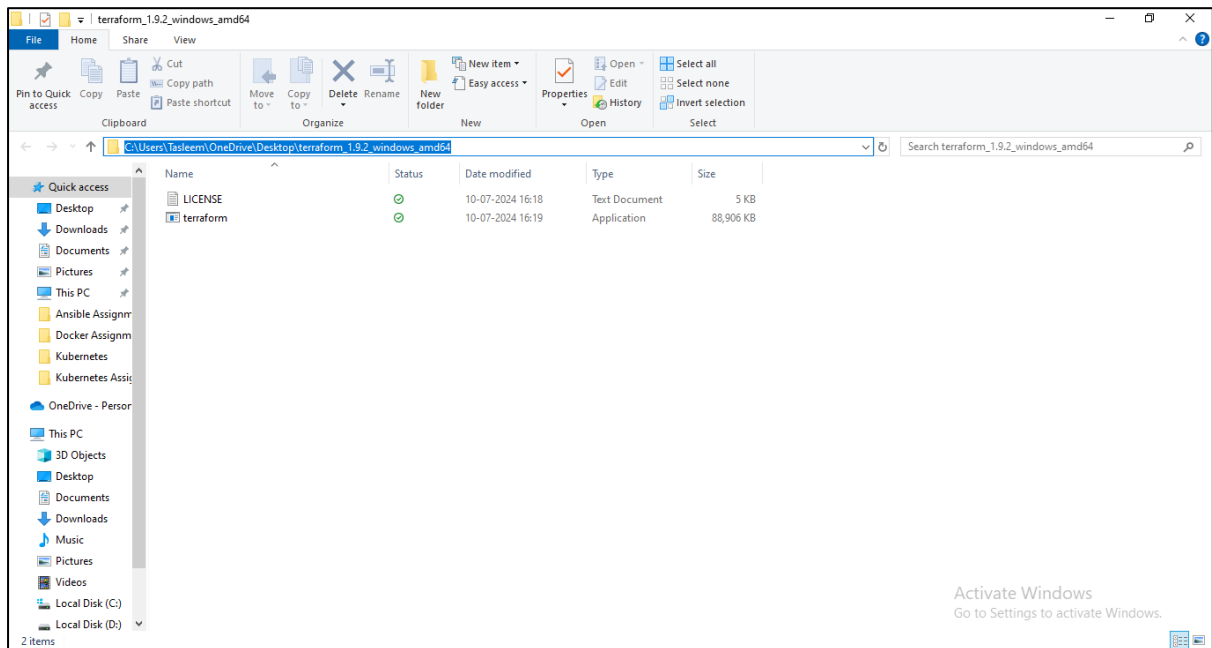**Step 2:** Set the environment variable for Terraform using the exec file path

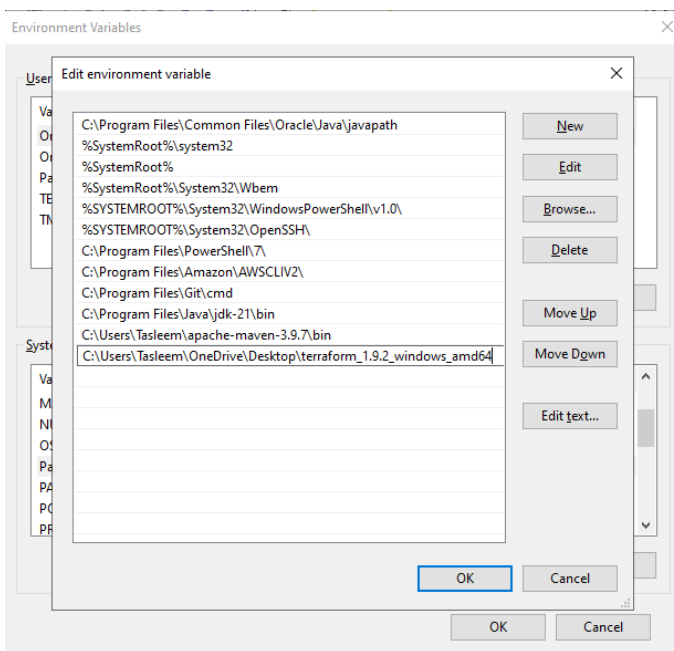Go to edit the system environment variable → advanced → Environment Variables..

In system variables → path → edit the path →

Step 3: Now add the exec path in the system variables.. – path

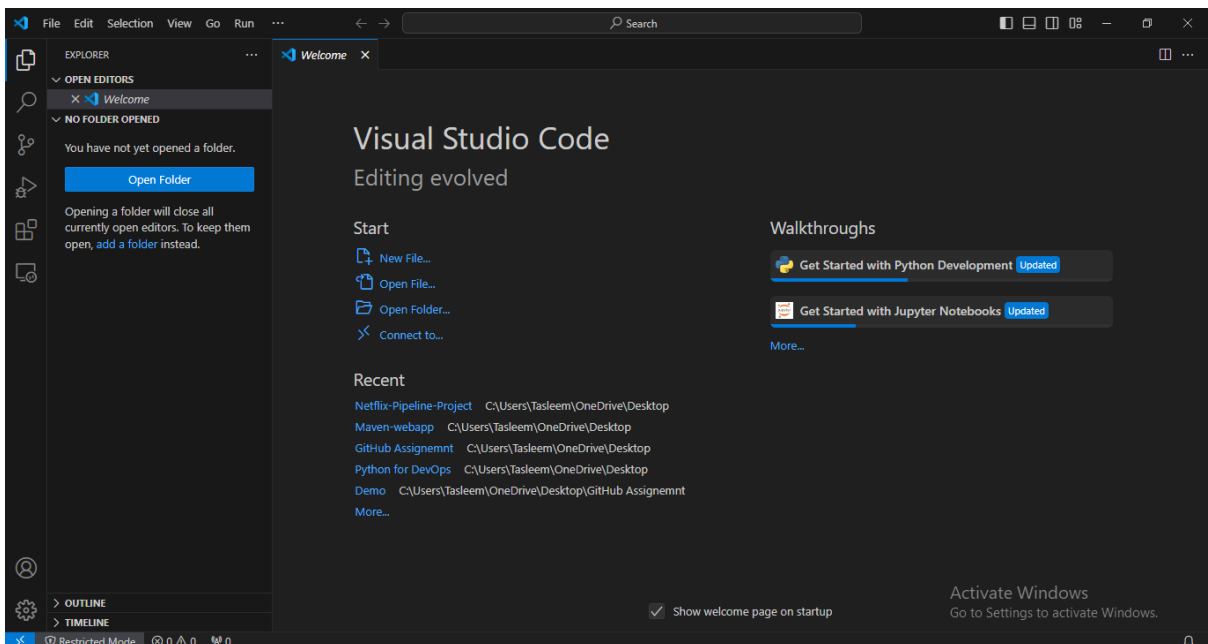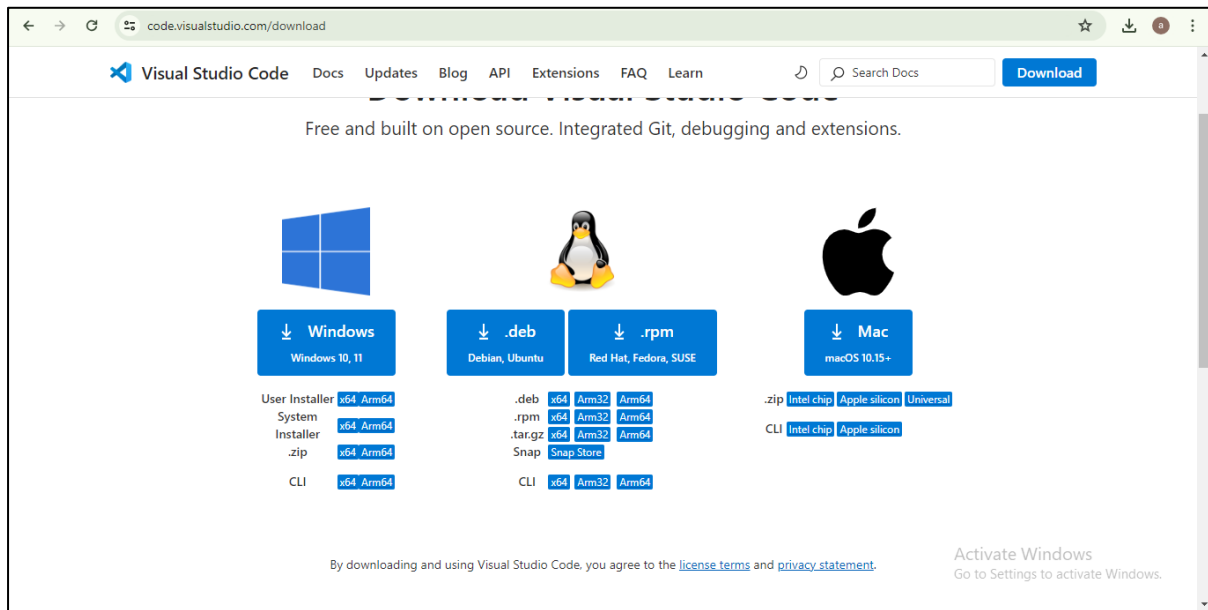**Step 4:** Copy the path of exec file and paste in the system variables path
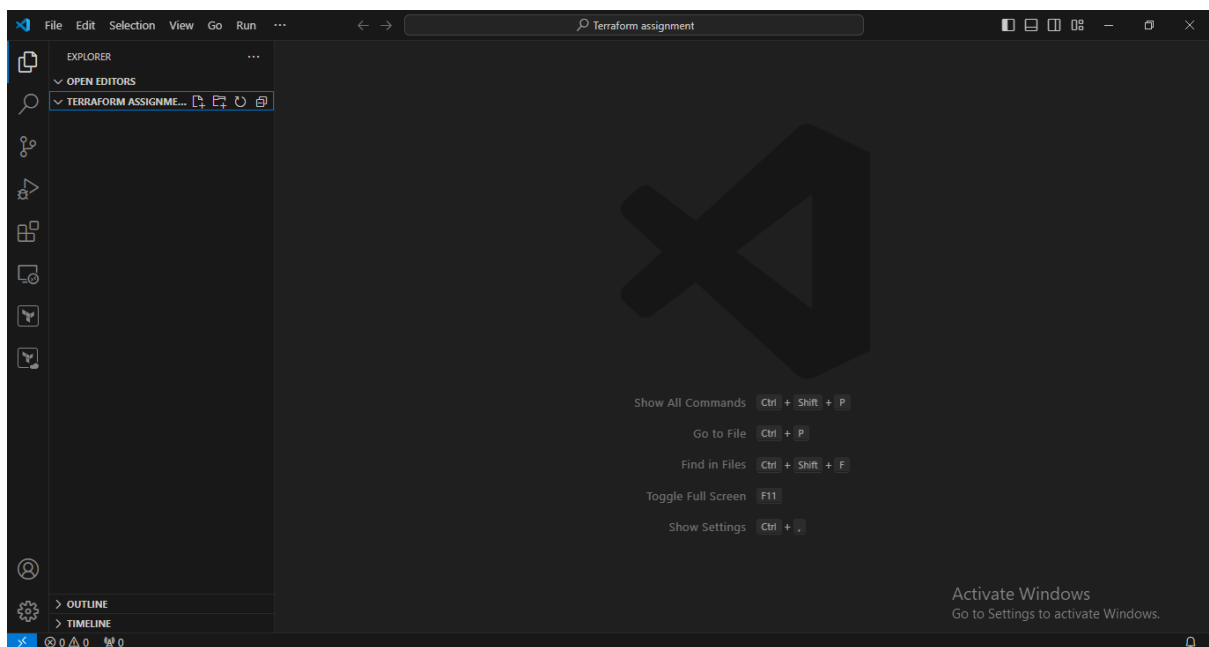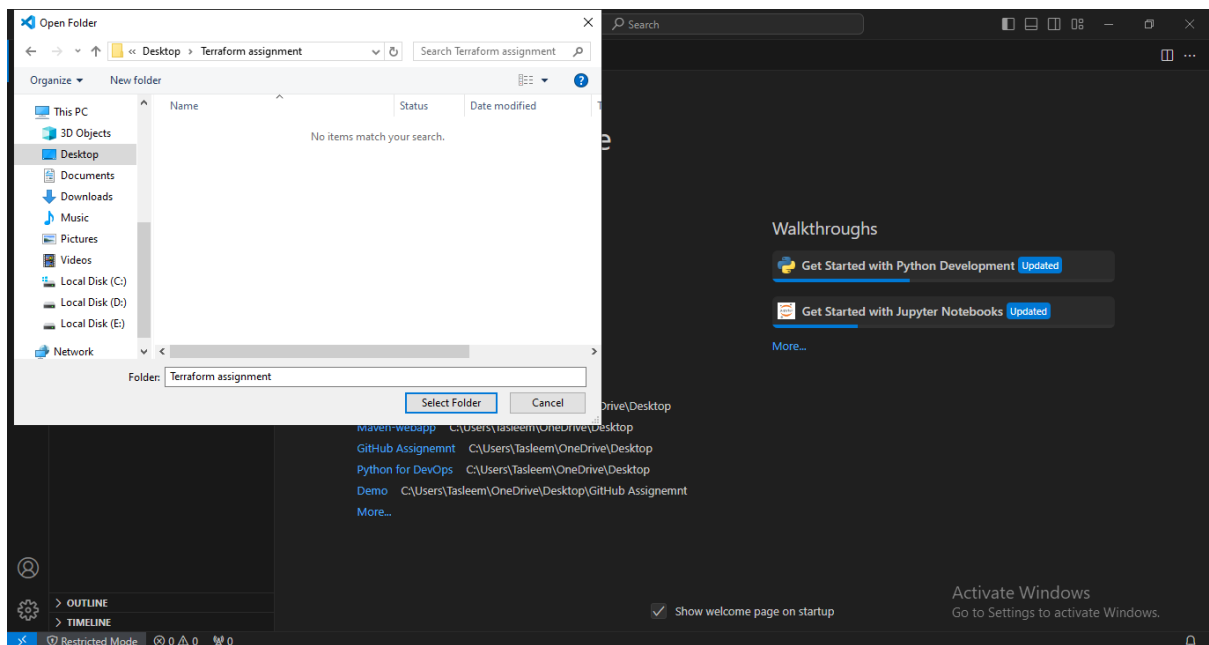

Click Ok.

Environment variable has been setup for Terraform.

Step 5: Download and Install Visual Studio Code.

Using download link: code.visualstudio.com/download

Step 6: Create a folder and open in VS Code

Step 7: Install Terraform Extension in VS Code
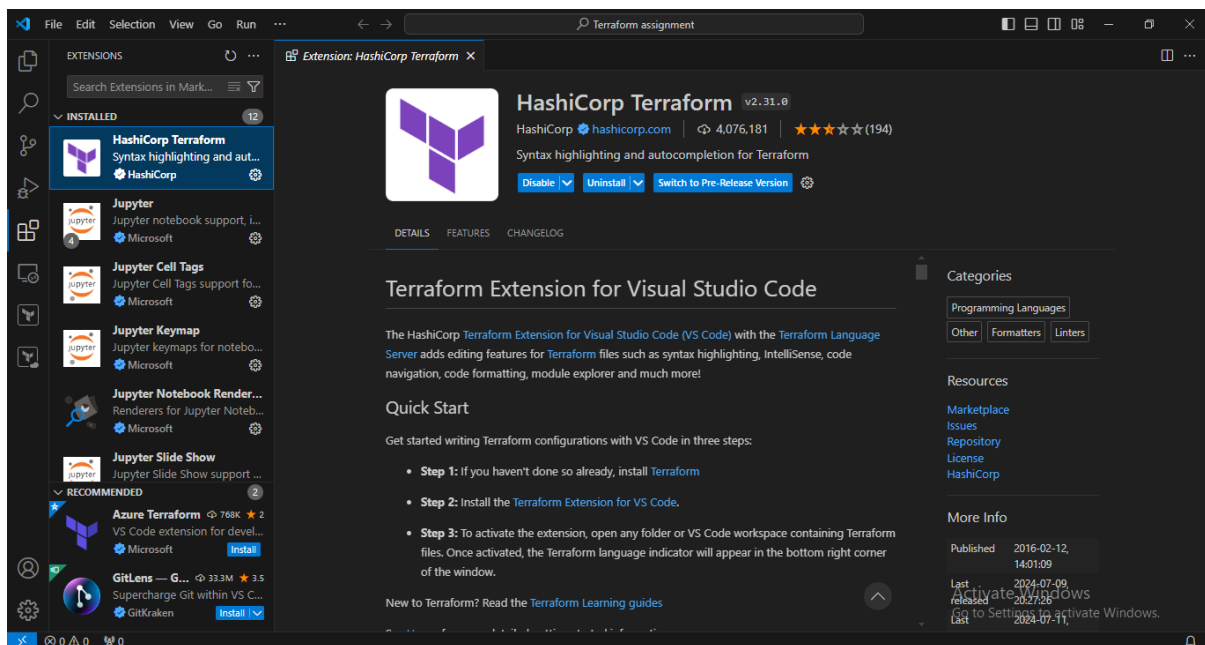
Step 8: Open the Terminal and query the Terraform version using command "terraform --version"
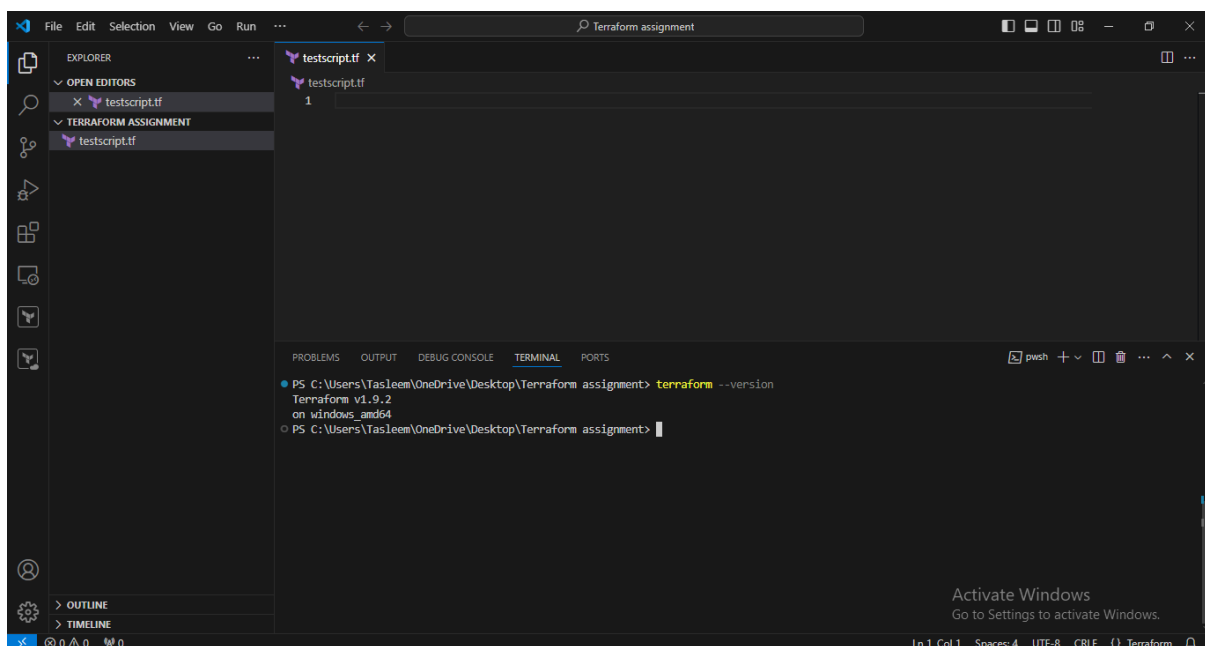
Terraform is successfully installed.

Step 9: Create a file to write the script to create an ec2-instance with .tf extension
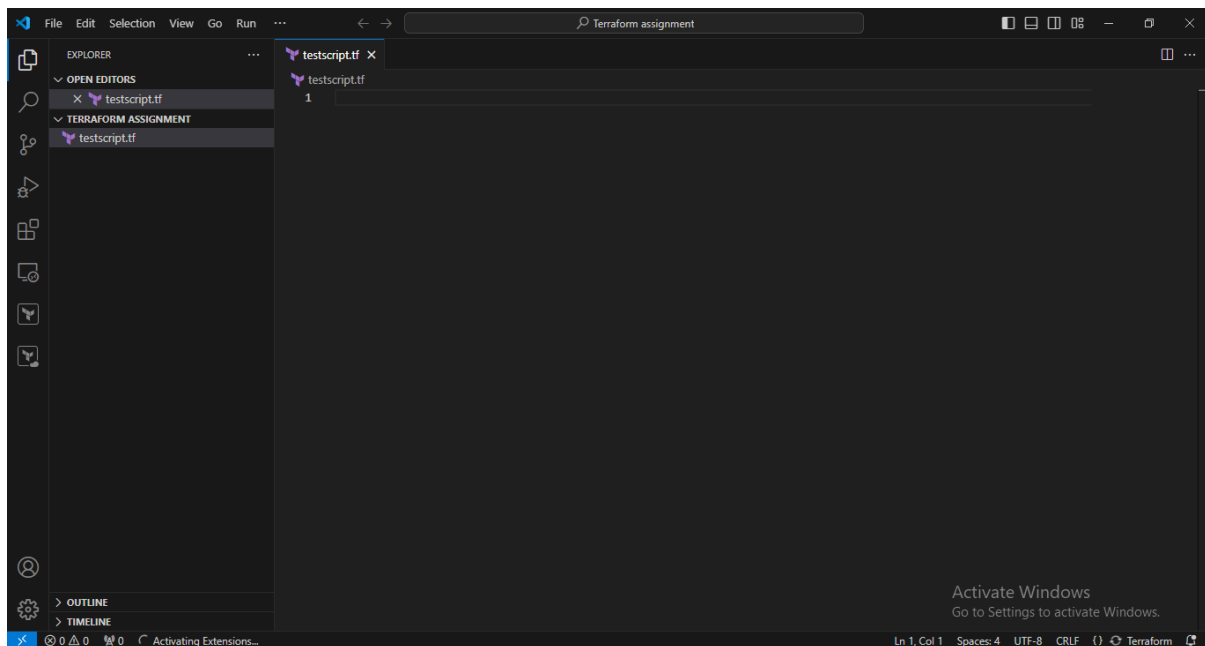
In Terraform Working Model we have 5 steps, which we follow:

1. Scope - Identify the Infrastructure
2. Author - Write Terraform script
3. Initialize - Initialize the Terraform
4. Plan - validate the script
5. Apply - Apply the changes

Step 10: Initially create Access key in AWS Console

Go to AWS Console → Login → Security credentials
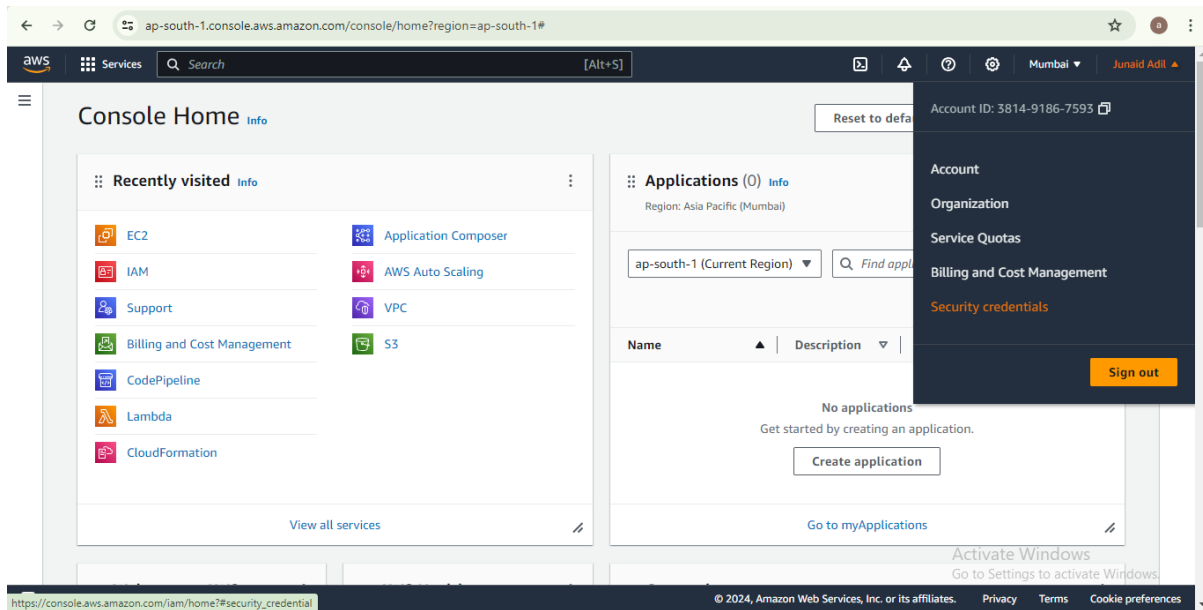
Step 11: Create access key
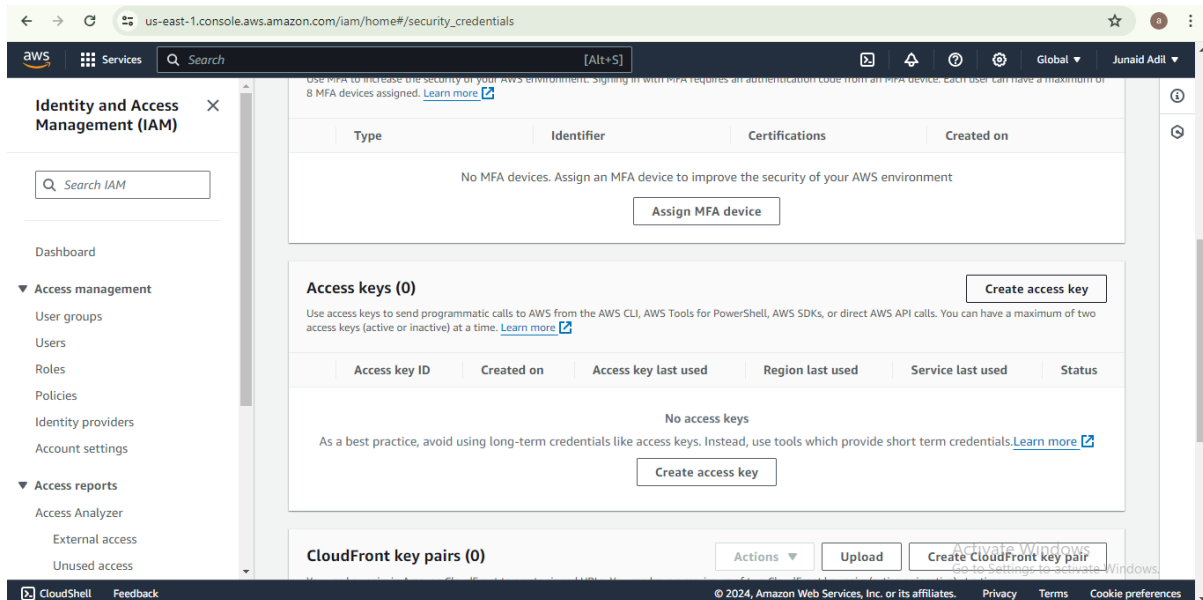
We have to save the Access keys and secret access key. Once we close the tab we can access only Access key. Or we can download the .csv file
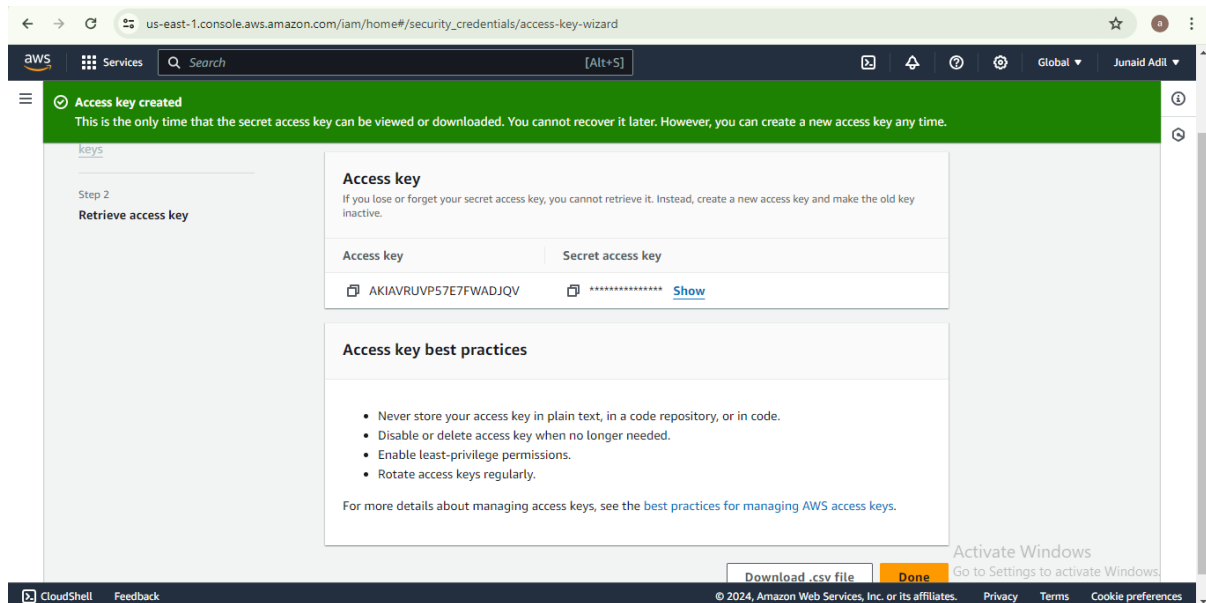
Step 12: Now writhe the script in VS code, in file "testscript.tf "
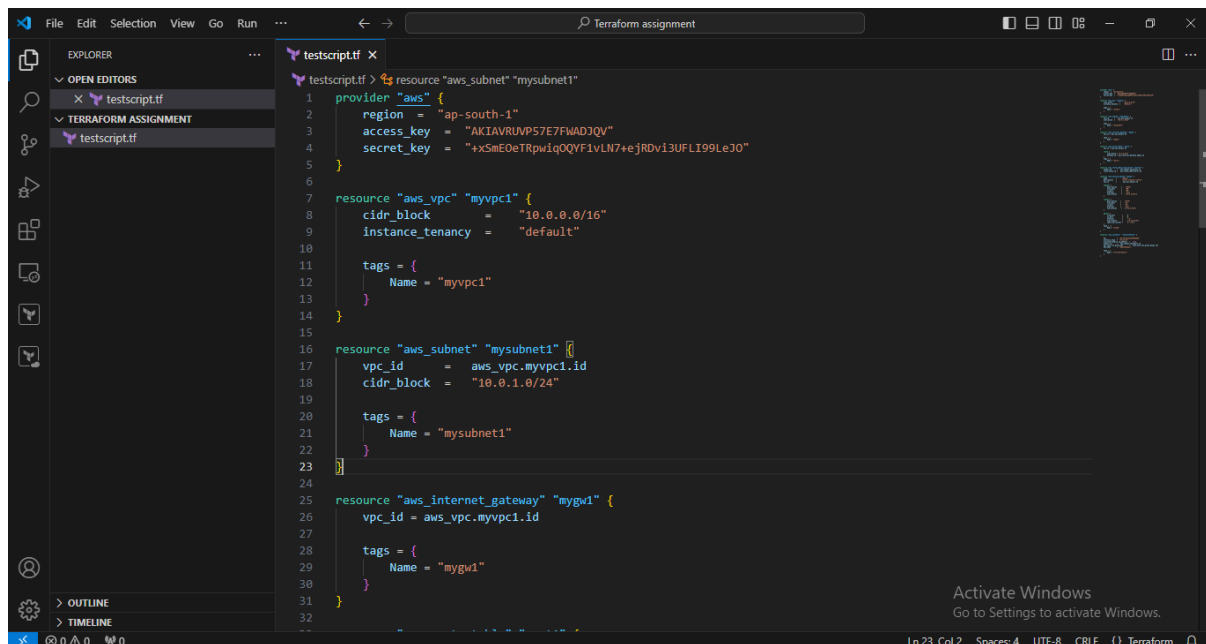
```
31    }
32
33    resource "aws_route_table" "myrt1" {
34        vpc_id = aws_vpc.myvpc1.id
35
36        route {
37            cidr_block = "0.0.0.0/0"
38            gateway_id = aws_internet_gateway.mygw1.id
39        }
40        tags = {
41            Name = "myrt1"
42        }
43    }
44
45    resource "aws_route_table_association" "myrta1" {
46        subnet_id      =    aws_subnet.mysubnet1.id
47        route_table_id =    aws_route_table.myrt1.id
48    }
49
50    resource "aws_security_group" "mysg1" {
51        name        =       "mysg1"
52        description =       "Allow inbound traffic"
53        vpc_id      =       aws_vpc.myvpc1.id
54
55        ingress {
56            description =    "HTTP"
57            from_port   =    "80"
58            to_port     =    "80"
59            protocol    =    "tcp"
60            cidr_blocks =    ["0.0.0.0/0"]
61        }
62
```
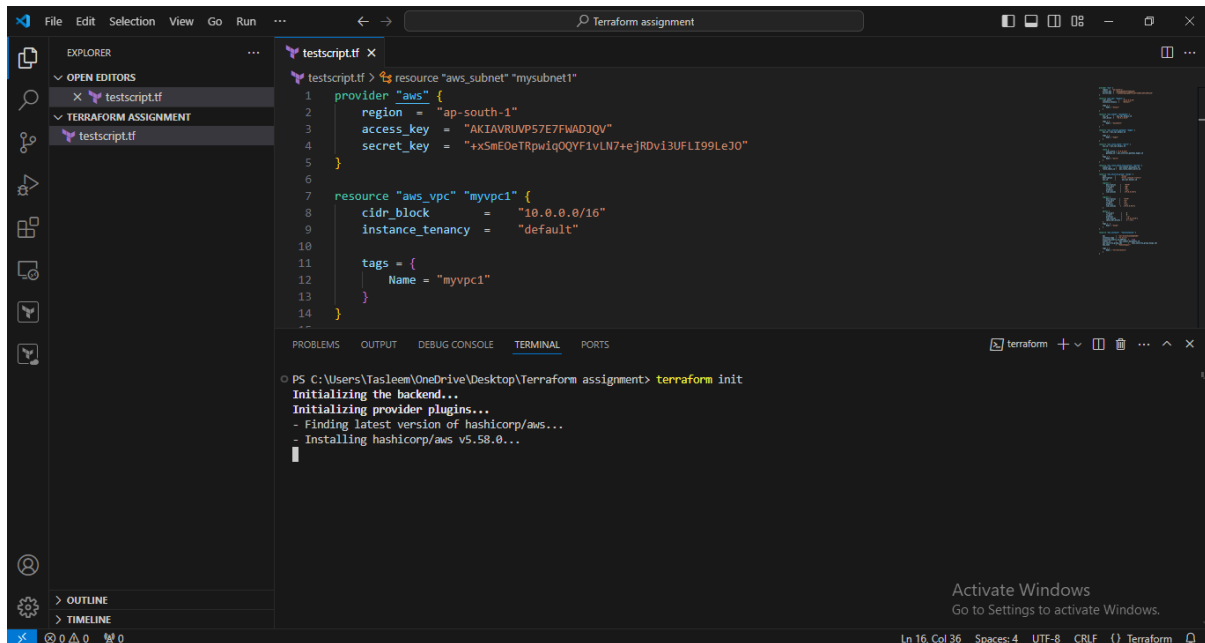


```
50    resource "aws_security_group" "mysg1" {
63        ingress {
64            description =    "HTTP"
65            from_port   =    "22"
66            to_port     =    "22"
67            protocol    =    "tcp"
68            cidr_blocks =    ["0.0.0.0/0"]
69        }
70
71        egress {
72            from_port       =    0
73            to_port         =    0
74            protocol        =    "-1"
75            cidr_blocks     =    ["0.0.0.0/0"]
76            ipv6_cidr_blocks =   ["::/0"]
77        }
78        tags = {
79            Name = "mysg1"
80        }
81    }
82
83    resource "aws_instance"  "testinstance1" {
84
85        ami             =    "ami-0c2af51e265bd5e0e"
86        instance_type   =    "t2.micro"
87        associate_public_ip_address =    true
88        subnet_id       =    aws_subnet.mysubnet1.id
89        vpc_security_group_ids    =    [aws_security_group.mysg1.id]
90        key_name        =    "Demo1keypair"
91
92        tags = {
93            Name = "Terraformtest1"
```
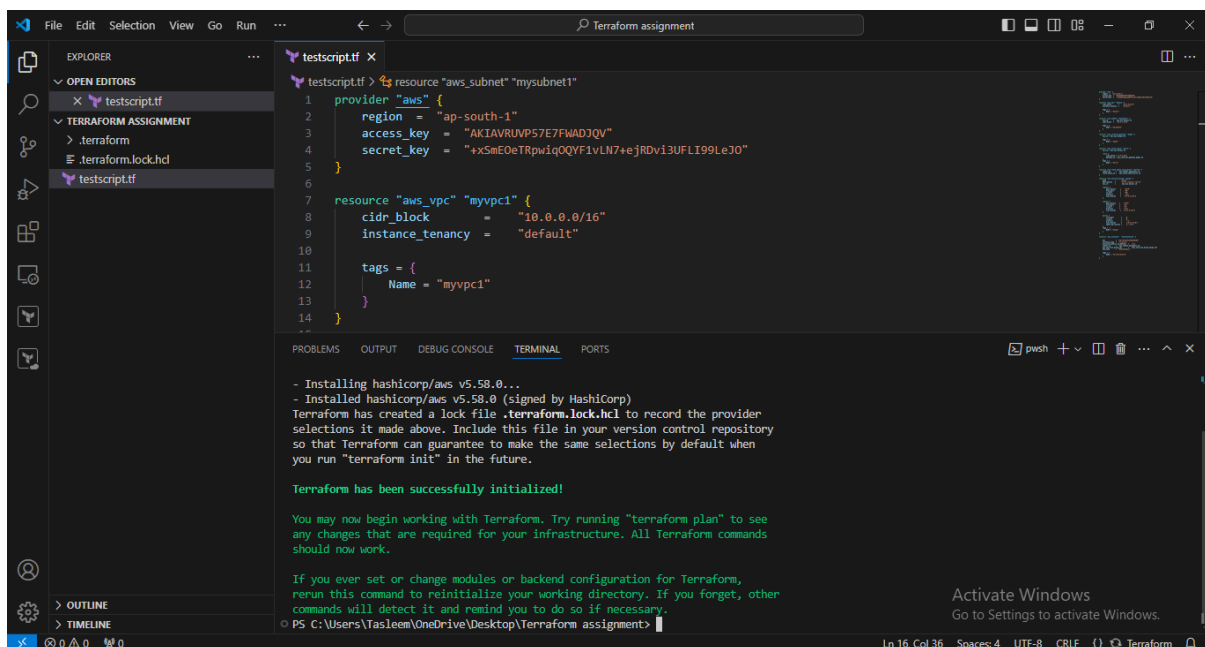
Step 13: Open new Terminal and initialize Terraform using command: "terraform init".

It creates a hidden directory and within that it maintains the latest version of terraform provider for aws.

Step 14: Execute "terraform plan" to see the changes that are required for infrastructure as per the script. It identifies the resource to be created.
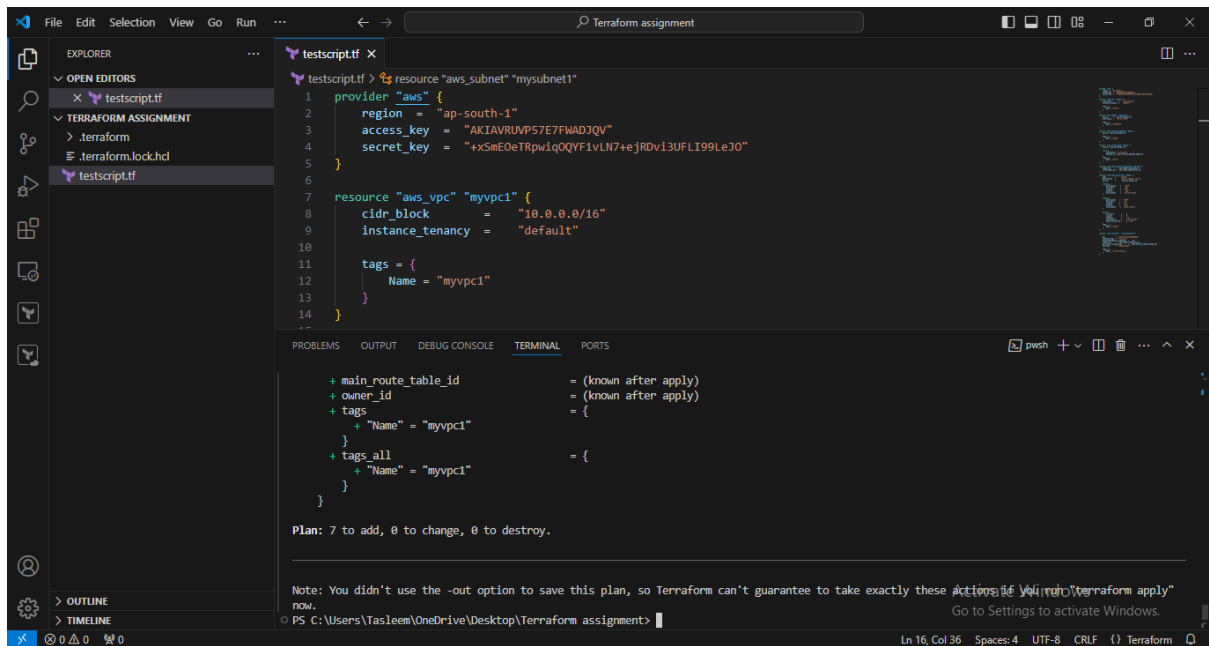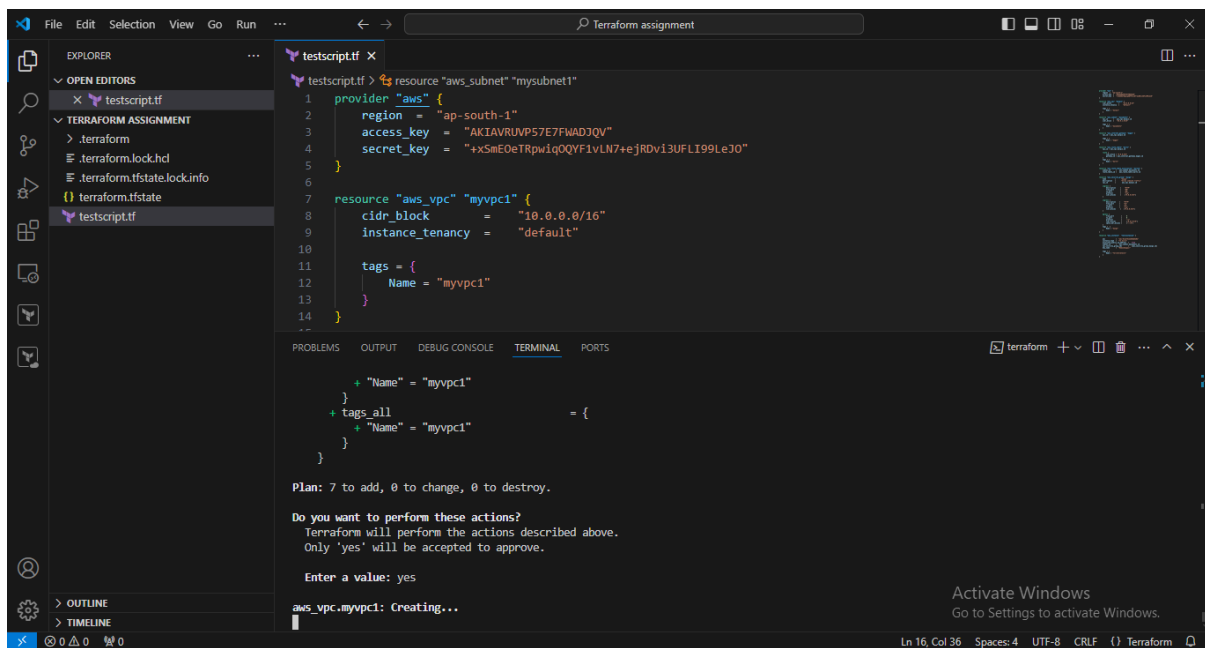
Step 15: Now to apply the changes, execute command: "terraform apply"
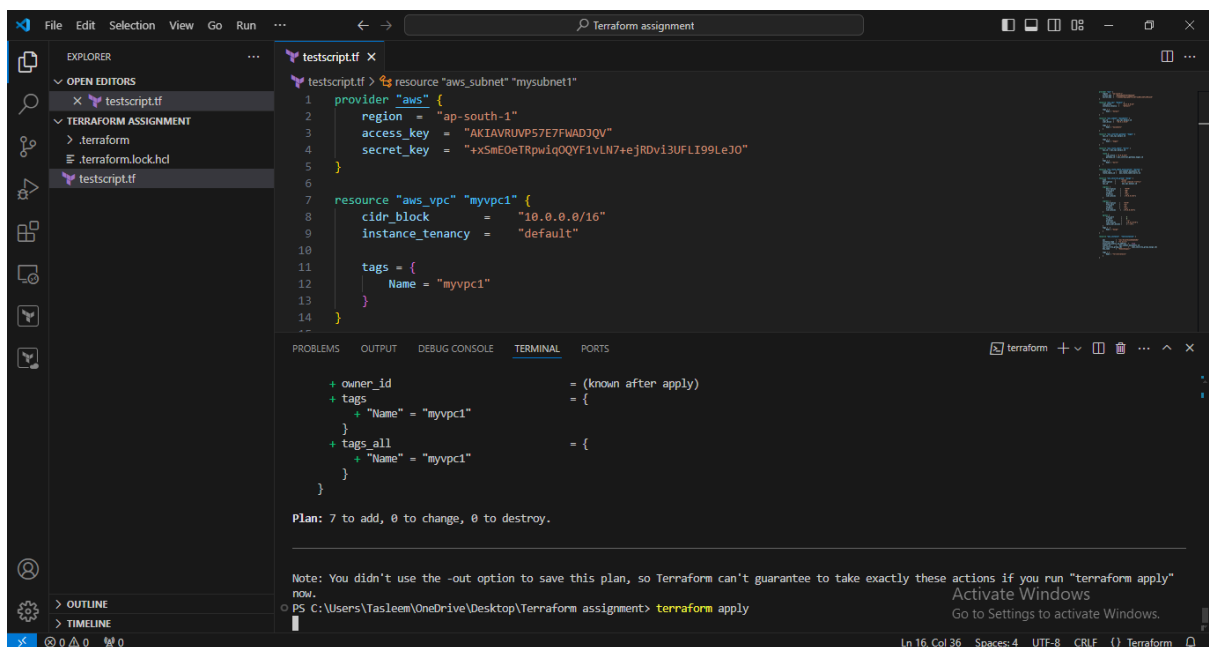
Step 16: It will ask to verify if we want to make the changes. Enter "yes"

We can see it applied successfully.

Step 17: Now go to the AWS Console → EC2 Instance and query the Instance.
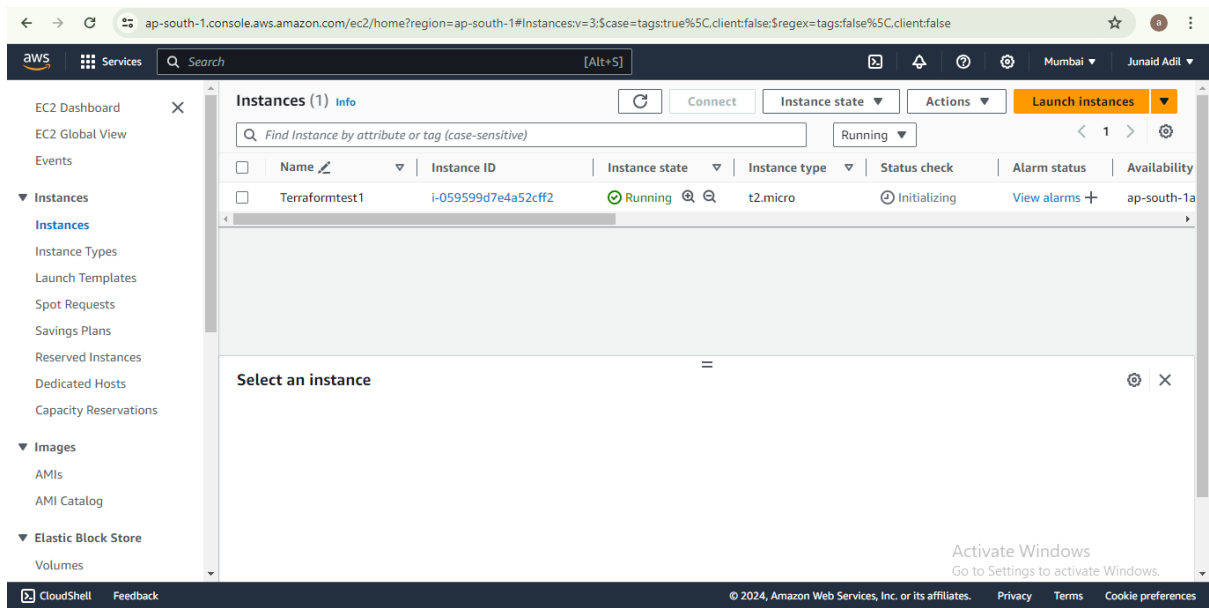
We can see an instance has been created "terraformtest1".
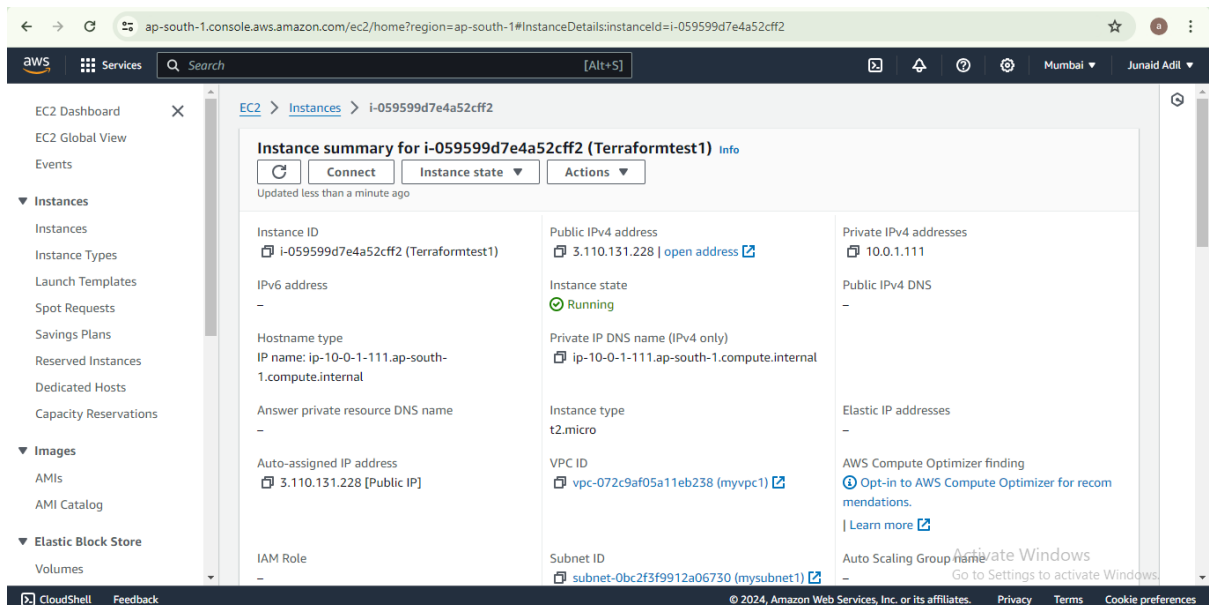
"myvpc1" VPC has been created

"mysubnet1" subnet has been created.

Security group is created with inbound rules for port range: 22 and 80 as mentioned in the script.