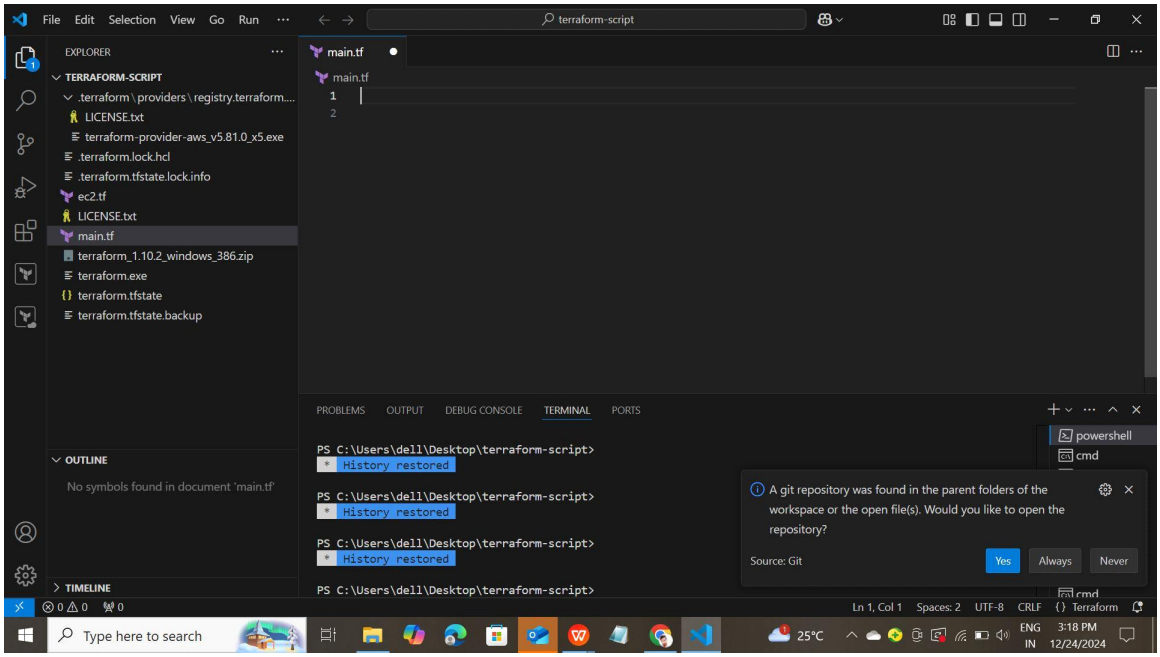
REDDY PRAKASH

Terraform Overview Assignments

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

- Successfully Terraform is installed in local machine like VScode
- with the help of VScode to create ec2 infrastructure
- Create a directory with the name terraform-script
- we mandatiry follow extension with .tf
- first we configure with command of aws cridentials



- inside that directory create file name main.tf
- below we use this script for create ec2 Infrastructure

```
provider "aws" {

   region = "us-east-1"

}
```

```
# 1. Create a VPC

resource "aws_vpc" "my_vpc" {
```

```
    cidr_block                  = "10.0.0.0/16"
    enable_dns_support      = true
    enable_dns_hostnames = true
    tags = {
        Name = "MyVPC"
    }
}


# 2. Create a Subnet
resource "aws_subnet" "my_subnet" {
    vpc_id                              = aws_vpc.my_vpc.id
    cidr_block                      = "10.0.1.0/24"
    map_public_ip_on_launch = true
    availability_zone           = "us-east-1a"      tags = {
        Name = "MySubnet"
    }
}
```

```
# 3. Create an Internet Gateway

resource "aws_internet_gateway" "my_igw" {

    vpc_id = aws_vpc.my_vpc.id

    tags = {

        Name = "MyInternetGateway"

    }

}



# 4. Create a Route Table

resource "aws_route_table" "my_route_table" {

    vpc_id = aws_vpc.my_vpc.id

    tags = {

        Name = "MyRouteTable"

    }

}
```

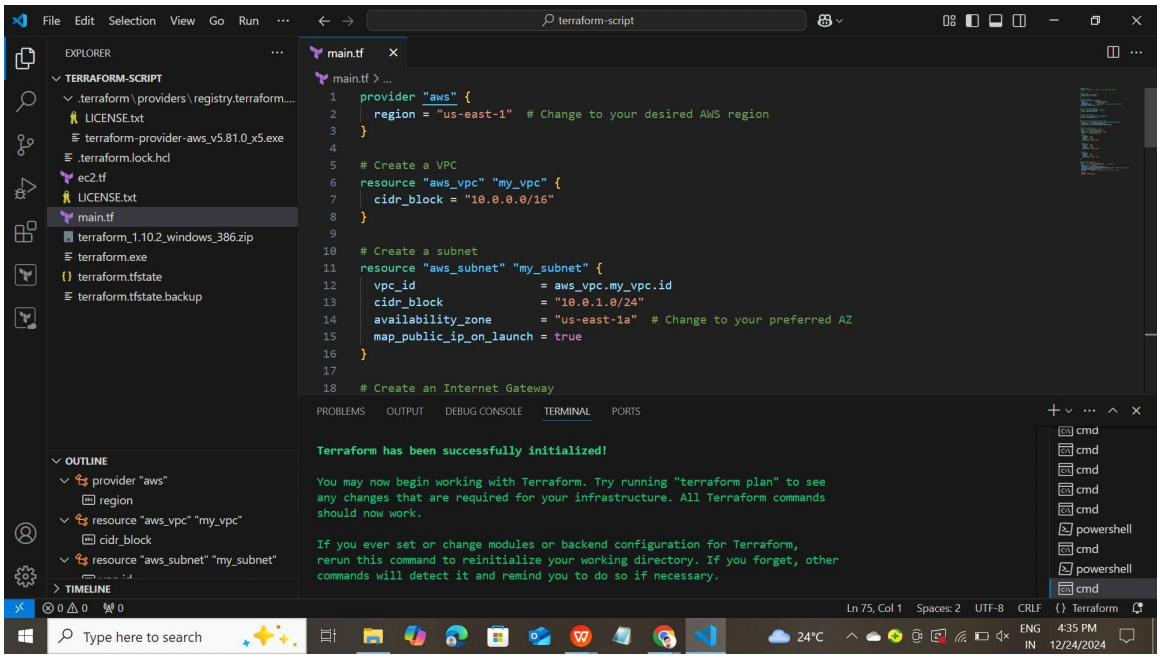
```
# 5. Create a Route for Internet Access

resource "aws_route" "internet_access" {

   route_table_id                =
aws_route_table.my_route_table.id

   destination_cidr_block = "0.0.0.0/0"

   gateway_id                       =
aws_internet_gateway.my_igw.id

}


# 6. Associate the Subnet with the Route Table

resource "aws_route_table_association"
"my_subnet_association" {

   subnet_id           = aws_subnet.my_subnet.id

   route_table_id = aws_route_table.my_route_table.id

}


# 7. Create a Security Group

resource "aws_security_group" "my_sg" {
```

```
vpc_id = aws_vpc.my_vpc.id

name      = "MySecurityGroup"

ingress {

   from_port      = 22

   to_port          = 22

   protocol        = "tcp"

   cidr_blocks = ["0.0.0.0/0"]

}

ingress {

   from_port      = 80

   to_port          = 80

   protocol        = "tcp"

   cidr_blocks = ["0.0.0.0/0"]

}

egress {

   from_port      = 0

   to_port          = 0
```

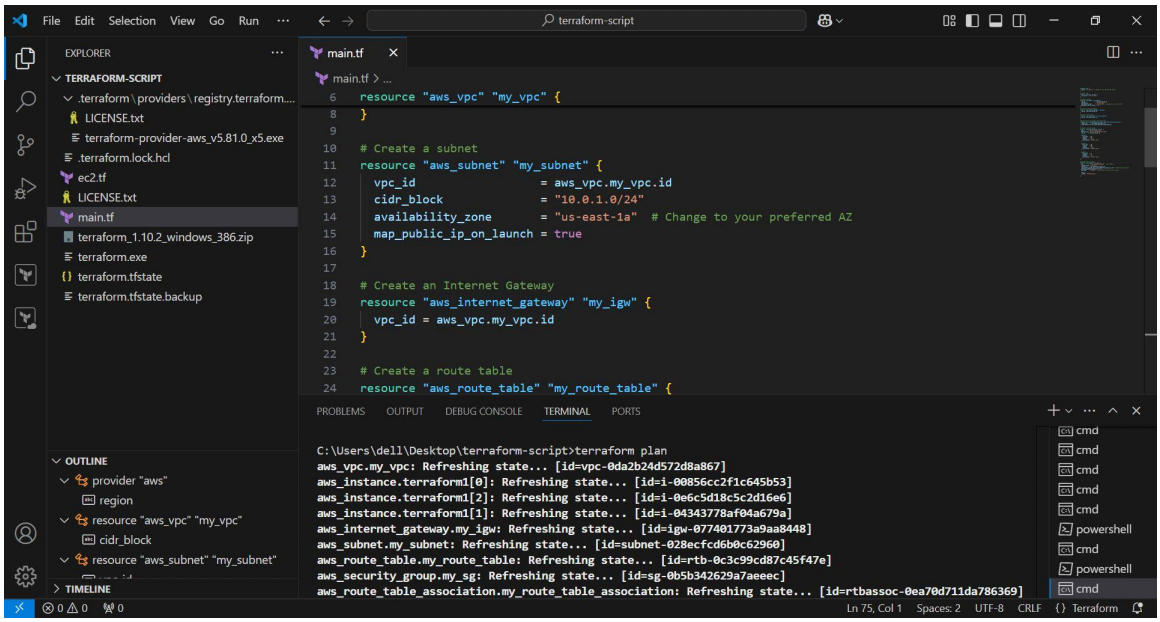
```
    protocol        = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  tags = {
    Name = "MySecurityGroup"
  }
}

# 8. Launch an EC2 Instance
resource "aws_instance" "my_instance" {
  ami                    = "ami-0c02fb55956c7d316"
us-east-1
  instance_type = "t2.micro"
  subnet_id          = aws_subnet.my_subnet.id
  security_groups = [aws_security_group.my_sg.name]

  key_name = "docker"
```

```
    tags = {
        Name = "MyInstance"
    }
}
```

With help of terraform init command to initial the terraform

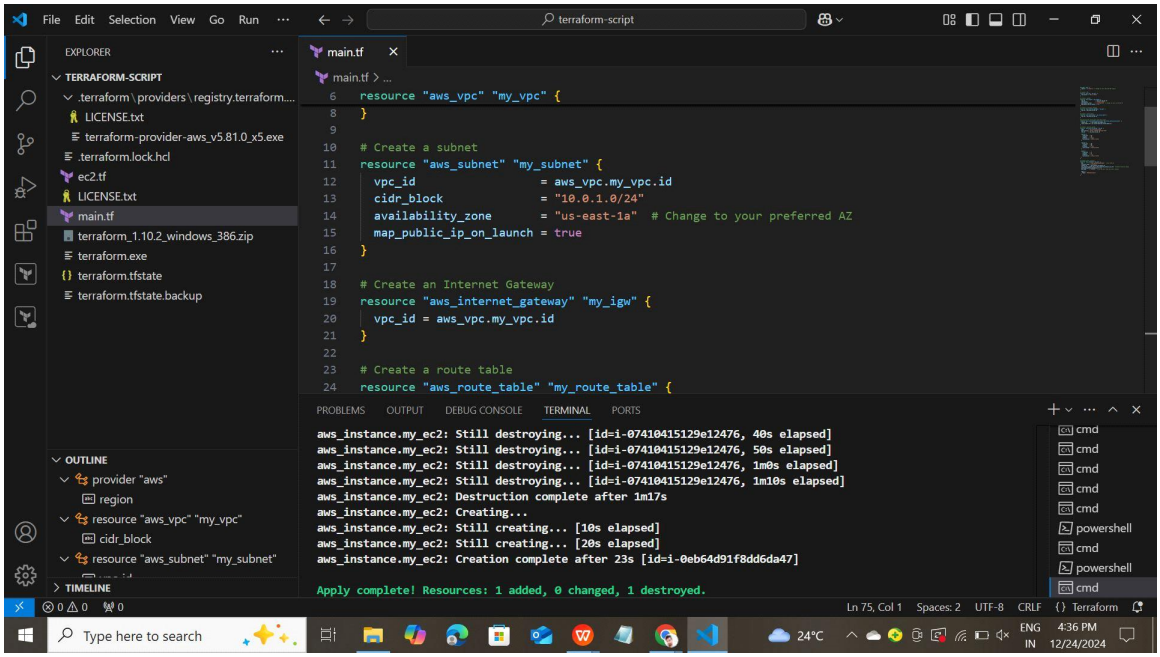


we plan the ec2 infrastructure with the help of terraform plan command



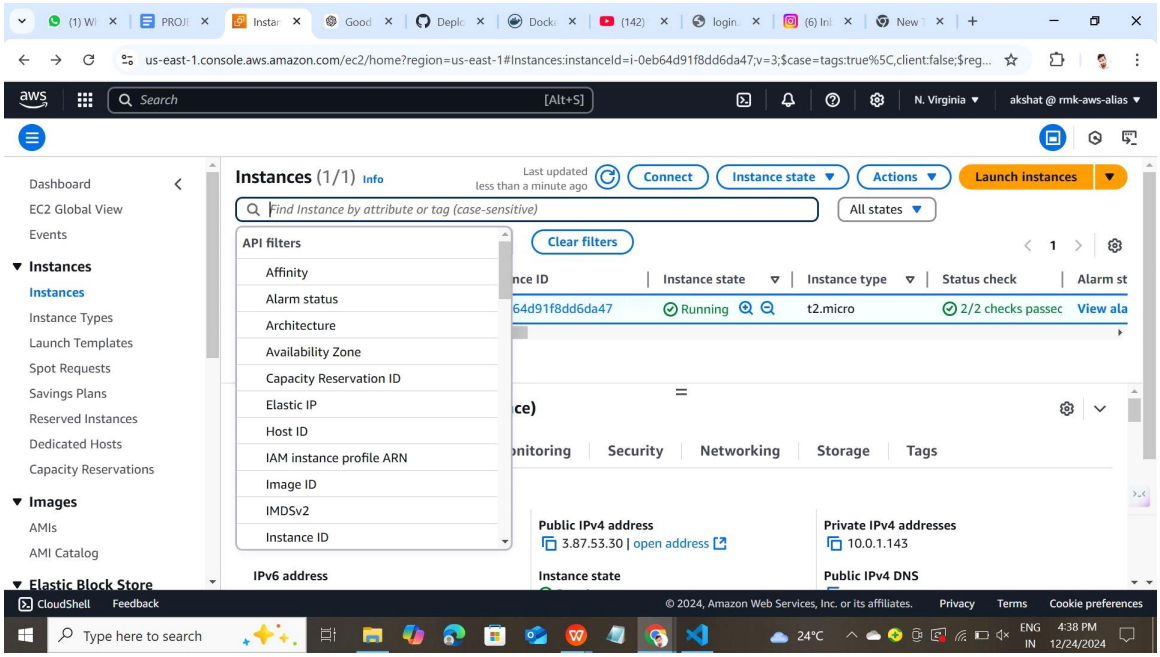
we succesfully apply the plan with command terraform apply

then we create ec2 infrastructure with above script



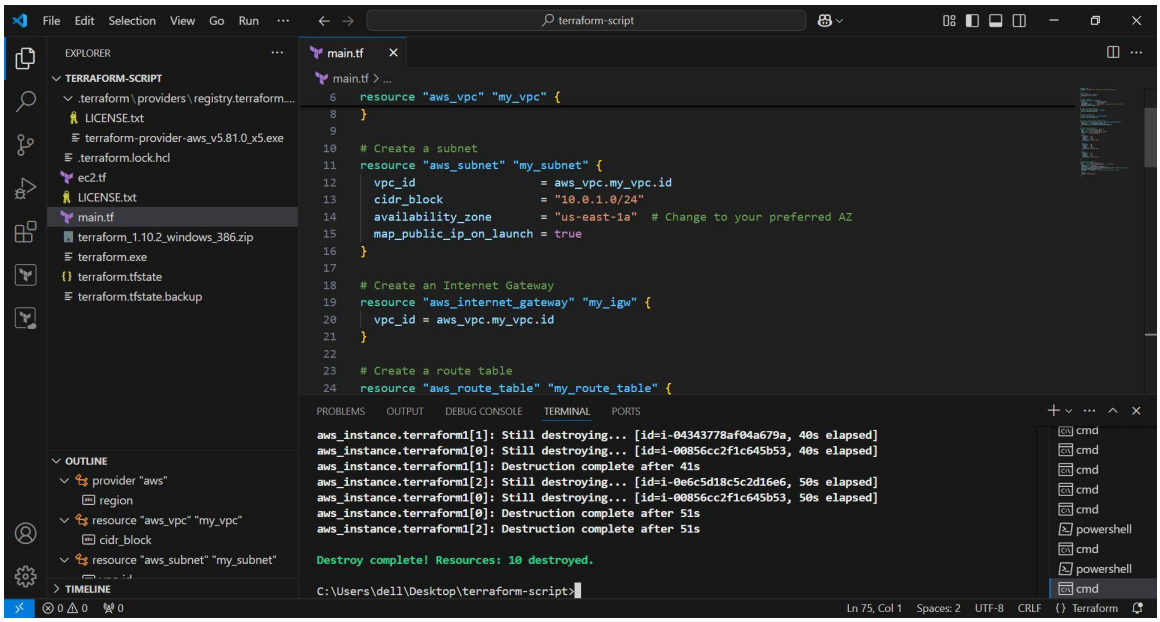
we saw successfully create ec2 instance in aws ec2 dashboard

we find that



Then we successfully destory the instance with command

terraform destroy


File Edit Selection View Go Run
terraform-script
EXPLORER
TERRAFORM-SCRIPT
.terraform \ providers \ registry.terraform....
LICENSE.txt
terraform-provider-aws_v5.81.0_x5.exe
.terraform.lock.hcl
ec2.tf
LICENSE.txt
main.tf
terraform_1.10.2_windows_386.zip
terraform.exe
terraform.tfstate
terraform.tfstate.backup
OUTLINE
provider "aws"
region
resource "aws_vpc" "my_vpc"
cidr_block
resource "aws_subnet" "my_subnet"
TIMELINE
main.tf
main.tf > ...
6 resource "aws_vpc" "my_vpc" {
8 }
9
10 # Create a subnet
11 resource "aws_subnet" "my_subnet" {
12 vpc_id = aws_vpc.my_vpc.id
13 cidr_block = "10.0.1.0/24"
14 availability_zone = "us-east-1a" # Change to your preferred AZ
15 map_public_ip_on_launch = true
16 }
17
18 # Create an Internet Gateway
19 resource "aws_internet_gateway" "my_igw" {
20 vpc_id = aws_vpc.my_vpc.id
21 }
22
23 # Create a route table
24 resource "aws_route_table" "my_route_table" {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
aws_instance.terraform1[1]: Still destroying... [id=i-04343778af04a679a, 40s elapsed]
aws_instance.terraform1[0]: Still destroying... [id=i-00856cc2f1c645b53, 40s elapsed]
aws_instance.terraform1[1]: Destruction complete after 41s
aws_instance.terraform1[2]: Still destroying... [id=i-0e6c5d18c5c2d16e6, 50s elapsed]
aws_instance.terraform1[0]: Still destroying... [id=i-00856cc2f1c645b53, 50s elapsed]
aws_instance.terraform1[0]: Destruction complete after 51s
aws_instance.terraform1[2]: Destruction complete after 51s
Destroy complete! Resources: 10 destroyed.
C:\Users\dell\Desktop\terraform-script>
cmd
cmd
cmd
cmd
cmd
powershell
cmd
powershell
cmd
Ln 75, Col 1 Spaces: 2 UTF-8 CRLF {} Terraform

THANK YOU