ASSIGNMENT 1

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

Name: Aditya Tomar

SAP:500106015

R.NO=R2142221060

Batch:DevOps(2 NH)

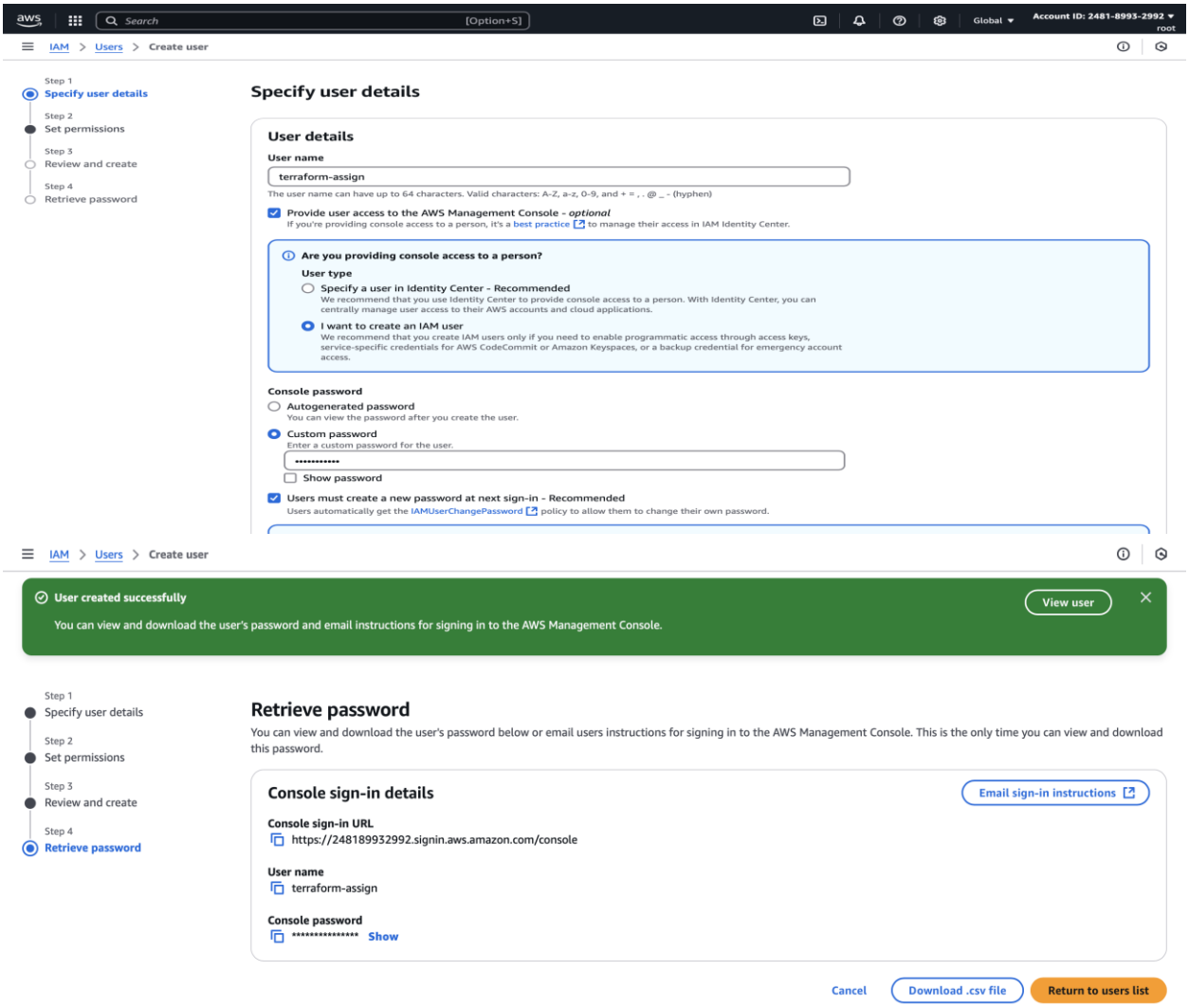
In this assignment, terraform — an open-source Infrastructure as Code (IaC) tool — is used to automate the provisioning of cloud resources on Amazon Web Services (AWS). The goal is to write a reusable, version-controlled script that sets up a basic cloud infrastructure in a matter of seconds.

The tasks covered in this assignment include:

- **Creating two EC2 instances of type t2.micro**
- **Provisioning a Customer Gateway and a VPN Gateway to simulate a VPN setup**
- **Creating an S3 bucket for cloud-based storage**

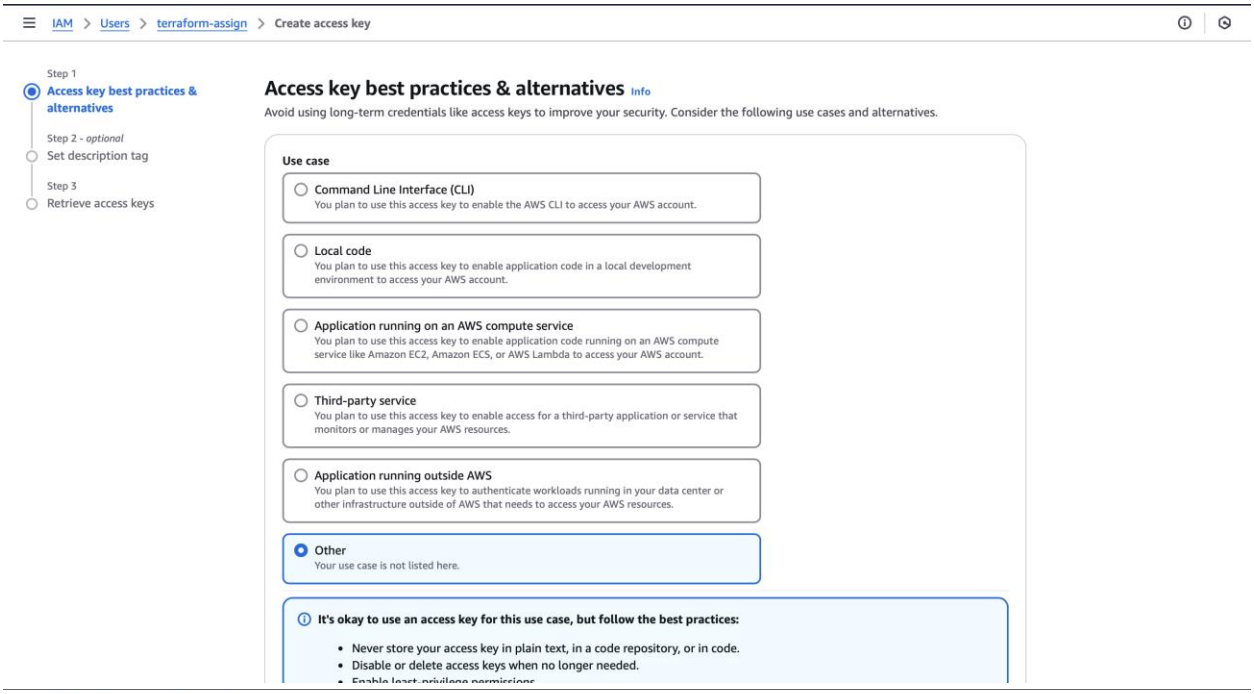
1. Creating IAM User

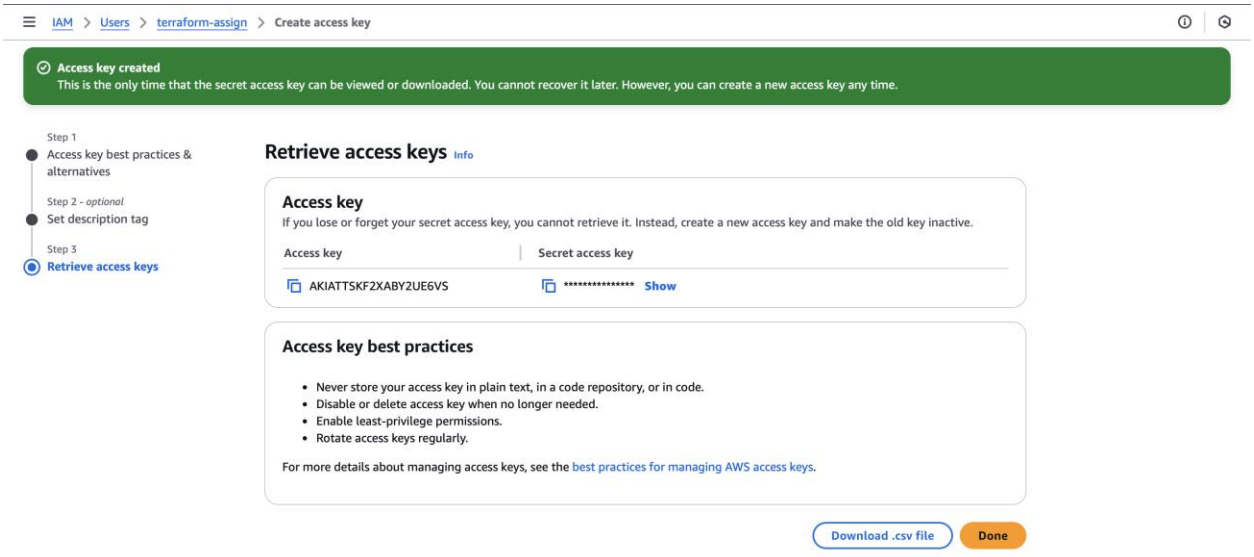
An IAM user named terraform-assign was created in the AWS Console with programmatic access. This user will be used to authenticate Terraform with AWS for infrastructure provisioning.



2. Creating Access Key for IAM User

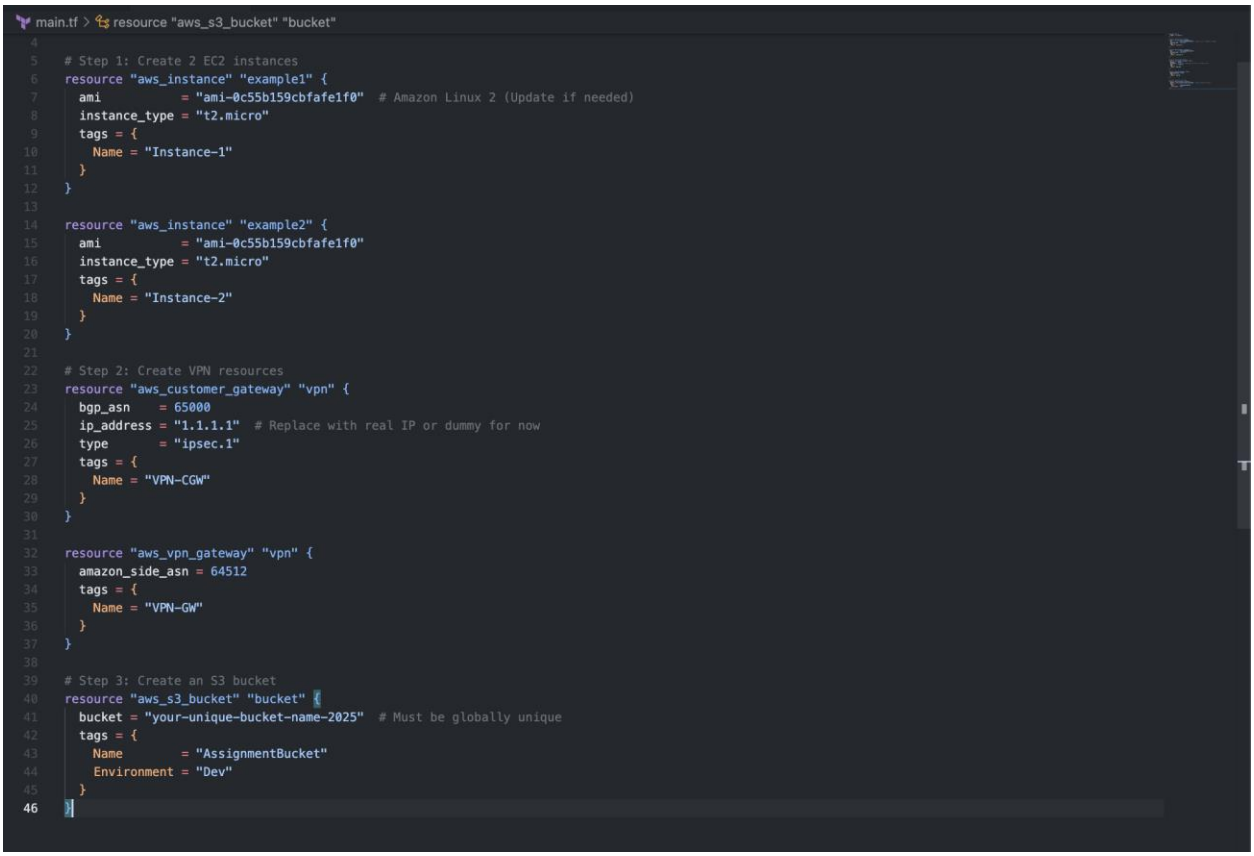
An Access Key and Secret Access Key were generated for the IAM user. These credentials were configured using the aws configure command to securely connect Terraform with the AWS account.





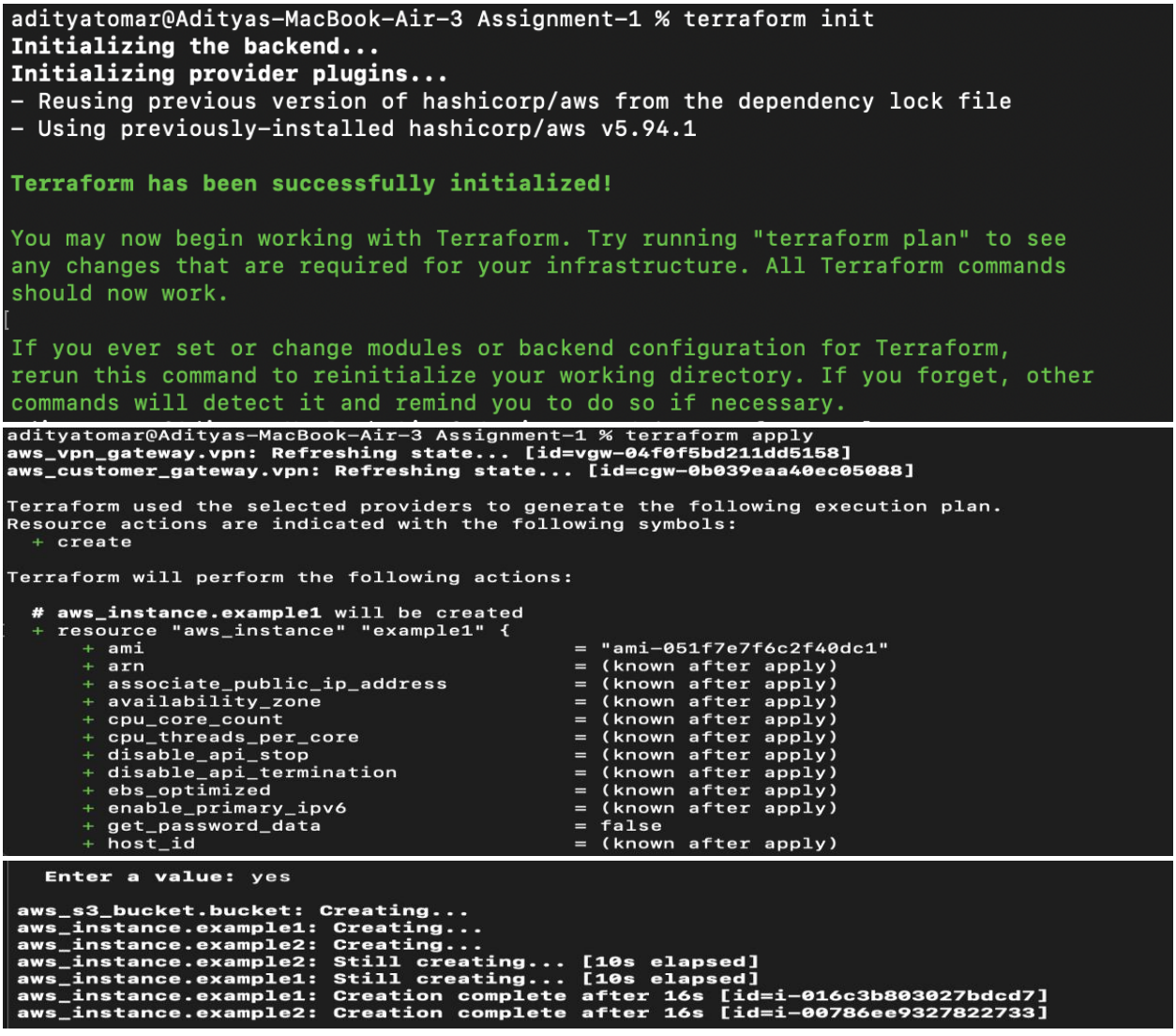
3. Writing Terraform Code in VS Code

The Terraform configuration was written in VS Code. It includes resources for EC2 instances, a VPN setup, and an S3 bucket — all defined as Infrastructure as Code in a single main.tf file.



4. Running terraform init, plan, and apply

Terraform CLI commands were used to initialize the working directory, preview the execution plan, and apply the infrastructure changes. This screenshot shows the commands being executed in the terminal.



```
adityatomar@Adityas-MacBook-Air-3 Assignment-1 % terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.94.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
adityatomar@Adityas-MacBook-Air-3 Assignment-1 % terraform apply
aws_vpn_gateway.vpn: Refreshing state... [id=vgw-04f0f5bd211dd5158]
aws_customer_gateway.vpn: Refreshing state... [id=cgw-0b039eaa40ec05088]

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.example1 will be created
  + resource "aws_instance" "example1" {
      + ami                                  = "ami-051f7e7f6c2f40dc1"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
```

```
  Enter a value: yes

aws_s3_bucket.bucket: Creating...
aws_instance.example1: Creating...
aws_instance.example2: Creating...
aws_instance.example2: Still creating... [10s elapsed]
aws_instance.example1: Still creating... [10s elapsed]
aws_instance.example1: Creation complete after 16s [id=i-016c3b803027bdcd7]
aws_instance.example2: Creation complete after 16s [id=i-00786ee9327822733]
```

5. AWS Console Showing Resources Created

This screenshot captures the AWS Console showing all the resources created — including EC2 instances, VPN components, and the S3 bucket — confirming successful provisioning using Terraform.