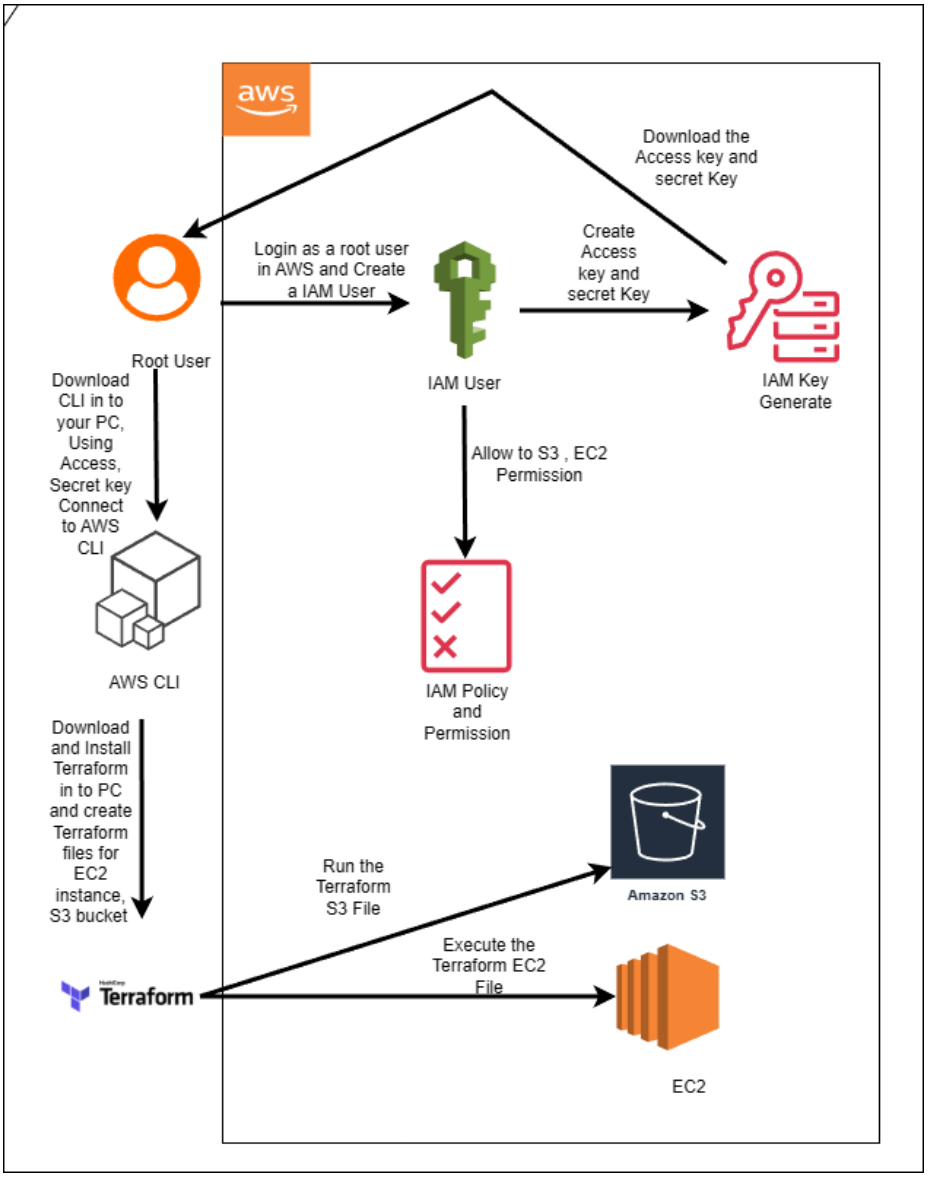
Step-by-step guide on how to create an EC2 instance and S3 bucket using Terraform

Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently. It takes your infrastructure you have defined in code and makes it real! The beauty of what Terraform does is that it does not ask you how to get from the infrastructure you have to the infrastructure you want, it just asks you what you want the world to look like and then it does the hard work.



- **Step 1: Download the Terraform** in to your PC. Install Chocolaty recommended for windows, Open power shell as administrator then paste the below command.

  - First install Chocolatey (if not already installed),

```
Set-ExecutionPolicy Bypass -Scope Process -Force
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072
iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

  - Install Terraform

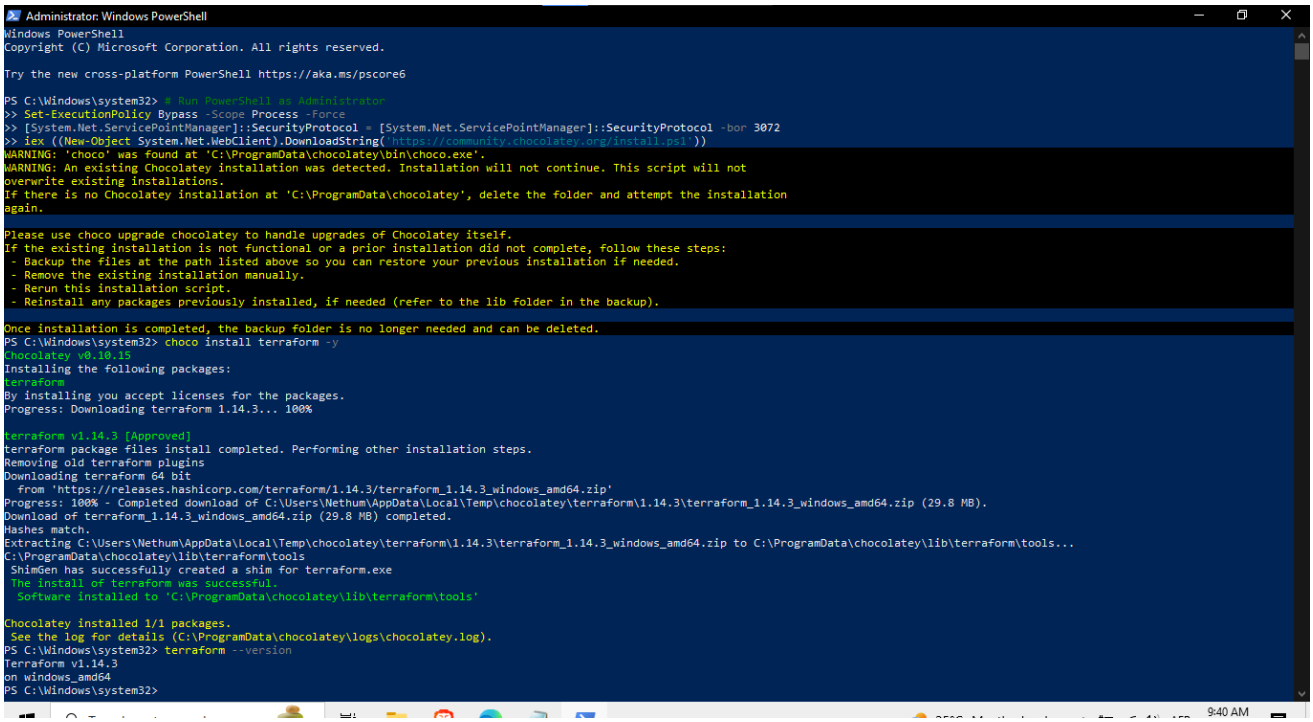
```
choco install terraform
```

  - Option 2: Go to the official Terraform website, select your operating system (Windows or macOS), choose the version you need, and download the package

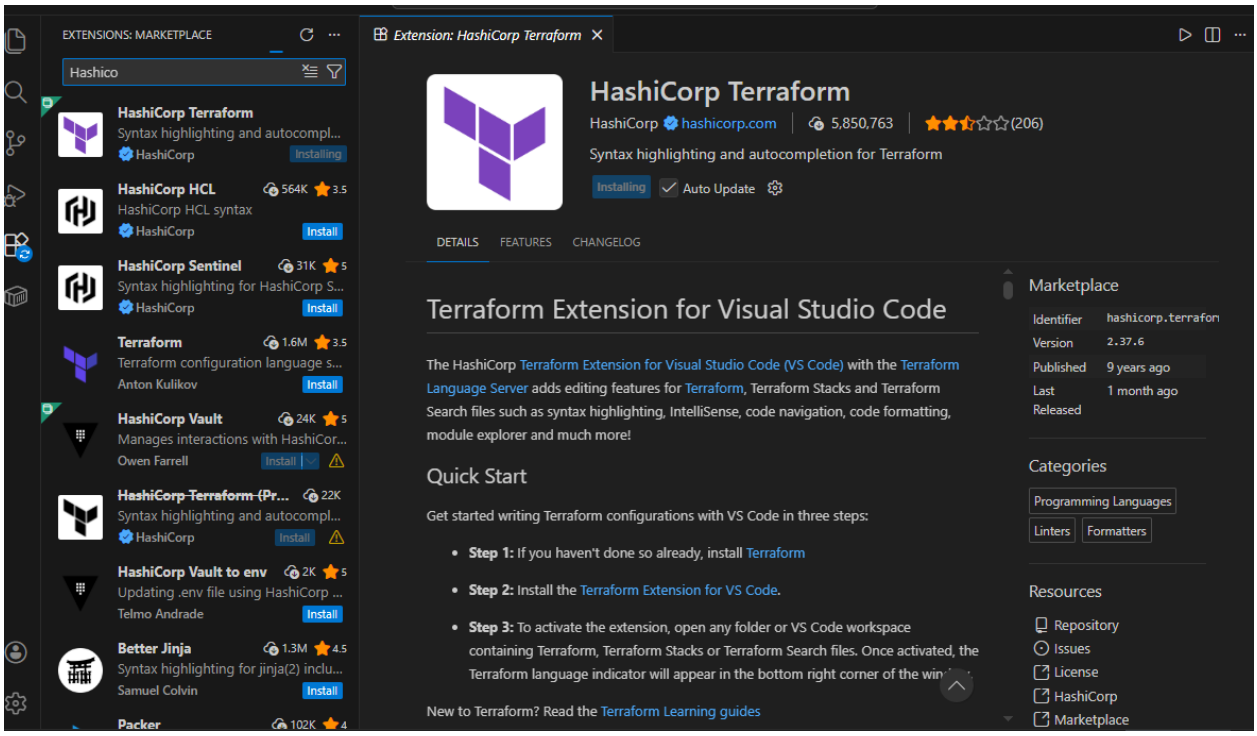
    https://developer.hashicorp.com/terraform/install

    After Installation done go to PowerShell or MacOs terminal just type below command then your Terraform version show like that,

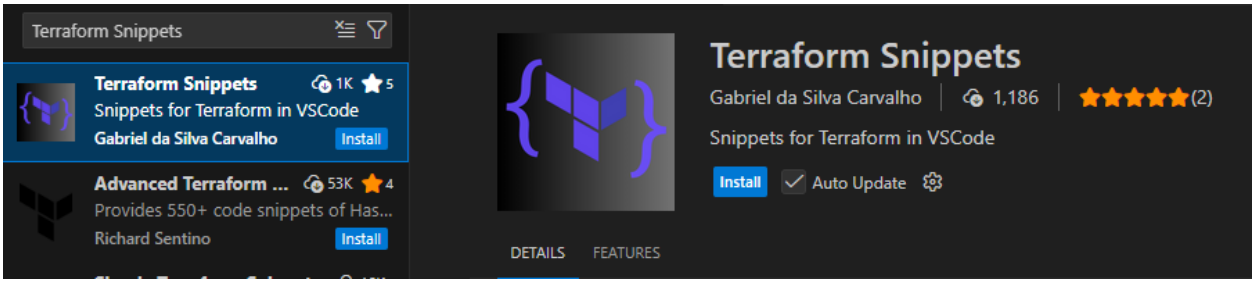
```
terraform --version
```



- **Step 2: Open VS code** and go to extensions then download supportive tools for properly run Terraform file,
  - Install Hashicope Terraform



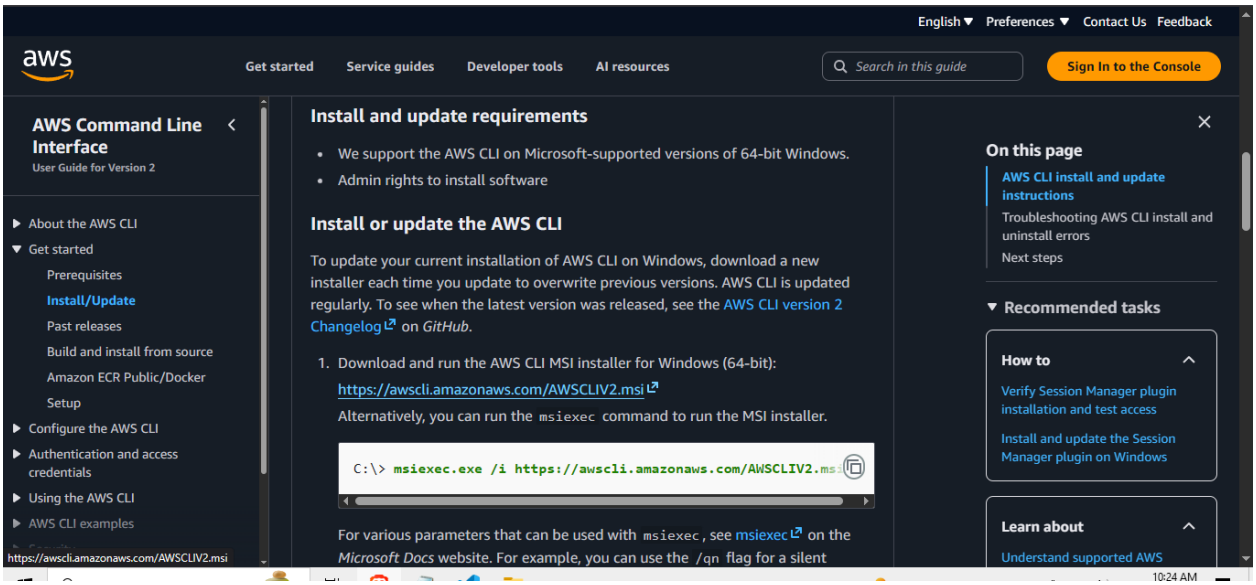
  - Install Terraform Snippets,



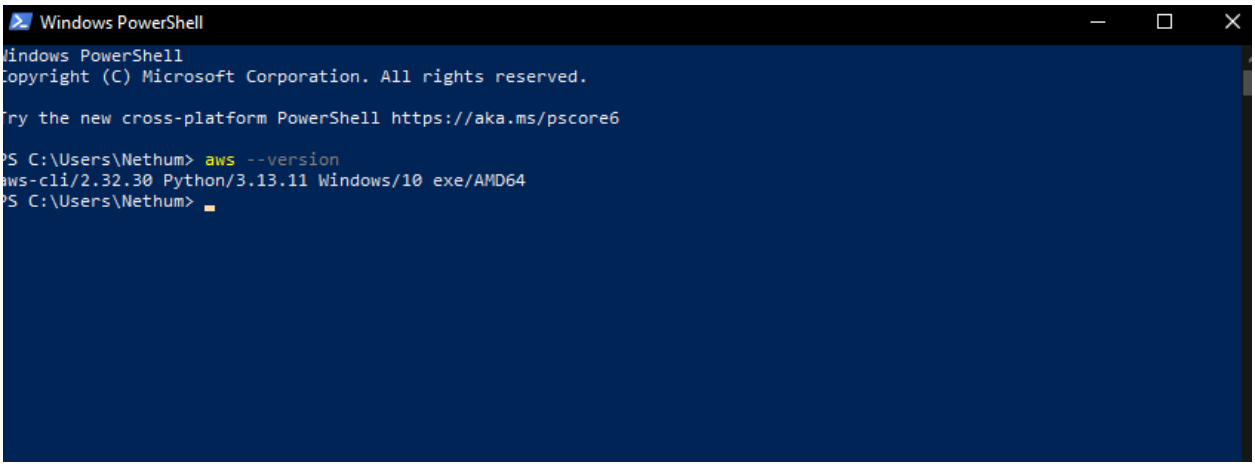
- **Step 3: Download AWS CLI :** Go to the official AWS website, download the latest version to your PC, and verify the download was successful.

  https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

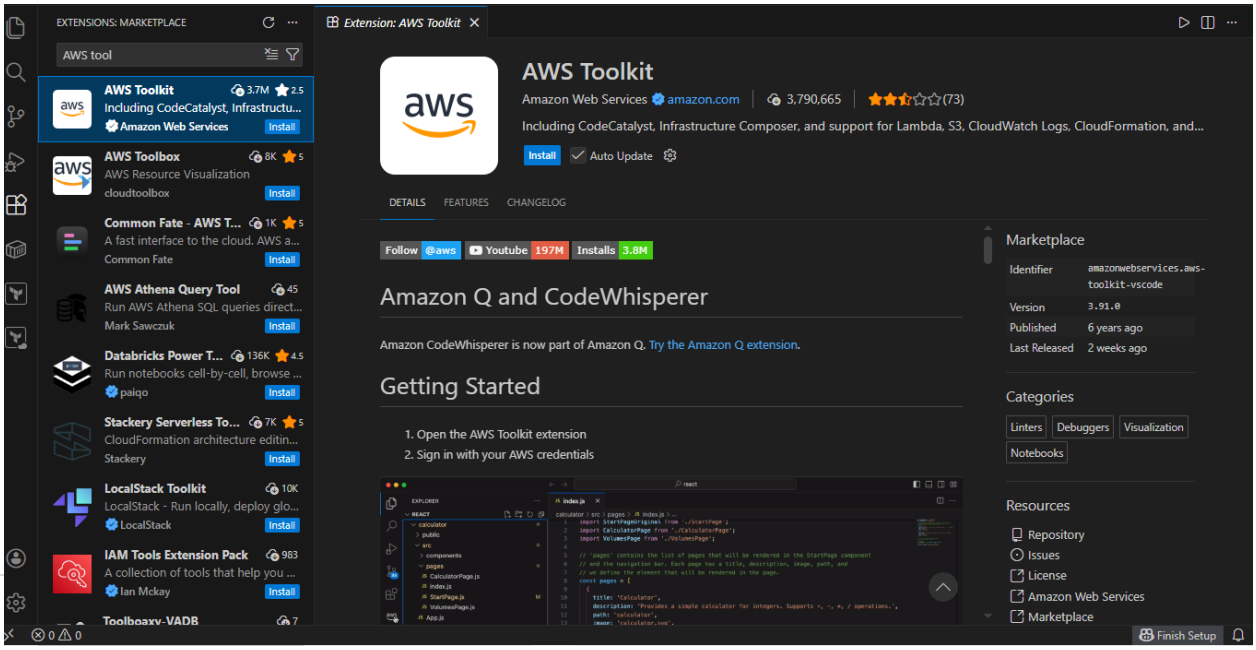
  check whether AWS status :- aws --version



❖ AWS Status

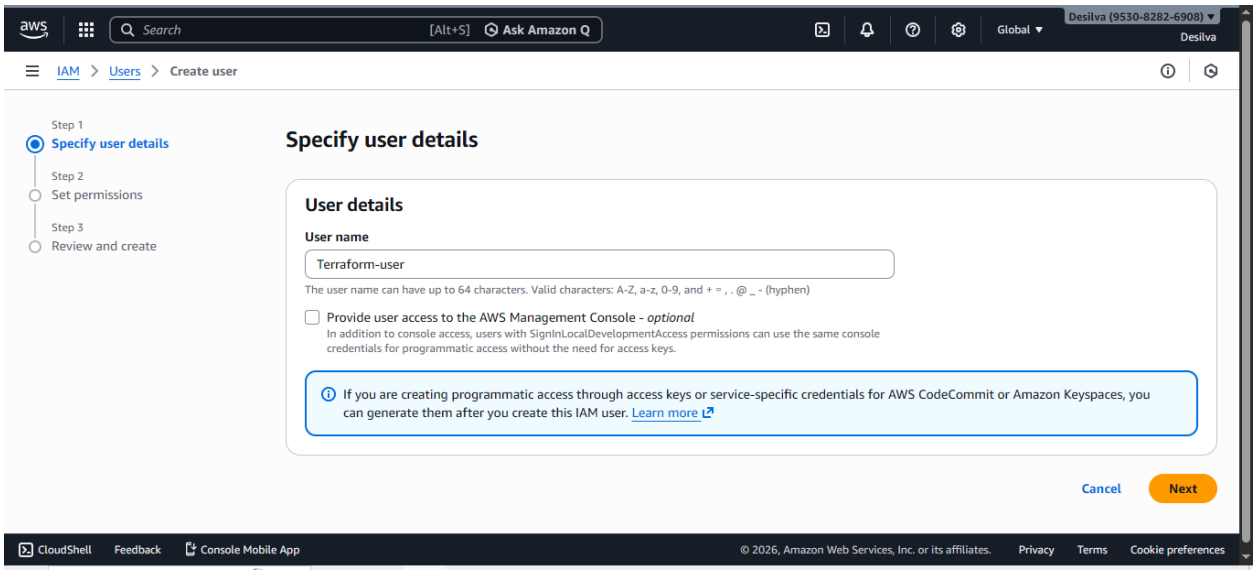


❖ Open VS code and Install AWS tool kit to configure AWS service.

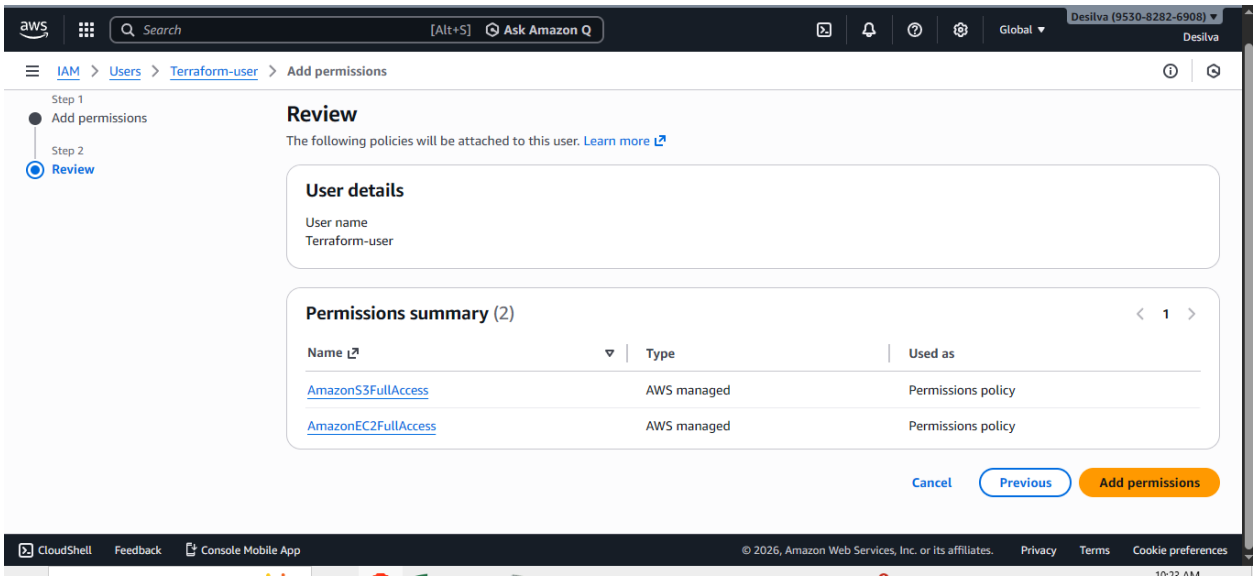


❖ **Step 4 Create an IAM user**. Under Permissions policies, attach the policies AmazonS3FullAccess and AmazonEC2FullAccess. Then, go to Security credentials, select Create access key, and generate an access key ID and a secret access key.

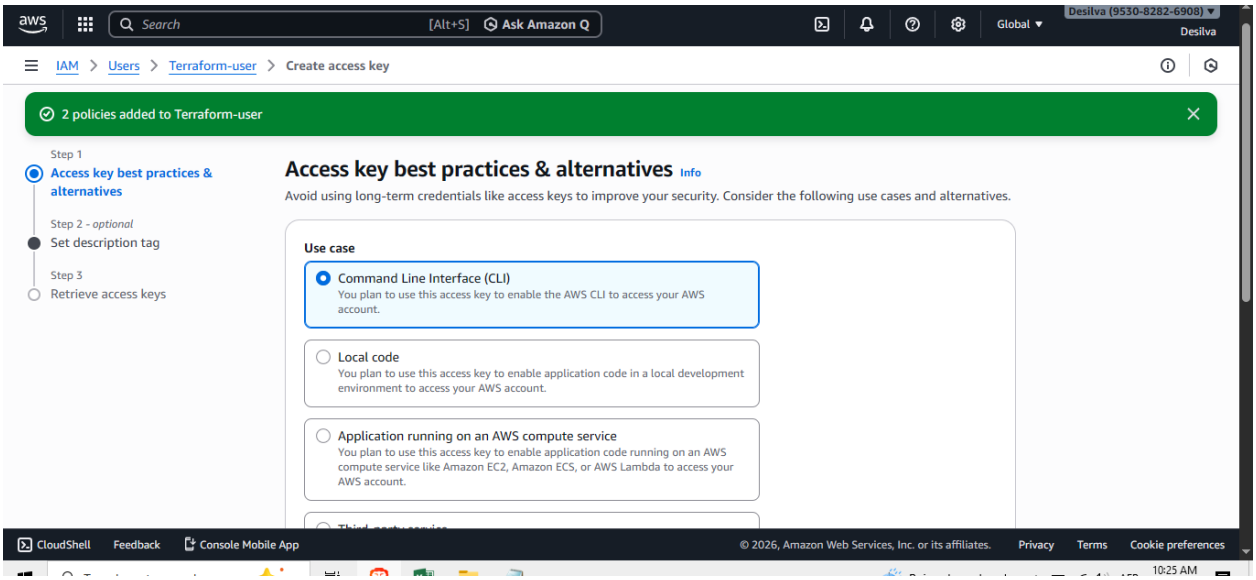
❖ IAM User Creation



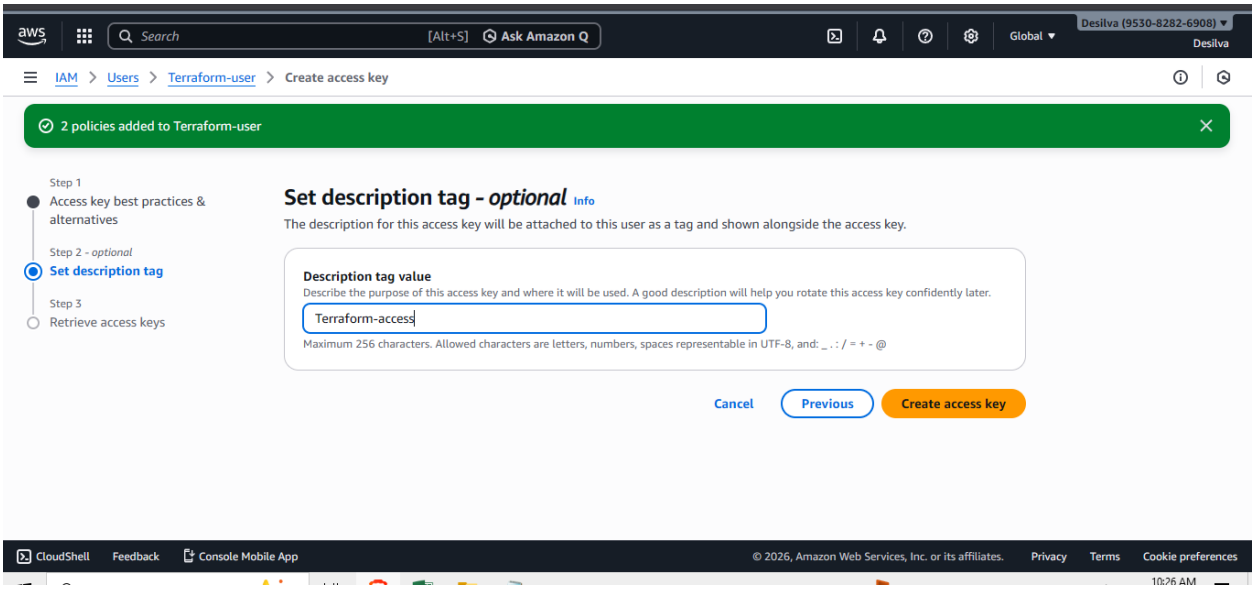
❖ Add permission and policy in to IAM user,



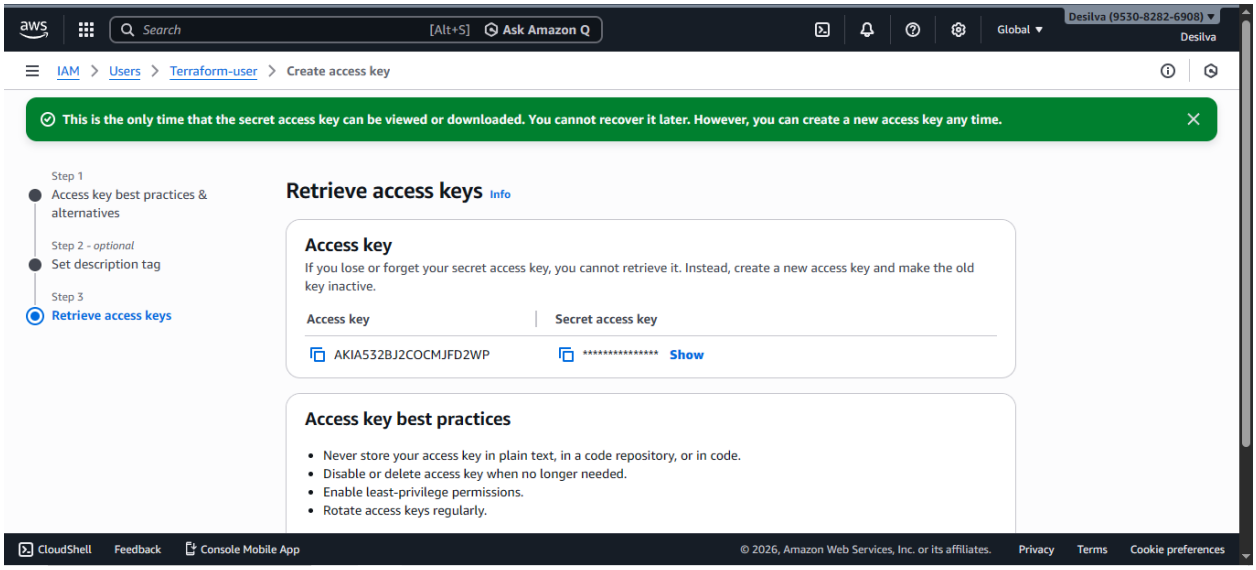
❖ Access Key Creation,



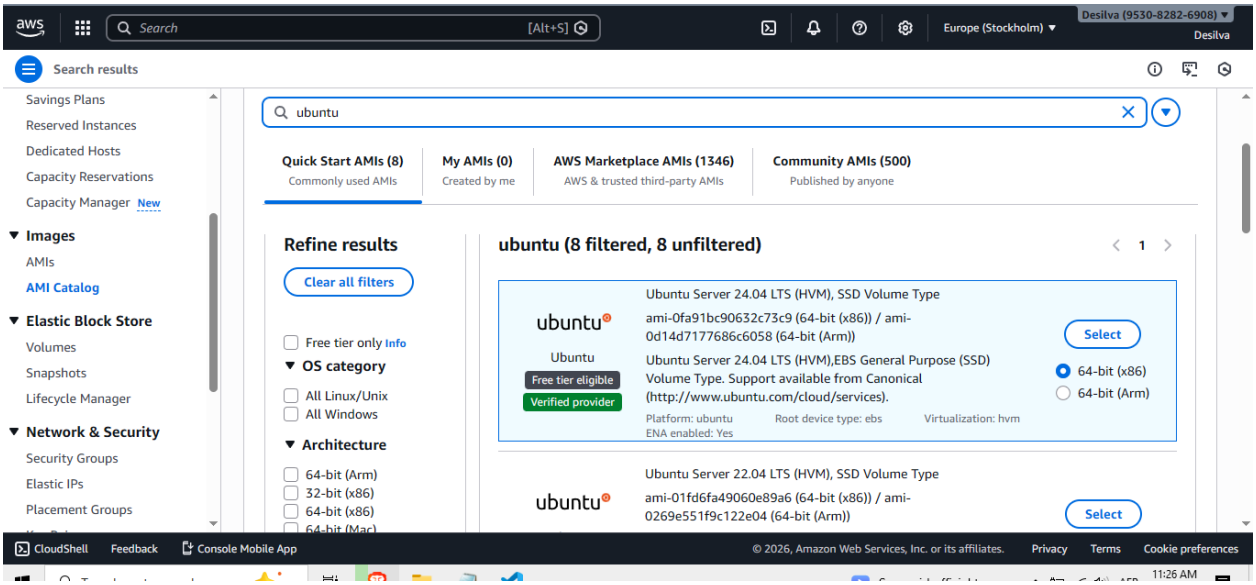
❖ Provide tag name for the Access Key creation,



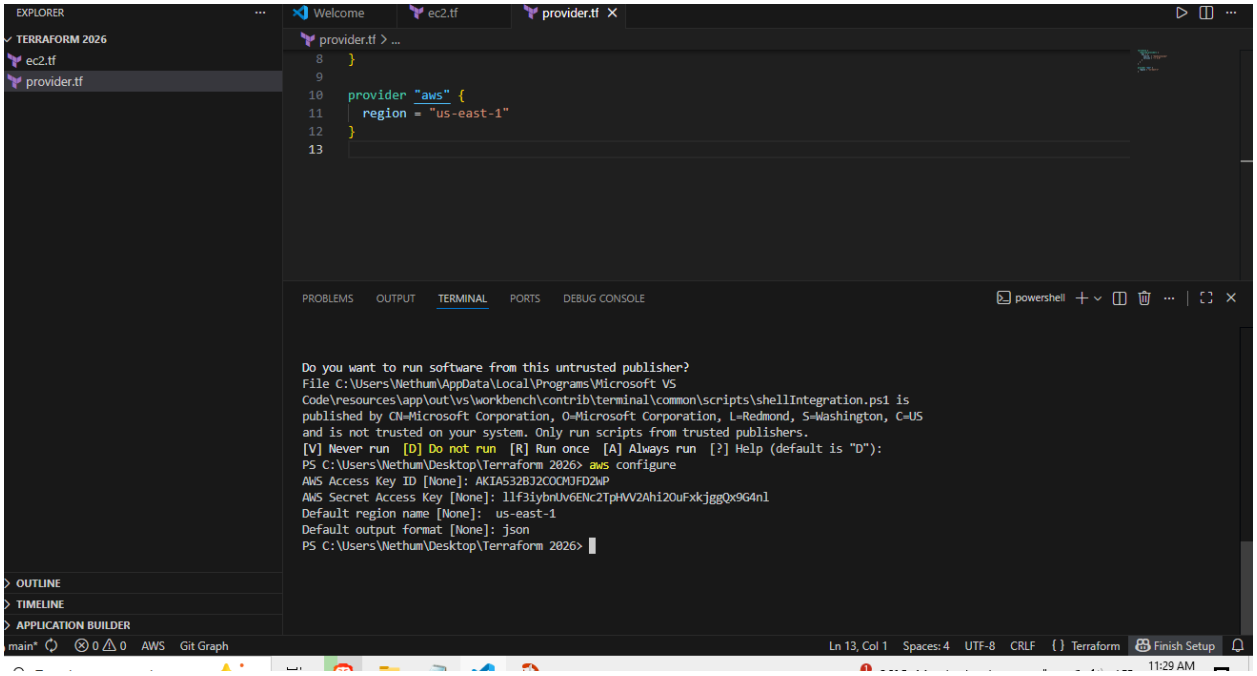
❖ Generated the Access key and secret key,



❖ **Step 5: Create a Terraform** file in VS Code to define an EC2 instance. Provide the required AMI ID from the AWS Management Console's AMI Catalog.

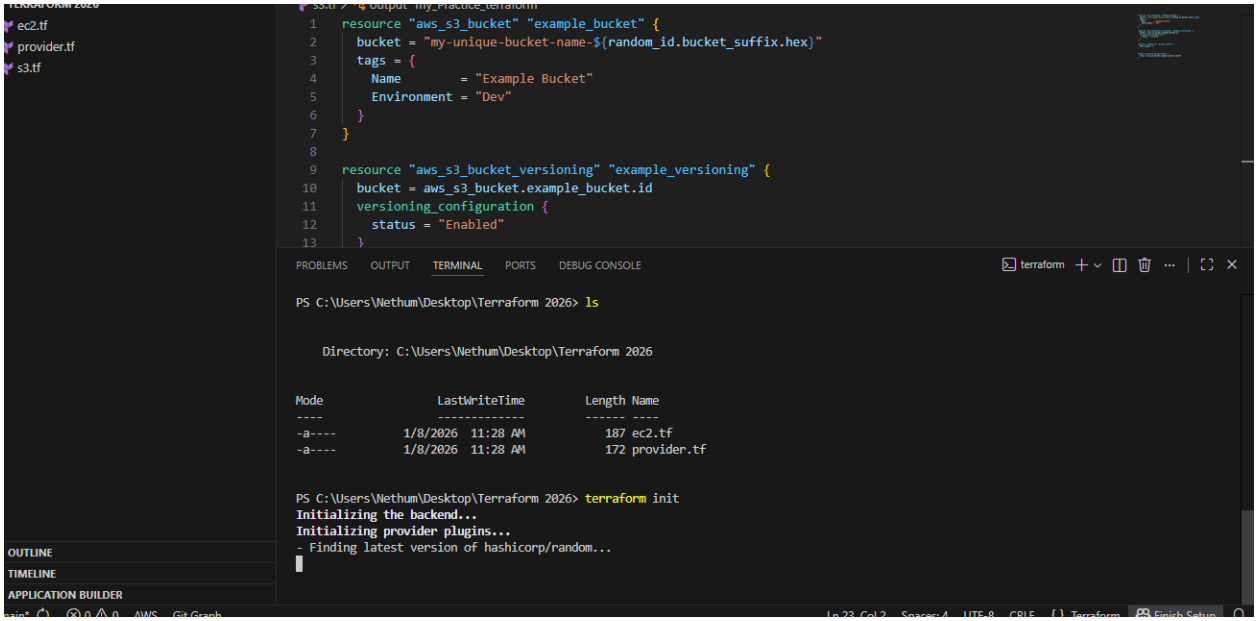


❖ **Step 6: Configure AWS CLI:** Open your terminal and configure the AWS CLI by providing your IAM user's access key, secret key, region, and output format. Then, create separate Terraform files for an S3 bucket and an EC2 instance,

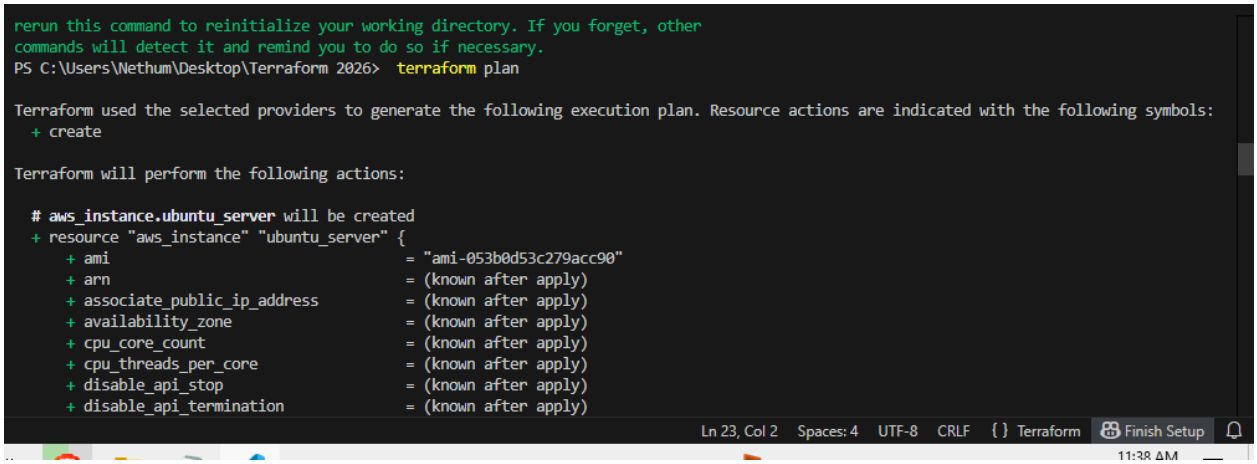


- After Configure AWS CLI and create Terraform files, Finally Execute the terraform Commands,

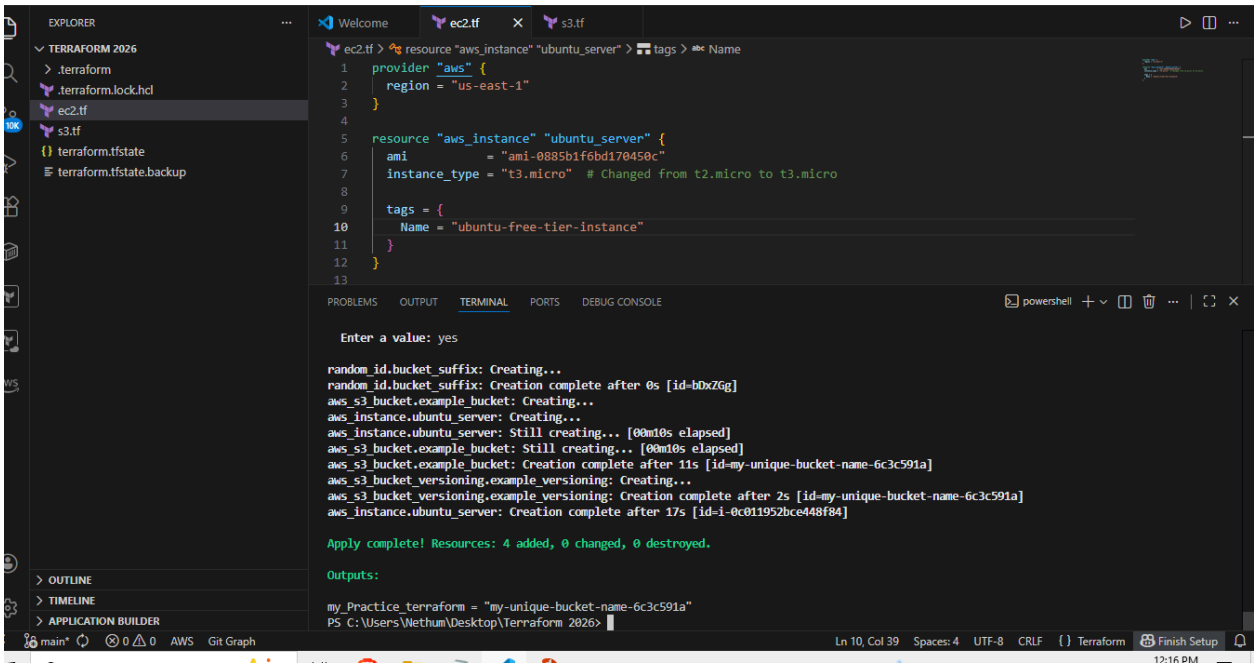
- Command : terraform init



- Command :- terraform plan

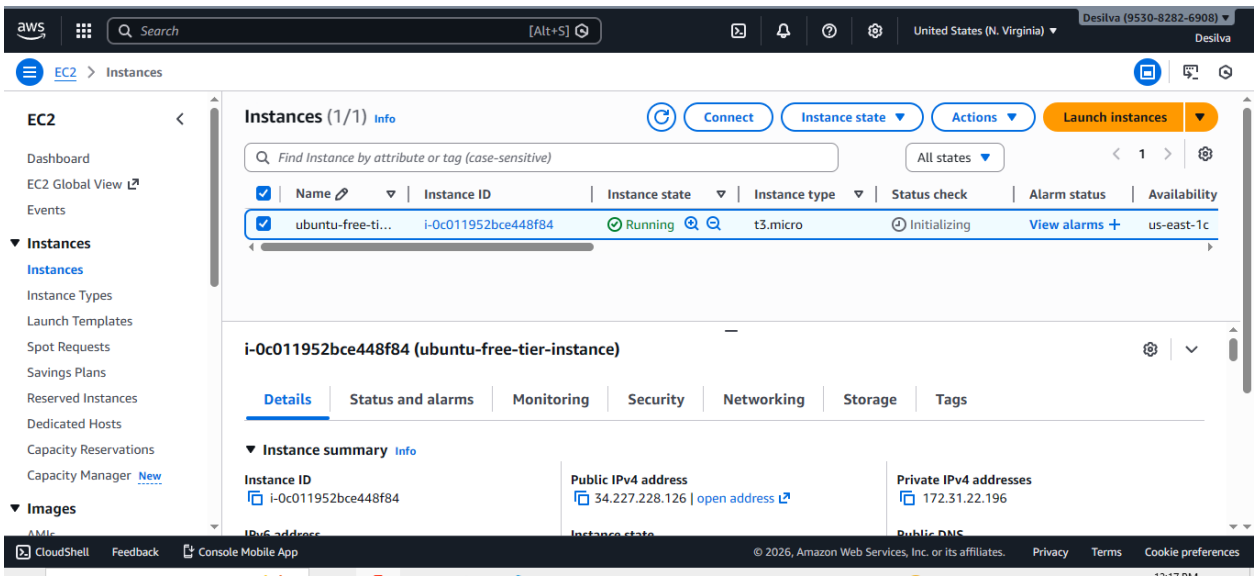


- Command : terraform apply

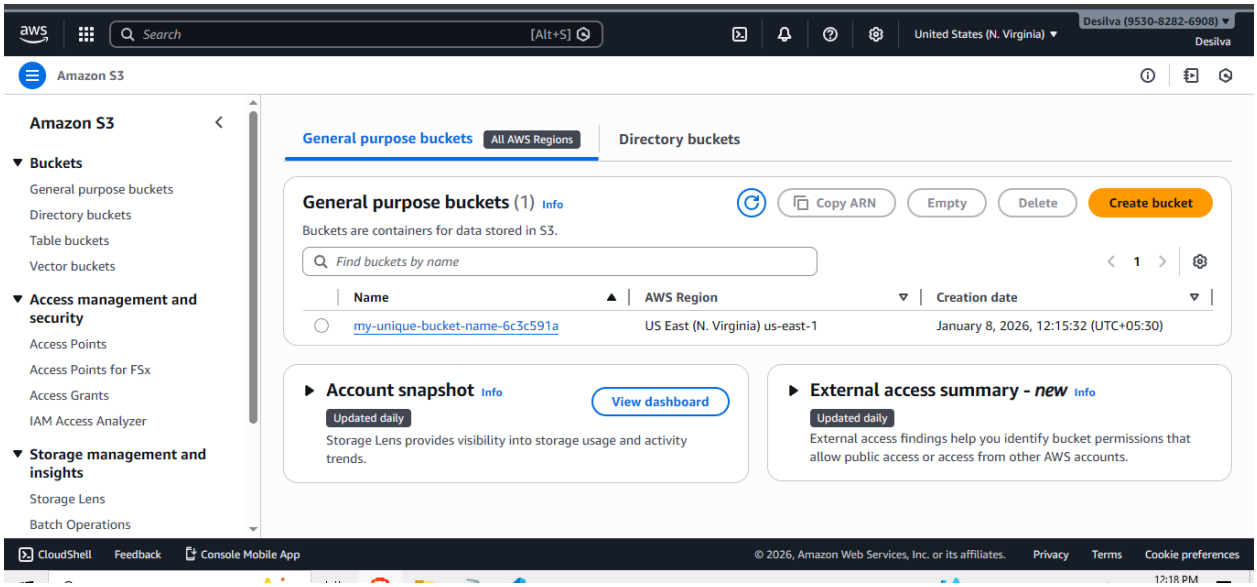


- After executing the Terraform commands, go to the AWS Management Console to verify whether the EC2 instance and S3 bucket were created successfully.

- EC2 Instance Status,



- S3 bucket Status



Terraform File Structure

Ec2.tf

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "ubuntu_server" {
  ami           = "ami-0885b1f6bd170450c"
  instance_type = "t3.micro"  # Changed from t2.micro to t3.micro

  tags = {
    Name = "ubuntu-free-tier-instance"
  }
}
```

S3.tf

```
resource "aws_s3_bucket" "example_bucket" {
  bucket = "my-unique-bucket-name-${random_id.bucket_suffix.hex}"
  tags = {
    Name        = "Example Bucket"
    Environment = "Dev"
  }
}

resource "aws_s3_bucket_versioning" "example_versioning" {
  bucket = aws_s3_bucket.example_bucket.id
  versioning_configuration {
    status = "Enabled"
  }
}

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


output "my_Practice_terraform" {
  value = aws_s3_bucket.example_bucket.bucket
}
```