

# Lab Exercise 14 – Provisioning an S3 Bucket On AWS

**Name:-Vansh Bhatt**

**SapId:- 500125395**

**R.No:- R2142231689**

**Batch:- DevOps B1**

**To:- Hitesh Kumar Sharma Sir**

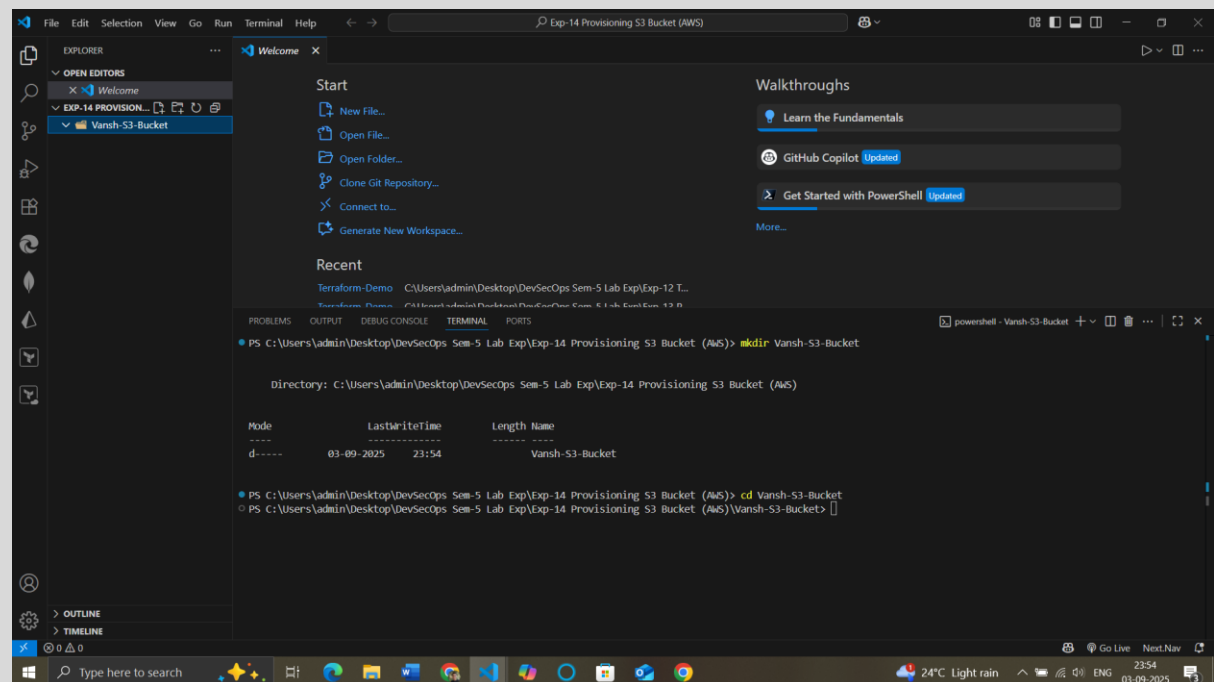
## **Exercise Steps:**

### **Step 1: Create a New Directory:**

Create a new directory to store your Terraform configuration:

```
mkdir Terraform-S3-Demo
```

```
cd Terraform-S3-Demo
```



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)> mkdir Vansh-S3-Bucket
```

Directory: C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)

Mode	LastWriteTime	Length	Name
d----	03-09-2025 23:54		Vansh-S3-Bucket

```
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)> cd Vansh-S3-Bucket
```

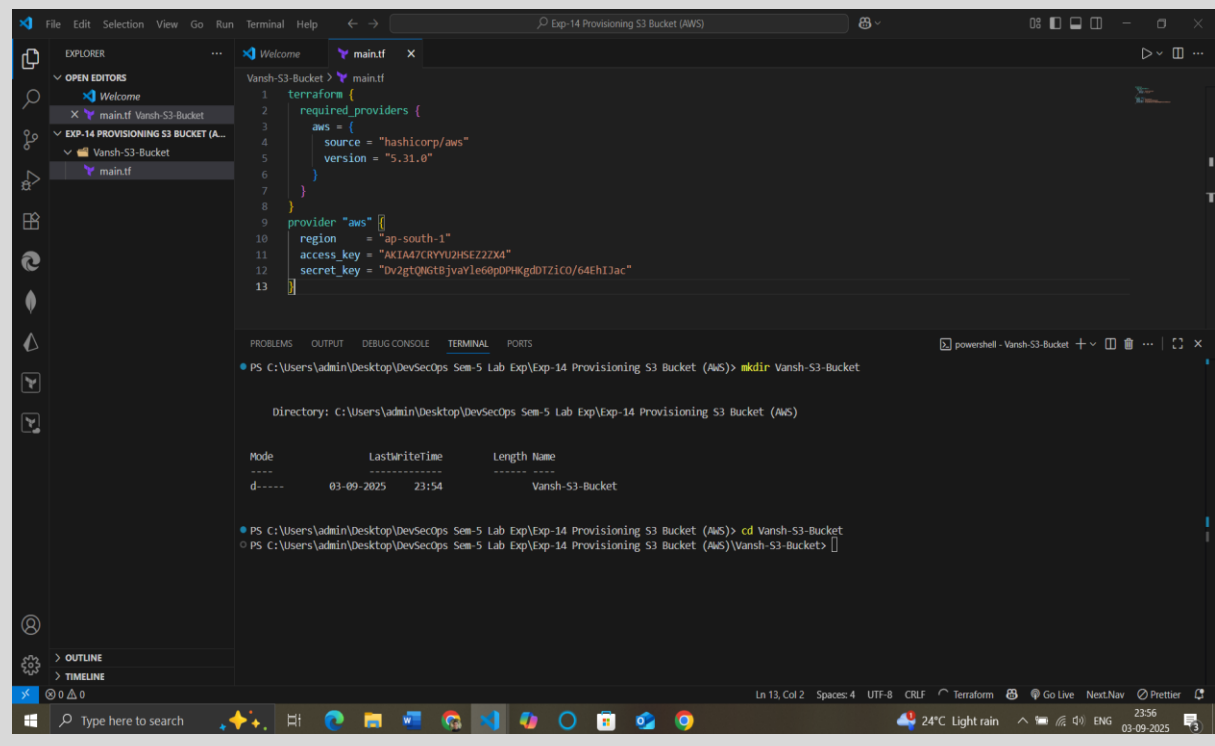
The Explorer pane on the left shows the file structure with 'Vansh-S3-Bucket' created under the 'EXP-14 PROVISION...' folder. The terminal window title is 'powershell - Vansh-S3-Bucket'.

## Step 2: Create the Terraform Configuration File (main.tf):

Create a file named main.tf with the following content:

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

provider "aws" {
  region     = "us-east-1" # Replace with your preferred region
  access_key = "your IAM access key" # Replace with your Access Key
  secret_key = "your secret access key" # Replace with your Secret Key
}
```



The screenshot shows a Visual Studio Code editor with a file named `main.tf` open. The file contains the Terraform configuration for the AWS provider. Below the editor, a terminal window is open, showing the output of the `mkdir` command, which creates a directory named `Vansh-S3-Bucket`.

```
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)> mkdir Vansh-S3-Bucket

Directory: C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)

Mode                LastWriteTime         Length Name
----                -
d-----          03-09-2025   23:54             Vansh-S3-Bucket

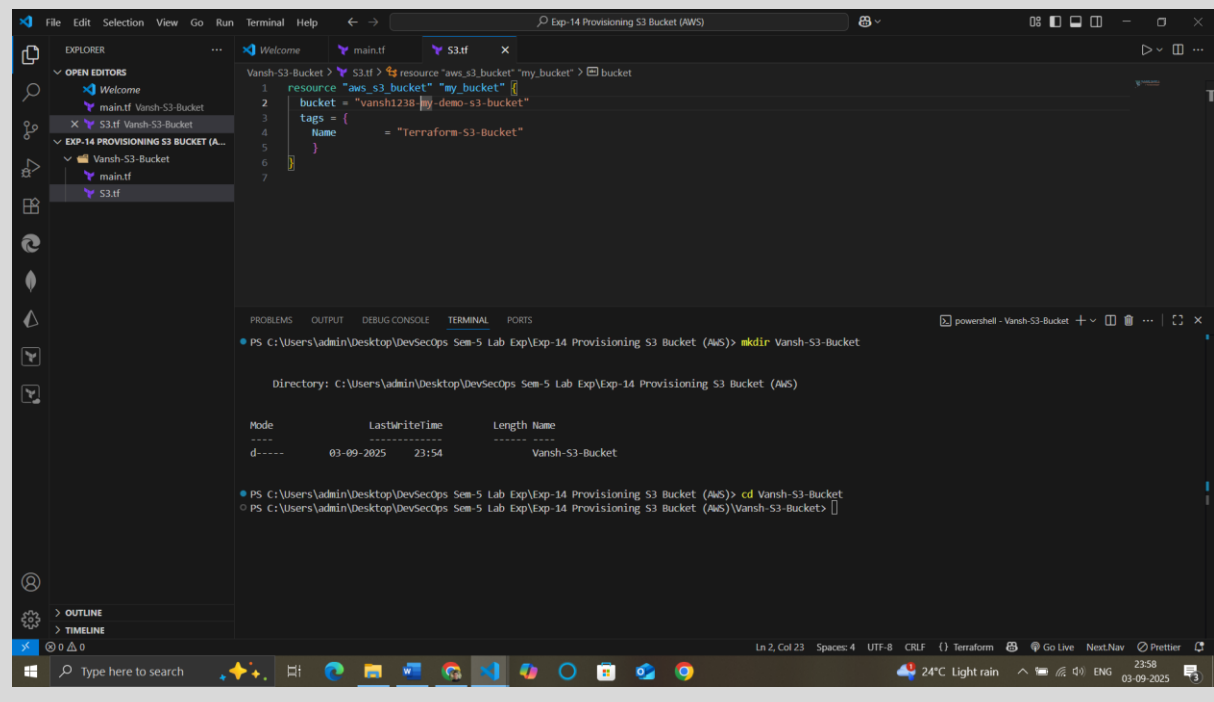
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)> cd Vansh-S3-Bucket
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)\Vansh-S3-Bucket>
```

This file sets up the Terraform AWS provider.

### Step 3: Create a Terraform Configuration File for the S3 Bucket (s3.tf):

Create another file named s3.tf with the following content:

```
resource "aws_s3_bucket" "my_bucket" {  
  bucket = "my-demo-s3-bucket"  
  tags = {  
    Name      = "Terraform-S3-Bucket"  
  }  
}
```

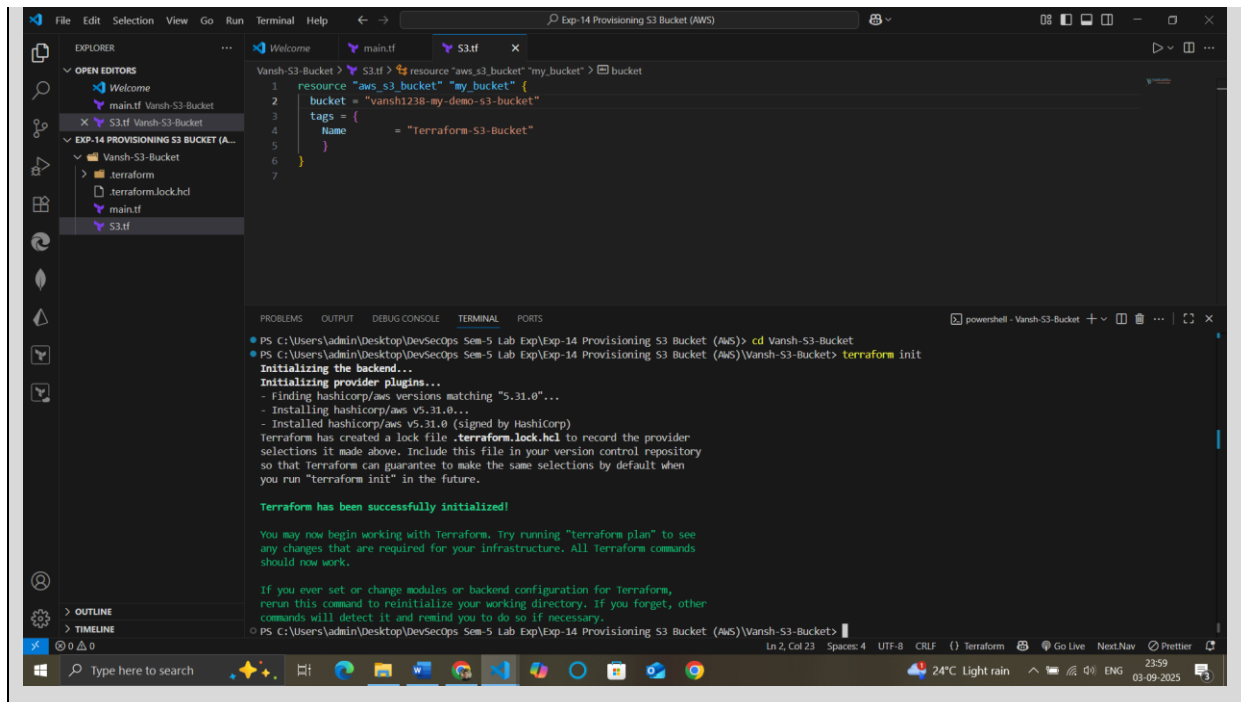


This file provisions an S3 bucket with a unique name using a random string suffix.

### Step 4: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

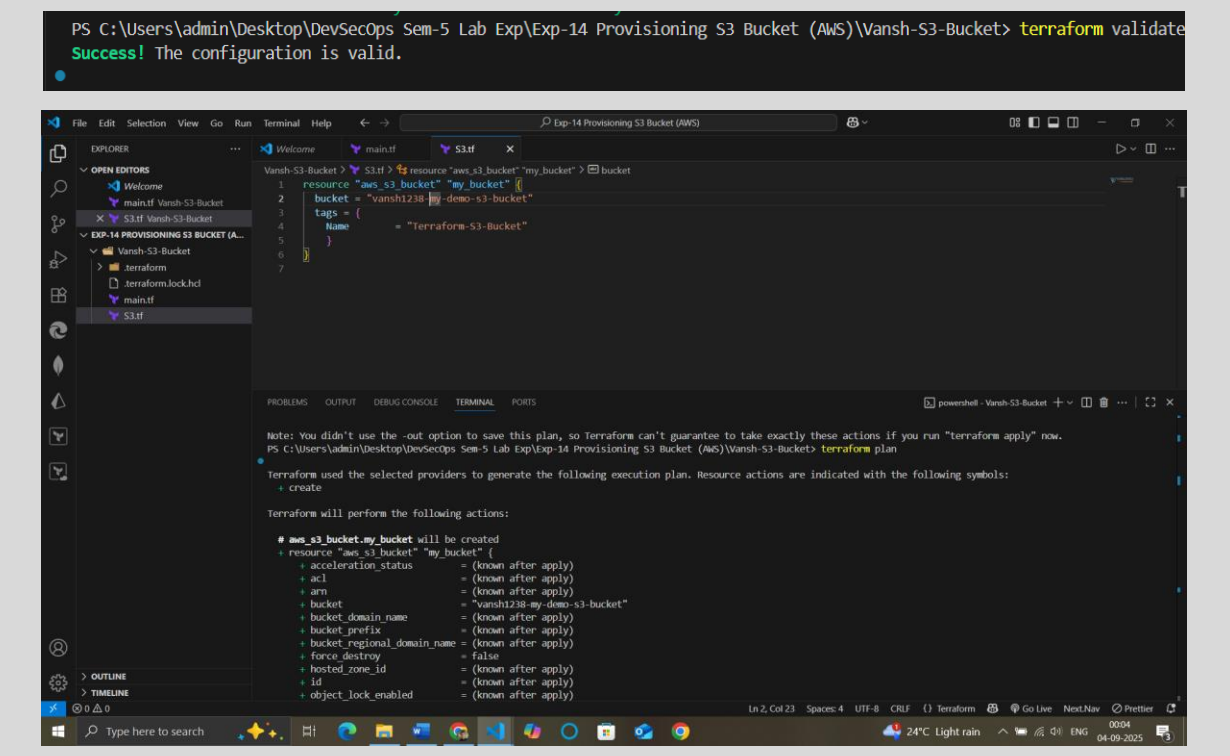
```
terraform init
```



## Step 5: Review the Plan:

Preview the changes Terraform will make:

### terraform plan



Review the output to ensure it meets your expectations.

## Step 6: Apply the Changes:

Create the resources:

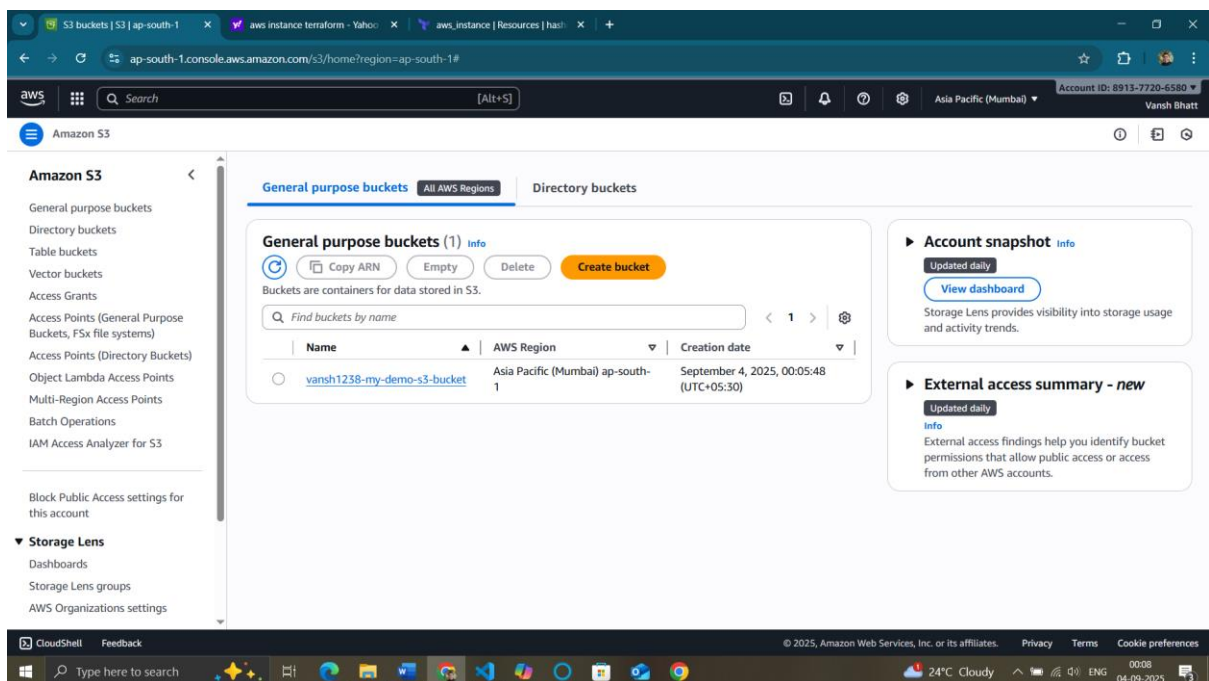
The first screenshot shows the initial state of the terminal. The title bar reads "Exp-14 Provisioning S3 Bucket (AWS)". The terminal content includes a note about the -out option, a list of providers (aws, localstack, null), and a detailed list of actions Terraform will perform for the resource "aws\_s3\_bucket.my\_bucket". The actions include creating the bucket with various attributes like acceleration\_status, acl, arn, bucket, bucket\_domain\_name, bucket\_prefix, bucket\_regional\_domain\_name, force\_destroy, hosted\_zone\_id, id, object\_lock\_enabled, policy, region, request\_payer, tags, tags\_all, website\_domain, website\_endpoint, cors\_rule, grant, and lifecycle\_rule.

The second screenshot shows the terminal after the apply command has been executed. The output shows the creation of the bucket: "aws\_s3\_bucket.my\_bucket: Creating..." followed by "aws\_s3\_bucket.my\_bucket: Creation complete after 5s [id=vanish1238-my-demo-s3-bucket]". The final summary states: "Apply complete! Resources: 1 added, 0 changed, 0 destroyed." The terminal prompt is now "PS C:\Users\Admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)\Vansh-S3-Bucket>".

When prompted, type yes to confirm.

## Step 7: Verify Resources:

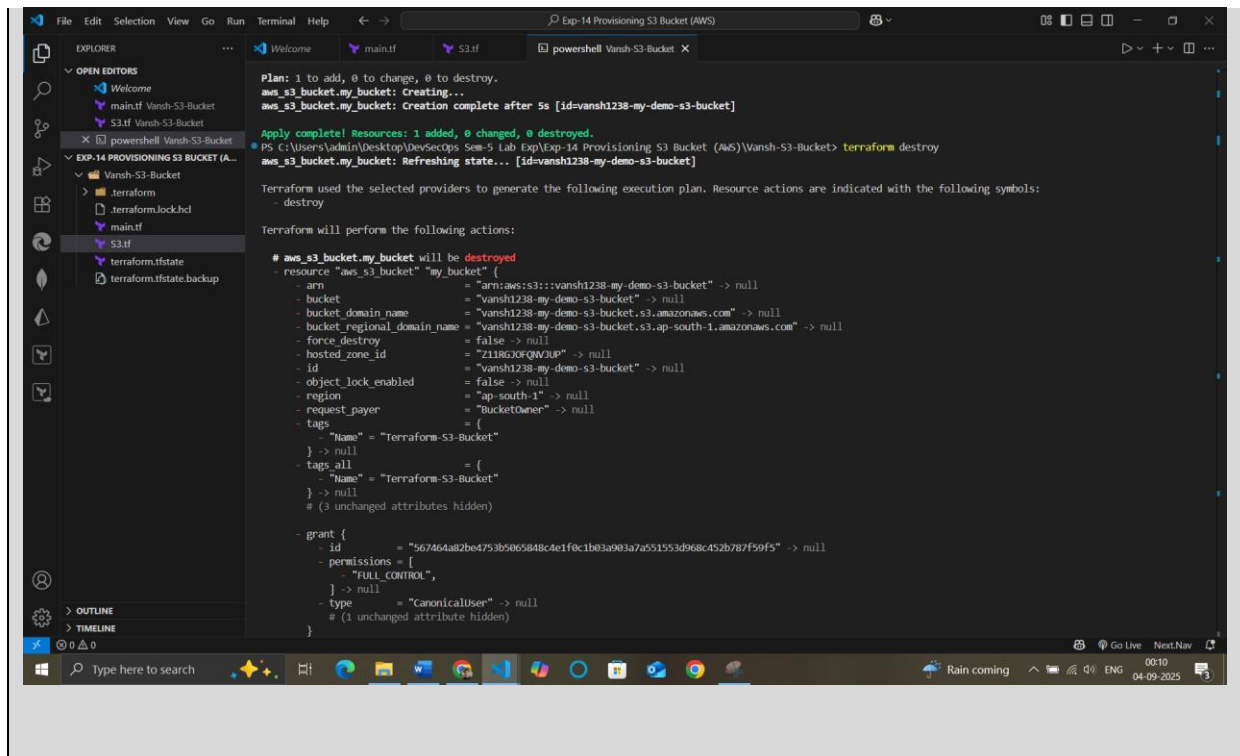
1. Log in to your AWS Management Console.
2. Navigate to the **S3** dashboard.
3. Verify that the S3 bucket has been created with the specified configuration.



## Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform destroy
```



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the project structure. The main editor shows the output of the Terraform destroy command. The output indicates that the bucket is being destroyed and provides details about the resource and its configuration.

```
Plan: 1 to add, 0 to change, 0 to destroy.
aws_s3_bucket.my_bucket: Creating...
aws_s3_bucket.my_bucket: Creation complete after 5s [id=vanshi238-my-demo-s3-bucket]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)\Vansh-S3-Bucket> terraform destroy
aws_s3_bucket.my_bucket: Refreshing state... [id=vanshi238-my-demo-s3-bucket]

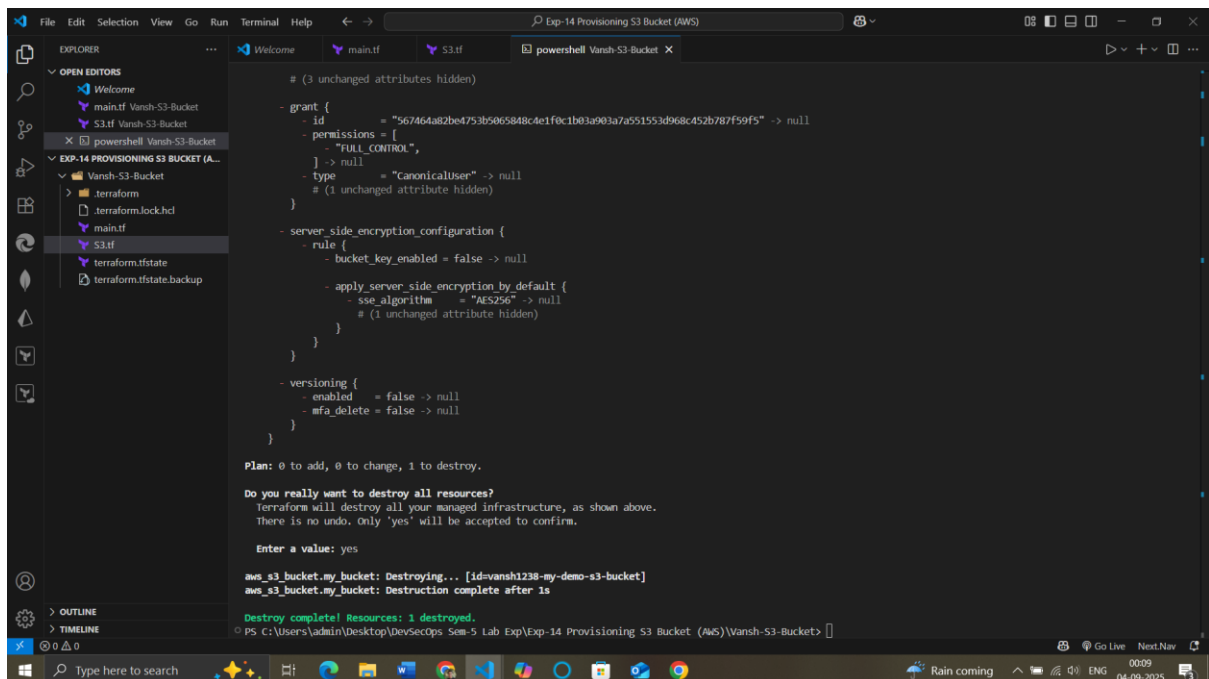
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_s3_bucket.my_bucket will be destroyed
- resource "aws_s3_bucket" "my_bucket" {
  - arn = "arn:aws:s3::vanshi238-my-demo-s3-bucket" -> null
  - bucket = "vanshi238-my-demo-s3-bucket" -> null
  - bucket_domain_name = "vanshi238-my-demo-s3-bucket.s3.amazonaws.com" -> null
  - bucket_regional_domain_name = "vanshi238-my-demo-s3-bucket.s3.ap-south-1.amazonaws.com" -> null
  - force_destroy = false -> null
  - hosted_zone_id = "Z1HC2RGW4UP" -> null
  - id = "vanshi238-my-demo-s3-bucket" -> null
  - object_lock_enabled = false -> null
  - region = "ap-south-1" -> null
  - request_payer = "BucketOwner" -> null
  - tags = {
    - "Name" = "Terraform-S3-Bucket"
  } -> null
  - tags_all = {
    - "Name" = "Terraform-S3-Bucket"
  } -> null
  # (3 unchanged attributes hidden)

  - grant {
    - id = "567464a82be4753b5865848c4e1fc1b03a903a7a551553d968c452b787f9f5" -> null
    - permissions = [
      - "FULL_CONTROL",
    ] -> null
    - type = "CanonicalUser" -> null
    # (1 unchanged attribute hidden)
  }
}
```

When prompted, type yes to confirm.



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the project structure. The main editor shows the output of the Terraform destroy command, including a confirmation prompt asking if the user really wants to destroy all resources. The user has entered 'yes' to confirm the destruction.

```
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_s3_bucket.my_bucket: Destroying... [id=vanshi238-my-demo-s3-bucket]
aws_s3_bucket.my_bucket: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-14 Provisioning S3 Bucket (AWS)\Vansh-S3-Bucket>
```