Lab Exercise 14–Provisioning an S3 Bucket on AWS

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
```

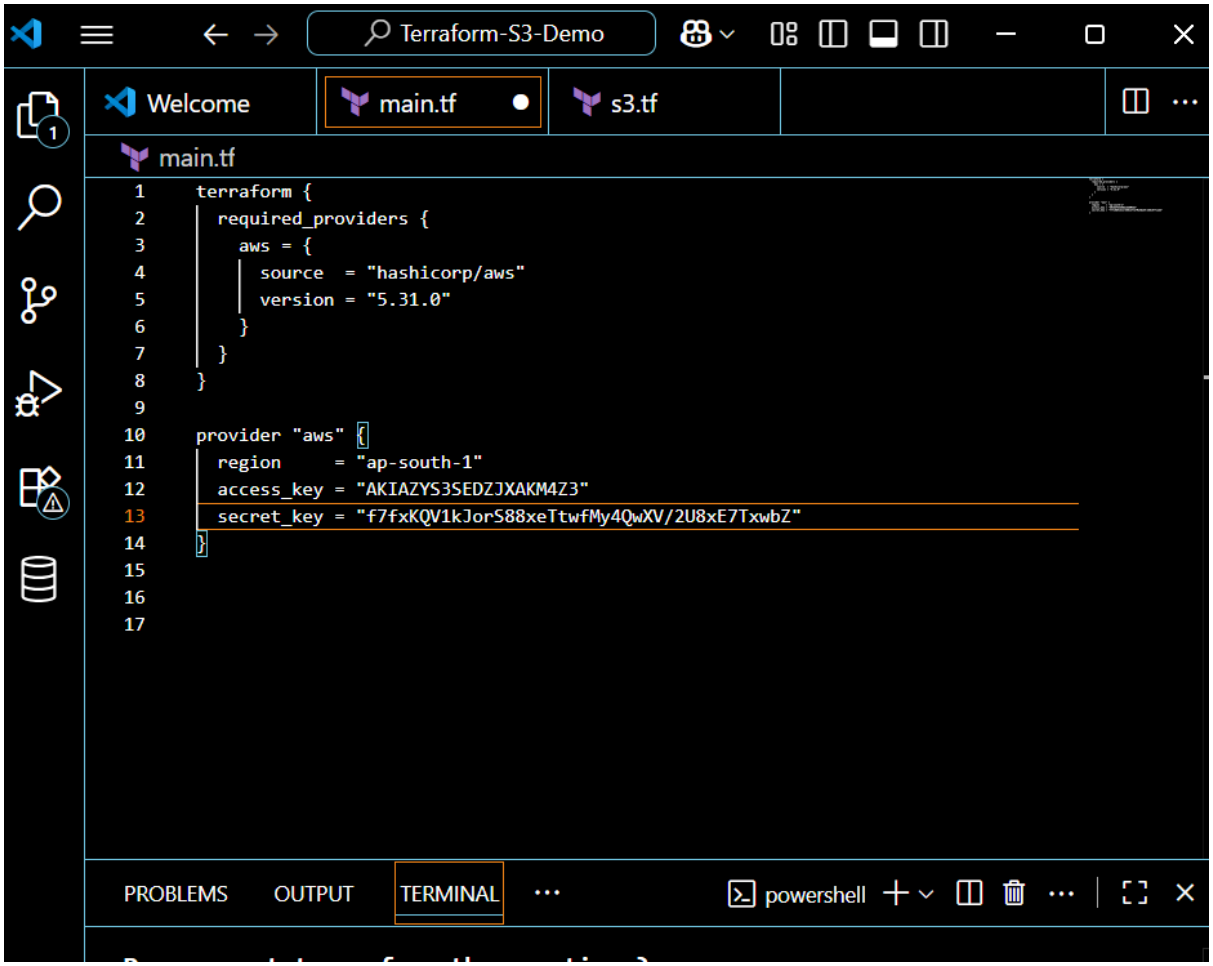
Step 2: Create the Terraform Configuration File (main.tf):

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
}
```

This file sets up the Terraform AWS provider.

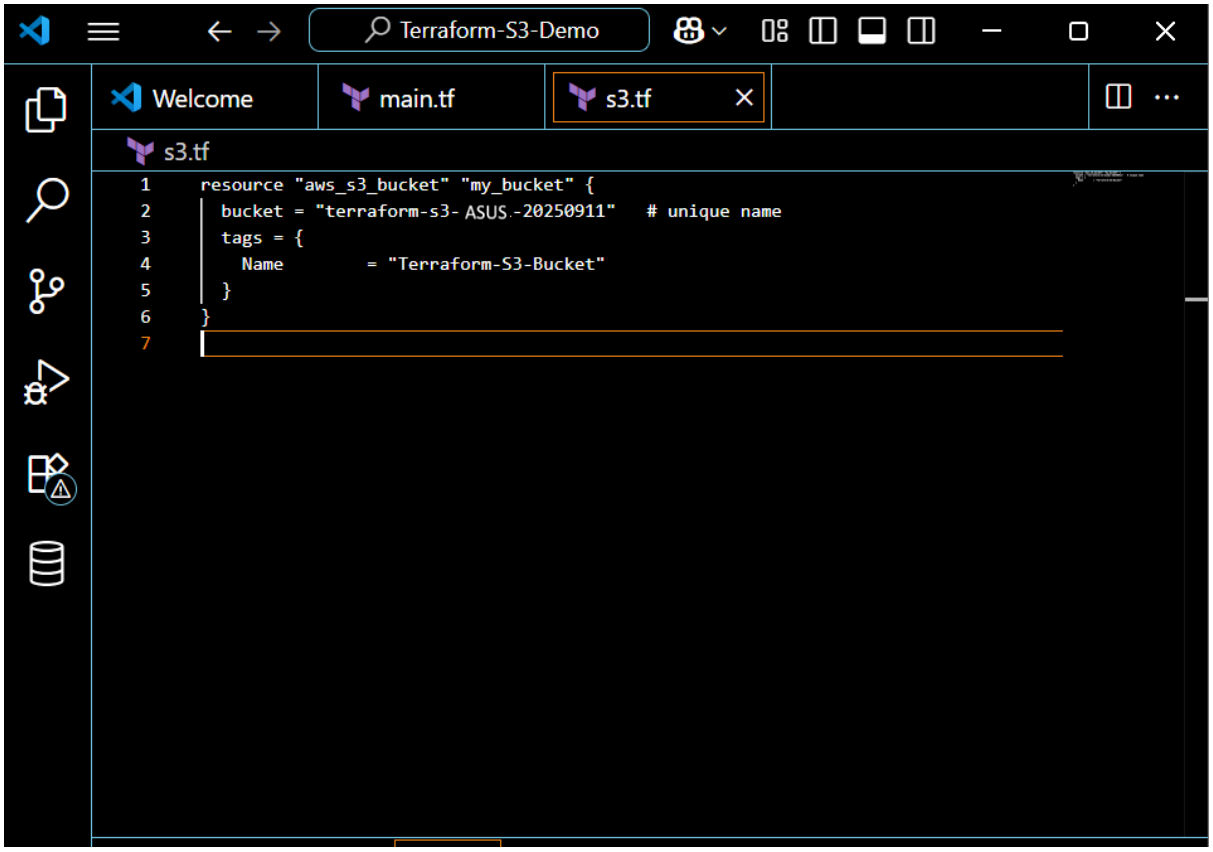


Step 3: Create a Terraform Configuration File for the S3 Bucket (s3.tf):

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
```

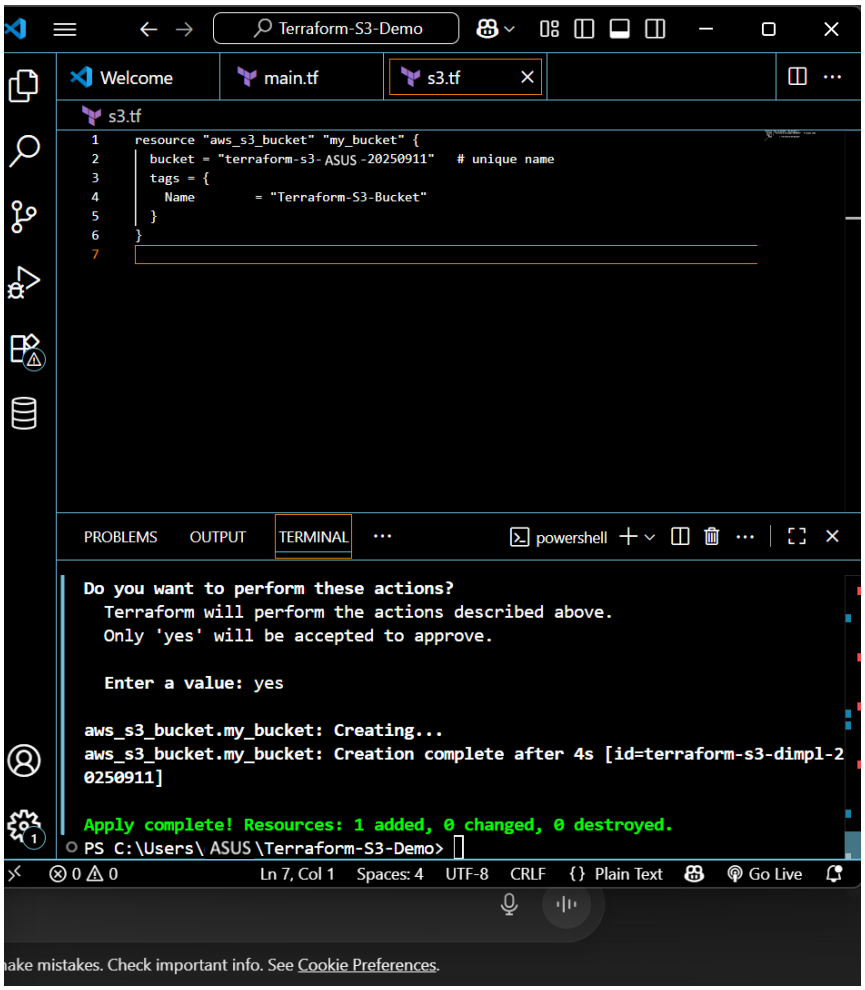
Review the output to ensure it meets your expectations.

Step 6: Apply the Changes:

Create the resources:

```
terraform apply
```

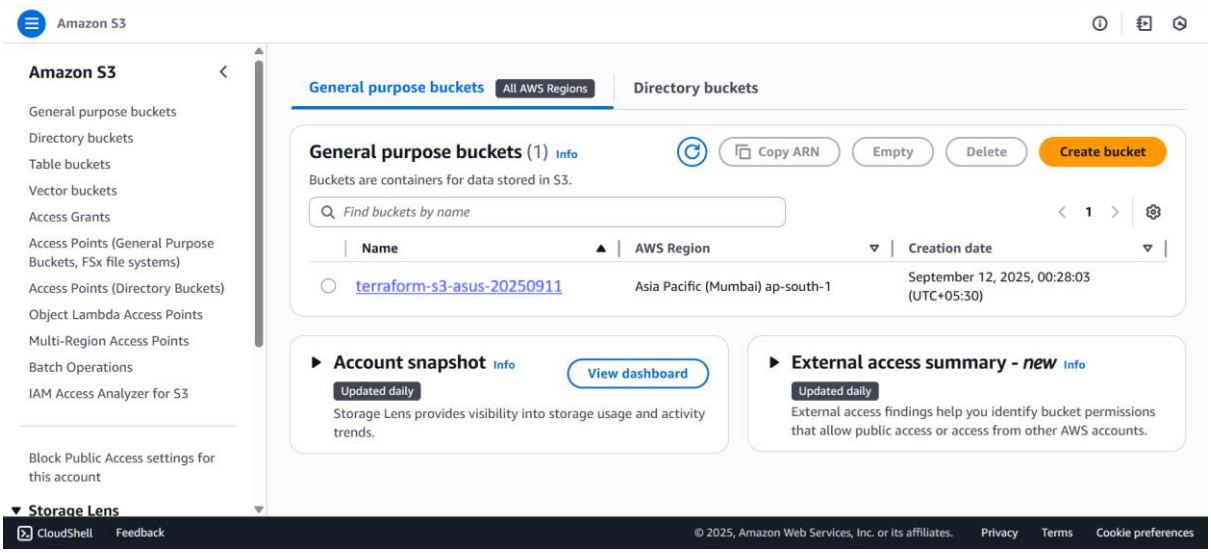
When prompted, type yes to confirm.



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.

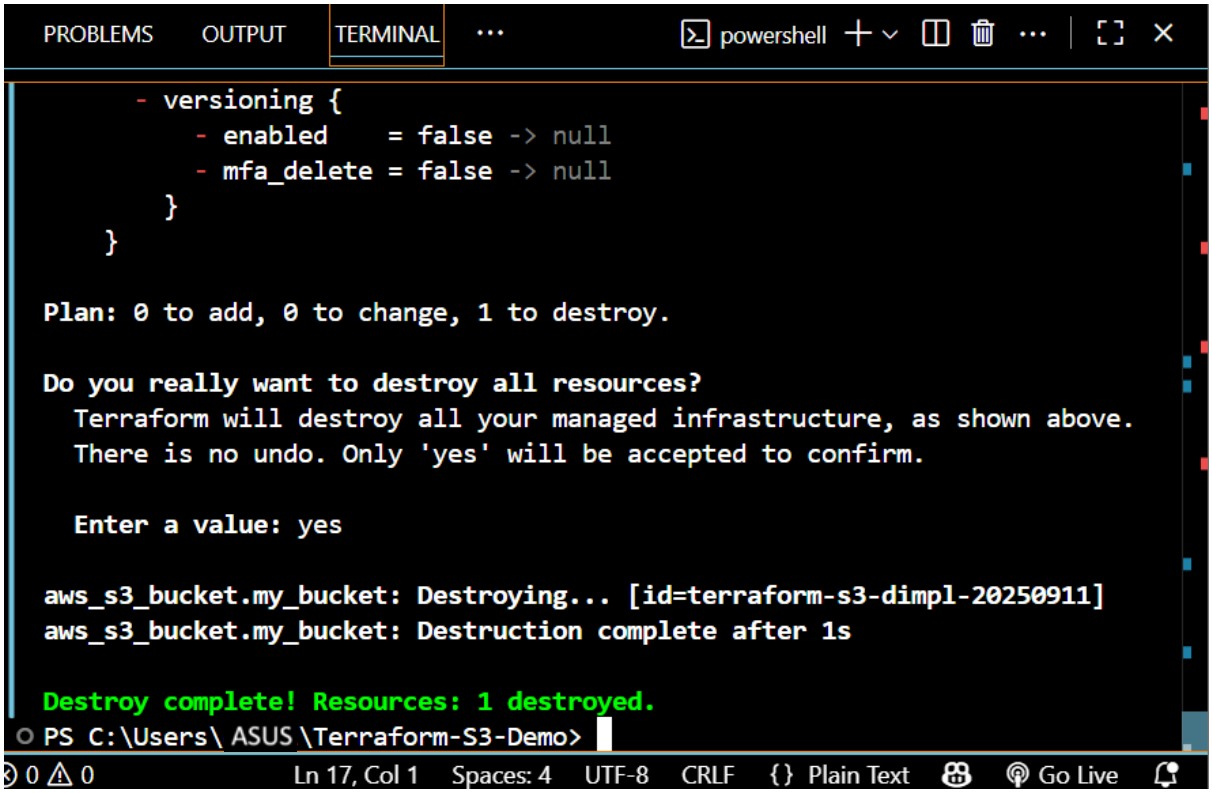


Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform destroy
```

When prompted, type yes to confirm.


PROBLEMS
OUTPUT
TERMINAL
powershell
- versioning {
- enabled = false -> null
- mfa_delete = false -> null
}
}
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=terraform-s3-dimpl-20250911]
aws_s3_bucket.my_bucket: Destruction complete after 1s
Destroy complete! Resources: 1 destroyed.
PS C:\Users\ ASUS \Terraform-S3-Demo>
0 0
Ln 17, Col 1
Spaces: 4
UTF-8
CRLF
{} Plain Text
Go Live