Lab Exercise 5–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:

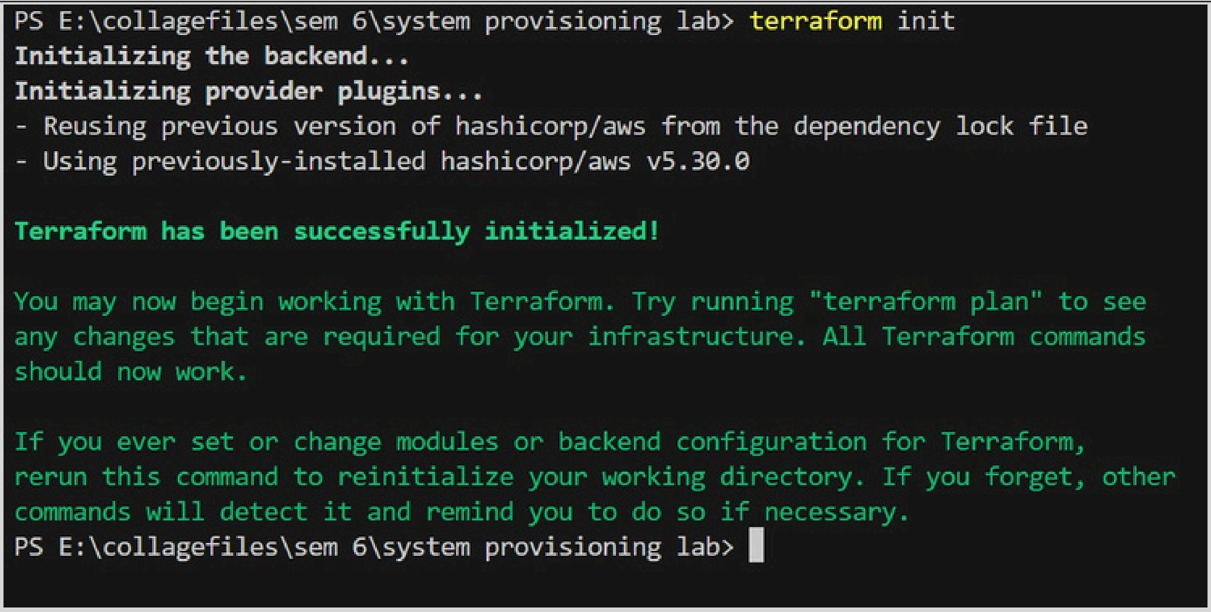
```
resource "aws_s3_bucket" "my_bucket" {
  bucket = "my-demo-s3-bucket"
  tags = {
    Name       = "Terraform-S3-Bucket"
    }
}
```

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
```

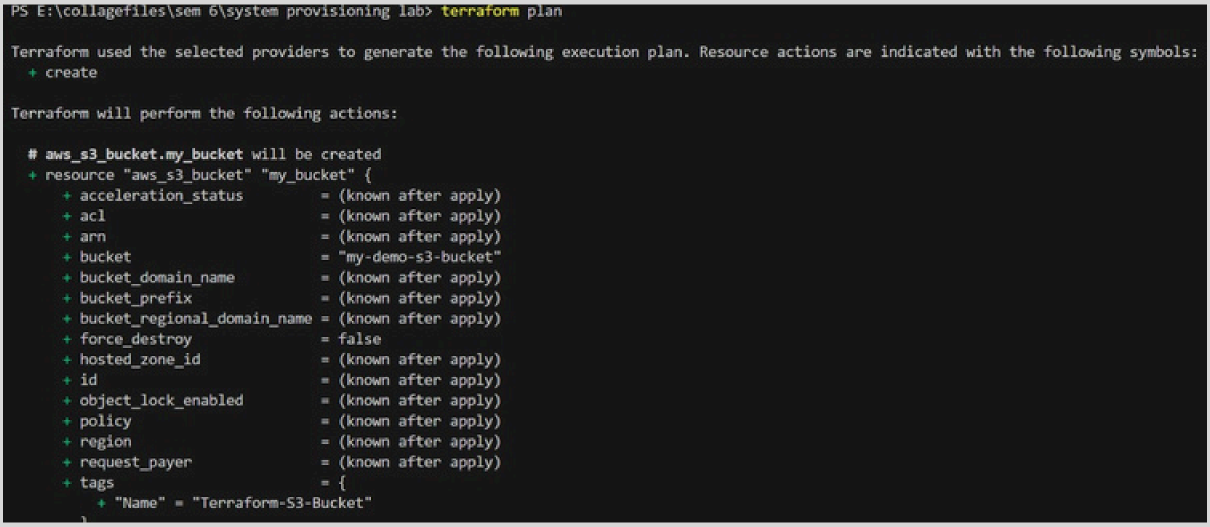


```
PS E:\collagefiles\sem 6\system provisioning lab> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.30.0

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.
PS E:\collagefiles\sem 6\system provisioning lab>
```

Step 5: Review the Plan:

Preview the changes Terraform will make:

```
terraform plan
```



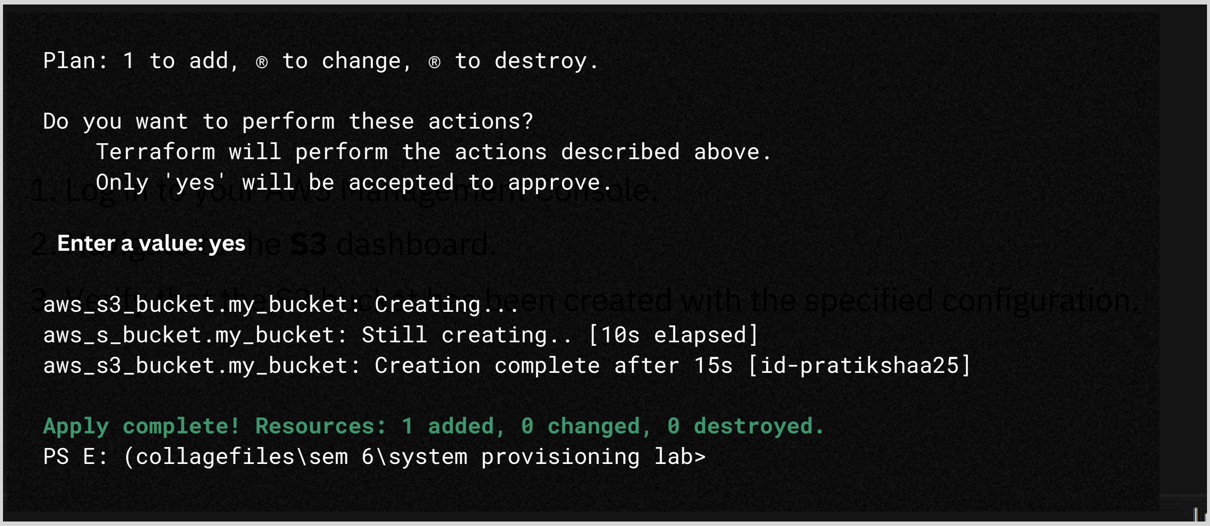
```
PS E:\collagefiles\sem 6\system provisioning lab> terraform plan

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_s3_bucket.my_bucket will be created
  + resource "aws_s3_bucket" "my_bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "my-demo-s3-bucket"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
      + object_lock_enabled         = (known after apply)
      + policy                      = (known after apply)
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags                        = {
          + "Name" = "Terraform-S3-Bucket"
```

Review the output to ensure it meets your expectations.

Step 6: Apply the Changes:

Create the resources:

```
terraform apply
```



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

Do you want to perform these actions?
    Terraform will perform the actions described above.
    Only 'yes' will be accepted to approve.

 Enter a value: yes

aws_s3_bucket.my_bucket: Creating...
aws_s_bucket.my_bucket: Still creating.. [10s elapsed]
aws_s3_bucket.my_bucket: Creation complete after 15s [id-pratikshaa25]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS E: (collagefiles\sem 6\system provisioning lab>
```

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.

When prompted, type yes to confirm.

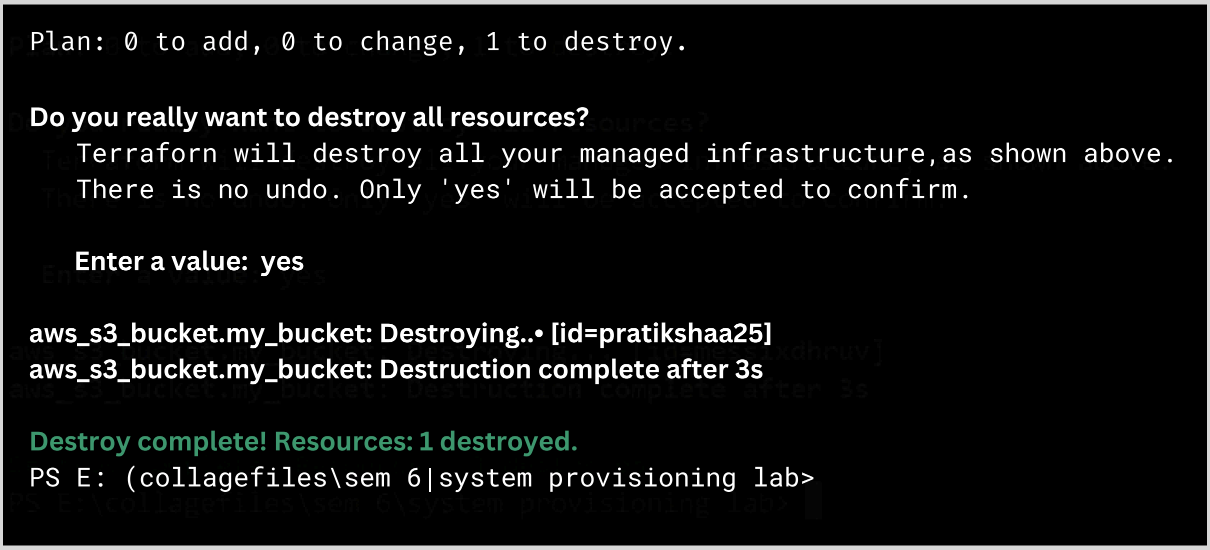
Step 7: Verify Resources:

○ pratikshaa25 US East (Ohio) us-east-2 View analyzer for us-east-2 January 17, 2025, 15:09:17 (UTC+05:30)

Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform destroy
```



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

Do you really want to destroy all resources?
   Terraforn 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=pratikshaa25]
aws_s3_bucket.my_bucket: Destruction complete after 3s

Destroy complete! Resources: 1 destroyed.
PS E: (collagefiles\sem 6|system provisioning lab>
```

When prompted, type yes to confirm.