

Lab Exercise 5–Provisioning an S3 Bucket on AWS

Aditya Tomar

500106015

R2142221060

Batch-2(DevOps)

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
```

```
Last login: Fri Jan 17 15:46:17 on ttys000
adityatomar@Adityas-MacBook-Air VS Code % mkdir Terraform-S3-Demo
adityatomar@Adityas-MacBook-Air VS Code % cd Terraform-S3-Demo
adityatomar@Adityas-MacBook-Air Terraform-S3-Demo % touch main.tf
```

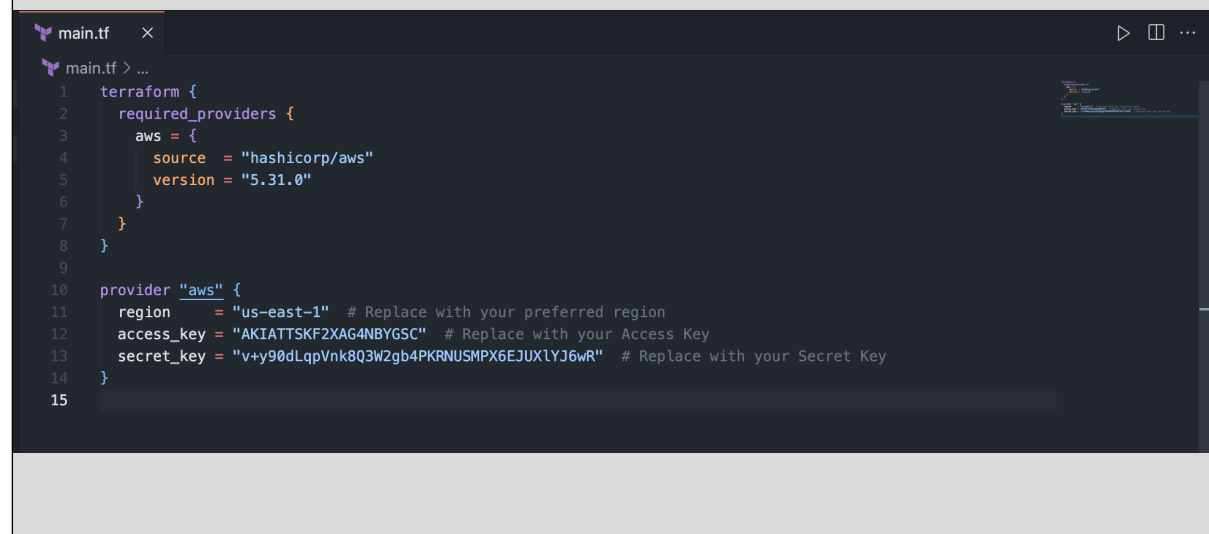
Step 2: Create the Terraform Configuration File (main.tf):

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
}
```

A screenshot of a code editor with a dark theme. The file is named 'main.tf'. The code defines the Terraform AWS provider configuration. It includes a 'required_providers' block for 'aws' with source 'hashicorp/aws' and version '5.31.0'. Below that is the 'provider "aws"' block with configuration for region, access_key, and secret_key, each with a comment indicating where to replace the values.

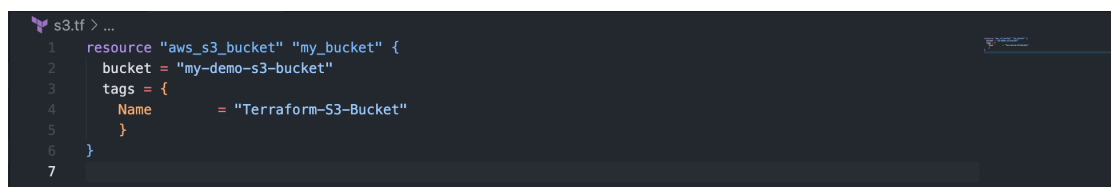
```
main.tf x
main.tf > ...
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region    = "us-east-1" # Replace with your preferred region
12   access_key = "AKIATTSKF2XAG4NBYGSC" # Replace with your Access Key
13   secret_key = "v+y90dLqpVnk8Q3W2gb4PKRNU SMPX6EJUXLYJ6wR" # Replace with your Secret Key
14 }
15
```

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"
  }
}
```

A screenshot of a code editor with a dark theme. The file is named 's3.tf'. The code defines an 'aws_s3_bucket' resource named 'my_bucket' with a bucket name 'my-demo-s3-bucket' and a tag 'Name' with value 'Terraform-S3-Bucket'.

```
s3.tf > ...
1 resource "aws_s3_bucket" "my_bucket" {
2   bucket = "my-demo-s3-bucket"
3   tags = {
4     Name      = "Terraform-S3-Bucket"
5   }
6 }
7
```

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
```

```
adityatomar@Adityas-MacBook-Air Terraform-S3-Demo % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

Step 5: Review the Plan:

Preview the changes Terraform will make:

```
terraform plan
```

```
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"
  }
  + tags_all            = {
    + "Name" = "Terraform-S3-Bucket"
  }
  + website_domain      = (known after apply)
  + website_endpoint    = (known after apply)

  + cors_rule (known after apply)
  + grant (known after apply)
  + lifecycle_rule (known after apply)
  + logging (known after apply)
```

Review the output to ensure it meets your expectations.

Step 6: Apply the Changes:

Create the resources:

terraform apply


```
Enter a value: yes

random_string.suffix: Creating...
random_string.suffix: Creation complete after 0s [id=gxc2zp]
aws_s3_bucket.my_bucket: Creating...
aws_s3_bucket.my_bucket: Creation complete after 5s [id=my-demo-s3-bucket-gxc2zp]
```

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.

General purpose buckets (1) Info All AWS Regions  Copy ARN Empty Delete Create bucket

Buckets are containers for data stored in S3.

☐

[my-demo-s3-bucket-gxc2zp](#)

AWS Region

Europe (Ireland) eu-west-1


IAM Access Analyzer

[View analyzer for eu-west-1](#)

Creation date

January 17, 2025, 22:22:53 (UTC+05:30)

< 1 >



Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform 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=my-demo-s3-bucket-gxc2zp]
aws_s3_bucket.my_bucket: Destruction complete after 2s
random_string.suffix: Destroying... [id=gxc2zp]
random_string.suffix: Destruction complete after 0s
```

When prompted, type yes to confirm.
