Lab Exercise 5–Provisioning an S3 Bucket on AWS

Exercise Steps:

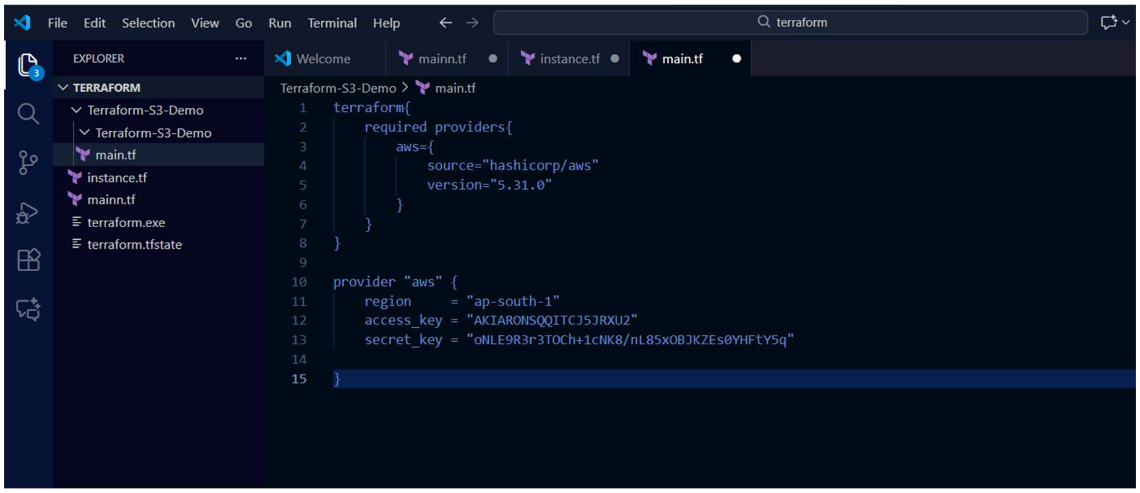
Step 1: Create a New Directory:

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:

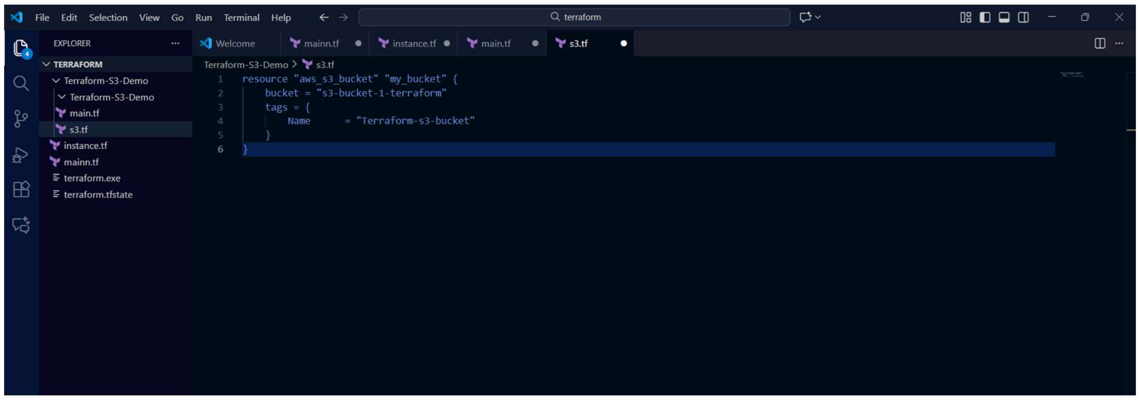


```
terraform{
    required providers{
        aws={
            source="hashicorp/aws"
            version="5.31.0"
        }
    }
}

provider "aws" {
    region     = "ap-south-1"
    access_key = "AKIARONSQQITCJ5JRXU2"
    secret_key = "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"

}
```

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 = "s3-bucket-1-terraform"
    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:

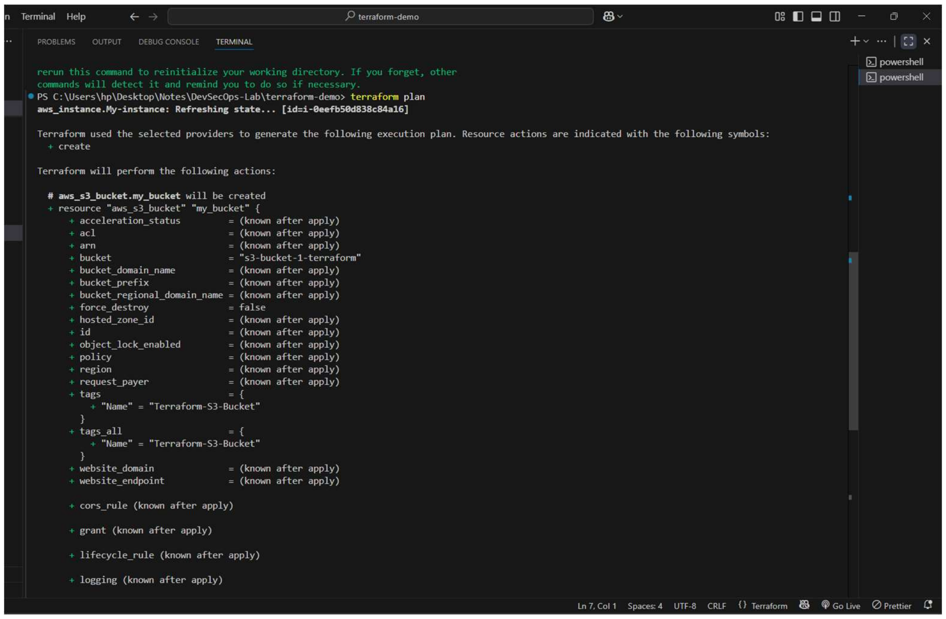
```
terraform init
```

Step 5: Review the Plan:

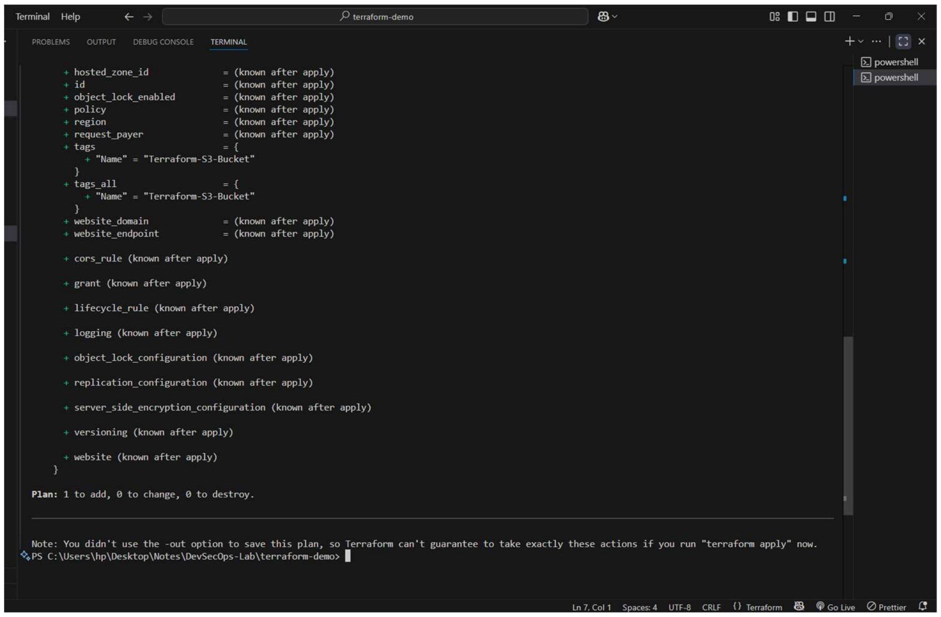
Preview the changes Terraform will make:

```
terraform plan
```

Output:



```
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 C:\Users\hp\Desktop\Notes\DevSecOps-Lab\terraform-demo> terraform plan
aws_instance.My-instance: Refreshing state... [id=i-0eefb50d838c84a16]

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                      = "s3-bucket-1-terraform"
      + 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)
```



```
      + 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)

      + object_lock_configuration (known after apply)

      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

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

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\hp\Desktop\Notes\DevSecOps-Lab\terraform-demo>
```

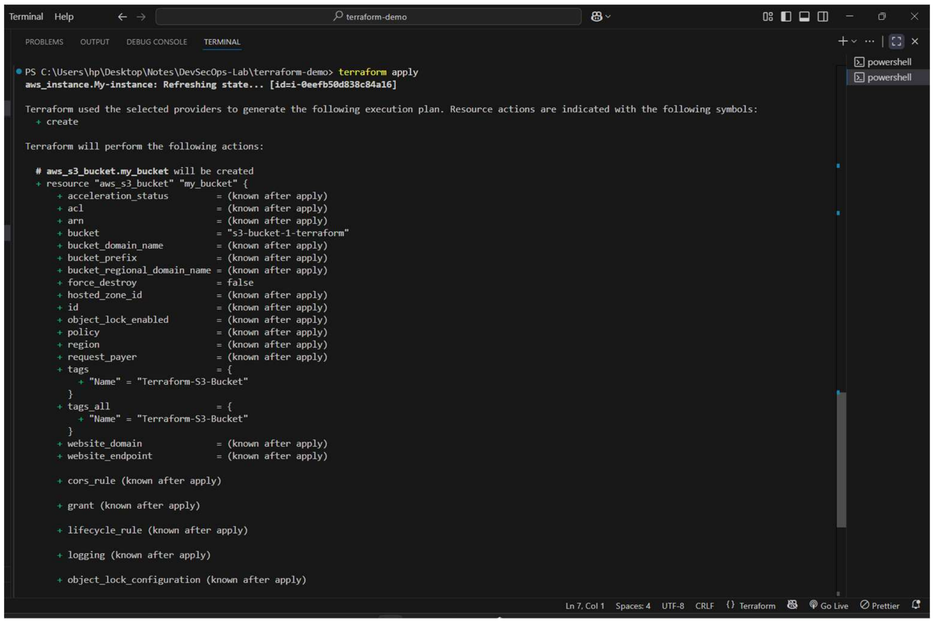
Step 6: Apply the Changes:

Create the resources:

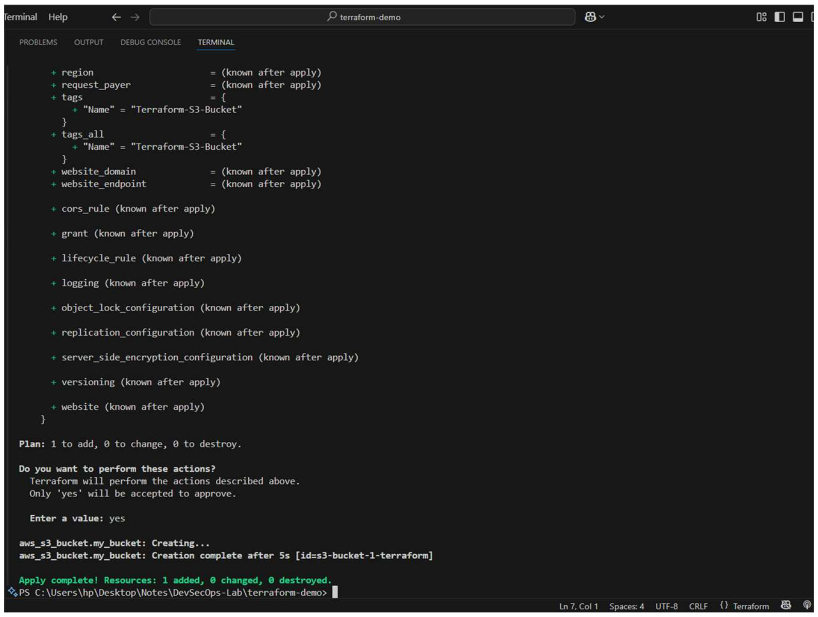
```
terraform apply
```

When prompted, type yes to confirm.

Output:



```
PS C:\Users\hp\Desktop\Notes\DevSecOps-Lab\terraform-demo> terraform apply
aws_instance.My-instance: Refreshing state... [id=i-0eefb50d838c84a16]

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                      = "s3-bucket-1-terraform"
      + 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)

      + object_lock_configuration (known after apply)
```

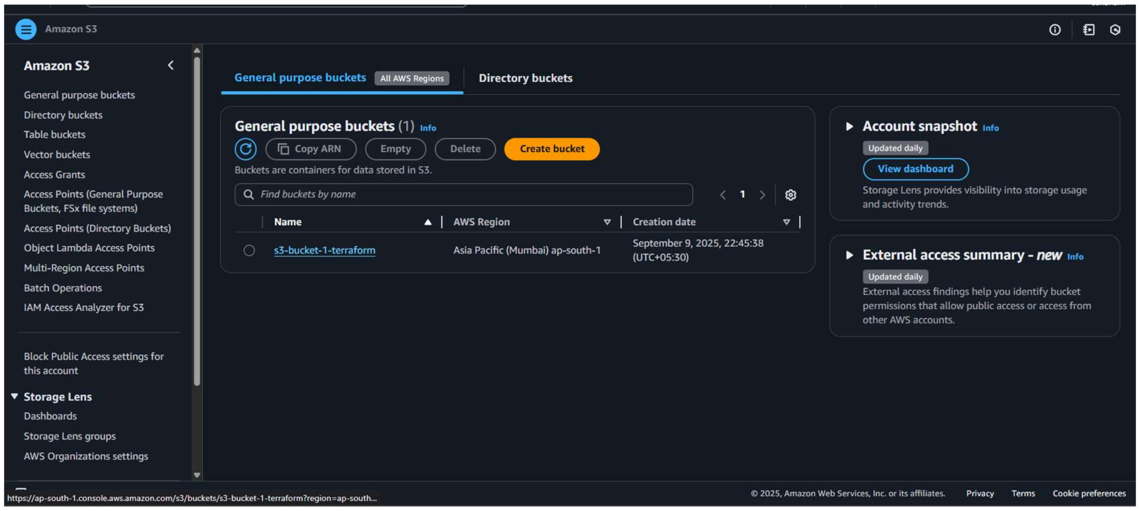


```
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags                        = {
          + "Name" = "Terraform-S3-Bucket"
        }
      + tags_all                    = {
          + "Name" = "Terraform-S3-Bucket"
        }
      + 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)

      + object_lock_configuration (known after apply)

      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

Plan: 1 to add, 0 to change, 0 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_s3_bucket.my_bucket: Creation complete after 5s [id=s3-bucket-1-terraform]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\hp\Desktop\Notes\DevSecOps-Lab\terraform-demo>
```

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.

Output:



Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
s3-bucket-1-terraform	Asia Pacific (Mumbai) ap-south-1	September 9, 2025, 22:45:38 (UTC+05:30)

Storage Lens provides visibility into storage usage and activity trends.

External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

https://ap-south-1.console.aws.amazon.com/s3/buckets/s3-bucket-1-terraform?region=ap-south...

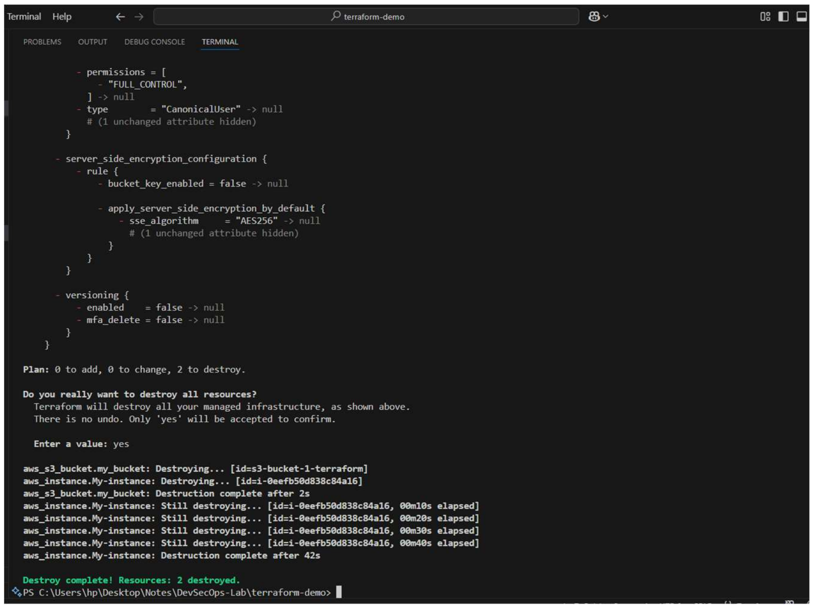
© 2025, Amazon Web Services, Inc. or its affiliates.

Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform destroy
```

When prompted, type yes to confirm.



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

      - server_side_encryption_configuration {
          - rule {
              - bucket_key_enabled = false -> null

              - apply_server_side_encryption_by_default {
                  - sse_algorithm     = "AES256" -> null
                    # (1 unchanged attribute hidden)
                }
            }
        }

      - versioning {
          - enabled    = false -> null
          - mfa_delete = false -> null
        }
    }

Plan: 0 to add, 0 to change, 2 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=s3-bucket-1-terraform]
aws_instance.My-instance: Destroying... [id=i-0eefb50d838c84a16]
aws_s3_bucket.my_bucket: Destruction complete after 2s
aws_instance.My-instance: Still destroying... [id=i-0eefb50d838c84a16, 00m10s elapsed]
aws_instance.My-instance: Still destroying... [id=i-0eefb50d838c84a16, 00m20s elapsed]
aws_instance.My-instance: Still destroying... [id=i-0eefb50d838c84a16, 00m30s elapsed]
aws_instance.My-instance: Still destroying... [id=i-0eefb50d838c84a16, 00m40s elapsed]
aws_instance.My-instance: Destruction complete after 42s

Destroy complete! Resources: 2 destroyed.
PS C:\Users\hp\Desktop\Notes\DevSecOps-Lab\terraform-demo>
```

Resources destroyed.