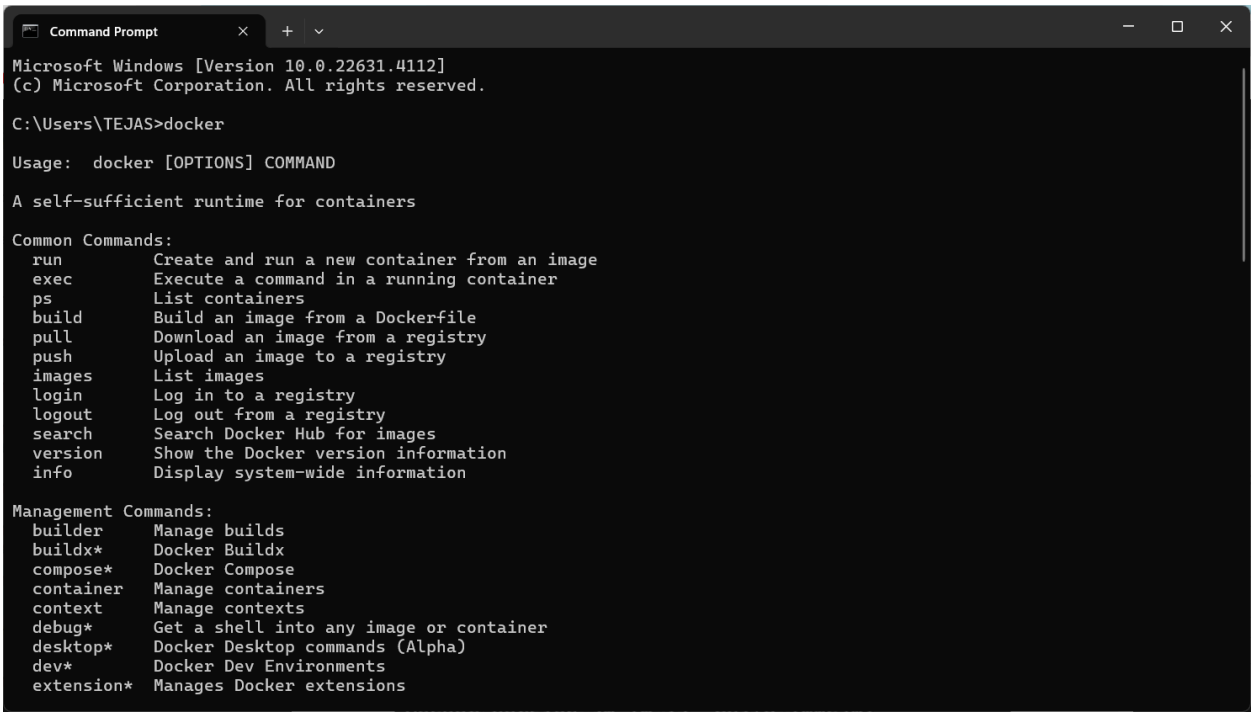
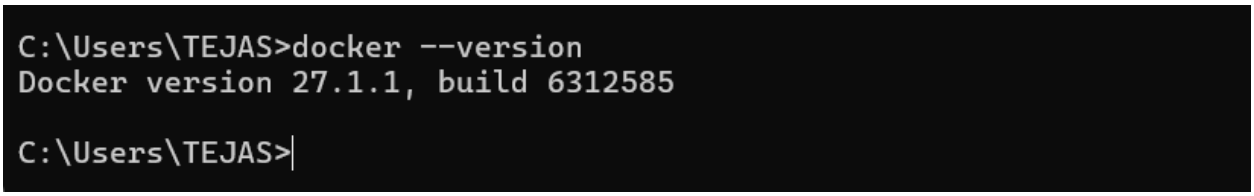
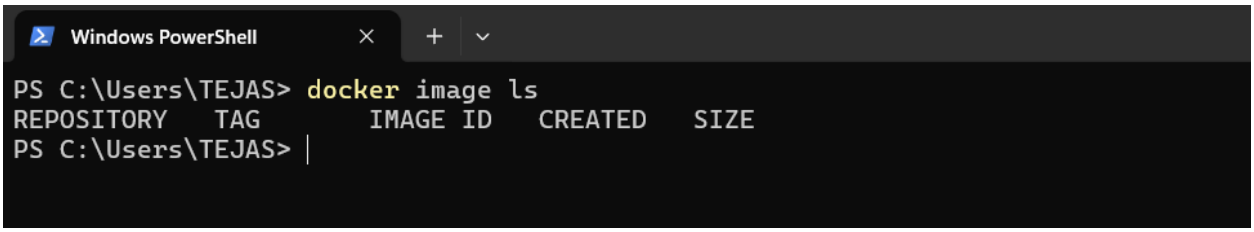
# PRACTICAL NO: - 06

## A] Create Docker Image using Terraform

- Checking the Docker Functionality



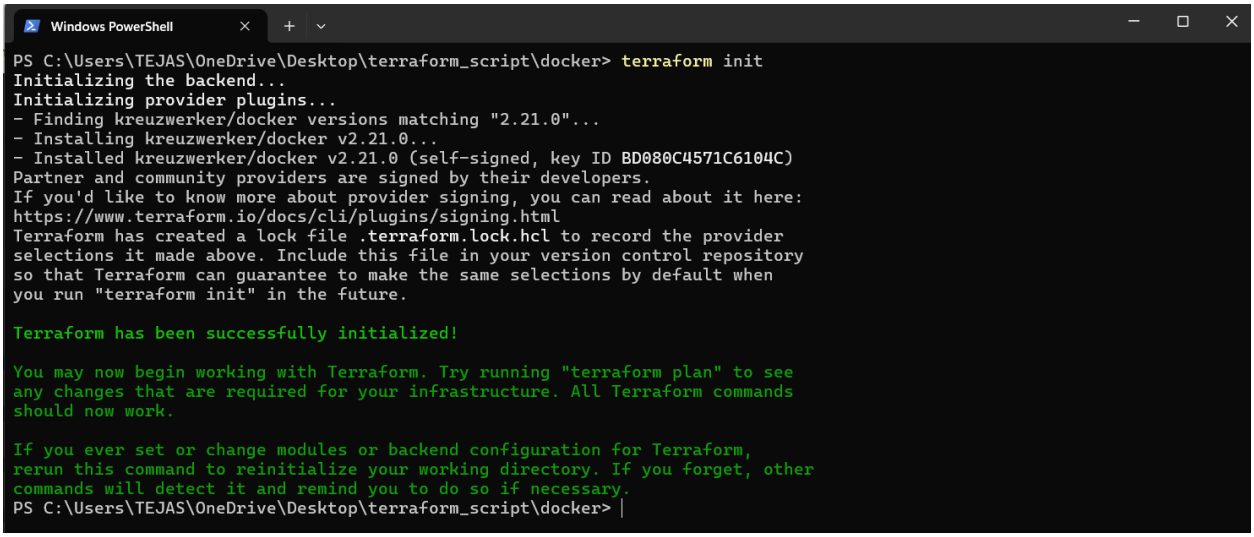

- Creation of docker.tf file

Create a folder named ‘Terraform Scripts’ in which we save our different types of scripts. Then, create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.
Then create a new docker.tf file to create a Ubuntu Linux container.

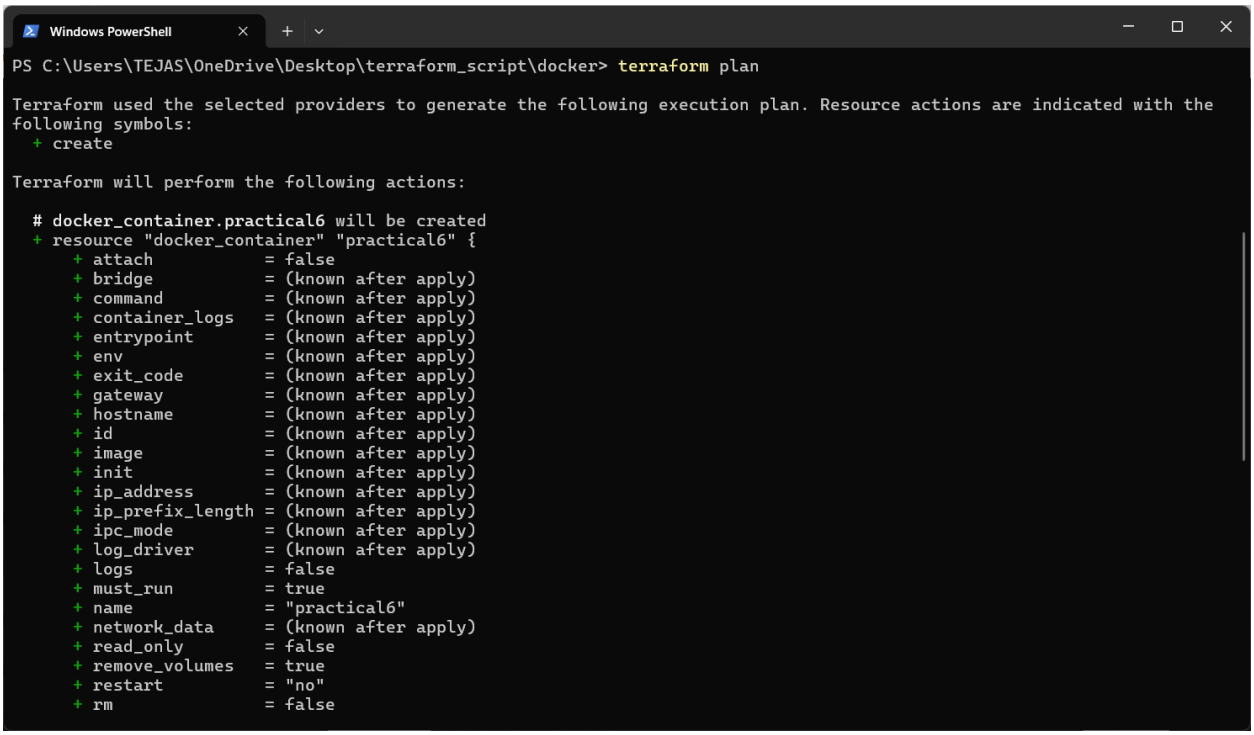
docker.tf > ...

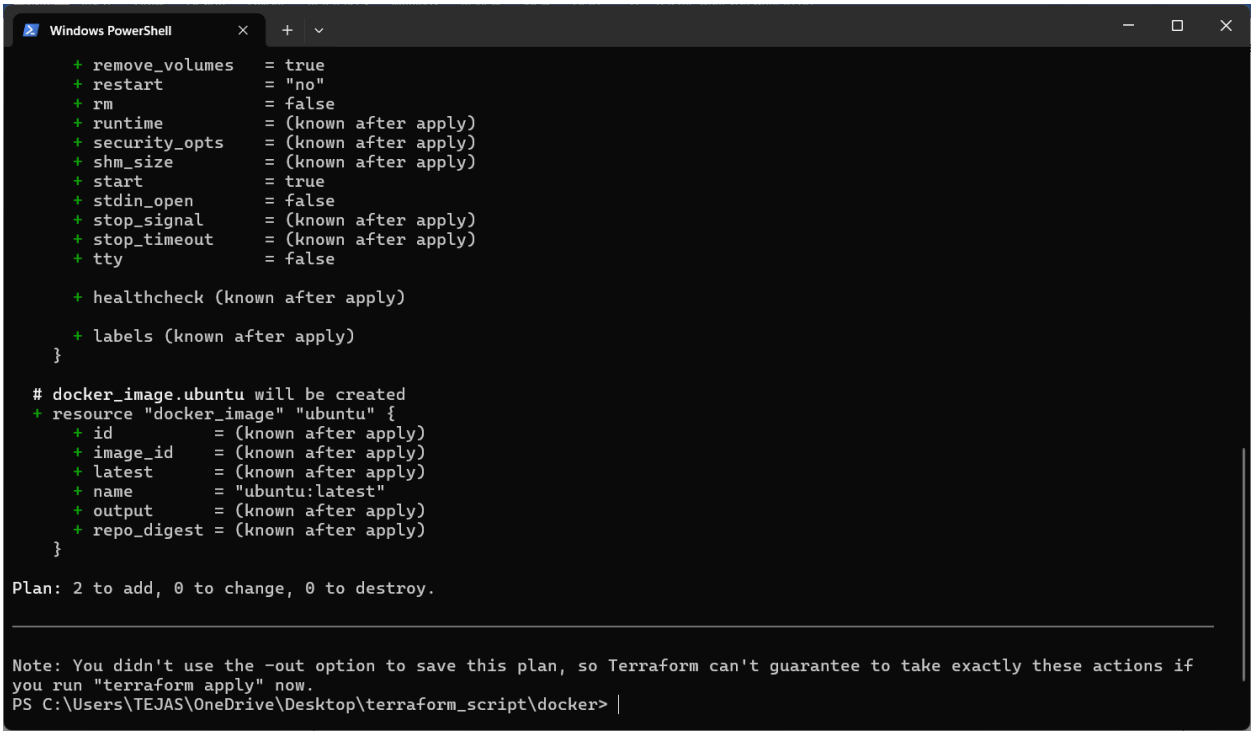
```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

provider "docker" {
  host = "npipe:////.//pipe//docker_engine"
}

resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

resource "docker_container" "practical6" {
  image = docker_image.ubuntu.image_id
  name  = "practical6"
}
```

- Execute Terraform Init command to initialize the resources

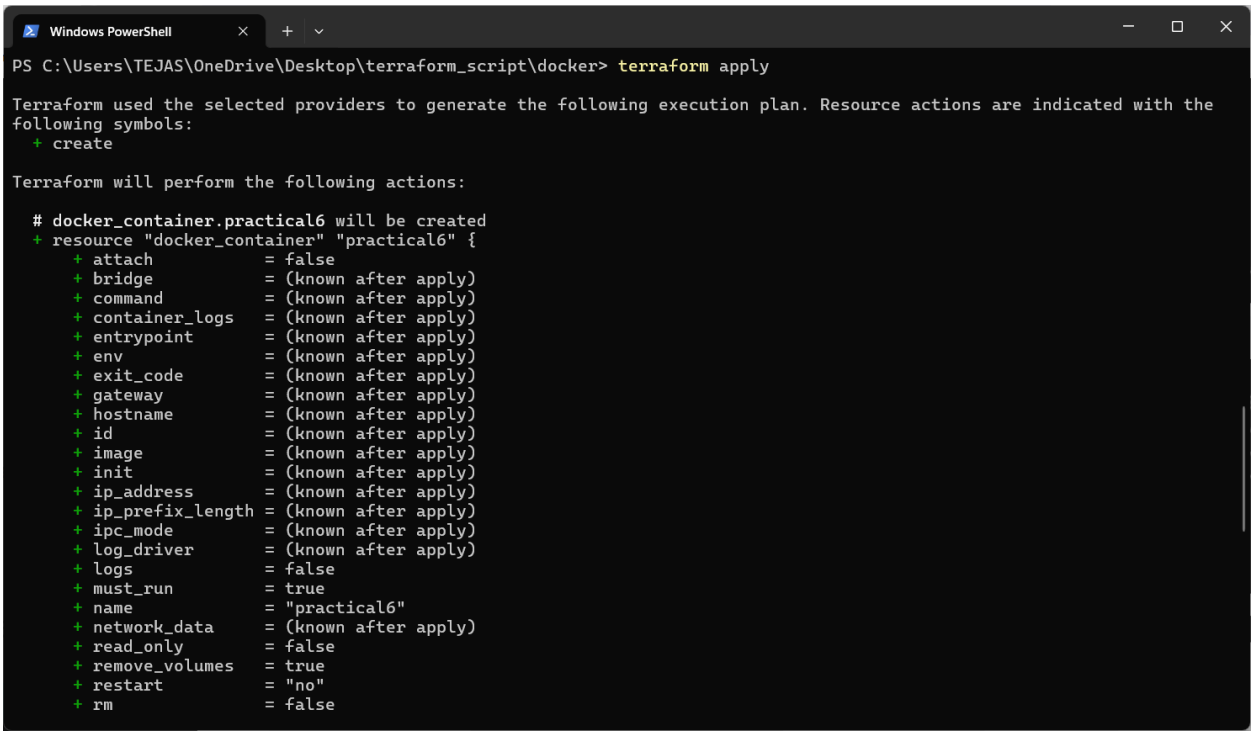


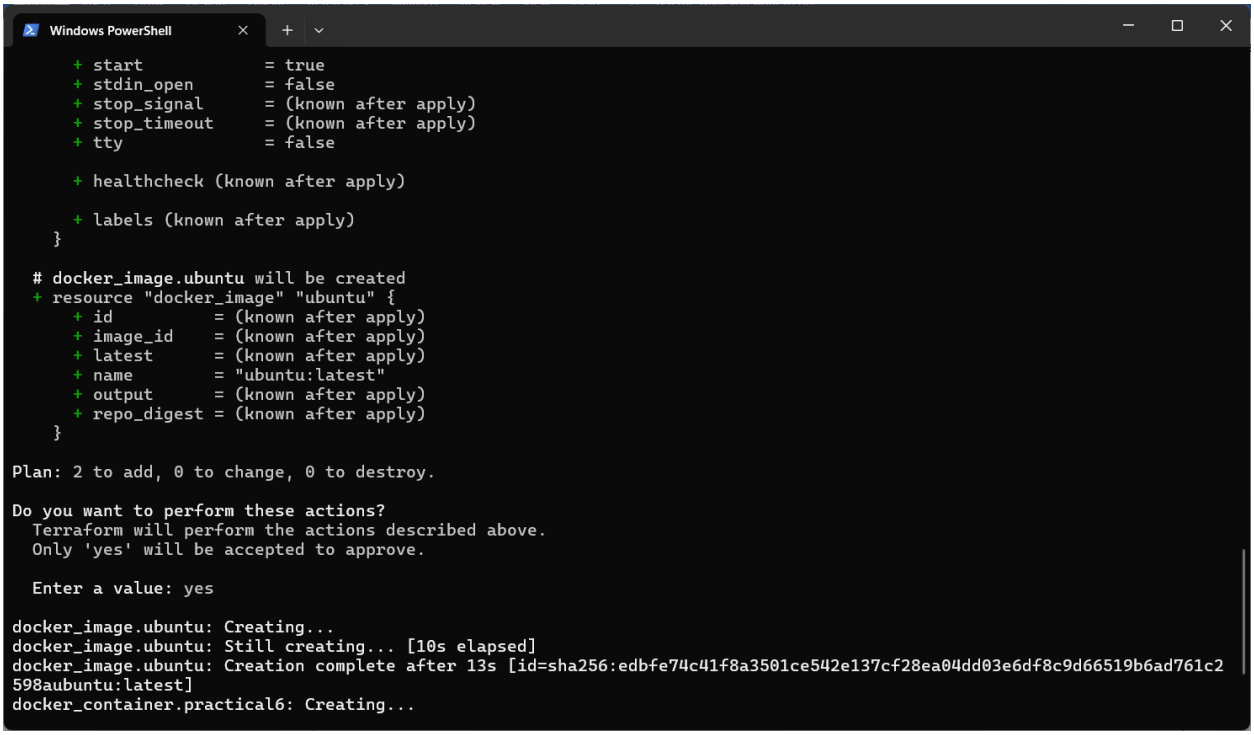

- Execute Terraform plan to see the available resources




- Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration.




- Docker images, Before Executing Apply step:

```
Windows PowerShell
PS C:\Users\TEJAS> docker image ls
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
PS C:\Users\TEJAS>
```

- Docker images, After Executing Apply step:

```
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    edbfe74c41f8   6 weeks ago   78.1MB
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker>
```

- Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
Windows PowerShell
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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:

  # docker_image.ubuntu will be destroyed
  - resource "docker_image" "ubuntu" {
      - id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
      - image_id    = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - latest      = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - name        = "ubuntu:latest" -> null
      - repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> 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

docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker>
```

- Docker images After Executing Destroy step

```
Destroy complete! Resources: 1 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker>
```

B] Creating S3 Bucket using terraform.

- Write a Terraform Script in Atom for creating S3 Bucket on Amazon AWS

aws-s3 > main.tf > ...

```
terraform {
  required_providers {
    aws={
        source="hashicorp/aws"
        version="5.64.0"
    }
    random = {
      source = "hashicorp/random"
      version = "3.6.2"
    }
  }
}

resource "random_id" "rand_id"{
    byte_length = 8
}

resource "aws_s3_bucket" "demo-bucket" {
  bucket = "demo-bucket-${random_id.rand_id.hex}"
}

resource "aws_s3_object" "bucket-data" {
  bucket = aws_s3_bucket.demo-bucket.bucket
  source = "./myfile.txt"
  key="newfile.txt"
}
```

- Create a new provider.tf file

aws-s3 > provider.tf > ...

```
provider "aws" {
  access_key="ASIAXDVAWUCTAHPBYALZ"
  secret_key="kdP24c5nfLcCbuWQe49jBthzJc4RefZBns6t6J8y"
  token="IQoJb3JpZ2luX2VjEJf//////////wEaCXVzLXdlc3QtMiJIMEYCIQCSJ3i5FhAaZU/xRQHF5L2DoNBiAuOG0PAemX7fCnC2bwIhAPIRX10U0Af6Bz3Qs7oX8Cnpn62XK7LhDo3/
  u4vx66juKsACCMD//////////wEQABoMNDg4ODg5NTU3MTU4IgzPDJgqgUUyZ9njLWoqlAKlv6HhXOV2Vhev/w51dDaiTqa9/
  lTgri1VOfAFoZhGHRcK5yuOAho5IXYG0thTErBx8N1n0i8fQFbqnI/dB/pEMSk02xh/5PYBrdulaHVSWUvV86LPqZzJ0wTS4iTn+l6fi6PyhM4kZtbPTPixhAjZGJuvNYzrPbe6t6lhGsw/
  Rsv51B9lHEjyVyW7LDQsXvz+YX4ie5DtO9+Xbxh+WMWAp2+GucNkQOKRg+IbkdRPf5njQHJ818h3xQYD72tIGmWwXZgBHrobKI5zVkKFzMseofXImgj1hlZQ8zGof9KOUlxdivnRpFzH
  +FhYPV/cCEM1jE8M1euRcDlRnaD5E7PdAg3k7F8FnMAfCu4qck9uhQCJOo0wqZyRtwY6nAGhyr0X3yVQrC1EWK3j6HbY5MOJC4lThme7fNTs5HyKfDVSFUGZkRzoJspESepbn/
  YnNlOAadFZCEbY4ayUfSSEhMnKHyZdl85xewRnp8kZTKSdxWzqyuTjoi/dL1vfq0B5jPSdJ4J0v453YIsyZmXRnWdh8AnYg2zAyw3kC5CdipQLDEti37zP20nciYcA1Oy+iF+e3Cw/
  vcgFR08="
  region = "us-east-1"
}
\end{code}
```

- Execute Terraform Init command to initialize the resources

```
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical>
cd aws-s3
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/aws v5.64.0
- Using previously-installed hashicorp/random v3.6.2

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 C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3>
```

- Execute Terraform plan to see the available resources

```
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> terraform plan
random_id.rand_id: Refreshing state... [id=mMJ9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

  # aws_s3_bucket.demo-bucket has been deleted
  - resource "aws_s3_bucket" "demo-bucket" {
      - bucket                      = "demo-bucket-98c27d2fdb4b7fe7" -> null
        id                          = "demo-bucket-98c27d2fdb4b7fe7"
        # (12 unchanged attributes hidden)

        # (3 unchanged blocks hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to
undo or respond to these changes.

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

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.demo-bucket will be created
  + resource "aws_s3_bucket" "demo-bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "demo-bucket-98c27d2fdb4b7fe7"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
```

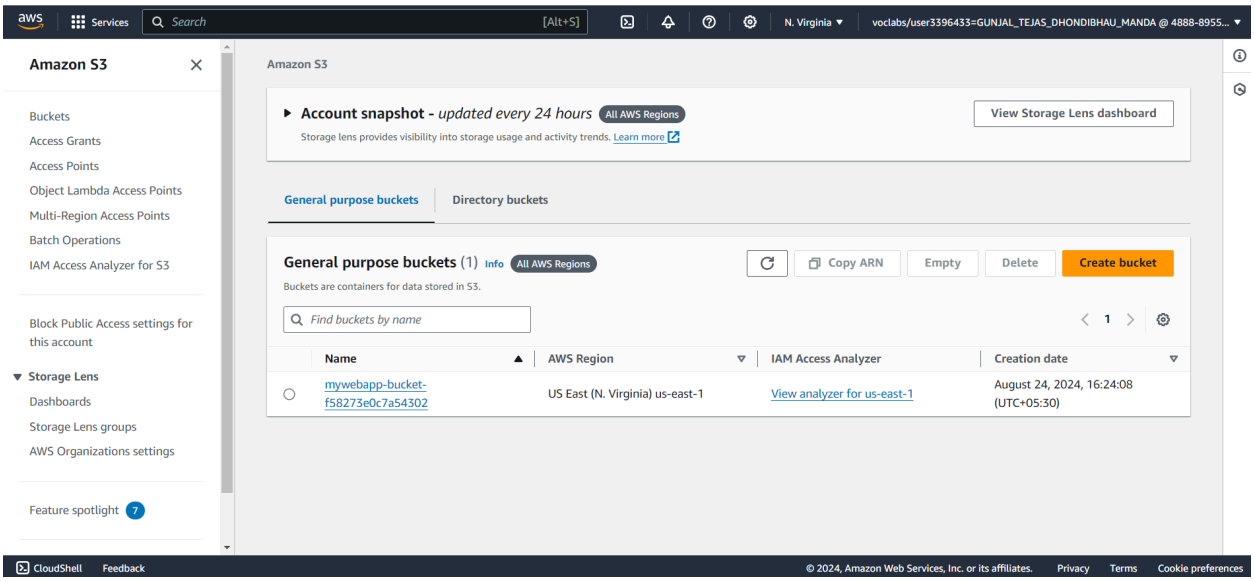
```
      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

  # aws_s3_object.bucket-data will be created
  + resource "aws_s3_object" "bucket-data" {
      + acl                    = (known after apply)
      + arn                    = (known after apply)
      + bucket                 = "demo-bucket-98c27d2fdb4b7fe7"
      + bucket_key_enabled     = (known after apply)
      + checksum_crc32         = (known after apply)
      + checksum_crc32c        = (known after apply)
      + checksum_sha1          = (known after apply)
      + checksum_sha256        = (known after apply)
      + content_type           = (known after apply)
      + etag                   = (known after apply)
      + force_destroy          = false
      + id                     = (known after apply)
      + key                    = "newfile.txt"
      + kms_key_id             = (known after apply)
      + server_side_encryption = (known after apply)
      + source                 = "./myfile.txt"
      + storage_class          = (known after apply)
      + tags_all               = (known after apply)
      + version_id             = (known after apply)
    }

Plan: 2 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\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3>
```

- Execute Terraform apply to apply the configuration, which will automatically create an S3 bucket based on our configuration.

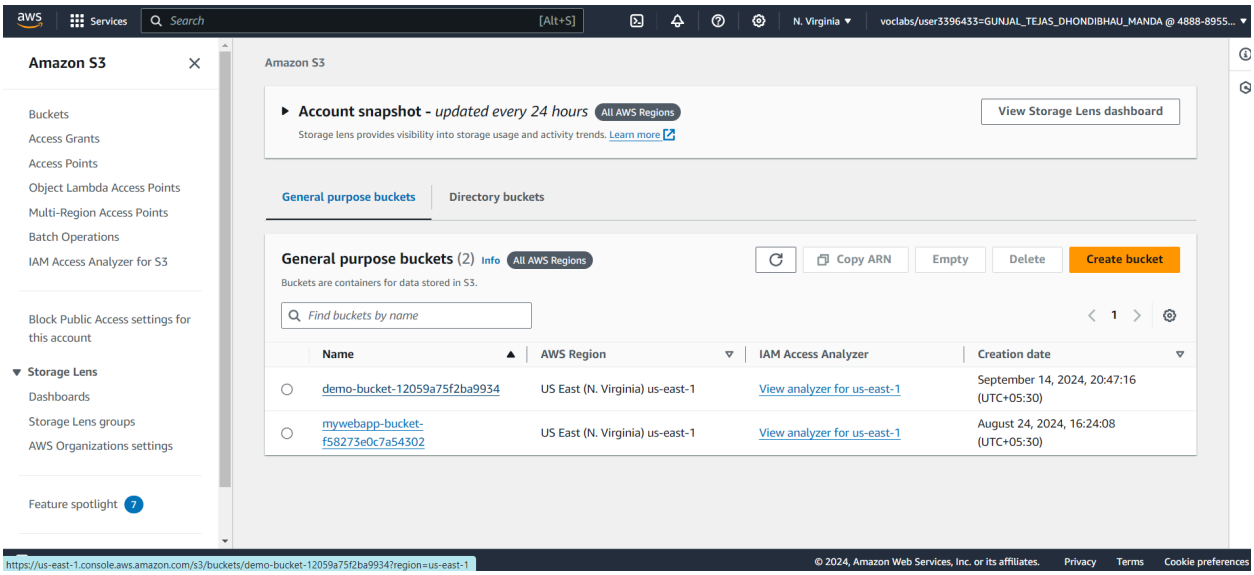
```
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> terraform apply
random_id.rand_id: Refreshing state... [id=mMJ9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

  # aws_s3_bucket.demo-bucket has been deleted
  - resource "aws_s3_bucket" "demo-bucket" {
      - bucket                      = "demo-bucket-98c27d2fdb4b7fe7" -> null
        id                          = "demo-bucket-98c27d2fdb4b7fe7"
        # (12 unchanged attributes hidden)

        # (3 unchanged blocks hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to
undo or respond to these changes.

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.demo-bucket will be created
  + resource "aws_s3_bucket" "demo-bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "demo-bucket-98c27d2fdb4b7fe7"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
```

```
  # aws_s3_object.bucket-data will be created
  + resource "aws_s3_object" "bucket-data" {
      + acl                    = (known after apply)
      + arn                    = (known after apply)
      + bucket                 = "demo-bucket-98c27d2fdb4b7fe7"
      + bucket_key_enabled     = (known after apply)
      + checksum_crc32         = (known after apply)
      + checksum_crc32c        = (known after apply)
      + checksum_sha1          = (known after apply)
      + checksum_sha256        = (known after apply)
      + content_type           = (known after apply)
      + etag                   = (known after apply)
      + force_destroy          = false
      + id                     = (known after apply)
      + key                    = "newfile.txt"
      + kms_key_id             = (known after apply)
      + server_side_encryption = (known after apply)
      + source                 = "./myfile.txt"
      + storage_class          = (known after apply)
      + tags_all               = (known after apply)
      + version_id             = (known after apply)
    }

Plan: 2 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.demo-bucket: Creating...
aws_s3_bucket.demo-bucket: Creation complete after 7s [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Creating...
aws_s3_object.bucket-data: Creation complete after 2s [id=newfile.txt]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3>
```

- AWS S3bucket dashboard, Before Executing Apply command:




- AWS S3 Bucket dashboard, After Executing Apply step:




- Execute Terraform destroy to delete the configuration, which will automatically delete the bucket

```
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> terraform destroy
random_id.rand_id: Refreshing state... [id=mMJ9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

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.demo-bucket will be destroyed
  - resource "aws_s3_bucket" "demo-bucket" {
      - arn                         = "arn:aws:s3:::demo-bucket-98c27d2fdb4b7fe7" -> null
      - bucket                      = "demo-bucket-98c27d2fdb4b7fe7" -> null
      - bucket_domain_name          = "demo-bucket-98c27d2fdb4b7fe7.s3.amazonaws.com" -> null
      - bucket_regional_domain_name = "demo-bucket-98c27d2fdb4b7fe7.s3.us-east-1.amazonaws.com" -> null
      - force_destroy               = false -> null
      - hosted_zone_id              = "Z3AQBSTGFYJSTF" -> null
      - id                          = "demo-bucket-98c27d2fdb4b7fe7" -> null
      - object_lock_enabled         = false -> null
      - region                      = "us-east-1" -> null
      - request_payer               = "BucketOwner" -> null
      - tags                        = {} -> null
      - tags_all                    = {} -> null
        # (3 unchanged attributes hidden)

      - grant {
          - id          = "89f1916699ac713a2d0a157aa4176d8ce16adda216e3bafb6e7f0811cb559dca" -> null
          - 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
```

```
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_object.bucket-data: Destroying... [id=newfile.txt]
aws_s3_object.bucket-data: Destruction complete after 1s
aws_s3_bucket.demo-bucket: Destroying... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_bucket.demo-bucket: Destruction complete after 1s
random_id.rand_id: Destroying... [id=mMJ9L9tLf-c]
random_id.rand_id: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3>
```

- AWS S3 Bucket dashboard, After Executing Destroy step: