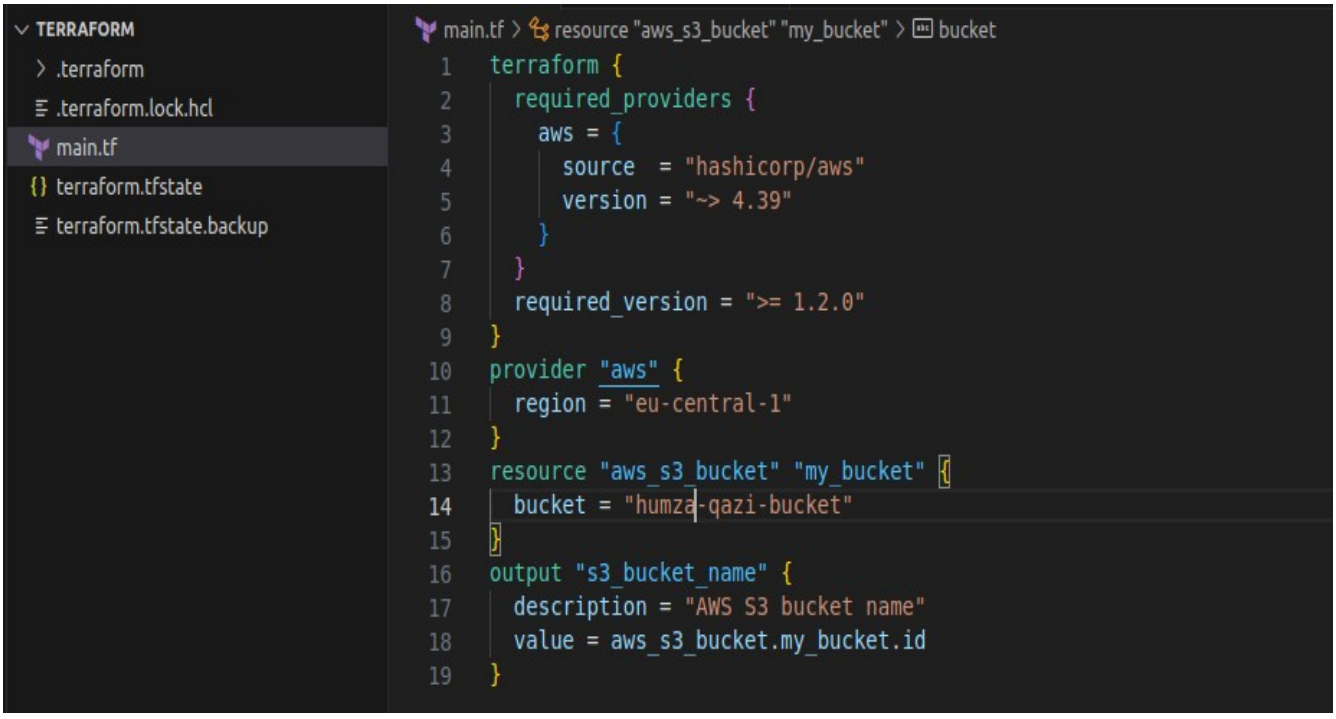
Name: Osama Abdul Razzak (2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui (2303.KHI.DEG.030)
Peer Name: M Humza Moeen (2303.KHI.DEG.019)

ASSIGNMENT 5.5(a)

Prepare a simple configuration to create an AWS S3 bucket with id set to "name-surname-bucket", and output this id to a screen.

Provide commands to create and erase this resource.

First we created a file named **main.tf** and here we applied this Code to create a bucket.



Name: Osama Abdul Razzak (2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui (2303.KHI.DEG.030)
Peer Name: M Humza Moeen (2303.KHI.DEG.019)

Then we apply Terraform init to initialize the bucket

```
muhammadhumza@all-Latitude-3490:~/Desktop/Terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.67.0

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.
```

Here we used Terraform apply to create a bucket successfully and here is the result .

```
muhammadhumza@all-Latitude-3490:~/Desktop/Terraform$ terraform apply

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                      = "humza-qazi-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_all                    = (known after apply)
      + website_domain              = (known after apply)
      + website_endpoint            = (known after apply)
    }

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

Changes to Outputs:
  + bucket_id = (known after apply)
```

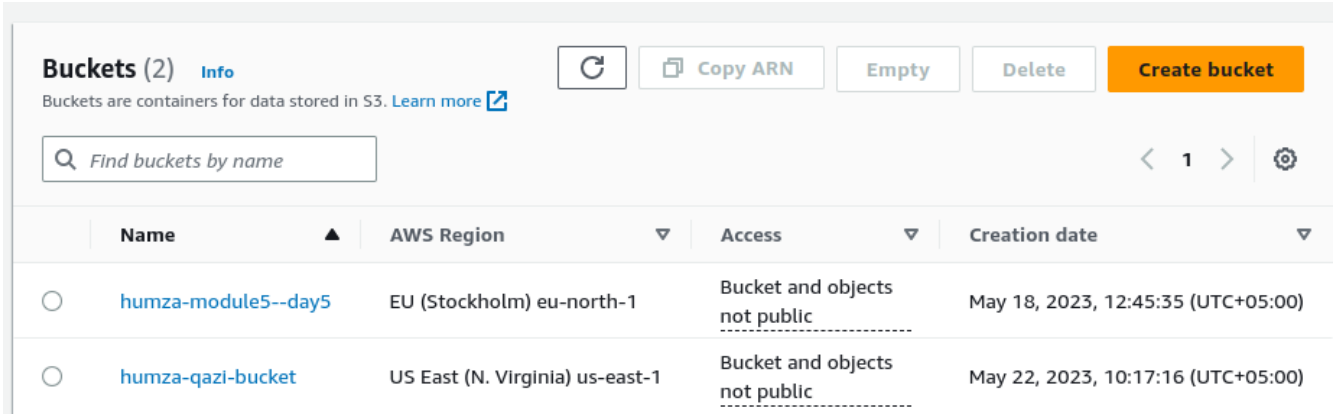
Name: Osama Abdul Razzak (2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui (2303.KHI.DEG.030)
Peer Name: M Humza Moeen (2303.KHI.DEG.019)

As you can see here is the output .

```
Outputs:

bucket_id = "humza-qazi-bucket"
```

Now Bucket has been created successfully in us-east-1 region.



After that we used Terraform destroy command to destroy bucket .

```
muhammadhumza@all-Latitude-3490:~/Desktop/Terraform$ terraform destroy
aws_s3_bucket.my_bucket: Refreshing state... [id=humza-qazi-bucket]

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.my_bucket will be destroyed
  - resource "aws_s3_bucket" "my_bucket" {
      - arn                         = "arn:aws:s3:::humza-qazi-bucket" -> null
      - bucket                      = "humza-qazi-bucket" -> null
      - bucket_domain_name          = "humza-qazi-bucket.s3.amazonaws.com" -> null
      - bucket_regional_domain_name = "humza-qazi-bucket.s3.amazonaws.com" -> null
      - force_destroy               = false -> null
      - hosted_zone_id              = "Z3AQBSTGFYJSTF" -> null
      - id                          = "humza-qazi-bucket" -> null
      - object_lock_enabled         = false -> null
      - region                      = "us-east-1" -> null
      - request_payer               = "BucketOwner" -> null
      - tags                        = {} -> null
      - tags_all                    = {} -> null

      - grant {
          - id          = "b74c5c8ce8058172124c17f83a3889c25eb352ddb0d83ad9ab760eb67bb3cfcc" -> null
          - permissions = [
              - "FULL_CONTROL",
            ] -> null
          - type        = "CanonicalUser" -> null
        }

      - server_side_encryption_configuration {
```

Name: Osama Abdul Razzak (2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui (2303.KHI.DEG.030)
Peer Name: M Humza Moeen (2303.KHI.DEG.019)

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

Changes to Outputs:
  - bucket_id = "humza-qazi-bucket" -> 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_bucket.my_bucket: Destroying... [id=humza-qazi-bucket]
aws_s3_bucket.my_bucket: Destruction complete after 3s

Destroy complete! Resources: 1 destroyed.
muhammadhumza@all-Latitude-3490:~/Desktop/Terraform$
```

After that bucket has been destroyed .