NAME: Mohammad Hussam (2303.KHI.DEG.020)
PARTNER :MAVIA ALAM KHAN (2303.KHI.DEG.017)

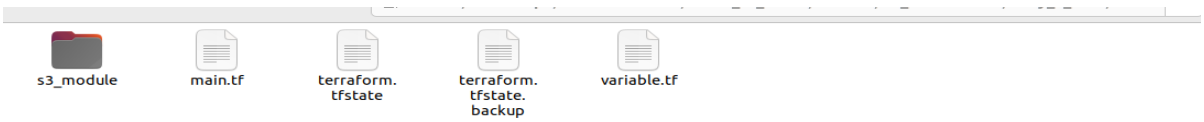
ASSIGNMENT 5.5 B

Make a module of yesterday's task, name it "s3_module". Pass bucket name to the module using variable "bucket_name". Use the returned "s3_bucket" variable to add "day2/IaC/" directory to this bucket in the main module.

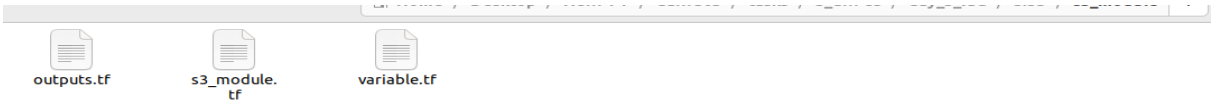
Solution:

Step 1 :

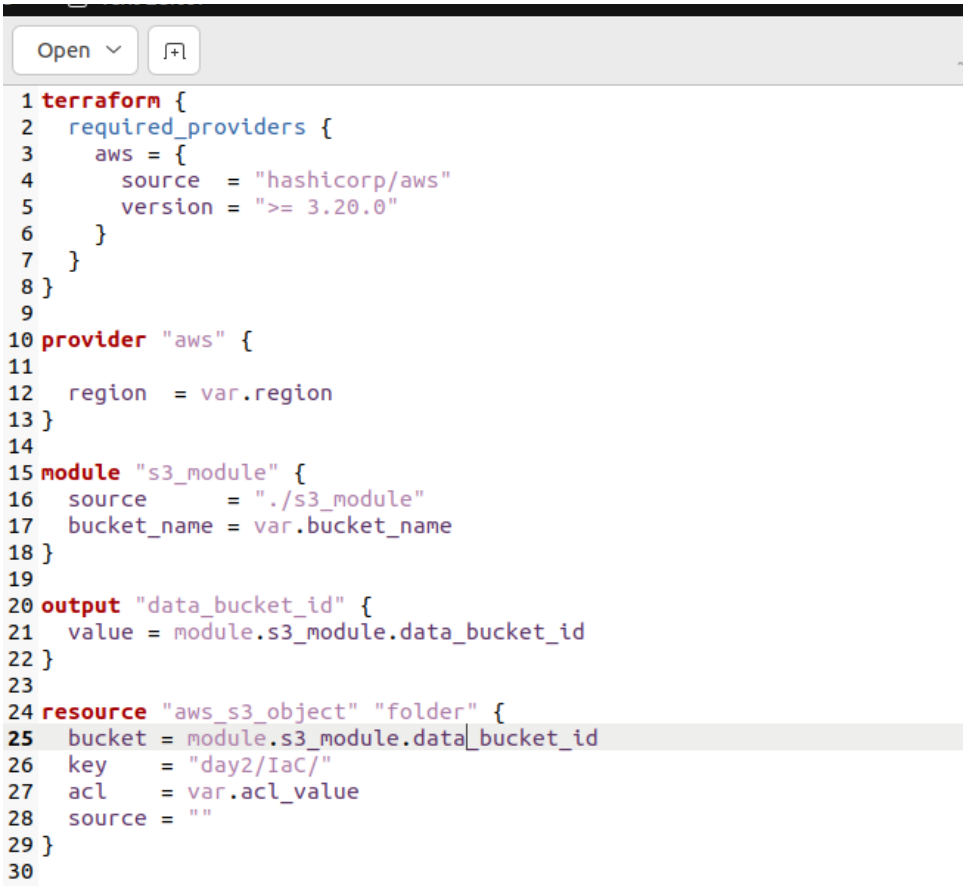
Creating the module name s3_module in the main directory :



Inside s3_module : we have outputs.tf, s3_module.tf and variable.tf files

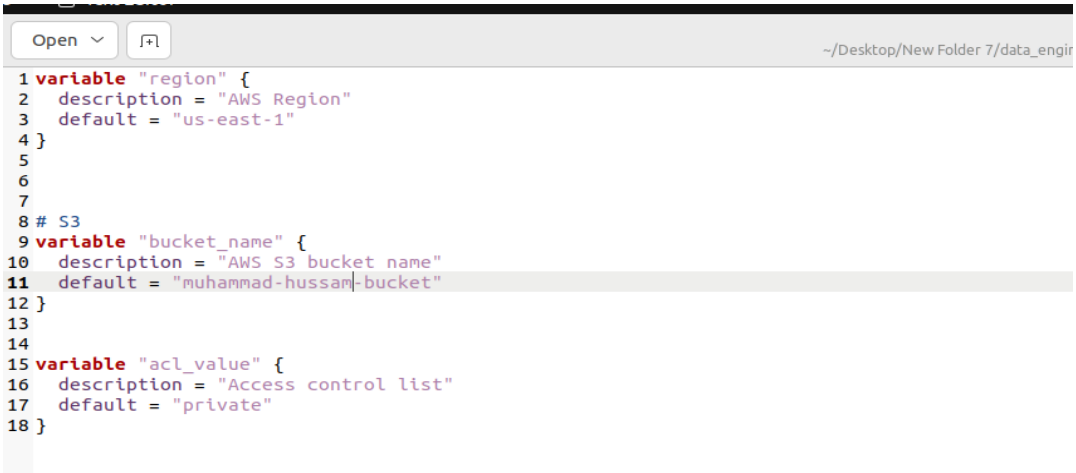


Main.tf :



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 3.20.0"
    }
  }
}

provider "aws" {

  region  = var.region
}

module "s3_module" {
  source      = "./s3_module"
  bucket_name = var.bucket_name
}

output "data_bucket_id" {
  value = module.s3_module.data_bucket_id
}

resource "aws_s3_object" "folder" {
  bucket = module.s3_module.data_bucket_id
  key    = "day2/IaC/"
  acl    = var.acl_value
  source = ""
}

```

Variable.tf:



```
variable "region" {
  description = "AWS Region"
  default = "us-east-1"
}



# S3
variable "bucket_name" {
  description = "AWS S3 bucket name"
  default = "muhammad-hussam-bucket"
}


variable "acl_value" {
  description = "Access control list"
  default = "private"
}
```

STEP 2:

Initializing the terraform in current directory

```
(base) muhammadhussam@all-MS-7D35:~/Desktop/New Folder 7/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_5_IaC/5.5b$ terraform init

Initializing the backend...
Initializing modules...
- s3_module in s3_module

Initializing provider plugins...
- Finding hashicorp/aws versions matching ">= 3.20.0"...
- Installing hashicorp/aws v4.67.0...
- Installed hashicorp/aws v4.67.0 (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 3:

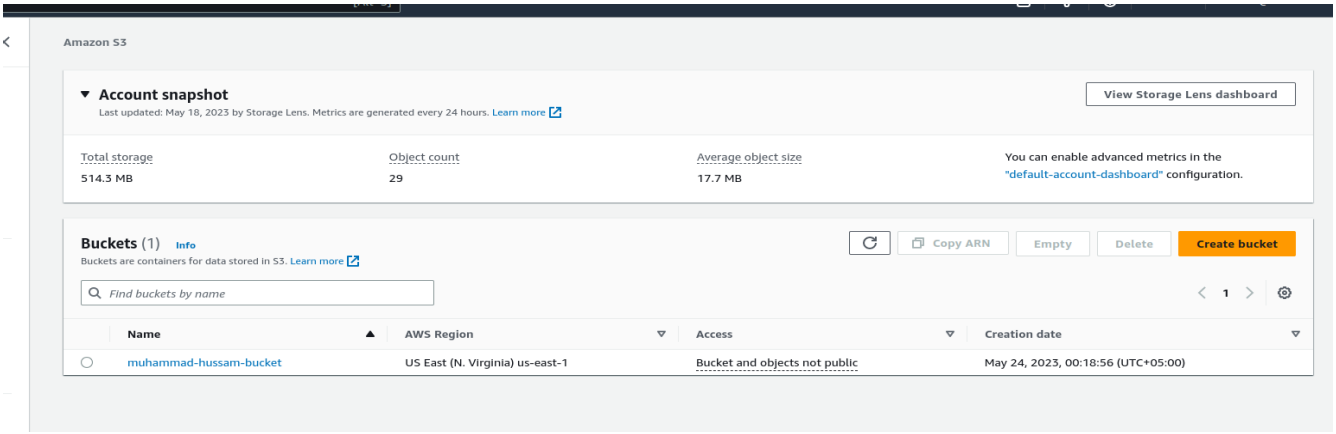
And then “terraform apply”

```
(base) muhammadhussam@all-MS-7D35:~/Desktop/New Folder 7/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_5_IaC/5.5b$ 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_object.folder will be created
  + resource "aws_s3_object" "folder" {
      + acl                    = "private"
      + bucket                 = (known after apply)
      + bucket_key_enabled     = (known after apply)
      + content_type           = (known after apply)
      + etag                   = (known after apply)
      + force_destroy          = false
      + id                     = (known after apply)
      + key                    = "day2/IaC/"
      + kms_key_id             = (known after apply)
      + server_side_encryption = (known after apply)
      + storage_class          = (known after apply)
      + tags_all               = (known after apply)
      + version_id             = (known after apply)
    }

  # module.s3_module.aws_s3_bucket.s3_module will be created
  + resource "aws_s3_bucket" "s3_module" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "muhammad-hussam-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: 2 to add, 0 to change, 0 to destroy.

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

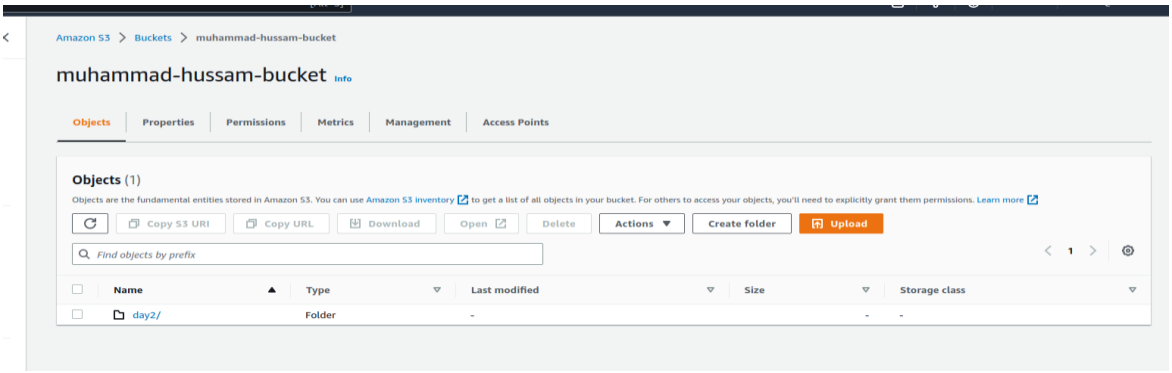
Bucket name is displayed on the terminal after we did “terraform apply” as:

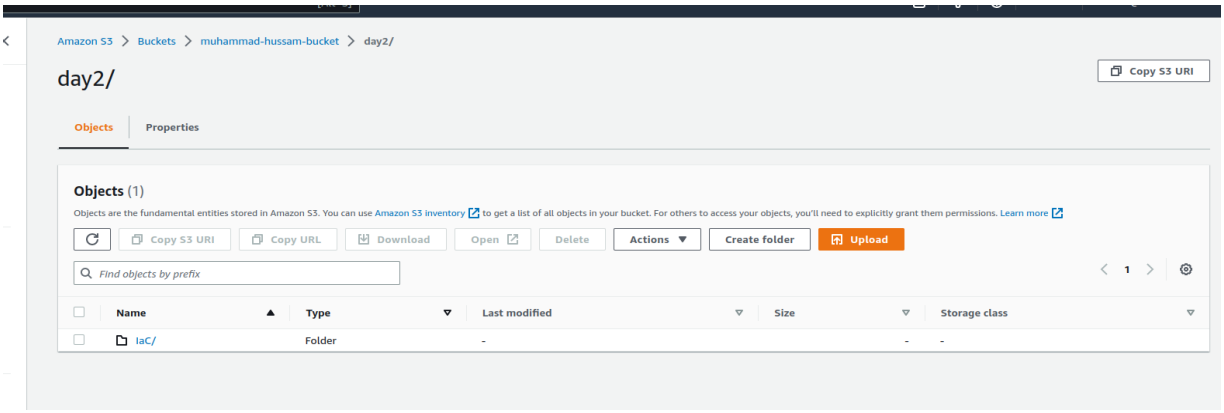
```
Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + data_bucket_id = (known after apply)

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

module.s3_module.aws_s3_bucket.s3_module: Creating...
module.s3_module.aws_s3_bucket.s3_module: Creation complete after 5s [id=muhammad-hussam-bucket]
aws_s3_object.folder: Creating...
aws_s3_object.folder: Creation complete after 0s [id=day2/IaC/]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

data_bucket_id = "muhammad-hussam-bucket"
(base) muhammadhussam@all-MS-7D35:~/Desktop/New Folder 7/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_5_IaC/5.5b$
```

Now we open AWS S3, and see that there is a bucket with the name “muhammad-hussam-bucket" :



As we go inside this bucket we”ll find an empty folder created with the path “day2/IaC”

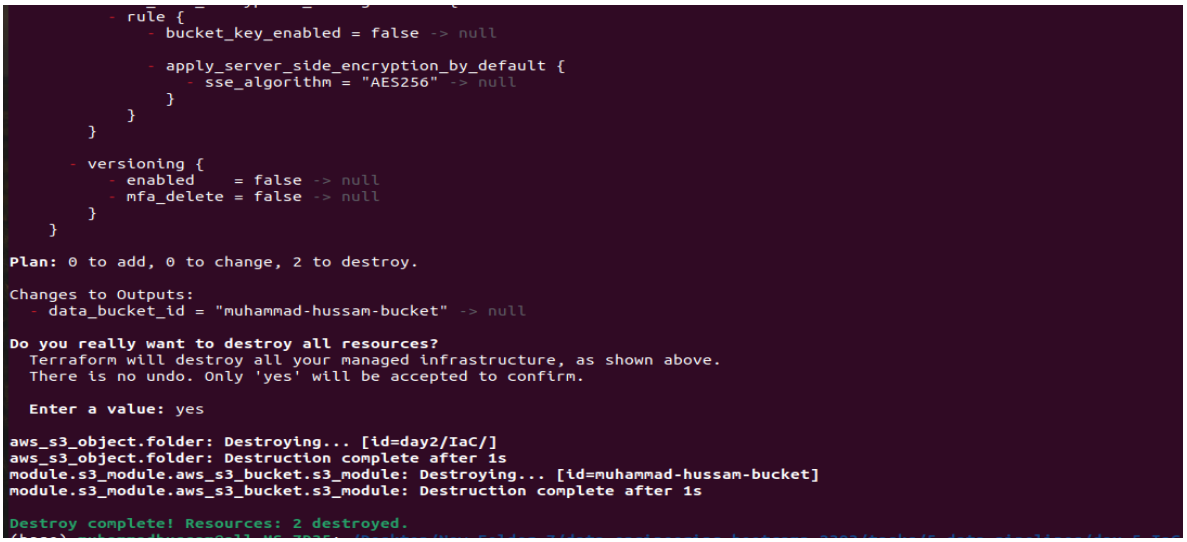




Step 4 :

Now destroying the resouces using “terraform destroy”:

```
(base) muhammadhussam@all-MS-7D35:~/Desktop/New Folder 7/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_5_IaC/5.5b$ terraform destroy
module.s3_module.aws_s3_bucket.s3_module: Refreshing state... [id=muhammad-hussam-bucket]
aws_s3_object.folder: Refreshing state... [id=day2/IaC/]

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_object.folder will be destroyed
  - resource "aws_s3_object" "folder" {
      - acl                    = "private" -> null
      - bucket                 = "muhammad-hussam-bucket" -> null
      - bucket_key_enabled     = false -> null
      - content_type           = "binary/octet-stream" -> null
      - etag                   = "d41d8cd98f00b204e9800998ecf8427e" -> null
      - force_destroy          = false -> null
      - id                     = "day2/IaC/" -> null
      - key                    = "day2/IaC/" -> null
      - metadata               = {} -> null
      - server_side_encryption = "AES256" -> null
      - storage_class          = "STANDARD" -> null
      - tags                   = {} -> null
      - tags_all               = {} -> null
    }

  # module.s3_module.aws_s3_bucket.s3_module will be destroyed
  - resource "aws_s3_bucket" "s3_module" {
      - arn                         = "arn:aws:s3:::muhammad-hussam-bucket" -> null
      - bucket                      = "muhammad-hussam-bucket" -> null
      - bucket_domain_name          = "muhammad-hussam-bucket.s3.amazonaws.com" -> null
      - bucket_regional_domain_name = "muhammad-hussam-bucket.s3.amazonaws.com" -> null
      - force_destroy               = false -> null
      - hosted_zone_id              = "Z3AQBSTGFYJSTF" -> null
      - id                          = "muhammad-hussam-bucket" -> null
      - object_lock_enabled         = false -> null
      - region                      = "us-east-1" -> null
      - request_payer               = "BucketOwner" -> null
      - tags                        = {} -> null
      - tags_all                    = {} -> null

      - grant {
          - id          = "0182cbf72cdf7566dded8d6c27be8b2f0f14853d266ea774dd299230536d78a9" -> null
          - permissions = [
              - "FULL_CONTROL",
            ] -> null
          - type        = "CanonicalUser" -> null
        }

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



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

              - apply_server_side_encryption_by_default {
                  - sse_algorithm = "AES256" -> null
                }
            }
        }

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

Plan: 0 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  - data_bucket_id = "muhammad-hussam-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_object.folder: Destroying... [id=day2/IaC/]
aws_s3_object.folder: Destruction complete after 1s
module.s3_module.aws_s3_bucket.s3_module: Destroying... [id=muhammad-hussam-bucket]
module.s3_module.aws_s3_bucket.s3_module: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```