UNIT 5.5(B) GRADED ASSIGNMENT

<u>Group members</u>

Ifra Saleem (2303.khi.deg.003)
Umaima Siddiqui (2023.KHI.DEG.033)

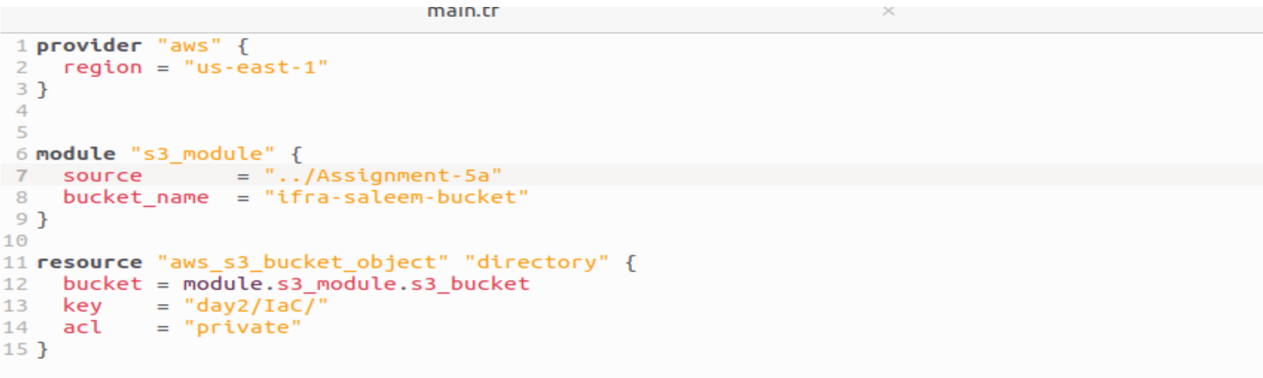
UNIT 5.5(B) GRADED ASSIGNMENT

Task:

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.

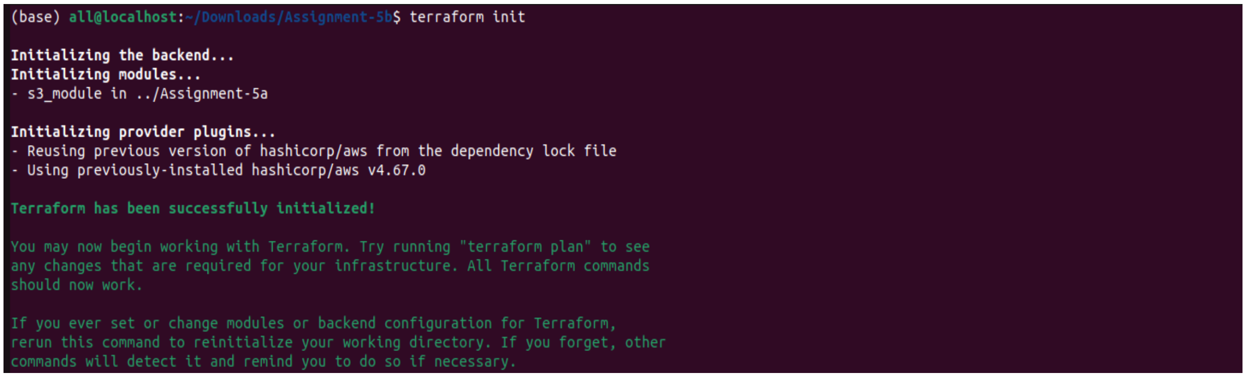
Solution:

main.tf:

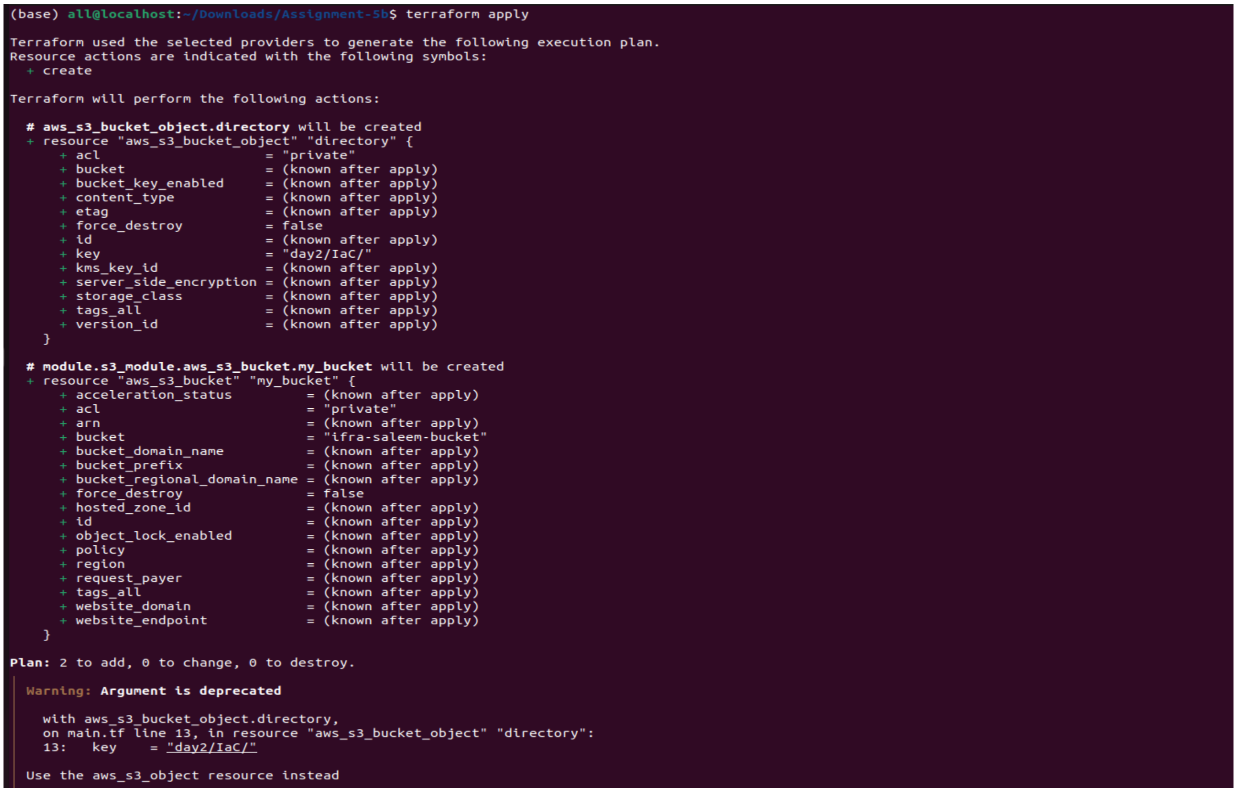


```
provider "aws" {
  region = "us-east-1"
}


module "s3_module" {
  source        = "../Assignment-5a"
  bucket_name   = "ifra-saleem-bucket"
}

resource "aws_s3_bucket_object" "directory" {
  bucket = module.s3_module.s3_bucket
  key    = "day2/IaC/"
  acl    = "private"
}
```

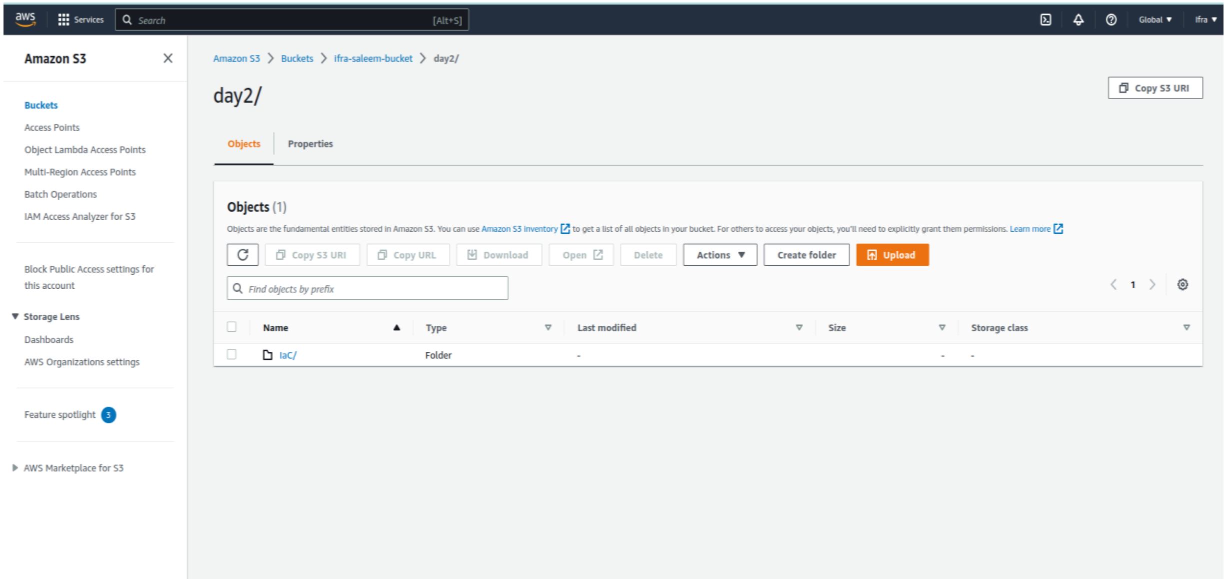
In the above file, we are simply creating a module named s3_module and we are giving Assignment-5a as a source and we are giving bucket name as ifra-saleem-bucket. It will simply create a bucket using the commands present in Assignment-5a and then inside that bucket, it will make a directory day2 and inside day2 another directory will be created named IaC.



```
(base) all@localhost:~/Downloads/Assignment-5b$ terraform init

Initializing the backend...
Initializing modules...
- s3_module in ../Assignment-5a

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.
```



```
(base) all@localhost:~/Downloads/Assignment-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_bucket_object.directory will be created
  + resource "aws_s3_bucket_object" "directory" {
      + 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.my_bucket will be created
  + resource "aws_s3_bucket" "my_bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = "private"
      + arn                         = (known after apply)
      + bucket                      = "ifra-saleem-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.
╷
│ Warning: Argument is deprecated
│
│   with aws_s3_bucket_object.directory,
│   on main.tf line 13, in resource "aws_s3_bucket_object" "directory":
│   13:   key     = "day2/IaC/"
│
│ Use the aws_s3_object resource instead
```

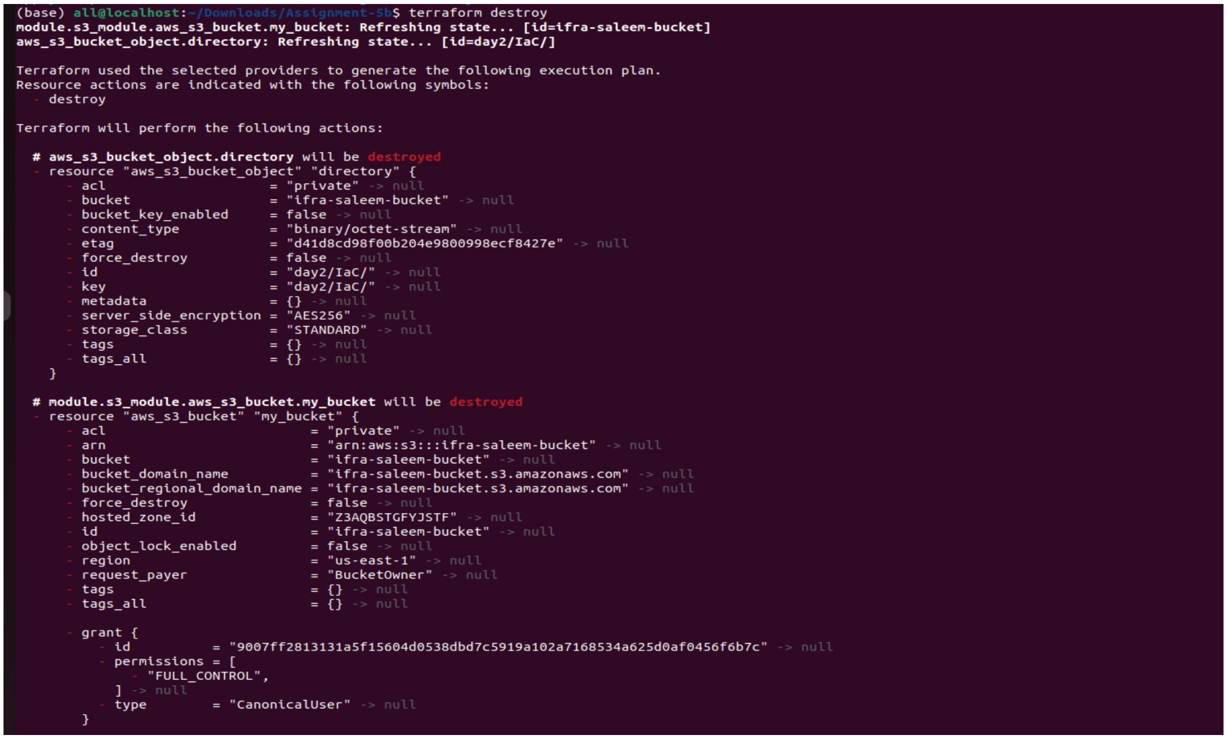


day2/

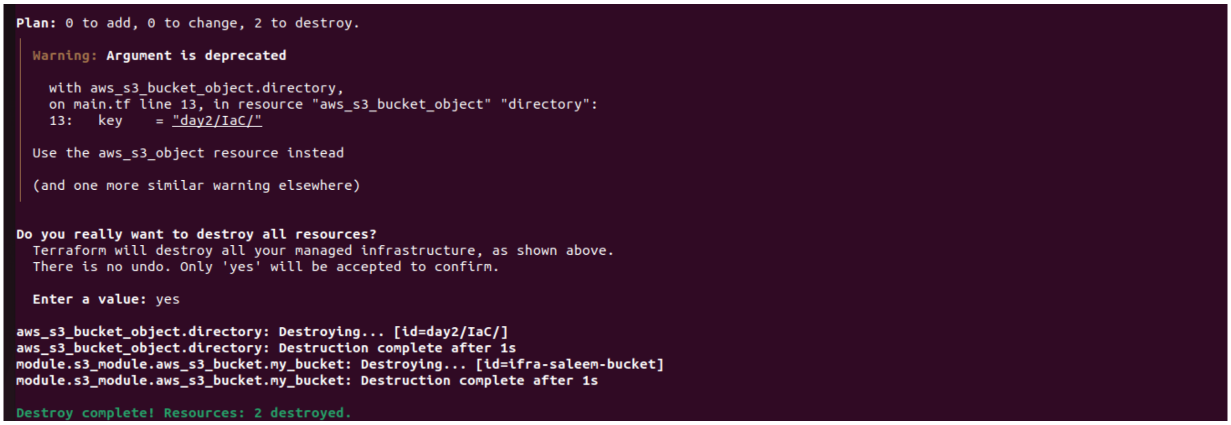
Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	IaC/	Folder	-	-	-



```
(base) all@localhost:~/Downloads/Assignment-5b$ terraform destroy
module.s3_module.aws_s3_bucket.my_bucket: Refreshing state... [id=ifra-saleem-bucket]
aws_s3_bucket_object.directory: 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_bucket_object.directory will be destroyed
  - resource "aws_s3_bucket_object" "directory" {
      - acl                    = "private" -> null
      - bucket                 = "ifra-saleem-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.my_bucket will be destroyed
  - resource "aws_s3_bucket" "my_bucket" {
      - acl                         = "private" -> null
      - arn                         = "arn:aws:s3:::ifra-saleem-bucket" -> null
      - bucket                      = "ifra-saleem-bucket" -> null
      - bucket_domain_name          = "ifra-saleem-bucket.s3.amazonaws.com" -> null
      - bucket_regional_domain_name = "ifra-saleem-bucket.s3.amazonaws.com" -> null
      - force_destroy               = false -> null
      - hosted_zone_id              = "Z3AQBSTGFYJSTF" -> null
      - id                          = "ifra-saleem-bucket" -> null
      - object_lock_enabled         = false -> null
      - region                      = "us-east-1" -> null
      - request_payer               = "BucketOwner" -> null
      - tags                        = {} -> null
      - tags_all                    = {} -> null

      - grant {
          - id          = "9007ff2813131a5f15604d0538dbd7c5919a102a7168534a625d0af0456f6b7c" -> null
          - permissions = [
              - "FULL_CONTROL",
            ] -> null
          - type        = "CanonicalUser" -> null
        }
```



```
Plan: 0 to add, 0 to change, 2 to destroy.
│
│ Warning: Argument is deprecated
│
│   with aws_s3_bucket_object.directory,
│   on main.tf line 13, in resource "aws_s3_bucket_object" "directory":
│   13:   key    = "day2/IaC/"
│
│ Use the aws_s3_object resource instead
│
│ (and one more similar warning elsewhere)
│

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_object.directory: Destroying... [id=day2/IaC/]
aws_s3_bucket_object.directory: Destruction complete after 1s
module.s3_module.aws_s3_bucket.my_bucket: Destroying... [id=ifra-saleem-bucket]
module.s3_module.aws_s3_bucket.my_bucket: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```