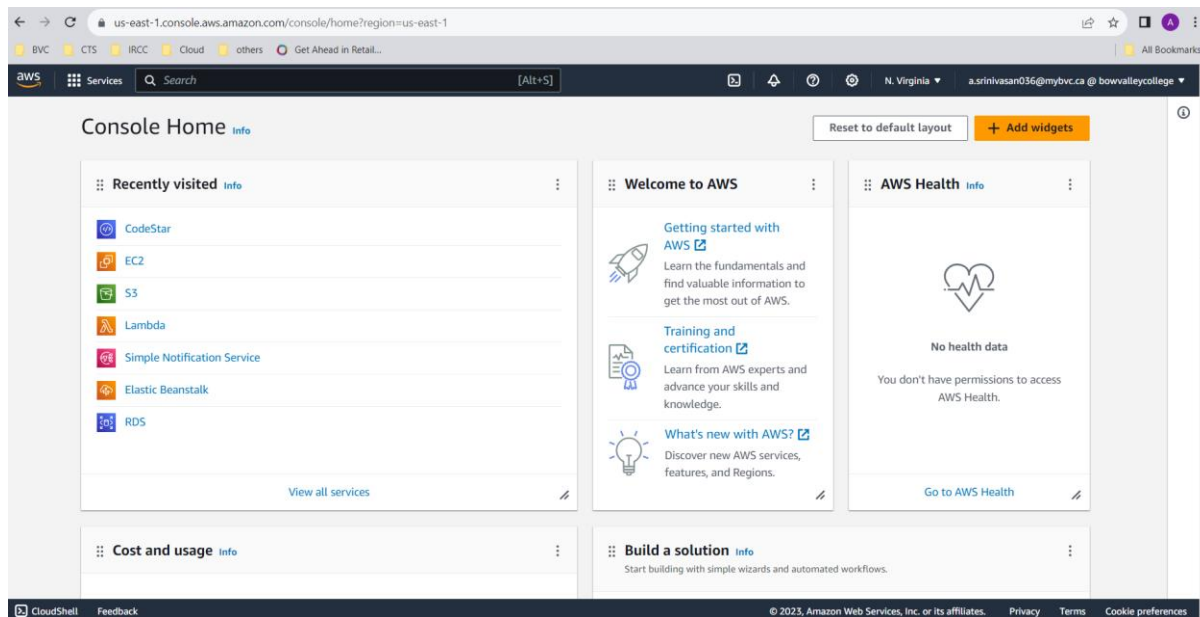
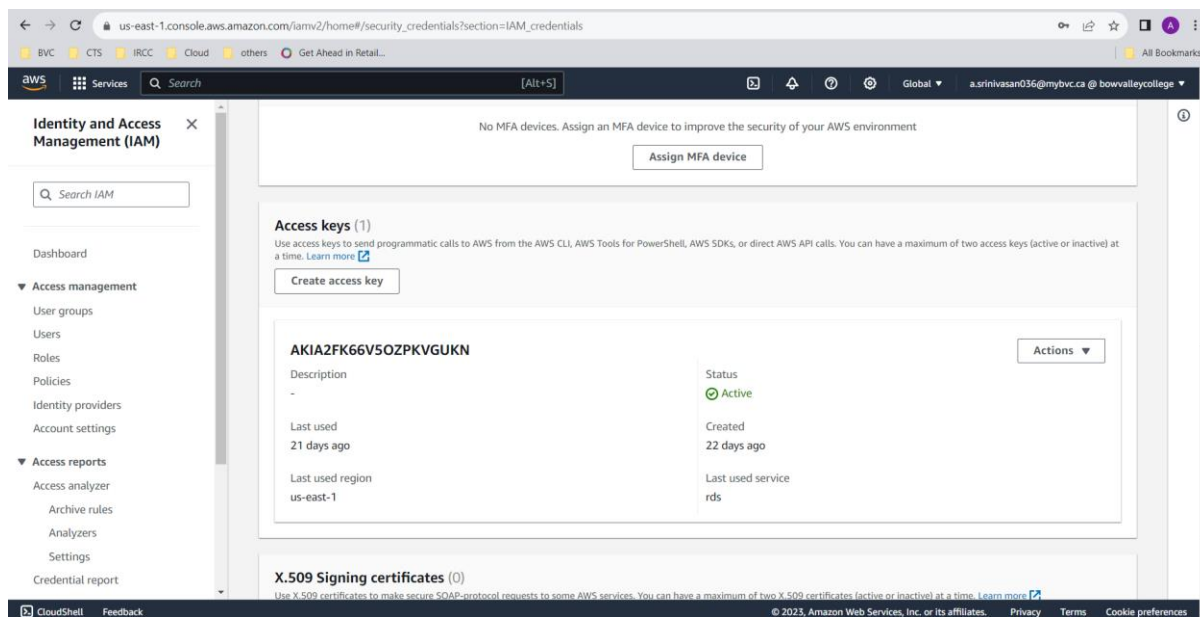


Inclass Exercise - Hosting static website in S3 with Modularity using Terraform (CLCM3504)

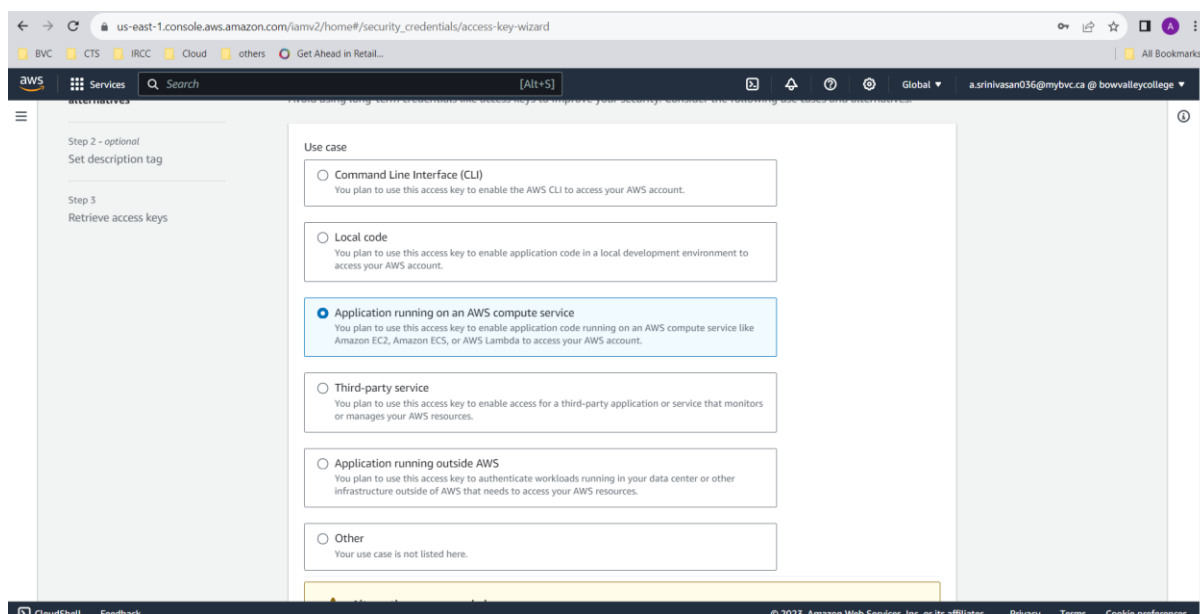
Open the AWS console and get the aws credentials in the profile.



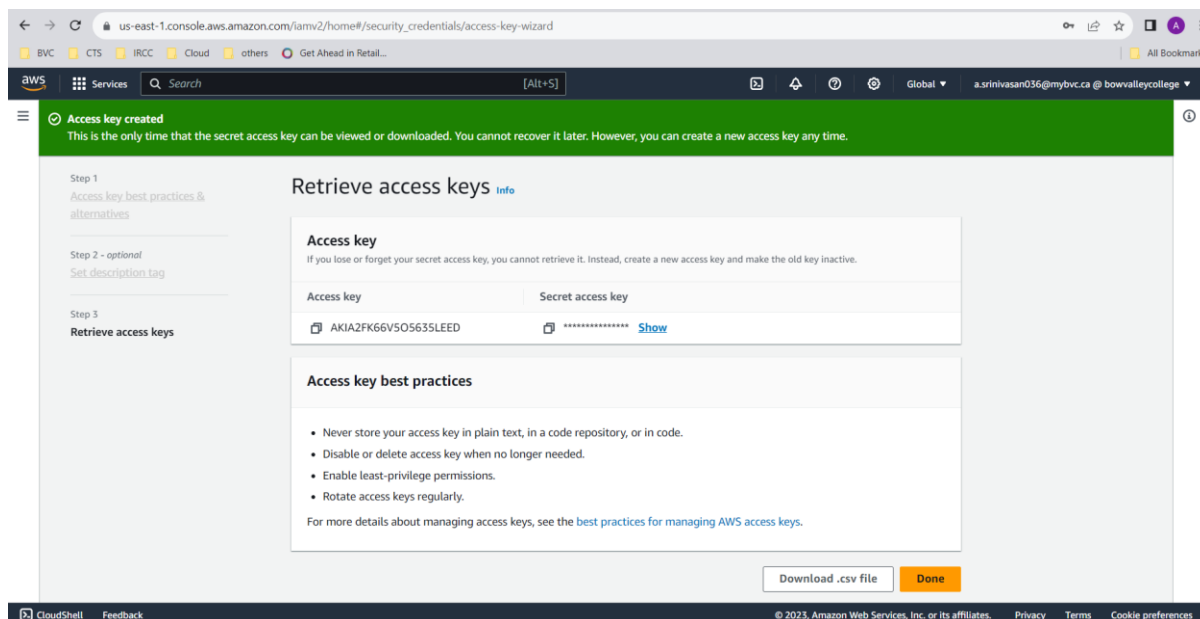
Create access key for the connection.



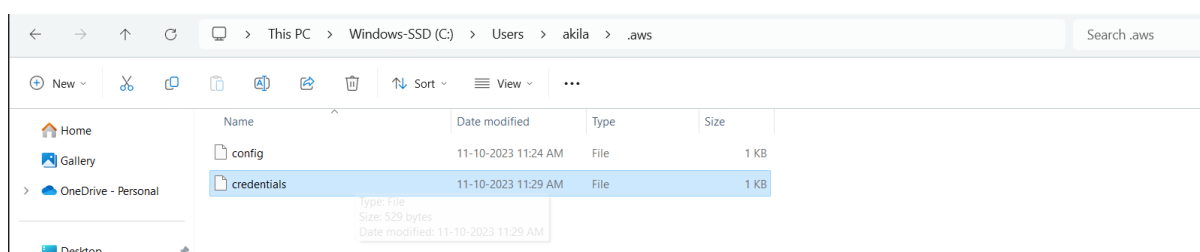
Click the application running on an AWS compute service.



Copy the access key and secret access key.



Now open the credentials and paste the access key and secret access key and save.



Now open the command prompt and give the command “aws configure” and paste the credentials which you have copied. And to initiate the terraform by giving “terraform init”

```
Command Prompt
(c) Microsoft Corporation. All rights reserved.

C:\Users\akila>aws configure
AWS Access Key ID [*****LEED]: AKIA2FK66V505635LEED
AWS Secret Access Key [*****pjKF]: +vGriy+0iiFe+IKEBZYf31eX69S+hd+eIdd4pjKF
Default region name [None]:
Default output format [None]:

C:\Users\akila>cd C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity

C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "4.0.0"...
- Finding hashicorp/random versions matching "3.1.0"...
- Installing hashicorp/aws v4.0.0...
- Installed hashicorp/aws v4.0.0 (signed by HashiCorp)
- Installing hashicorp/random v3.1.0...
- Installed hashicorp/random v3.1.0 (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.

C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>
```

After that “terraform Plan”

```
Command Prompt

rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>terraform plan

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.akila_s3_static_bucket will be created
+ resource "aws_s3_bucket" "akila_s3_static_bucket" {
+   acceleration_status      = (known after apply)
+   acl                      = (known after apply)
+   arn                     = (known after apply)
+   bucket                  = "akila_s3_static_bucket"
+   bucket_domain_name      = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   cors_rule               = (known after apply)
+   force_destroy           = false
+   grant                   = (known after apply)
+   hosted_zone_id          = (known after apply)
+   id                      = (known after apply)
+   lifecycle_rule          = (known after apply)
+   logging                 = (known after apply)
+   policy                  = (known after apply)
+   region                  = (known after apply)
+   replication_configuration = (known after apply)
+   request_payer           = (known after apply)
+   server_side_encryption_configuration = (known after apply)
+   tags_all                = (known after apply)
+   versioning              = (known after apply)
+   website                 = (known after apply)
+   website_domain          = (known after apply)
+   website_endpoint        = (known after apply)
}

# aws_s3_bucket_acl.akila_s3_static_bucket will be created
+ resource "aws_s3_bucket_acl" "akila_s3_static_bucket" {
+   acl = "public-read"
}
```

Then give “terraform apply”

```
Command Prompt
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.

C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>terraform apply
aws_s3_object.object: Refreshing state... [id=index.html]
aws_s3_bucket.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket]
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket]
aws_s3_bucket_public_access_block.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket]
aws_s3_bucket_ownership_controls.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket]
aws_s3_bucket_policy.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket]
aws_s3_bucket_acl.akila_s3_static_bucket: Refreshing state... [id=akila-s3-static-bucket,public-read]

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.object will be created
+ resource "aws_s3_object" "object" {
  + acl                = "private"
  + bucket             = "akila-s3-static-bucket"
  + bucket_key_enabled = (known after apply)
  + content_type       = (known after apply)
  + etag               = (known after apply)
  + force_destroy      = false
  + id                 = (known after apply)
  + key                = "index.html"
  + kms_key_id         = (known after apply)
  + server_side_encryption = (known after apply)
  + source              = "../html/index.html"
  + storage_class       = (known after apply)
  + tags_all           = (known after apply)
  + version_id         = (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.
```

```
Command Prompt

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_website_configuration.akila_s3_static_bucket will be created
+ resource "aws_s3_bucket_website_configuration" "akila_s3_static_bucket" {
  + bucket           = "akila-s3-static-bucket"
  + id               = (known after apply)
  + website_domain   = (known after apply)
  + website_endpoint = (known after apply)

  + index_document {
    + suffix = "index.html"
  }
}

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

Changes to Outputs:
+ bucket_endpoint = (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

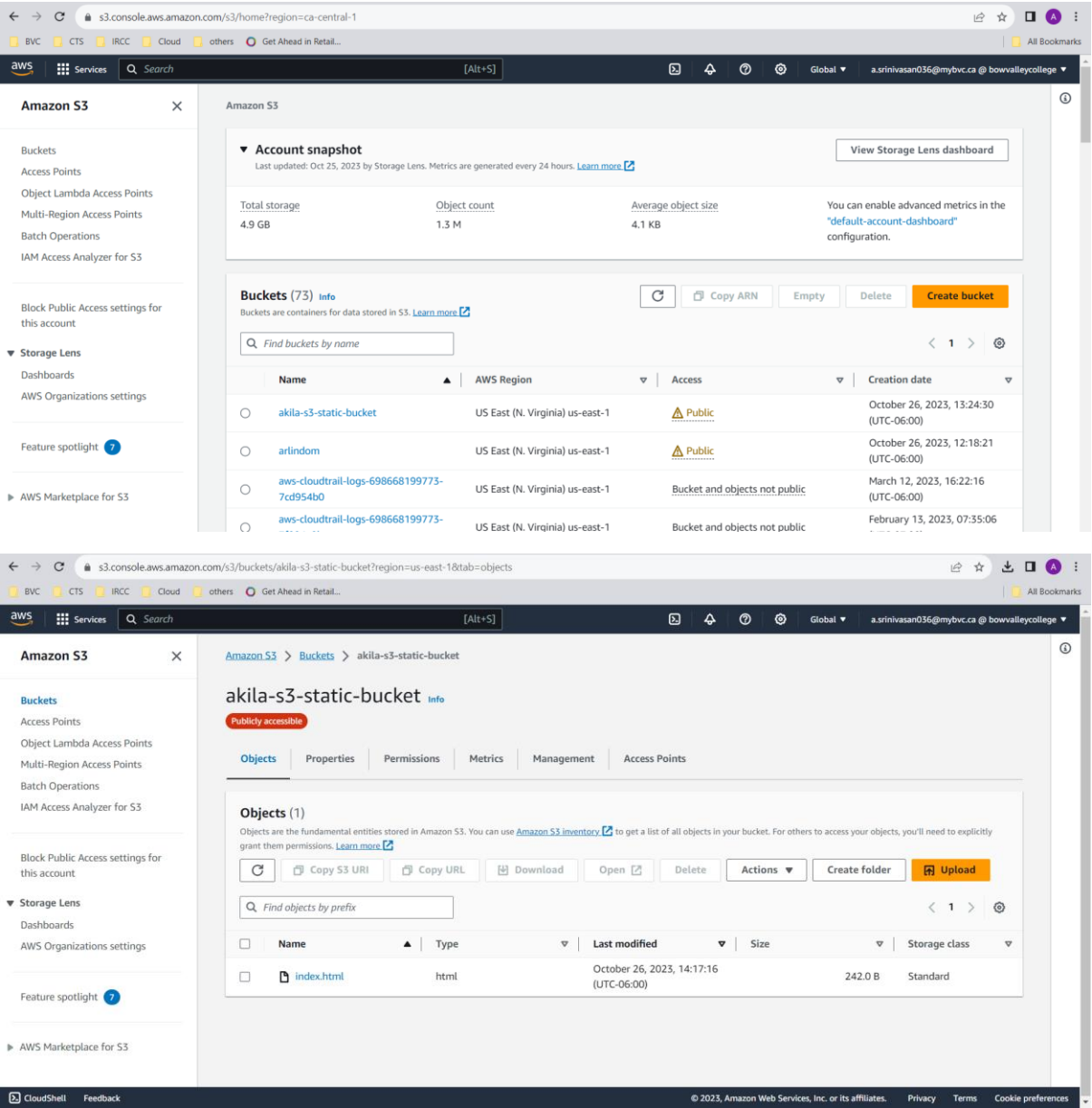
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Creating...
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Creation complete after 1s [id=akila-s3-static-bucket]

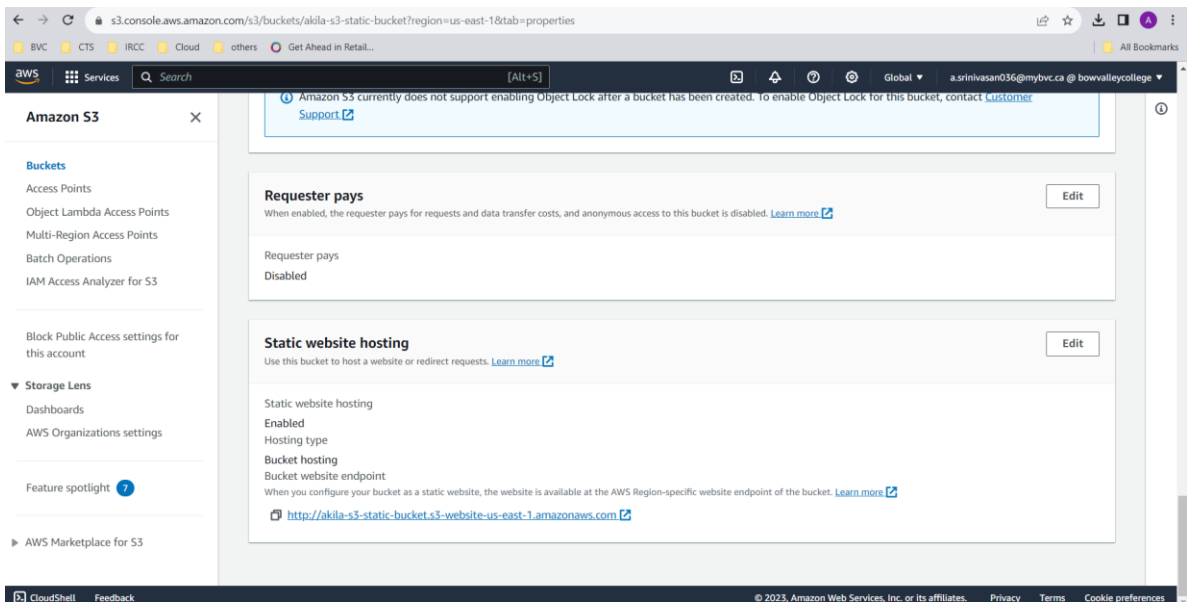
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
bucket_endpoint = "akila-s3-static-bucket.s3-website-us-east-1.amazonaws.com"
domain_name     = "akila-s3-static-bucket"
website_bucket_name = "akila-s3-static-bucket"

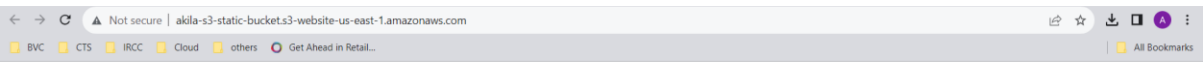
C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>
```

Here we can see the bucket has created





Here is the website which we have in the index.html



task completed....

Destroy the bucket using “terraform destroy”

```
Command Prompt
- key = "index.html" -> null
- metadata = {} -> null
- server_side_encryption = "AES256" -> null
- source = "./html/index.html" -> null
- storage_class = "STANDARD" -> null
- tags = {} -> null
- tags_all = {} -> null
}

Plan: 0 to add, 0 to change, 7 to destroy.

Changes to Outputs:
- bucket_endpoint = "akila-s3-static-bucket.s3-website-us-east-1.amazonaws.com" -> null
- domain_name = "akila-s3-static-bucket" -> null
- website_bucket_name = "akila-s3-static-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_website_configuration.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket]
aws_s3_bucket_acl.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket,public-read]
aws_s3_bucket_policy.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket]
aws_s3_object.object: Destroying... [id=index.html]
aws_s3_bucket_acl.akila_s3_static_bucket: Destruction complete after 0s
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket]
aws_s3_object.object: Destruction complete after 0s
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Destruction complete after 0s
aws_s3_bucket_policy.akila_s3_static_bucket: Destruction complete after 0s
aws_s3_bucket_public_access_block.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket]
aws_s3_bucket_public_access_block.akila_s3_static_bucket: Destruction complete after 1s
aws_s3_bucket_website_configuration.akila_s3_static_bucket: Destruction complete after 1s
aws_s3_bucket.akila_s3_static_bucket: Destroying... [id=akila-s3-static-bucket]
aws_s3_bucket.akila_s3_static_bucket: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.

C:\Users\akila\Desktop\BVC\in-class assignments\s3 with modularity>
```

Reflection:

In my case initially I have missed the object configuration in s3 so because of this the bucket is created but the object is not pushed into the bucket. After gone through the official documentation I have learnt that “Resource: aws_s3_bucket_object” after adding this I have got the output and this is what I have learned.