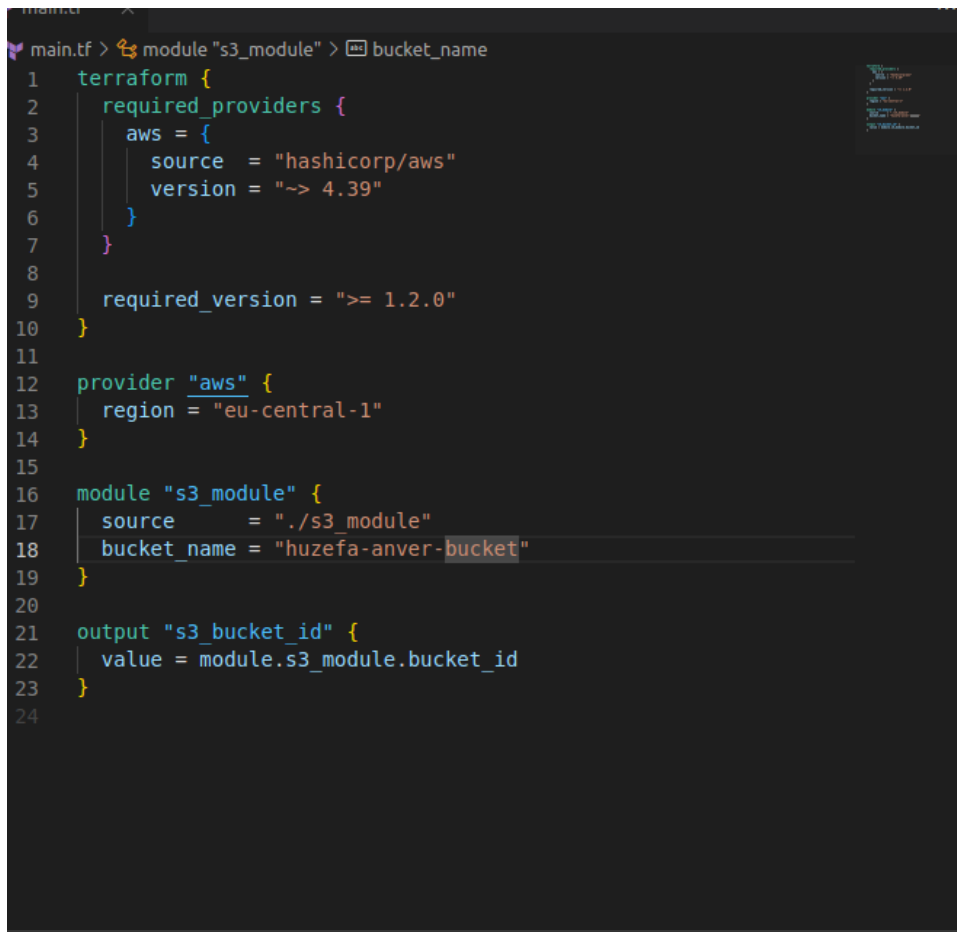


Members

Huzefa Anver (2303.KHI.DEG.002)

Syed Mohammad Anjil Hussain Rizvi (2303.KHI.DEG.031)

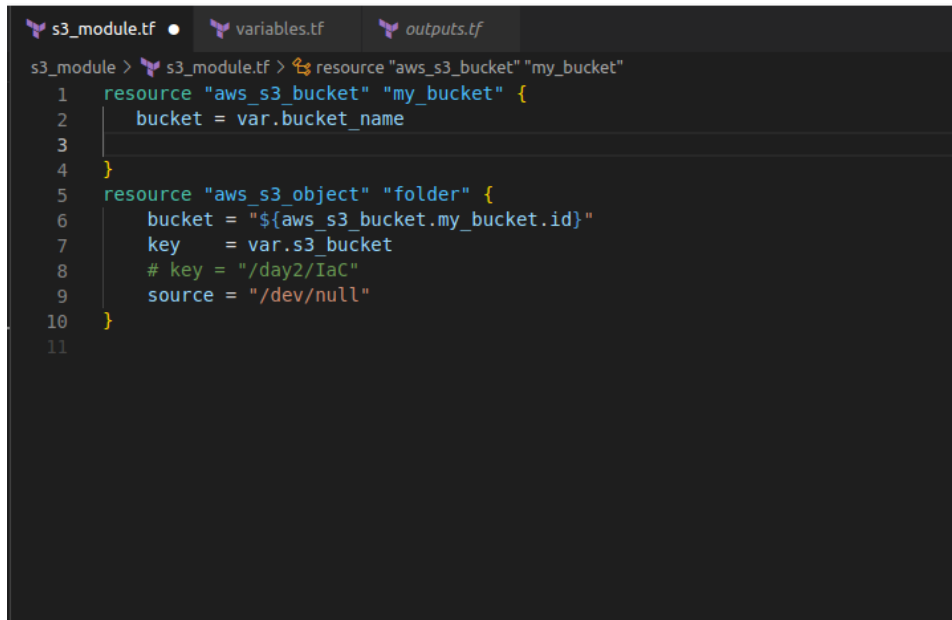
Main.tf



```
main.tf > module "s3_module" > bucket_name
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "~> 4.39"
6     }
7   }
8
9   required_version = ">= 1.2.0"
10 }
11
12 provider "aws" {
13   region = "eu-central-1"
14 }
15
16 module "s3_module" {
17   source = "./s3_module"
18   bucket_name = "huzefa-anver-bucket"
19 }
20
21 output "s3_bucket_id" {
22   value = module.s3_module.bucket_id
23 }
24
```

This is our main.tf file we are providing a module over here which is directing towards the folder we created in s3_module located in the same directory and name of the bucket. We will state our resources in separate files and variables in our separate file. The rest is same as what we did in 5.5A.

Resources.tf

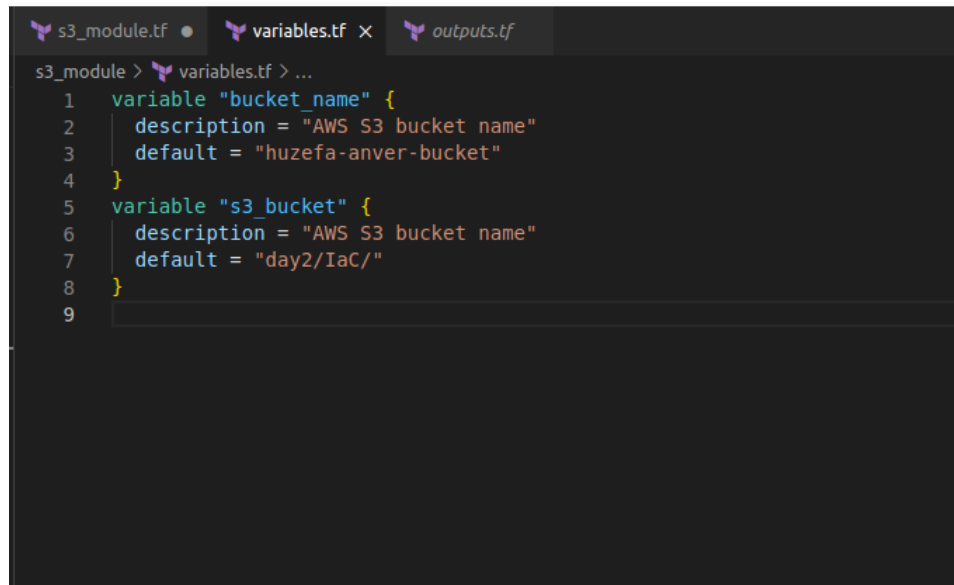


```
s3_module > s3_module.tf > resource "aws_s3_bucket" "my_bucket"
1  resource "aws_s3_bucket" "my_bucket" {
2      bucket = var.bucket_name
3
4  }
5  resource "aws_s3_object" "folder" {
6      bucket = "${aws_s3_bucket.my_bucket.id}"
7      key    = var.s3_bucket
8      # key = "/day2/IaC"
9      source = "/dev/null"
10 }
11
```

Over here we have created our resources directing towards the variable we created in our variable.tf which will be shown below. Instead of giving the resources directly over here we are using variable.tf file to define our resources we want to create and assigning a variable name to it. That variable name is then used in this file “var.bucket_name” to reference those resources we created which is an s3 bucket.

The second resource is creating directory in our s3 bucket /day2/lac.

Variables.tf

A screenshot of a code editor with three tabs: 's3_module.tf', 'variables.tf' (active), and 'outputs.tf'. The 'variables.tf' tab is selected, showing Terraform variable definitions. The code is as follows:

```
s3_module > variables.tf > ...
1  variable "bucket_name" {
2      description = "AWS S3 bucket name"
3      default = "huzefa-anver-bucket"
4  }
5  variable "s3_bucket" {
6      description = "AWS S3 bucket name"
7      default = "day2/IaC/"
8  }
9
```

As said above, we are defining resources and assigning them variable name and then that variable is used in our resources file.

Terraform Apply

```
huzefa@anverBall-40-7055: ~/Desktop/data_engineering_projects/2402/Week5/Data_pipelines/Day_5_TAC/Assignment_5_50$ 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:

# module.s3_module.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               = "huzefa-anver-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)
}

# module.s3_module.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/tac/"
+   kms_key_id   = (known after apply)
+   server_side_encryption = (known after apply)
+   source       = "/dev/null"
+   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.

Changes to Outputs:
  s3_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

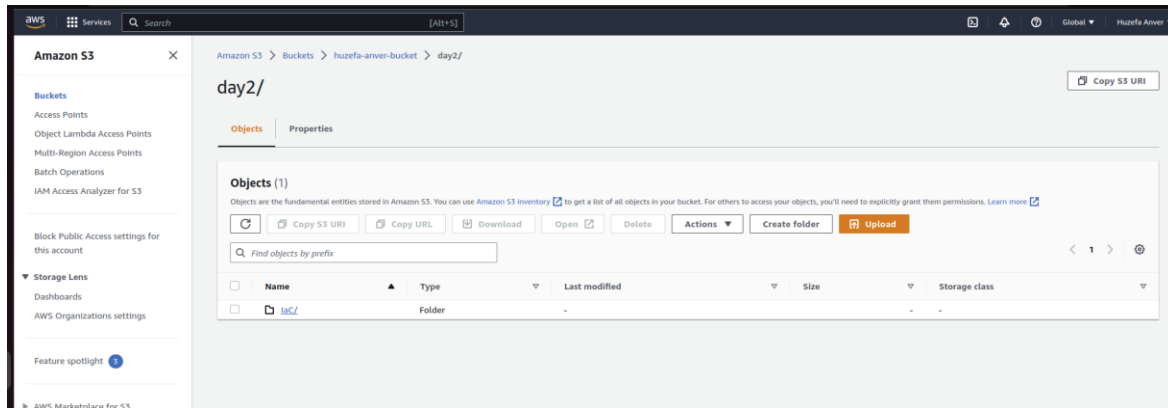
module.s3_module.aws_s3_bucket.my_bucket: Creating...
module.s3_module.aws_s3_bucket.my_bucket: Creation complete after 4s [id=huzefa-anver-bucket]
module.s3_module.aws_s3_object.folder: Creating...
module.s3_module.aws_s3_object.folder: Creation complete after 0s [id=day2/tac/]

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

Outputs:
s3_bucket_id = "huzefa-anver-bucket"
```

After terraform init using terraform apply to apply all the code we wrote in our main.tf. We can see it was successful and is printing the output as we desired.

AWS Cloud



Checking the output in our aws cloud. We can see there is a bucket containing directories as we intended.