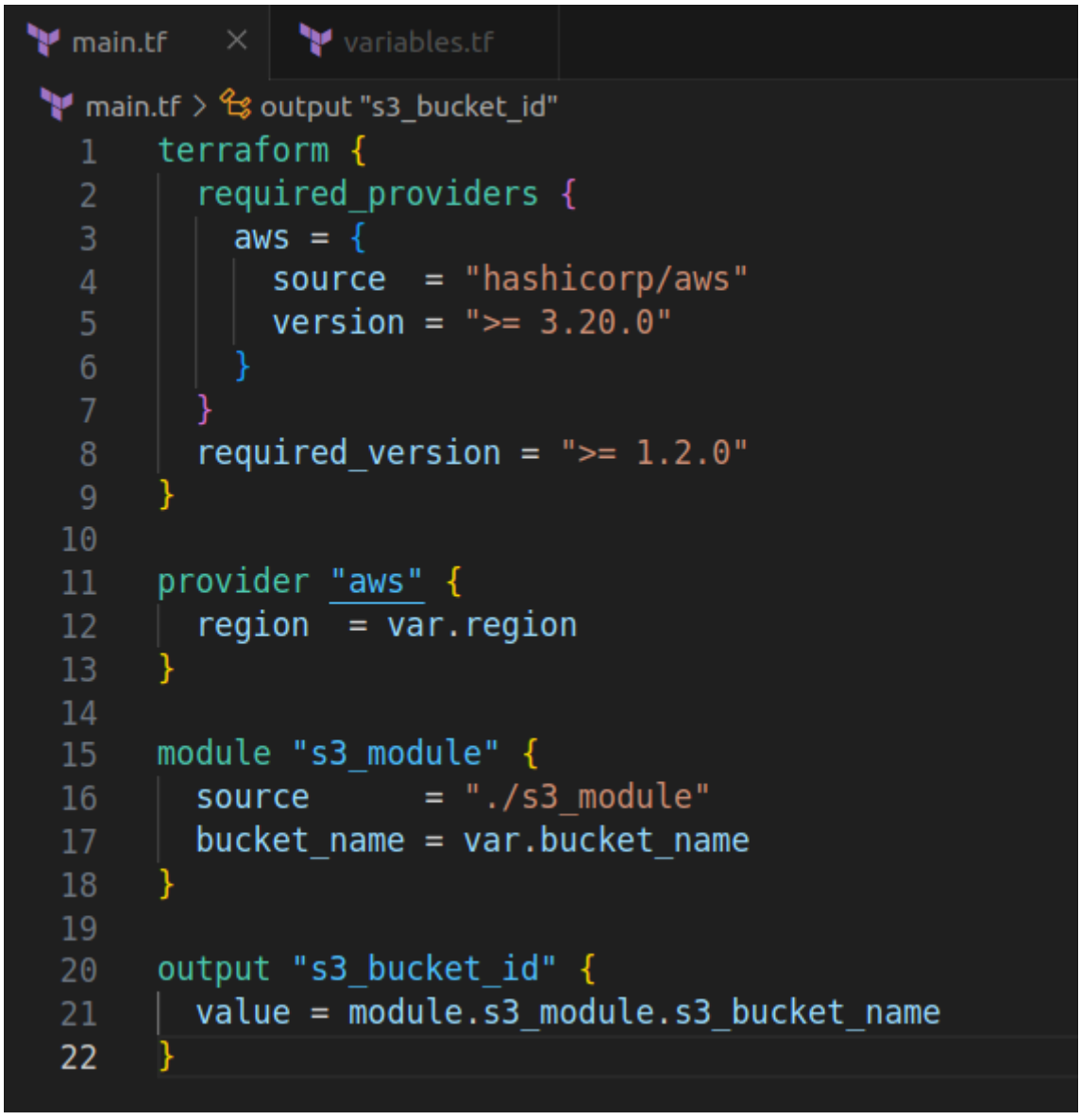
Graded Assignment 5.5 (a) & (b)

Name: Mohammad Hamza Asim
Employee#: 2303.KHI.DEG.014
Collaborated with: Saad Sameer Khan (2303.KHI.DEG.034)

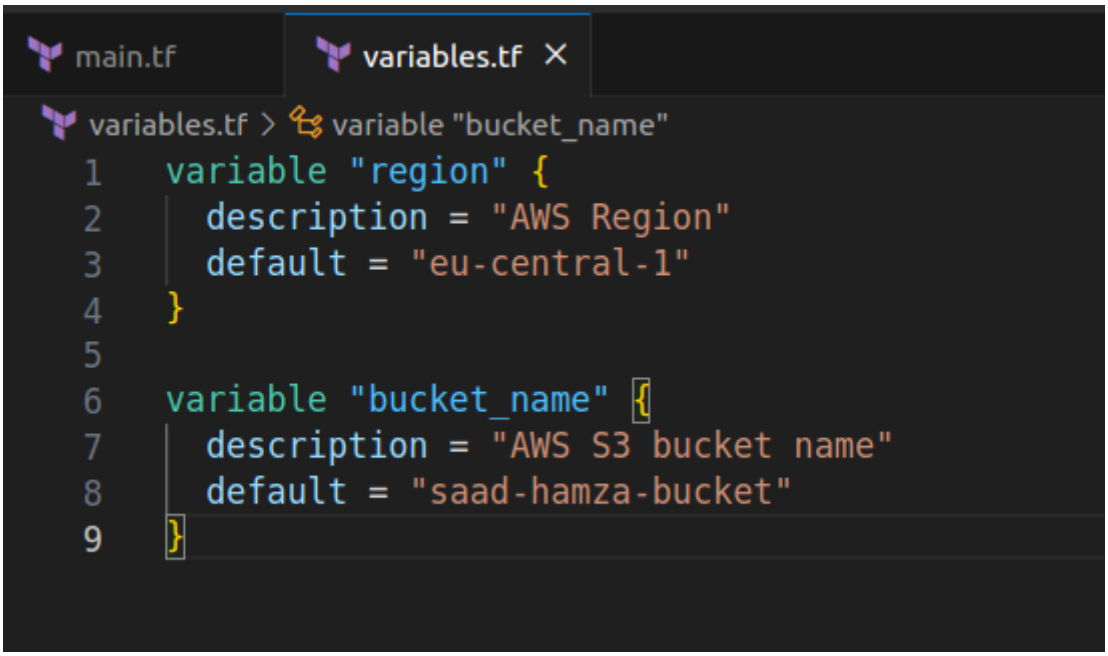
main.tf

- First we defined the providers we're using, AWS in our case.
- Next, we defined a region for our provider, via a variable.
- Next, we defined the module which contains the resources, and outputs we're going to be using. We also gave a name to our bucket via a variable.
- And finally, we provided an output block which will display the bucket ID on the terminal.



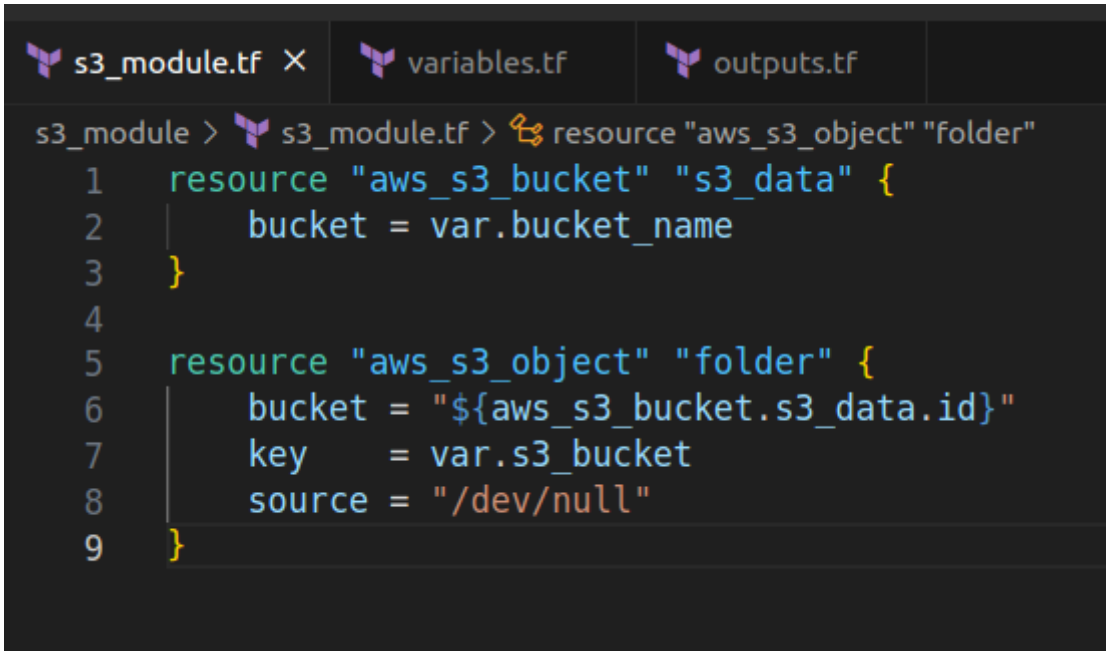
variables.tf (of main file)

- Here, we have just set the region name, “eu-central-1”
- and the name for our bucket “saad-hamza-bucket”



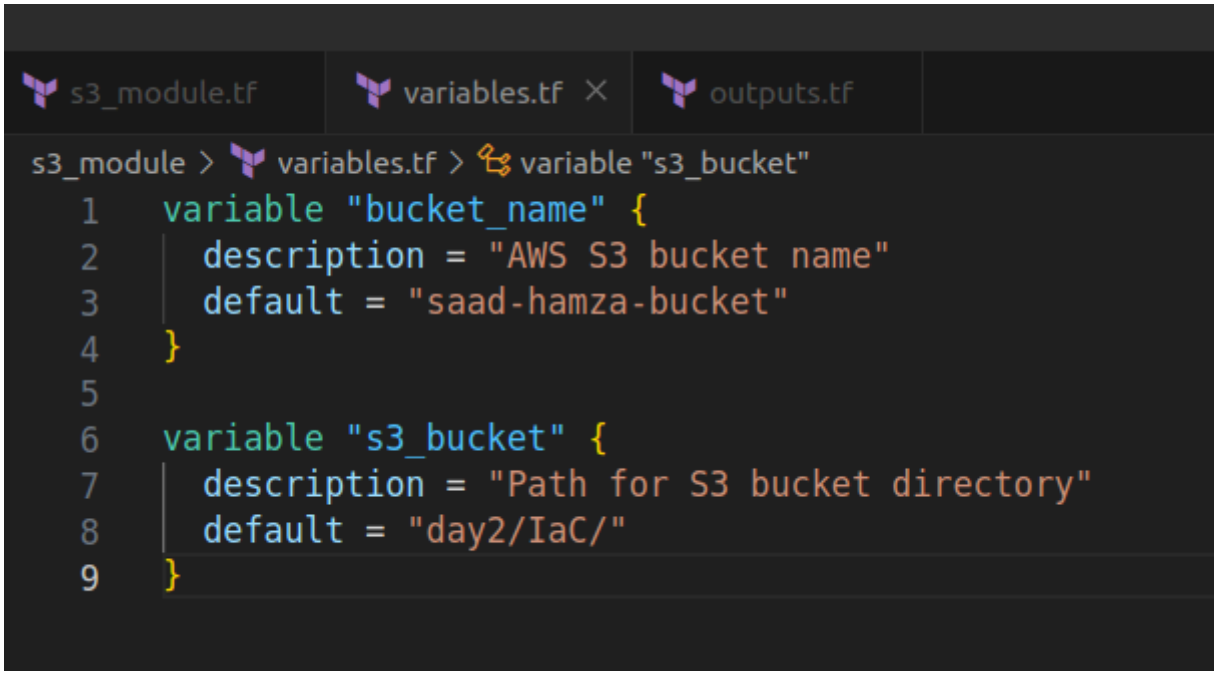
s3_module.tf

- We first create the bucket and also give a name to this resource “s3_data” which can be referenced by other resources.
- Next, we create a folder inside this bucket.
  - we first define which bucket we want to create the folder in.
  - the path of the folder inside the bucket is given via a variable called “s3_bucket”
  - finally, “/dev/null” defines that this will be an empty folder.



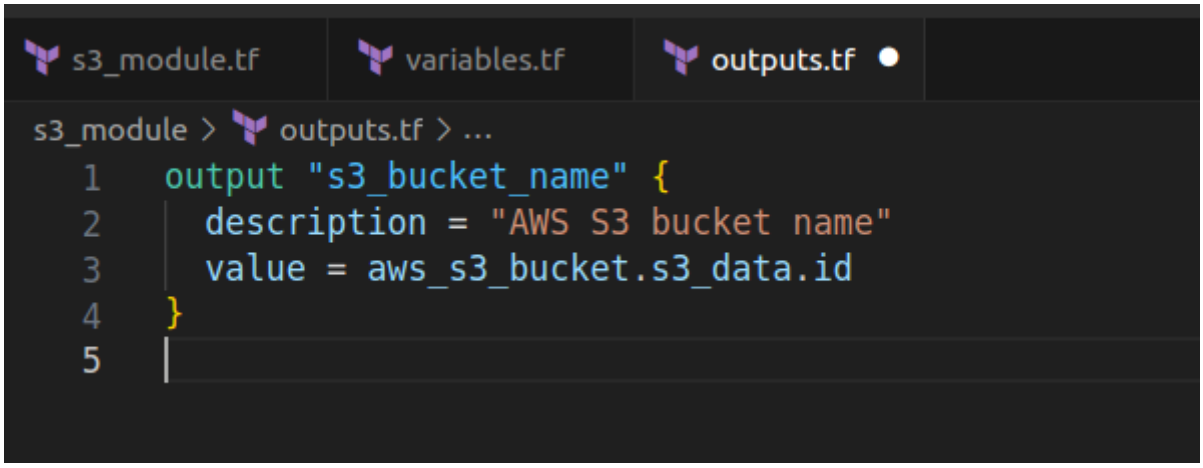
variables.tf (of s3_module)

- We have given a “bucket_name” variable again, so that it can be referenced in the s3_module.tf file.
- Then, we have defined a path for our folder inside the bucket “day2/IaC”. This is also referenced in the above s3_module.tf



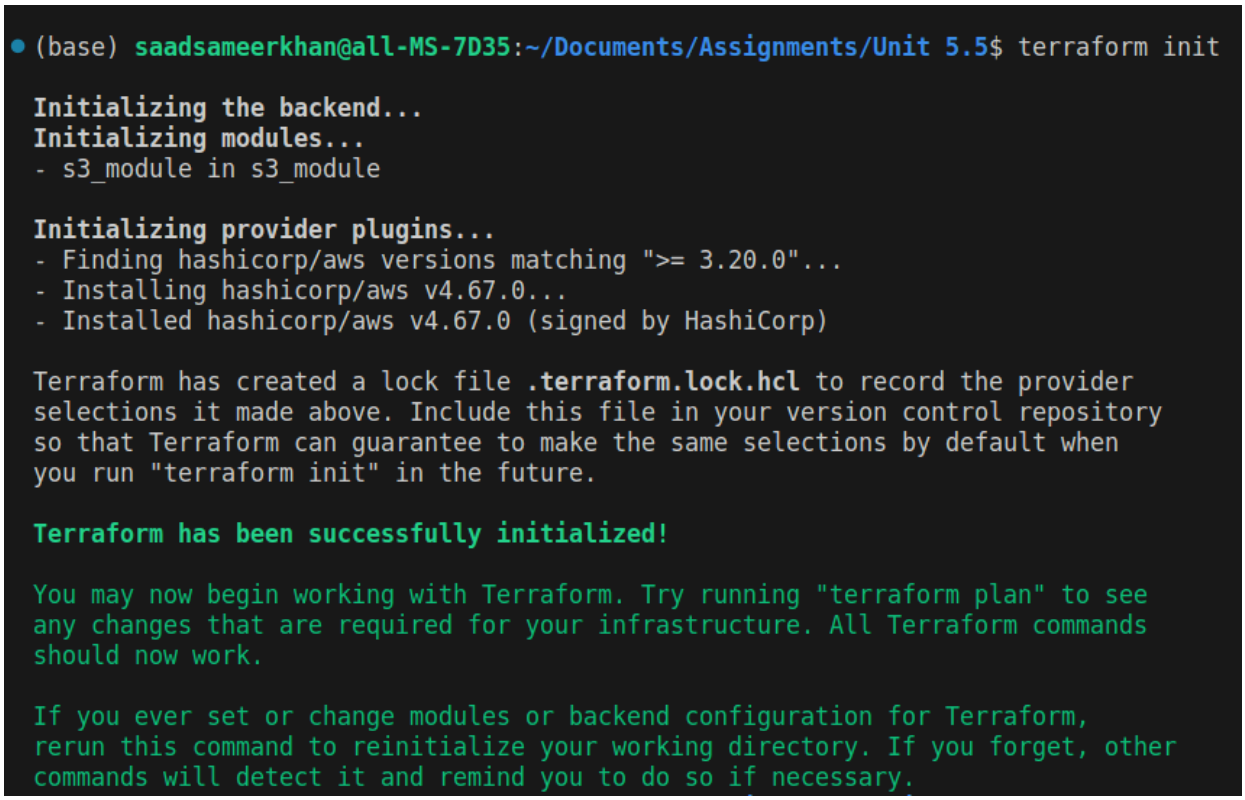
outputs.tf

- here, we have simply given the value that will be output to the screen. In our case, we’re outputting the bucket ID (“saad-hamza-bucket”)

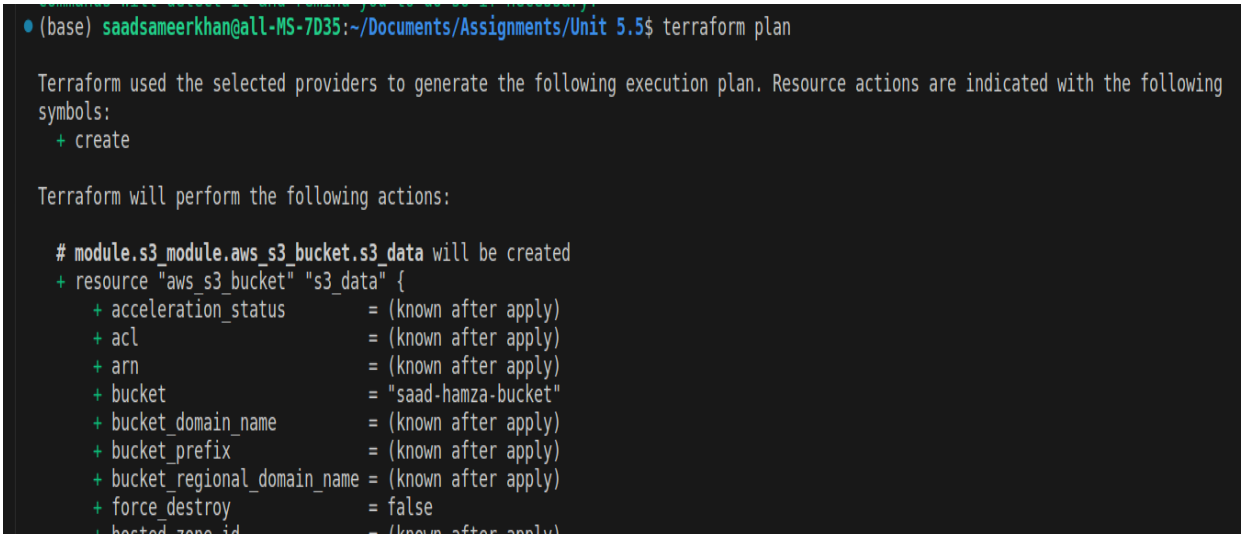


Results:

- First, we use terraform init to initialize terraform in our current folder.



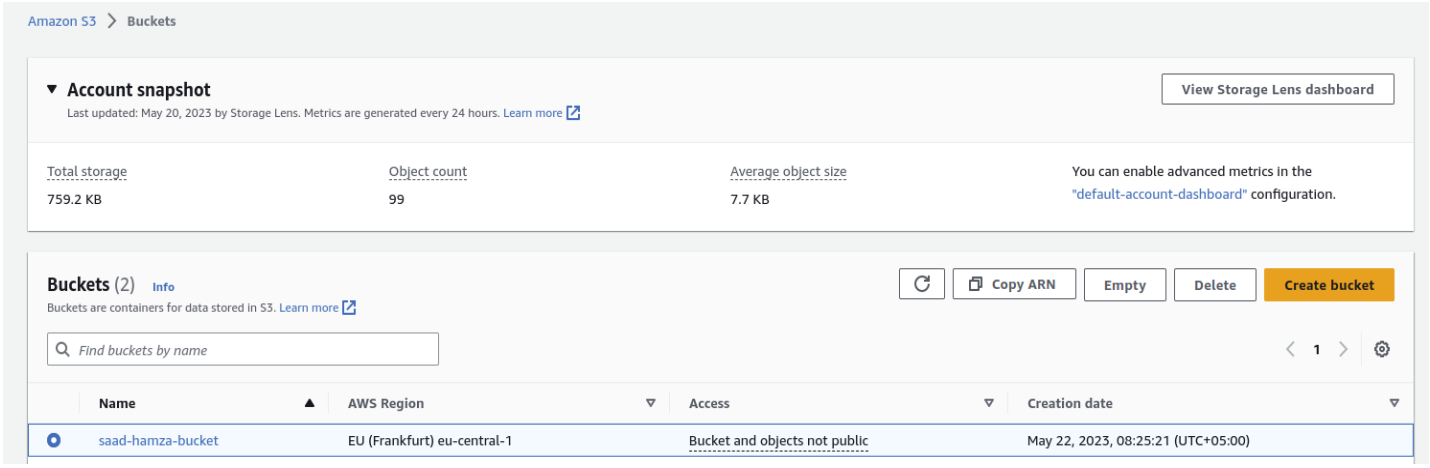
- Next, we do “terraform apply” to see what changes will take place.



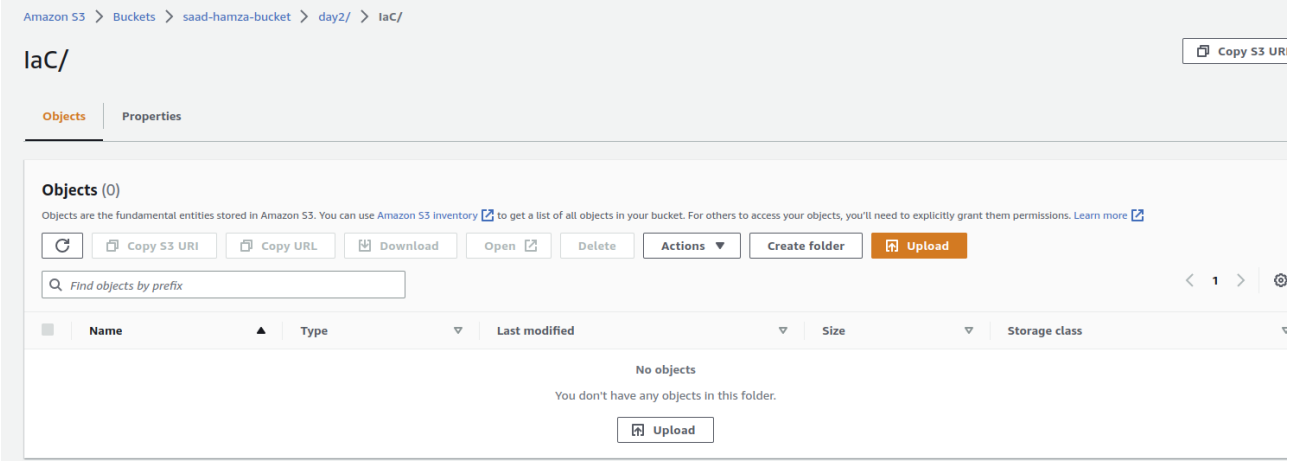
- We also see that our bucket name is displayed on the terminal after we did “terraform apply”

```
  Only 'yes' will be accepted to approve.

  Enter a value: yes

module.s3_module.aws_s3_bucket.s3_data: Creating...
module.s3_module.aws_s3_bucket.s3_data: Creation complete after 4s [id=saad-hamza-bucket]
module.s3_module.aws_s3_object.folder: Creating...
module.s3_module.aws_s3_object.folder: Creation complete after 1s [id=day2/IaC/]

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

Outputs:

s3_bucket_id = "saad-hamza-bucket"
```

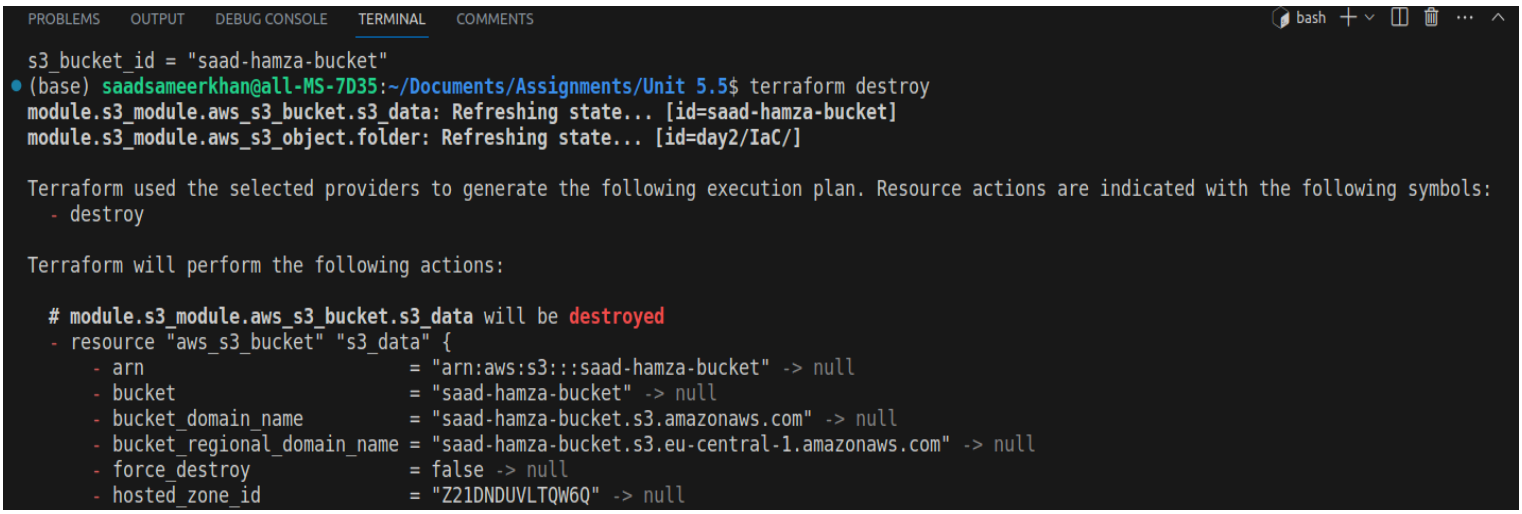
- Now we open AWS S3, and see that there is a bucket with the name “saad-hamza-bucket”

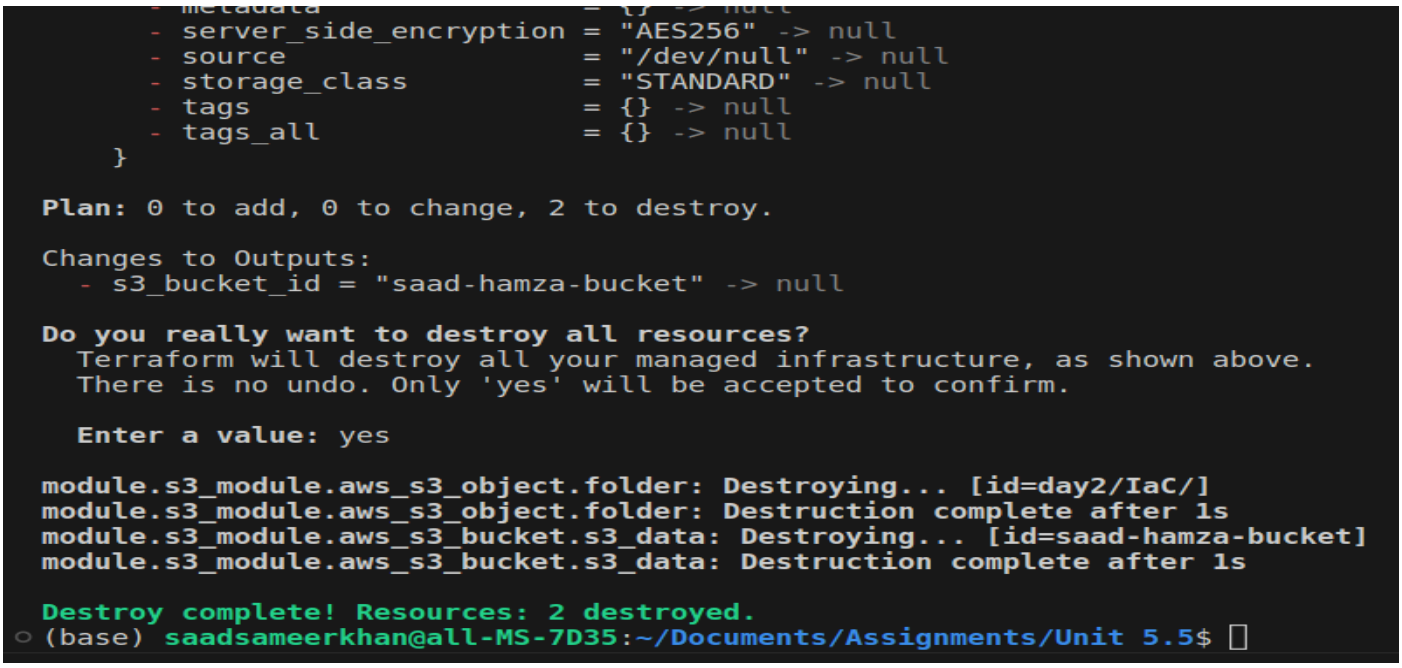


- We go inside this bucket and find that there is an empty folder created with the path “day2/IaC”



- Finally, we do “terraform destroy” to get rid of all the resources created using terraform.




- server_side_encryption = "AES256" -> null
- source = "/dev/null" -> null
- storage_class = "STANDARD" -> null
- tags = {} -> null
- tags_all = {} -> null
}
Plan: 0 to add, 0 to change, 2 to destroy.
Changes to Outputs:
- s3_bucket_id = "saad-hamza-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
module.s3_module.aws_s3_object.folder: Destroying... [id=day2/IaC/]
module.s3_module.aws_s3_object.folder: Destruction complete after 1s
module.s3_module.aws_s3_bucket.s3_data: Destroying... [id=saad-hamza-bucket]
module.s3_module.aws_s3_bucket.s3_data: Destruction complete after 1s
Destroy complete! Resources: 2 destroyed.
(base) saadsameerkhan@all-MS-7D35:~/Documents/Assignments/Unit 5.5$