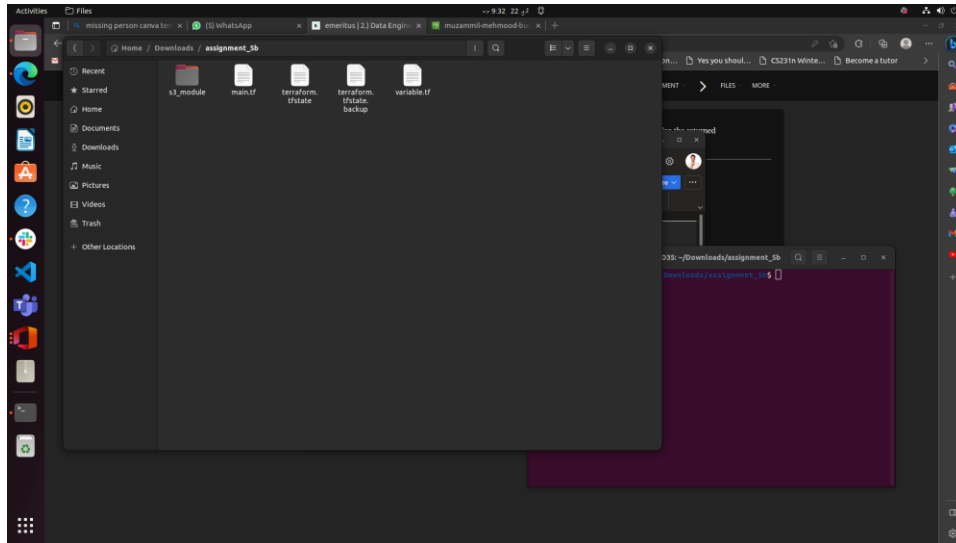


Make a module of yesterday's task, name it "s3_module". Pass bucket name to the module using variable "bucket_name". Use the returned "s3_bucket" variable to add "day2/laC/" directory to this bucket in the main module.

Created the module name s3_module



Initailizing the terraform in assignment directory

```
(base) muzammil@all-MS-7D35:~/Downloads/assignment_5b$ terraform init

Initializing the backend...
Initializing modules...

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) muzammil@all-MS-7D35:~/Downloads/assignment_5b$
```

Applying the terraform file

```
Activities Terminal -- 9:33 22 p.f. muzammil@ell-M5-7D35: ~/Downloads/assignment_5b

(base) muzammil@ell-M5-7D35:~/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_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/fac/"
  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)
}

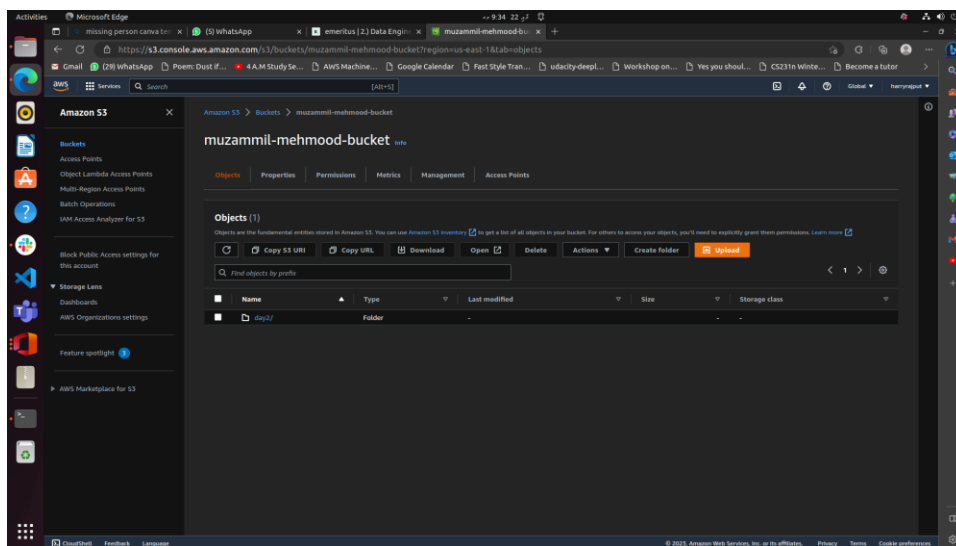
# module.s3_module.aws_s3_bucket.s3_module will be created
resource "aws_s3_bucket" "s3_module" {
  acceleration_status = (known after apply)
  acl                 = (known after apply)
  arns                = (known after apply)
  bucket              = "muzammil-mehmood-bucket"
  bucket_domain_name = (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)
}

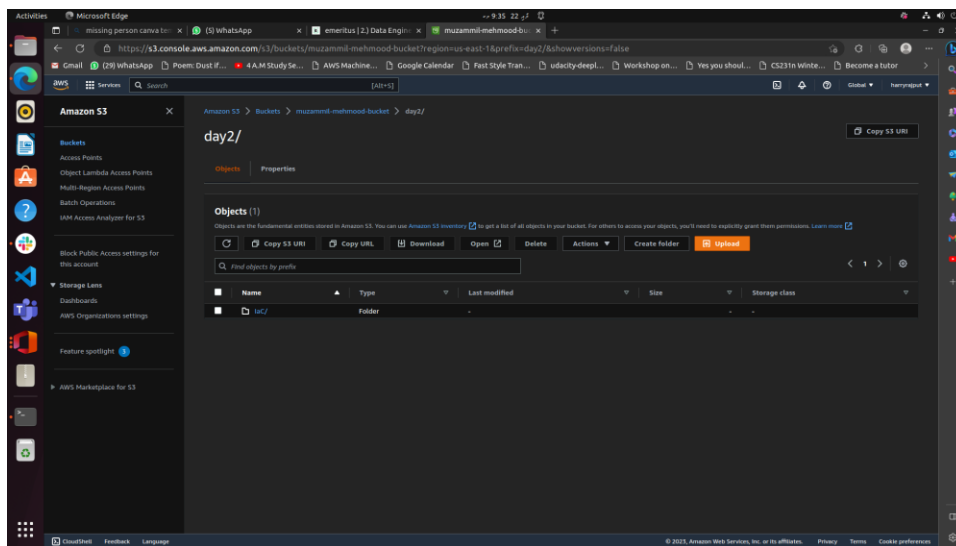
Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  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
```

This file has created the bucket with day2 folder in which there is another folder named laC





Now destroying the resources

