C:\Windows\System32\cmd.e X

Microsoft Windows [Version 10.0.22631.3880]

(c) Microsoft Corporation. All rights reserved.

C:\terra>terraform init

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.63.1...
- Installed hashicorp/aws v5.63.1 (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.

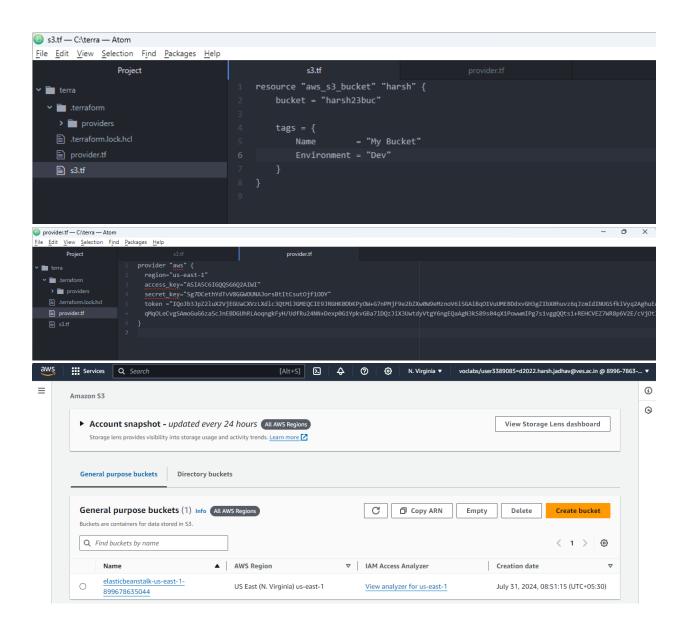
Terraform has been successfully initialized!

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:\terra>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:
       region = (I
request_payer = (I
tags = {
    "Environment" = "Dev"
    + "Name" = "My Bucket"
        fags_all = {
    "Environment" = "Dev"
    "Name" = "My Bucket"
      + website_domain
+ website_endpoint
                                     = (known after apply)
= (known after apply)
      + cors_rule (known after apply)
      + grant (known after apply)
      + lifecycle_rule (known after apply)
      + logging (known after apply)
      + object_lock_configuration (known after apply)
      + replication_configuration (known after apply)
      + server_side_encryption_configuration (known after apply)
      + versioning (known after apply)
      + website (known after apply)
Plan: 1 to add, 0 to change, 0 to destroy.
```

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:\terra>terraform apply
Terraferm used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbol + create
Terraform will perform the following actions:
  # ams_s3_bucket.harsh will be created
+ resource 'ams_s3_bucket' 'harsh' {
+ acceleration_status = ()
+ acl = ()
         - 'Environment' = 'Dev'
+ 'Name' = 'Hy Bucket'

}
tags_all = {
    'Environment' = 'Dev'
    'Name' = 'Hy Bucket'
}

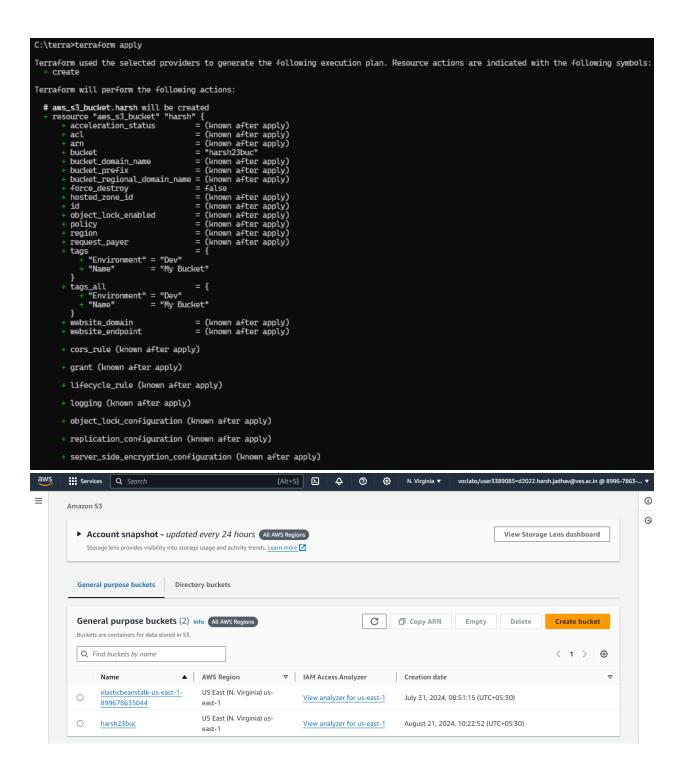
        + website_domain
+ website_endpoint
                                     = (known after apply)
= (known after apply)
       + cors_rule (known after apply)

    grant (known after apply)

       + lifecycle_rule (known after apply)
      + logging (known after apply)
      + object_lock_configuration (known after apply)
      + replication_configuration (known after apply)
       + server_side_encryption_configuration (known after apply)

    versioning (known after apply)

       + website (known after apply)
Plan: 1 to add, \theta to change, \theta to destroy.
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
ams_s3_bucket.harsh: Creating...
ams_s3_bucket.harsh: Creation complete after 5s [id=harsh23buc]
C:\terra>
```



```
C:\terra>terraform destroy
aws_s3_bucket.harsh: Refreshing state... [id=harsh23buc]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
 Terraform will perform the following actions:
         bucket = "harsh23buc" -> null
bucket_domain_name = "harsh23buc.s3.amazonaws.com" -> null
bucket_regional_domain_name = "harsh23buc.s3.us-east-1.amazonaws.com" -> null
force_destroy = false -> null
hosted_zone_id = "Z3AOBSTGFYJSTE" -> null
                                             = "harsh23buc" -> null
= false -> null
= "us-east-1" -> null
= "Bucket0wner" -> null
          object_lock_enabled
          request_payer
          tags

- "Environment" = "Dev"

- "Name" = "My Bucket"
          } -> null
tags_all
               _all
"Environment" = "Dev"
"Name" = "My Bucket"
          - "EINJITONNERT = "My Bucket"
} -> null
# (3 unchanged attributes hidden)
               grant {
— id
               - "FULL_CONTROL",
] -> null
type = "CanonicalUser" -> null
# (1 unchanged attribute hidden)
          server_side_encryption_configuration {
               rule {
- bucket_key_enabled = false -> null
                  - apply_server_side_encryption_by_default {
    - sse_algorithm = "AES256" -> null
    # (1 unchanged attribute hidden)
                                    # (1 unchanged attribute hidden)
                             }
                      }
               }
               versioning {
                    enabled 
                                       = false -> null
                      mfa_delete = false -> null
               3
        }
```

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

Do you really want to destroy all resources?

Destroy complete! Resources: 1 destroyed.

Enter a value: yes

C:\terra>

Terraform will destroy all your managed infrastructure, as shown above.

There is no undo. Only 'yes' will be accepted to confirm.

aws_s3_bucket.harsh: Destroying... [id=harsh23buc]
aws_s3_bucket.harsh: Destruction complete after 1s

