Terraform Assignment: EC2 and S3 Resource Lifecycle

📌 Objective: Use Terraform to provision and then clean up AWS resources (EC2 + S3).

📂 Project Structure

```
terraform-assignment/
|── provider.tf
|── ec2.tf
|── s3.tf
|── variables.tf
|── outputs.tf
|── README.md
```

⚙️ Steps to Run

1. Initialize Terraform

```
terraform init
```

2. Apply (Provision Resources)

```
terraform apply -auto-approve
```

✅ This will create:

- One **t2.micro EC2 instance** in `us-east-1`
- One **unique S3 bucket**

3. Check Outputs

```
terraform output
```

- Shows EC2 Public IP
- Shows S3 Bucket Name

4. Destroy Resources

```
terraform destroy -auto-approve
```

✅ This will delete all resources.

🖼️ Deliverables (for Assignment Submission)

- Screenshot of terraform installation on windows
- Terraform code files (`.tf`)
- Screenshot of `terraform apply` in terminal
- Screenshot of EC2 instance in AWS Console
- Screenshot of S3 bucket in AWS Console
- Screenshot of `terraform destroy` in terminal

🎁 Bonus (Tags)

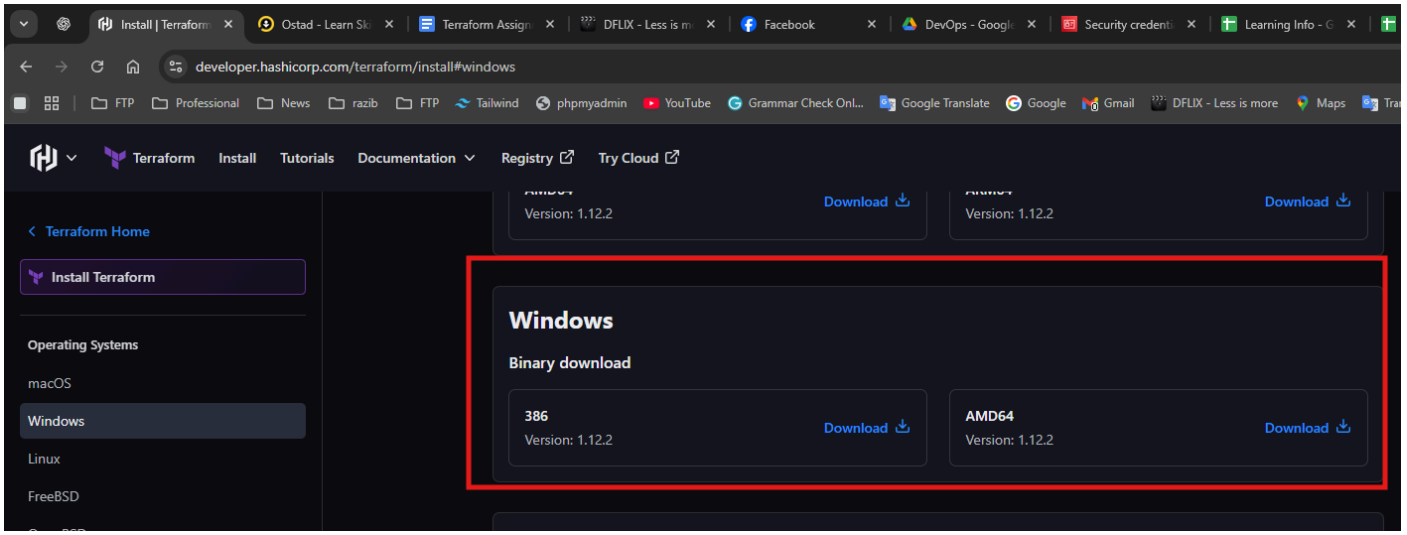
- EC2 Instance → `Name = TerraformAssignment-EC2`
- S3 Bucket → `Name = TerraformAssignment-S3`

🎁 Github Repo:

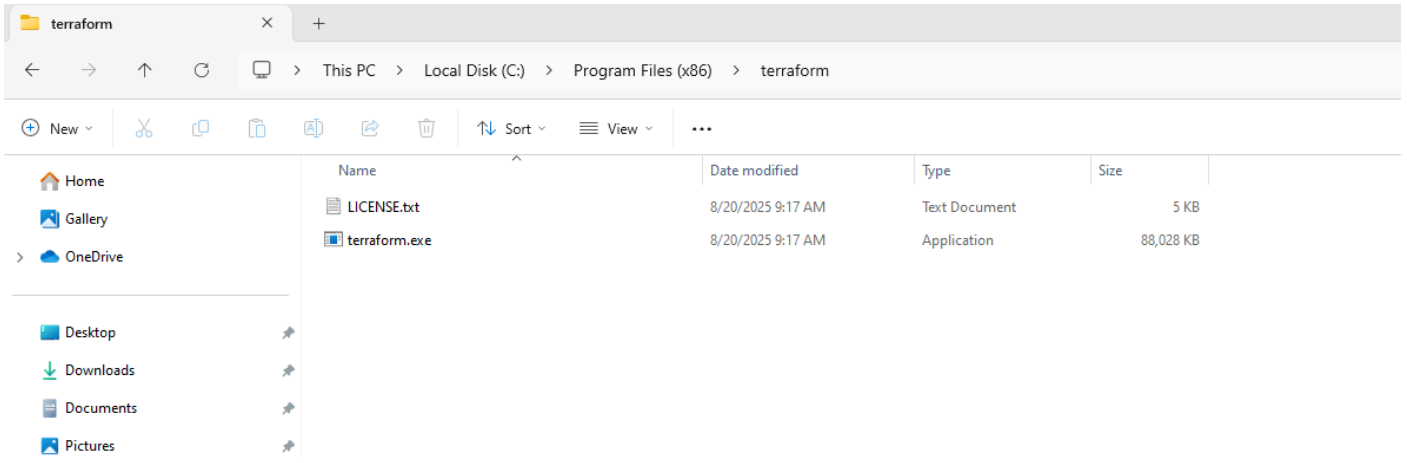
- Link: https://github.com/engr-razib/OstadDevOps/tree/main/TerraformAssignment

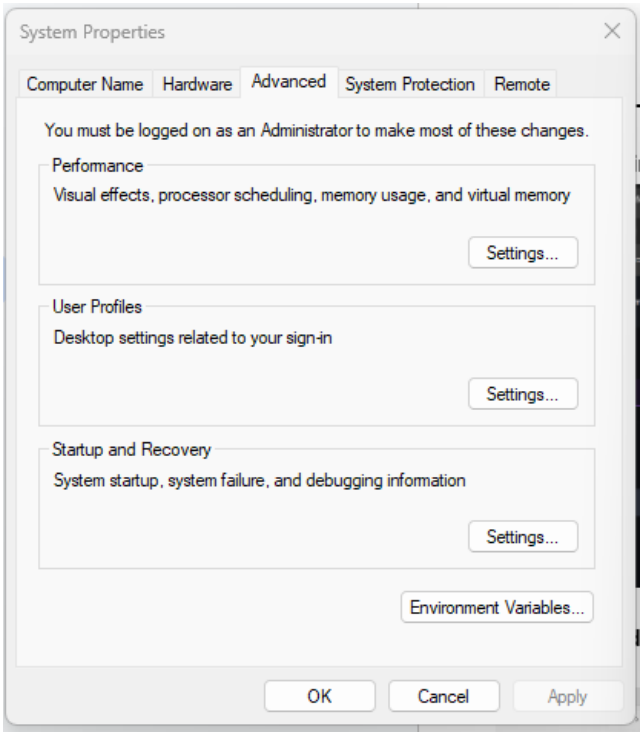
---------- **Screenshot of terraform installation on windows** ----------

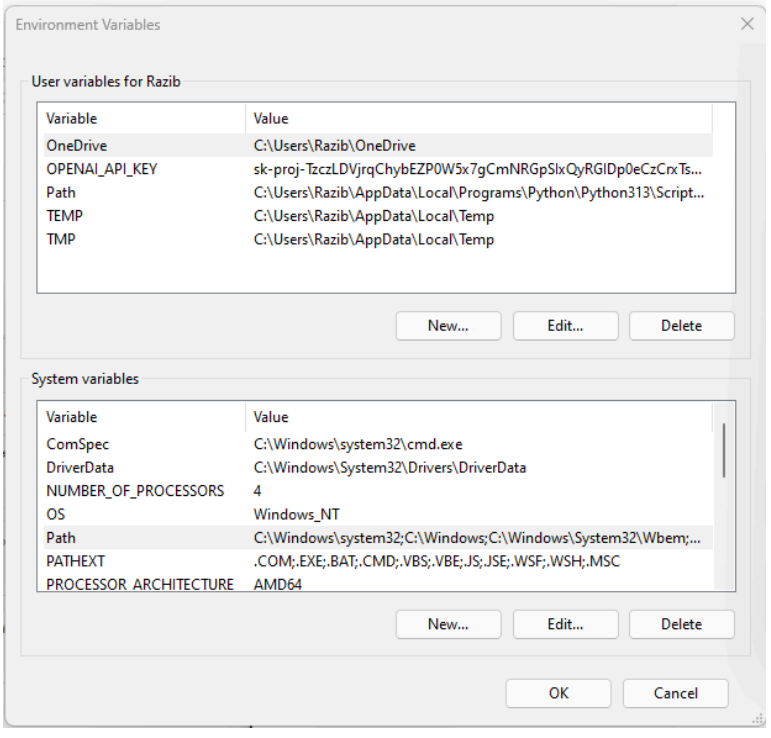
1. Download installation file form internet:

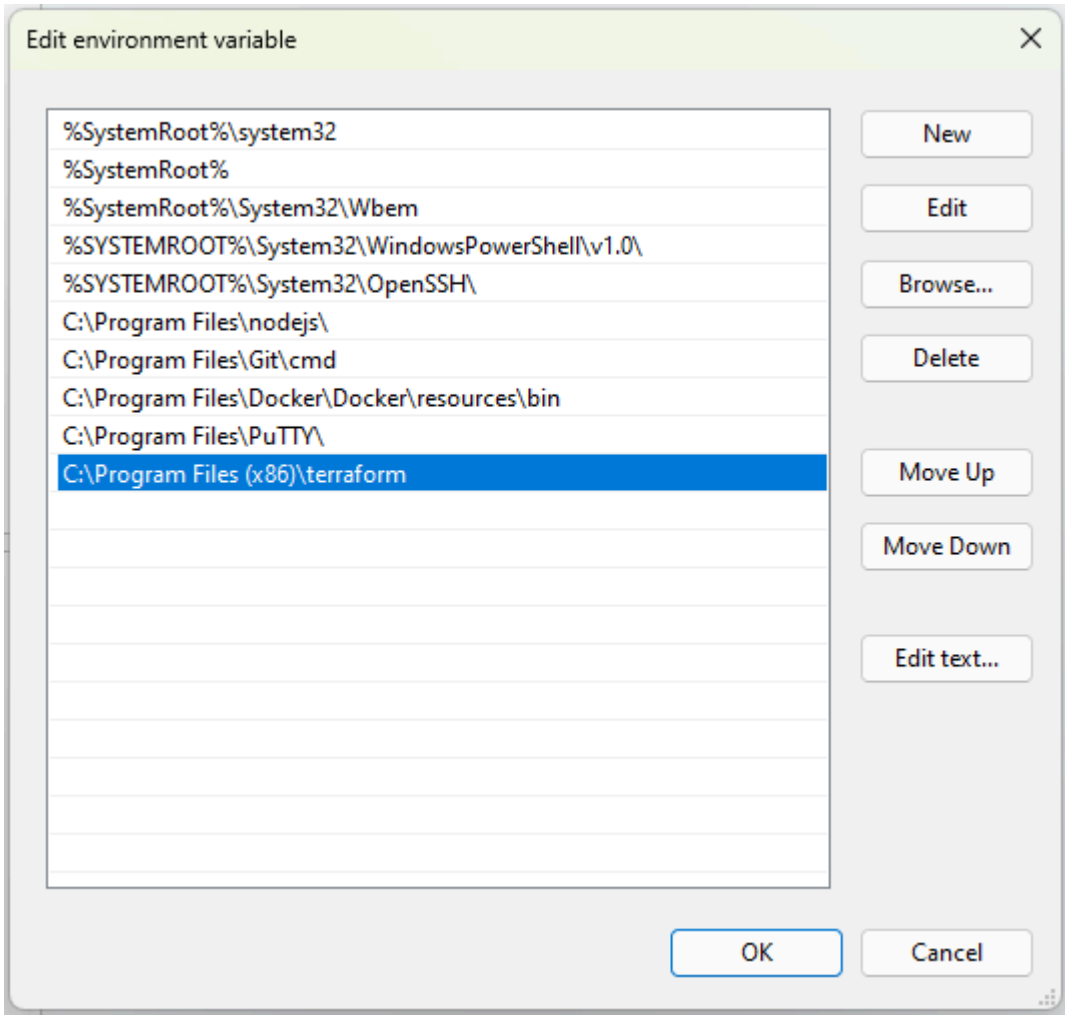


2. Extract and Environment variable setup:








Edit environment variable
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\Program Files\Docker\Docker\resources\bin
C:\Program Files\PuTTY\
C:\Program Files (x86)\terraform
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

---------- Screenshot of `terraform apply` in terminal ----------

```
E:\Server\github\OstadDevOps\TerraformAssignment>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/aws v6.9.0...
- Installed hashicorp/aws v6.9.0 (signed by HashiCorp)
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (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.

E:\Server\github\OstadDevOps\TerraformAssignment>
```

---------- Screenshot of `Setup AWS & AWS Profile` in terminal ----------

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.5768]
(c) Microsoft Corporation. All rights reserved.

E:\Server\github\OstadDevOps\TerraformAssignment>aws --version
aws-cli/2.28.13 Python/3.13.4 Windows/11 exe/AMD64

E:\Server\github\OstadDevOps\TerraformAssignment>aws --version
aws-cli/2.28.13 Python/3.13.4 Windows/11 exe/AMD64

E:\Server\github\OstadDevOps\TerraformAssignment>aws configure --profile ostad
AWS Access Key ID [None]: AKIA2YC4YENLOBY5OYFZ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json

E:\Server\github\OstadDevOps\TerraformAssignment>
```

---------- Screenshot of `Terraform Execution` in terminal ----------

1. Terraform init:

```
E:\Server\github\OstadDevOps\TerraformAssignment>terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/aws v6.9.0
- Using previously-installed hashicorp/random v3.7.2

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.
```

2. Terraform plan :

```
E:\Server\github\OstadDevOps\TerraformAssignment>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_instance.my_ec2 will be created
  + resource "aws_instance" "my_ec2" {
      + ami                                  = "ami-08c40ec9ead489470"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + region                               = "us-east-1"
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "TerraformAssignment-EC2"
        }
      + tags_all                             = {
```

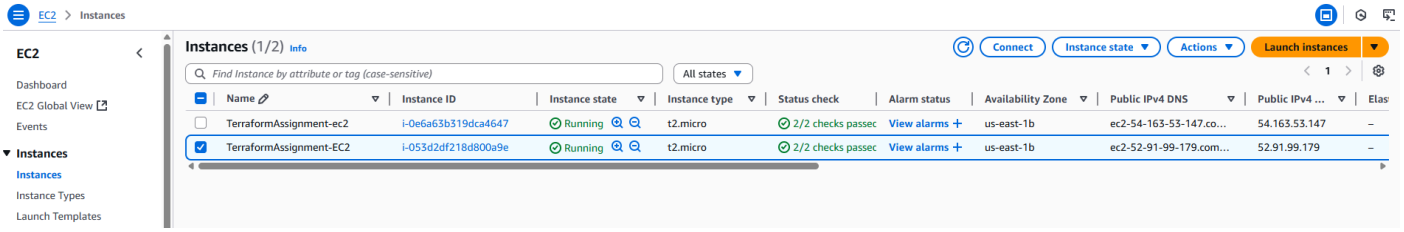
```
      + grant (known after apply)

      + 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)
    }

  # random_id.bucket_suffix will be created
  + resource "random_id" "bucket_suffix" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 4
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + ec2_public_ip  = (known after apply)
  + s3_bucket_name = (known after apply)

Warning: Argument is deprecated

  with aws_s3_bucket.my_bucket,
  on s3.tf line 9, in resource "aws_s3_bucket" "my_bucket":
   9:   acl    = "private"

acl is deprecated. Use the aws_s3_bucket_acl resource instead.

(and one more similar warning elsewhere)

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.
```

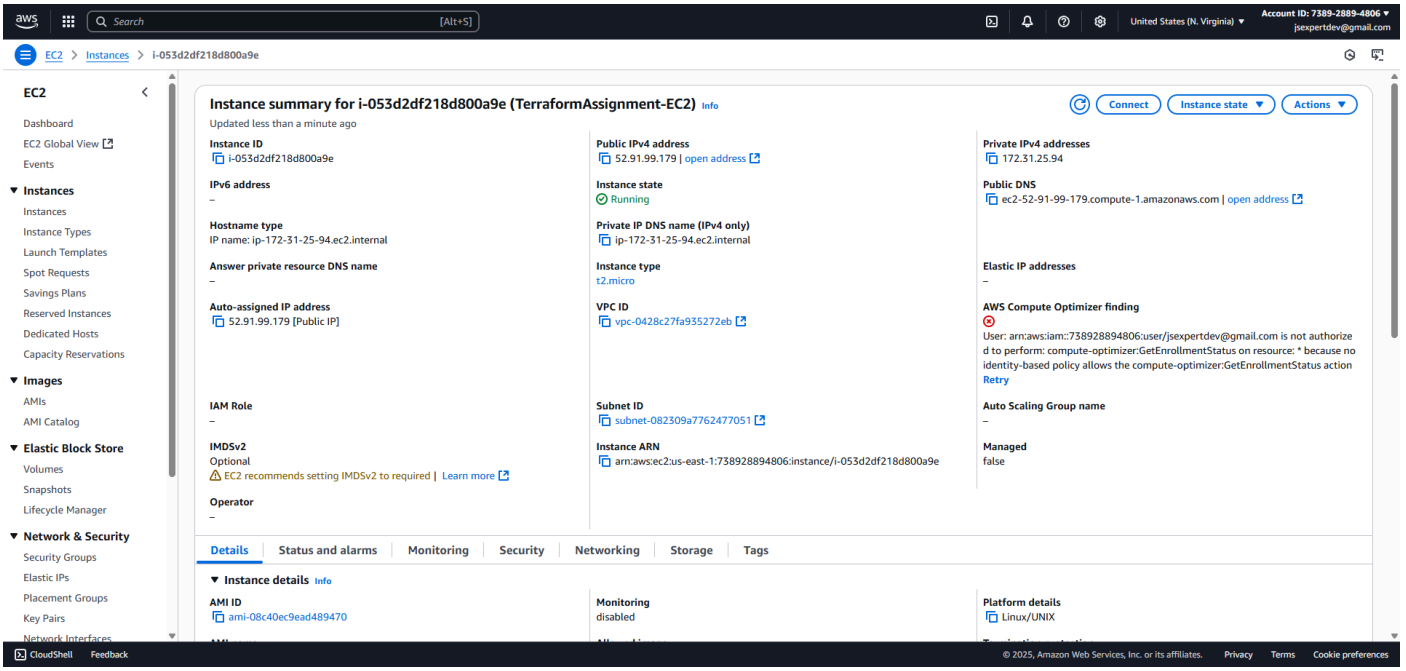
3. Terraform apply:

```
E:\Server\github\OstadDevOps\TerraformAssignment>
E:\Server\github\OstadDevOps\TerraformAssignment>terraform apply -auto-approve

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_instance.my_ec2 will be created
  + resource "aws_instance" "my_ec2" {
      + ami                                  = "ami-08c40ec9ead489470"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + region                               = "us-east-1"
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
```

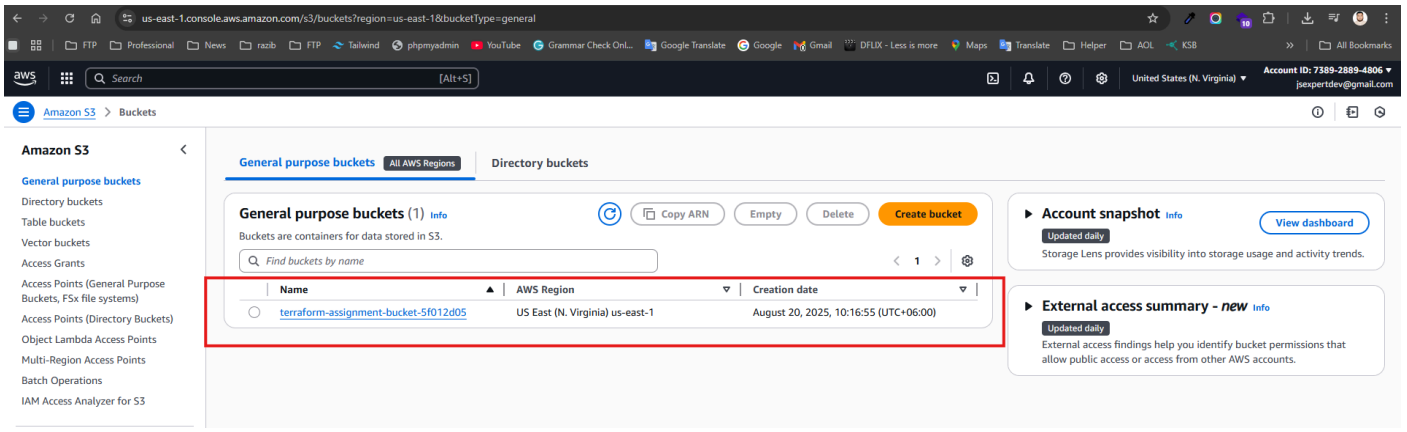
```
      + replication_configuration (known after apply)

      + server_side_encryption_configuration (known after apply)

      + versioning (known after apply)

      + website (known after apply)
    }

  # random_id.bucket_suffix will be created
  + resource "random_id" "bucket_suffix" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 4
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + ec2_public_ip  = (known after apply)
  + s3_bucket_name = (known after apply)
random_id.bucket_suffix: Creating...
random_id.bucket_suffix: Creation complete after 0s [id=XwEtBQ]
aws_s3_bucket.my_bucket: Creating...
aws_instance.my_ec2: Creating...
aws_s3_bucket.my_bucket: Creation complete after 8s [id=terraform-assignment-bucket-5f012d05]
aws_instance.my_ec2: Still creating... [00m10s elapsed]
aws_instance.my_ec2: Creation complete after 18s [id=i-053d2df218d800a9e]

  Warning: Argument is deprecated

    with aws_s3_bucket.my_bucket,
    on s3.tf line 9, in resource "aws_s3_bucket" "my_bucket":
     9:   acl    = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.

  (and 2 more similar warnings elsewhere)


Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

ec2_public_ip = "52.91.99.179"
s3_bucket_name = "terraform-assignment-bucket-5f012d05"

E:\Server\github\OstadDevOps\TerraformAssignment>
```

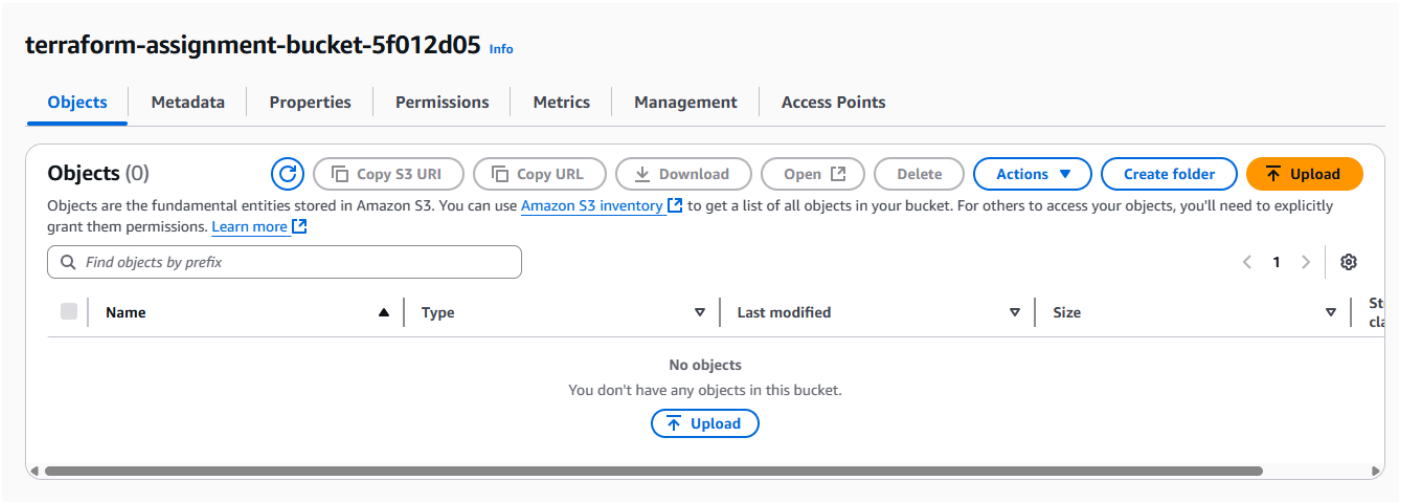
---------- Screenshot of EC2 Instance Created By Terraform on AWS ----------





---------- **Screenshot of S3 bucket Created By Terraform on AWS** ----------

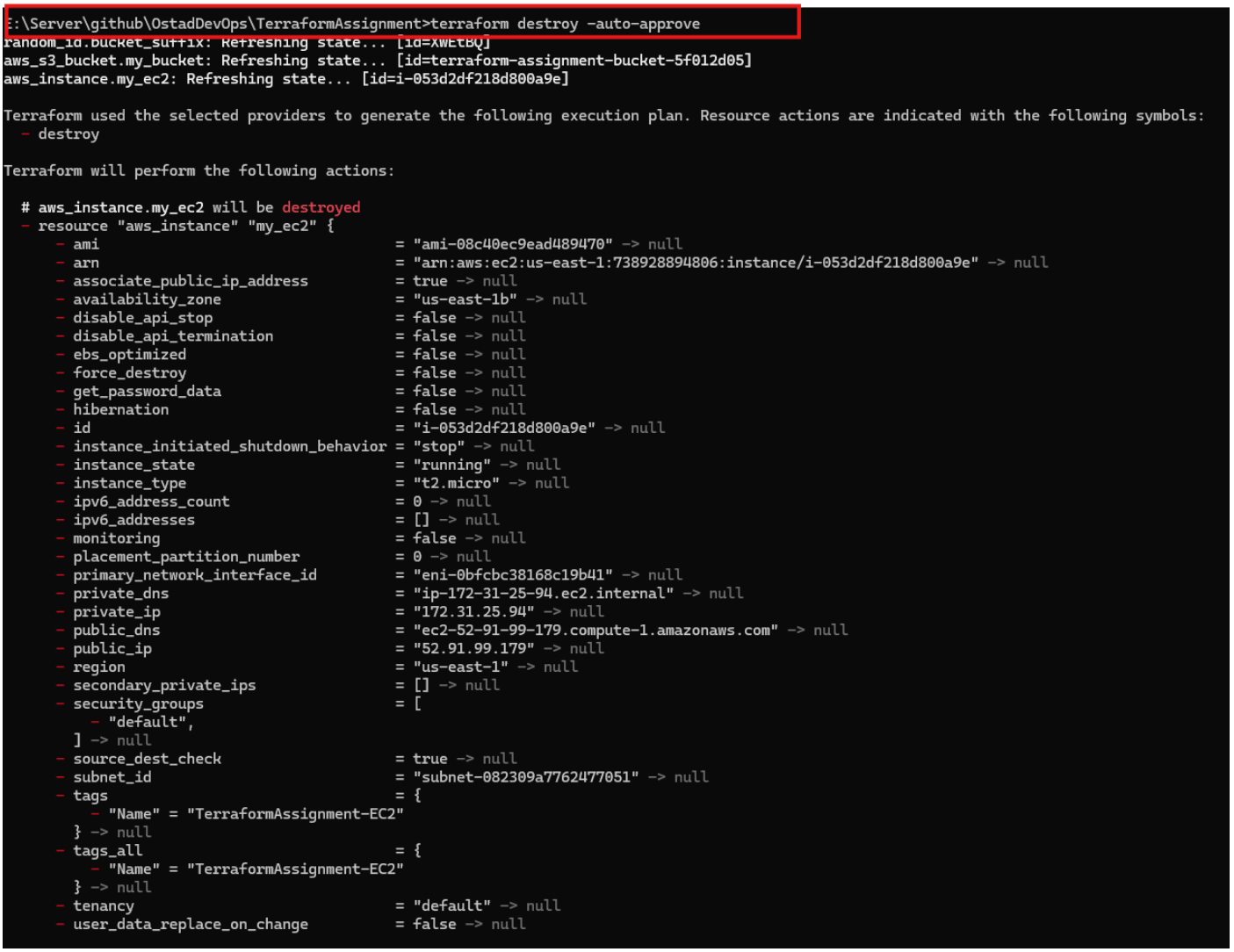




---------- **Screenshot of the terminal showing successful terraform destroy.** ----------

Terraform Destroy :

AWS Instance Destroy



```
E:\Server\github\OstadDevOps\TerraformAssignment>terraform destroy -auto-approve
random_id.bucket_suffix: Refreshing state... [id=xWEtBQ]
aws_s3_bucket.my_bucket: Refreshing state... [id=terraform-assignment-bucket-5f012d05]
aws_instance.my_ec2: Refreshing state... [id=i-053d2df218d800a9e]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.my_ec2 will be destroyed
  - resource "aws_instance" "my_ec2" {
      - ami                                  = "ami-08c40ec9ead489470" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:738928894806:instance/i-053d2df218d800a9e" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1b" -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - force_destroy                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-053d2df218d800a9e" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - monitoring                           = false -> null
      - placement_partition_number           = 0 -> null
      - primary_network_interface_id         = "eni-0bfcbc38168c19b41" -> null
      - private_dns                          = "ip-172-31-25-94.ec2.internal" -> null
      - private_ip                           = "172.31.25.94" -> null
      - public_dns                           = "ec2-52-91-99-179.compute-1.amazonaws.com" -> null
      - public_ip                            = "52.91.99.179" -> null
      - region                               = "us-east-1" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
          - "default",
        ] -> null
      - source_dest_check                    = true -> null
      - subnet_id                            = "subnet-082309a7762477051" -> null
      - tags                                 = {
          - "Name" = "TerraformAssignment-EC2"
        } -> null
      - tags_all                             = {
          - "Name" = "TerraformAssignment-EC2"
        } -> null
      - tenancy                              = "default" -> null
      - user_data_replace_on_change          = false -> null
```

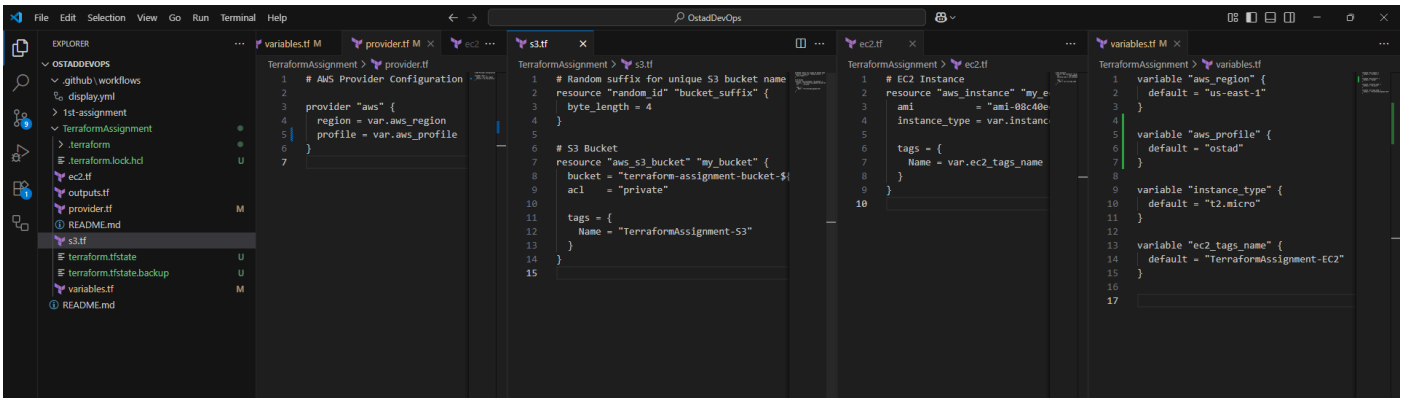
S3 Bucket Destroy:

```
# aws_s3_bucket.my_bucket will be destroyed
- resource "aws_s3_bucket" "my_bucket" {
    - acl                         = "private" -> null
    - arn                         = "arn:aws:s3:::terraform-assignment-bucket-5f012d05" -> null
    - bucket                      = "terraform-assignment-bucket-5f012d05" -> null
    - bucket_domain_name          = "terraform-assignment-bucket-5f012d05.s3.amazonaws.com" -> null
    - bucket_region               = "us-east-1" -> null
    - bucket_regional_domain_name = "terraform-assignment-bucket-5f012d05.s3.us-east-1.amazonaws.com" -> null
    - force_destroy               = false -> null
    - hosted_zone_id              = "Z3AQBSTGFYJSTF" -> null
    - id                          = "terraform-assignment-bucket-5f012d05" -> null
    - object_lock_enabled         = false -> null
    - region                      = "us-east-1" -> null
    - request_payer               = "BucketOwner" -> null
    - tags                        = {
        - "Name" = "TerraformAssignment-S3"
      } -> null
    - tags_all                    = {
        - "Name" = "TerraformAssignment-S3"
      } -> null
      # (3 unchanged attributes hidden)

    - grant {
        - id          = "920472250e8af413fe3af7e29f0203235bf5d3991e05cb698b40f75d3011e0be" -> null
        - permissions = [
            - "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)
              }
          }
      }

    - versioning {
        - enabled    = false -> null
        - mfa_delete = false -> null
      }
  }
```

Destroy Complete:

```
  # random_id.bucket_suffix will be destroyed
  - resource "random_id" "bucket_suffix" {
      - b64_std     = "XwEtBQ==" -> null
      - b64_url     = "XwEtBQ" -> null
      - byte_length = 4 -> null
      - dec         = "1593912581" -> null
      - hex         = "5f012d05" -> null
      - id          = "XwEtBQ" -> null
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Changes to Outputs:
  - ec2_public_ip  = "52.91.99.179" -> null
  - s3_bucket_name = "terraform-assignment-bucket-5f012d05" -> null
aws_s3_bucket.my_bucket: Destroying... [id=terraform-assignment-bucket-5f012d05]
aws_instance.my_ec2: Destroying... [id=i-053d2df218d800a9e]
aws_s3_bucket.my_bucket: Destruction complete after 1s
random_id.bucket_suffix: Destroying... [id=XwEtBQ]
random_id.bucket_suffix: Destruction complete after 0s
aws_instance.my_ec2: Still destroying... [id=i-053d2df218d800a9e, 00m10s elapsed]
aws_instance.my_ec2: Still destroying... [id=i-053d2df218d800a9e, 00m20s elapsed]
aws_instance.my_ec2: Still destroying... [id=i-053d2df218d800a9e, 00m30s elapsed]
aws_instance.my_ec2: Destruction complete after 32s

Warning: Argument is deprecated

  with aws_s3_bucket.my_bucket,
  on s3.tf line 9, in resource "aws_s3_bucket" "my_bucket":
   9:   acl    = "private"

acl is deprecated. Use the aws_s3_bucket_acl resource instead.


Destroy complete! Resources: 3 destroyed.

E:\Server\github\OstadDevOps\TerraformAssignment>
```

---------- **Screenshot of Terraform .ty code base** ----------



Terraform Assignment: EC2 and S3 Resource Lifecycle

Objective:
Use Terraform to provision and then clean up AWS resources.

Task Description:
You are required to write Terraform configuration files to perform the following tasks:

1. Create Resources:
- o Launch an EC2 instance using the t2.micro type in the us-east-1 region.
- o Create an S3 bucket with a unique name.
- o Ensure both resources are defined in your Terraform configuration using appropriate AWS provider settings.

2. **Provision Resources:**
- o Use terraform init and terraform apply to provision the EC2 instance and S3 bucket.
- o Validate that both resources are successfully created in your AWS account.

3. Destroy Resources:
- o After verification, use terraform destroy to remove all created resources.

Deliverables:
· Submit the following as a single PDF:
- o Screenshots of the created EC2 instance and S3 bucket in the AWS Console.
- o Screenshot of the terminal showing terraform apply execution.
- o Screenshot of the terminal showing successful terraform destroy.
- o Your Terraform .tf configuration files (as code snippets or screenshots).

Bonus:
Add tags to your EC2 instance and S3 bucket (e.g., Name = "TerraformAssignment").