UPES

UNIVERSITY OF TOMORROW

System Provisioning & Configuration Management

ASSIGNMENT-1

SUBMITTED BY:

NAME: DIYA TOMAR

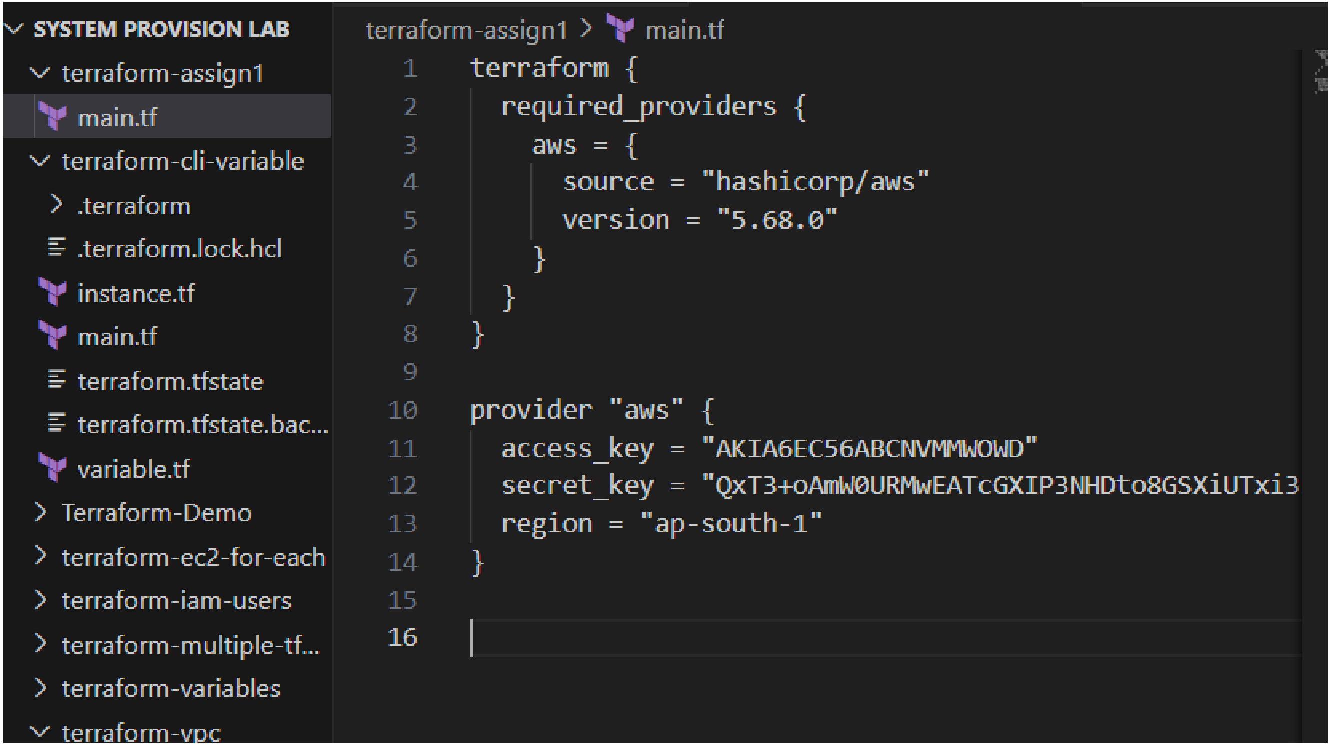
SAP ID: 500107068

R.NO: R2142220860

BATCH: 2 (DevOps)

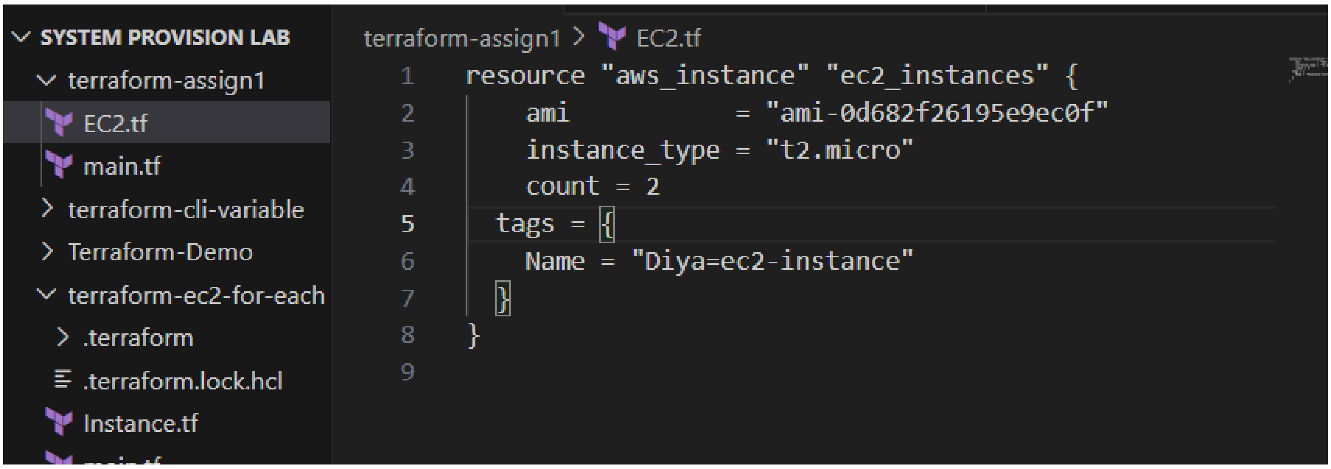
Write Terraform script to do perform following tasks on AWS cloud Platform

Step 1: Create Terraform Configuration File (main.tf):



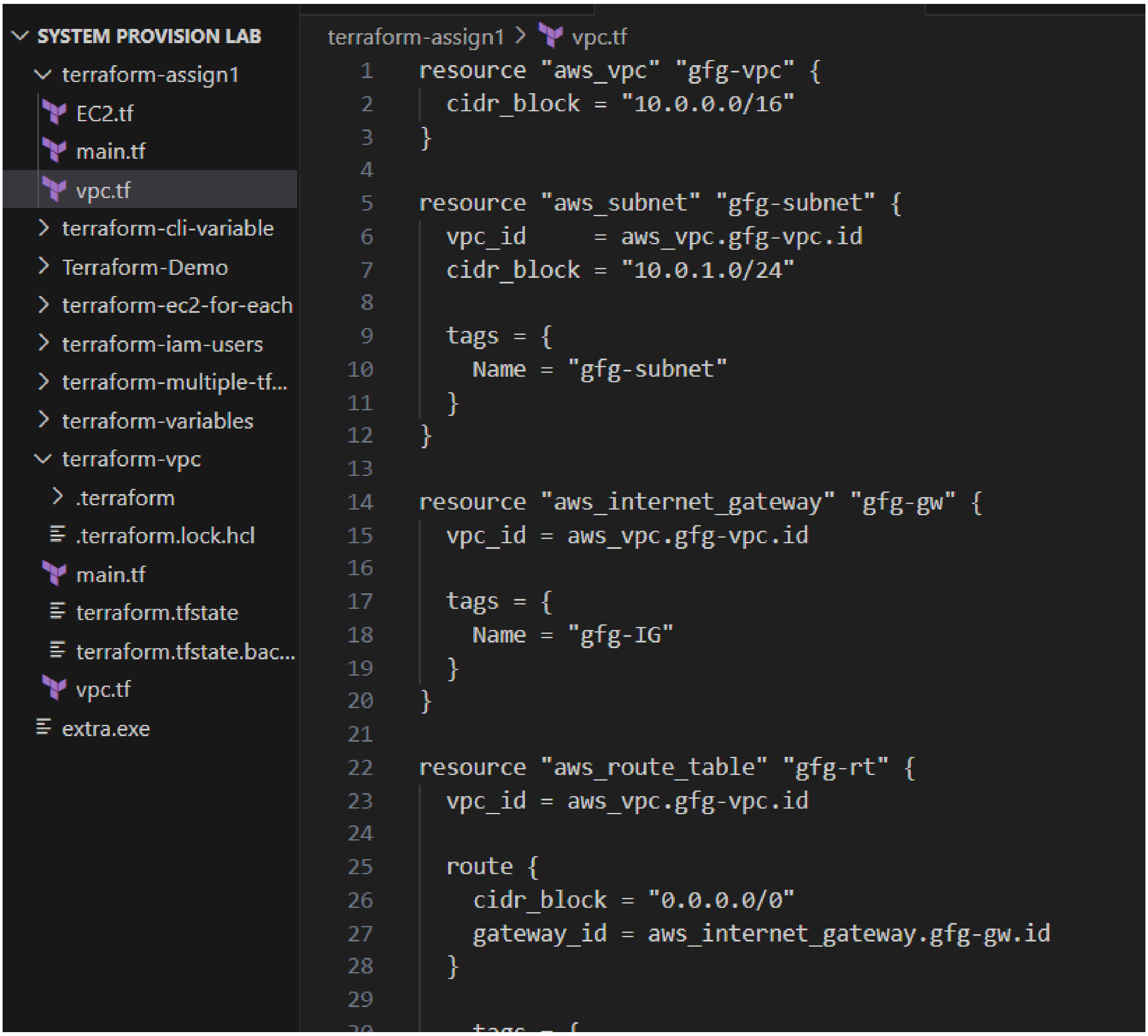
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.68.0"
    }
  }
}

provider "aws" {
  access_key = "AKIA6EC56ABCNVMMWOWD"
  secret_key = "QxT3+oAmW0URMwEATcGXIP3NHDto8GSXiUTxi3
  region = "ap-south-1"
}

```

Step 2: To create two T2 Micro EC2 Instances create Ec2.tf :



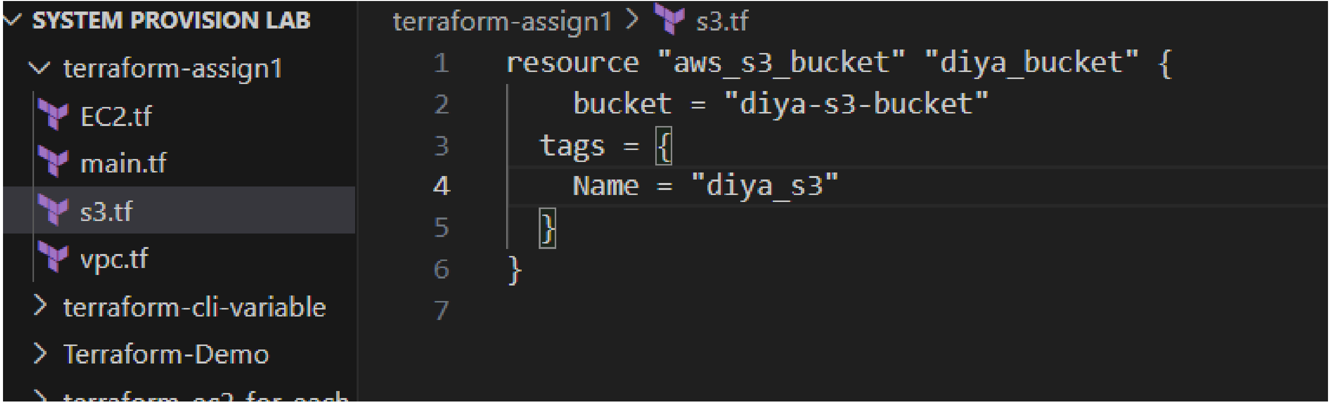
```
resource "aws_instance" "ec2_instances" {
    ami           = "ami-0d682f26195e9ec0f"
    instance_type = "t2.micro"
    count = 2
  tags = {
    Name = "Diya=ec2-instance"
  }
}

```

Step 3: To Create a VPC on AWS create VPC.tf :



```
resource "aws_vpc" "gfg-vpc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "gfg-subnet" {
  vpc_id     = aws_vpc.gfg-vpc.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "gfg-subnet"
  }
}

resource "aws_internet_gateway" "gfg-gw" {
  vpc_id = aws_vpc.gfg-vpc.id

  tags = {
    Name = "gfg-IG"
  }
}

resource "aws_route_table" "gfg-rt" {
  vpc_id = aws_vpc.gfg-vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gfg-gw.id
  }

    tags = {
```

Step 4: To Create a S3 Bucket create S3.tf :

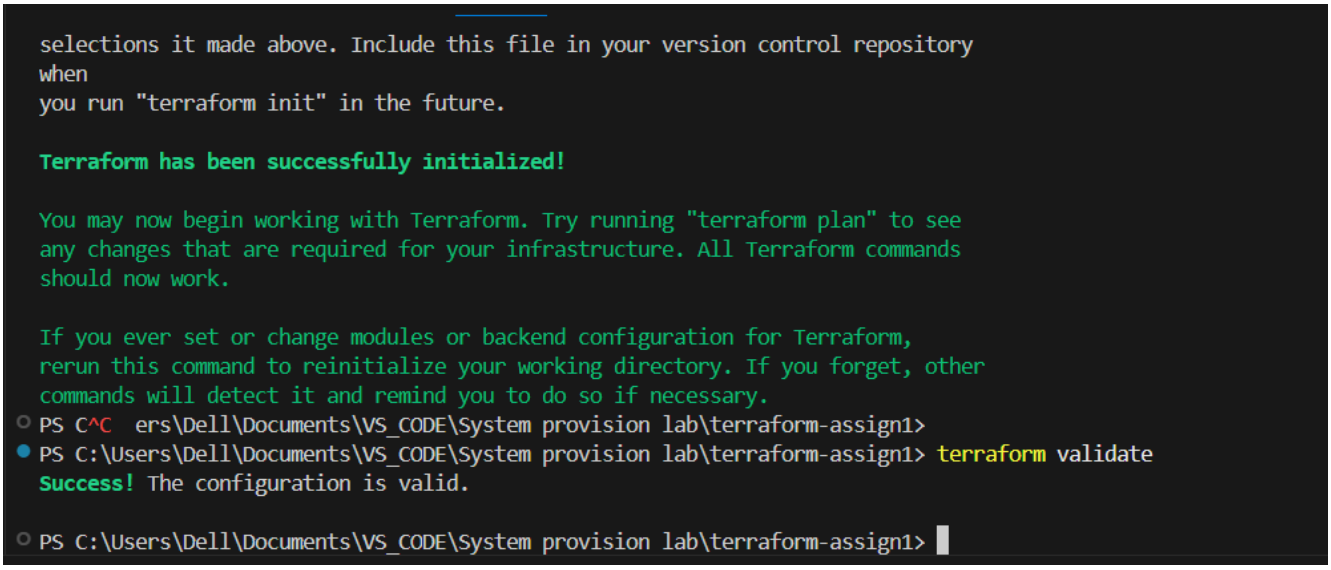


```
resource "aws_s3_bucket" "diya_bucket" {
    bucket = "diya-s3-bucket"
  tags = {
    Name = "diya_s3"
  }
}
```

Step 5: Initialize Terraform

Run the following command to initialize your Terraform working directory:

Terraform init

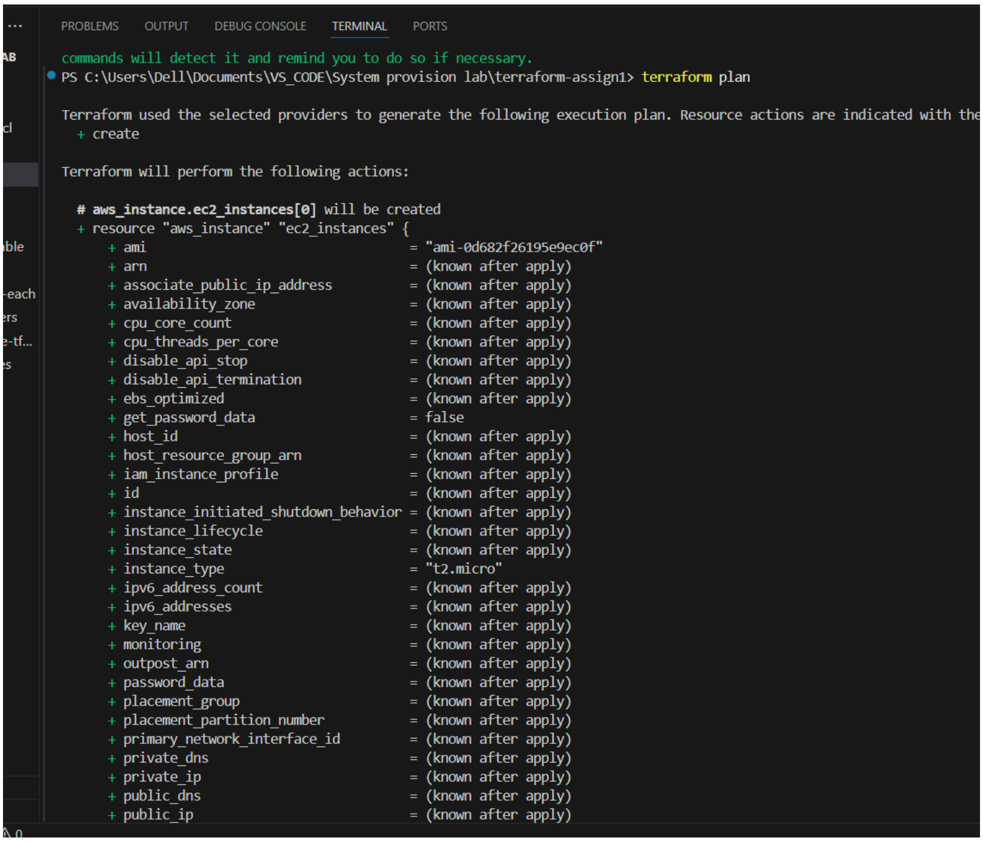


```
selections it made above. Include this file in your version control repository
when
you run "terraform init" in the future.

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.
PS C^C  ers\Dell\Documents\VS_CODE\System provision lab\terraform-assign1>
PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1> terraform validate
Success! The configuration is valid.

PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1>
```

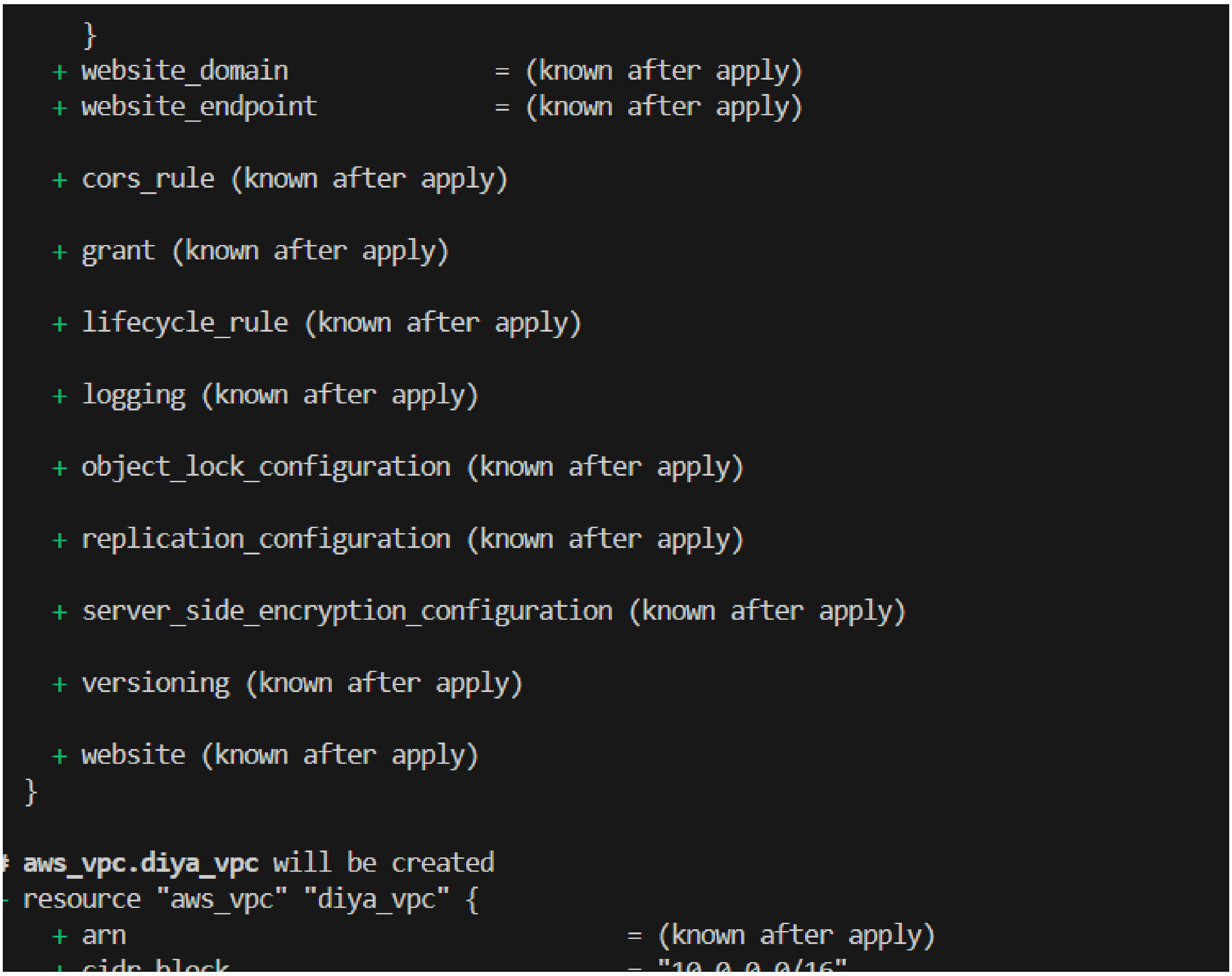
Step 6: Review Plan

Run the following command to see what Terraform will do:

Terraform plan



```
commands will detect it and remind you to do so if necessary.
PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
  + create

Terraform will perform the following actions:

  # aws_instance.ec2_instances[0] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0d682f26195e9ec0f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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)
```

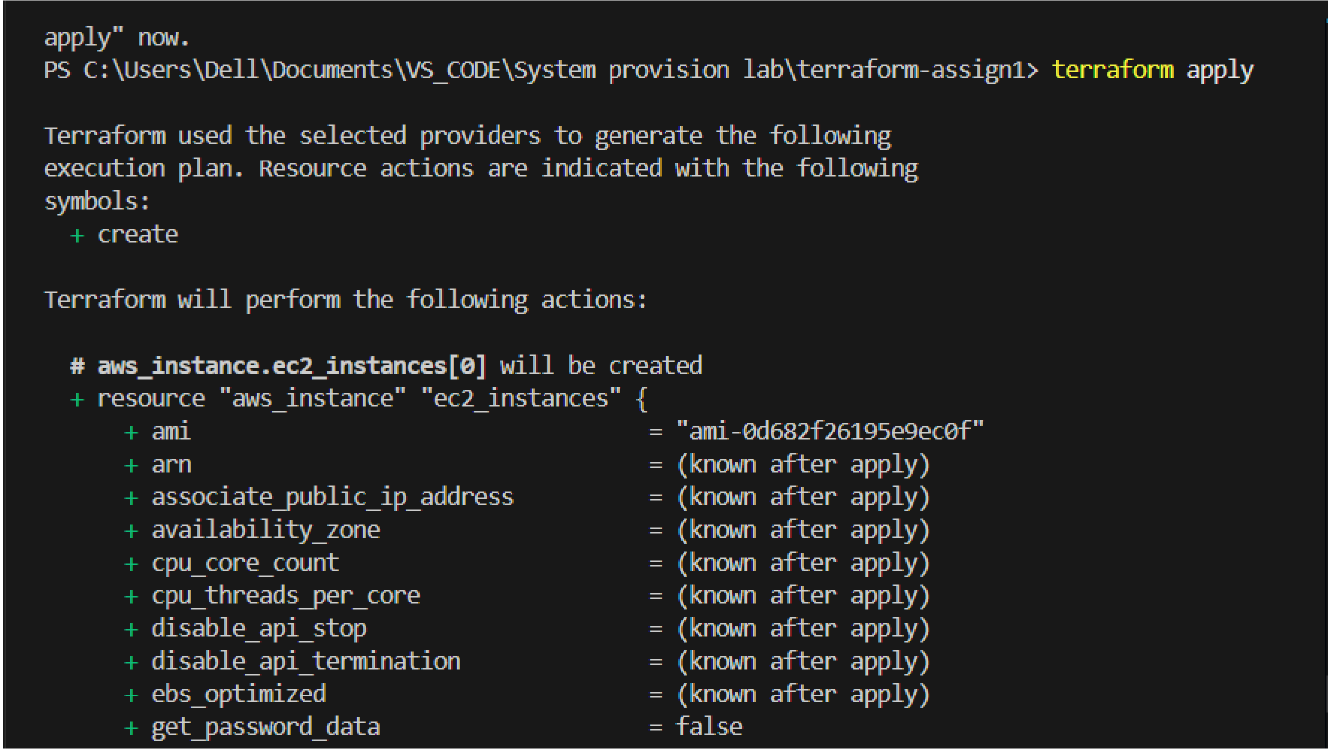


```
      }
    + website_domain                = (known after apply)
    + website_endpoint              = (known after apply)

    + 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)
  }

# aws_vpc.diya_vpc will be created
+ resource "aws_vpc" "diya_vpc" {
    + arn                                  = (known after apply)
    + cidr_block                           = "10.0.0.0/16"
```

Step 7: Apply Changes:

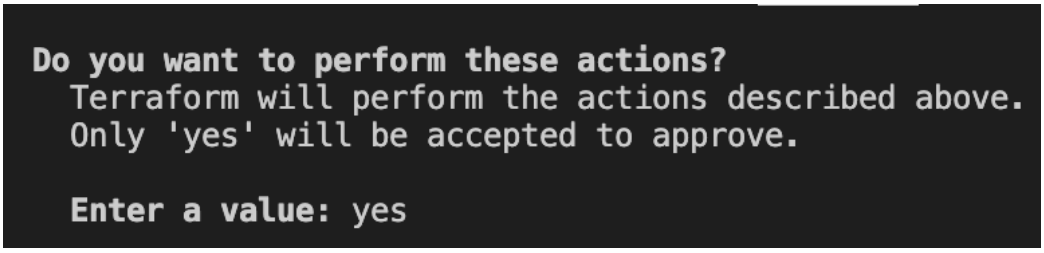
Apply the changes to create the AWS resources:

Terraform apply



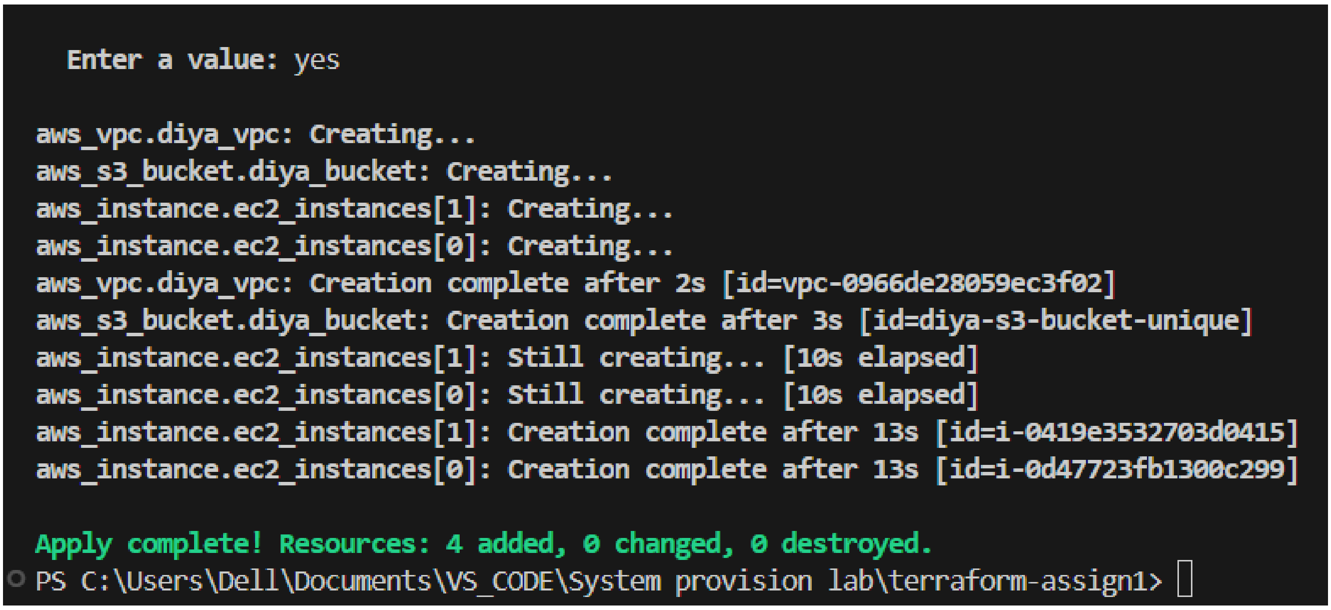
```
apply" now.
PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1> terraform apply

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_instance.ec2_instances[0] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0d682f26195e9ec0f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
```

It will ask for approval before creating, enter "yes" to continue.



```
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
```

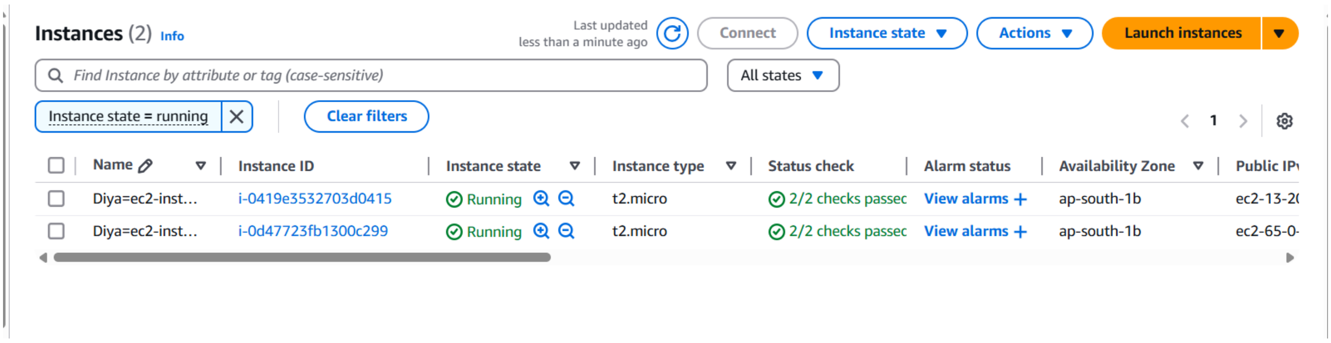
After approval, it will start creating.



```
  Enter a value: yes

aws_vpc.diya_vpc: Creating...
aws_s3_bucket.diya_bucket: Creating...
aws_instance.ec2_instances[1]: Creating...
aws_instance.ec2_instances[0]: Creating...
aws_vpc.diya_vpc: Creation complete after 2s [id=vpc-0966de28059ec3f02]
aws_s3_bucket.diya_bucket: Creation complete after 3s [id=diya-s3-bucket-unique]
aws_instance.ec2_instances[1]: Still creating... [10s elapsed]
aws_instance.ec2_instances[0]: Still creating... [10s elapsed]
aws_instance.ec2_instances[1]: Creation complete after 13s [id=i-0419e3532703d0415]
aws_instance.ec2_instances[0]: Creation complete after 13s [id=i-0d47723fb1300c299]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1>
```

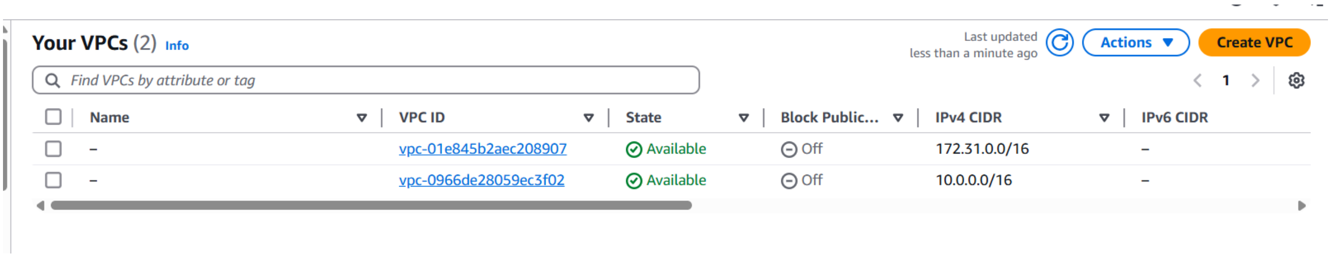
You can verify by logging into the AWS Console,

- Creation of 2 instances



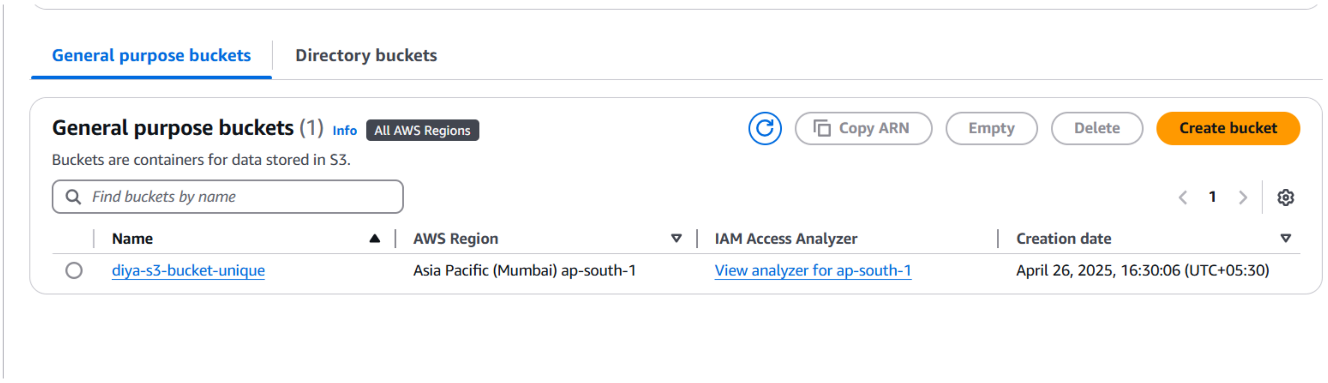
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
□	Diya=ec2-inst…	i-0419e3532703d0415	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(
□	Diya=ec2-inst…	i-0d47723fb1300c299	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-65-0-

- Creation of a VPC



□	Name	VPC ID	State	Block Public…	IPv4 CIDR	IPv6 CIDR
□	–	vpc-01e845b2aec208907	✓ Available	⊖ Off	172.31.0.0/16	–
□	–	vpc-0966de28059ec3f02	✓ Available	⊖ Off	10.0.0.0/16	–

- Creation of S3 Bucket



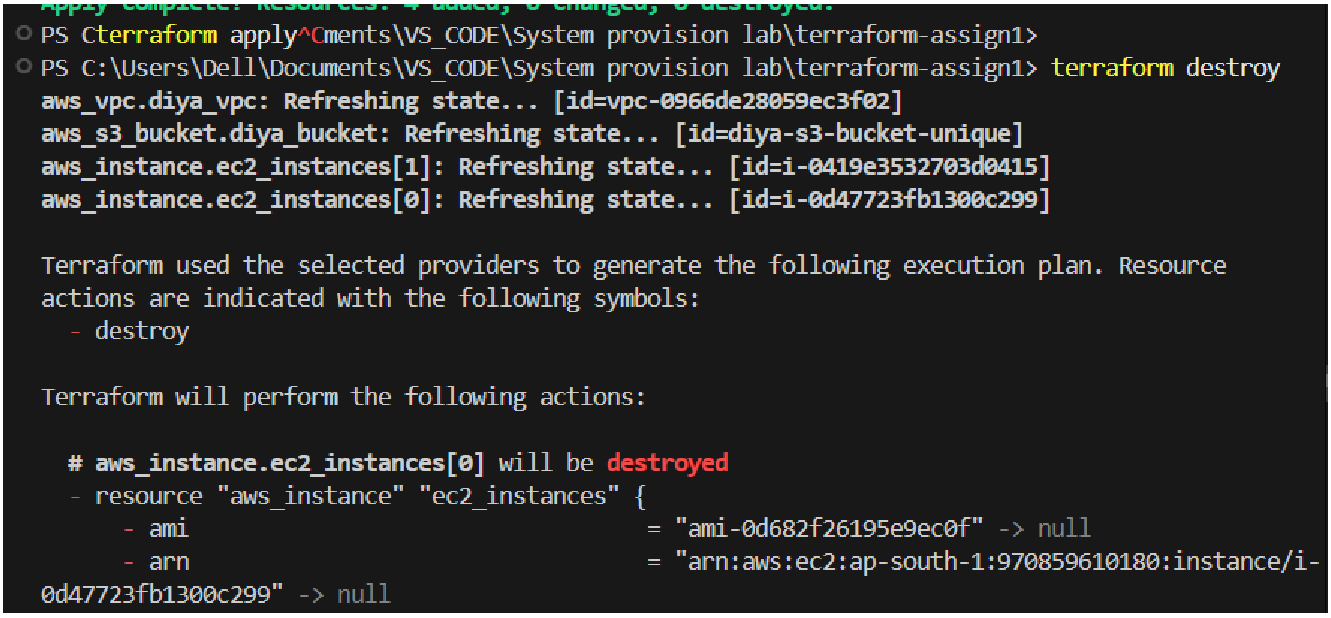
Buckets are containers for data stored in S3.

○	Name ▲	AWS Region	IAM Access Analyzer	Creation date
○	diya-s3-bucket-unique	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 26, 2025, 16:30:06 (UTC+05:30)

Step 8: Cleanup Resources

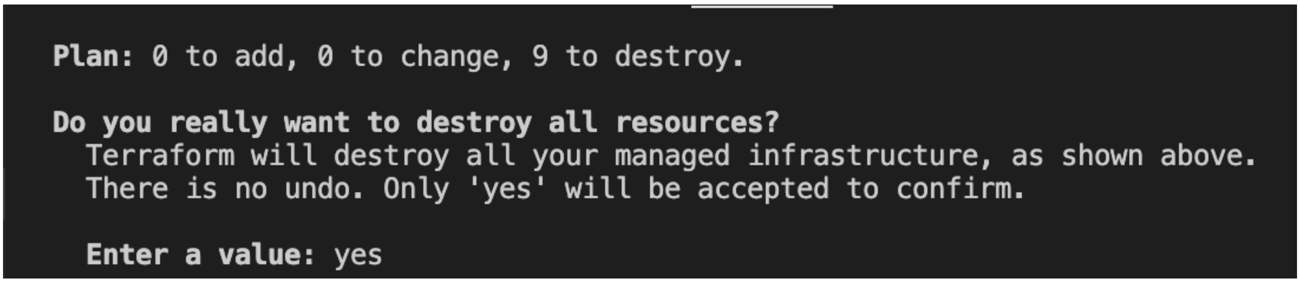
When you are done experimenting, run the following command to destroy the created resources:

Terraform destroy



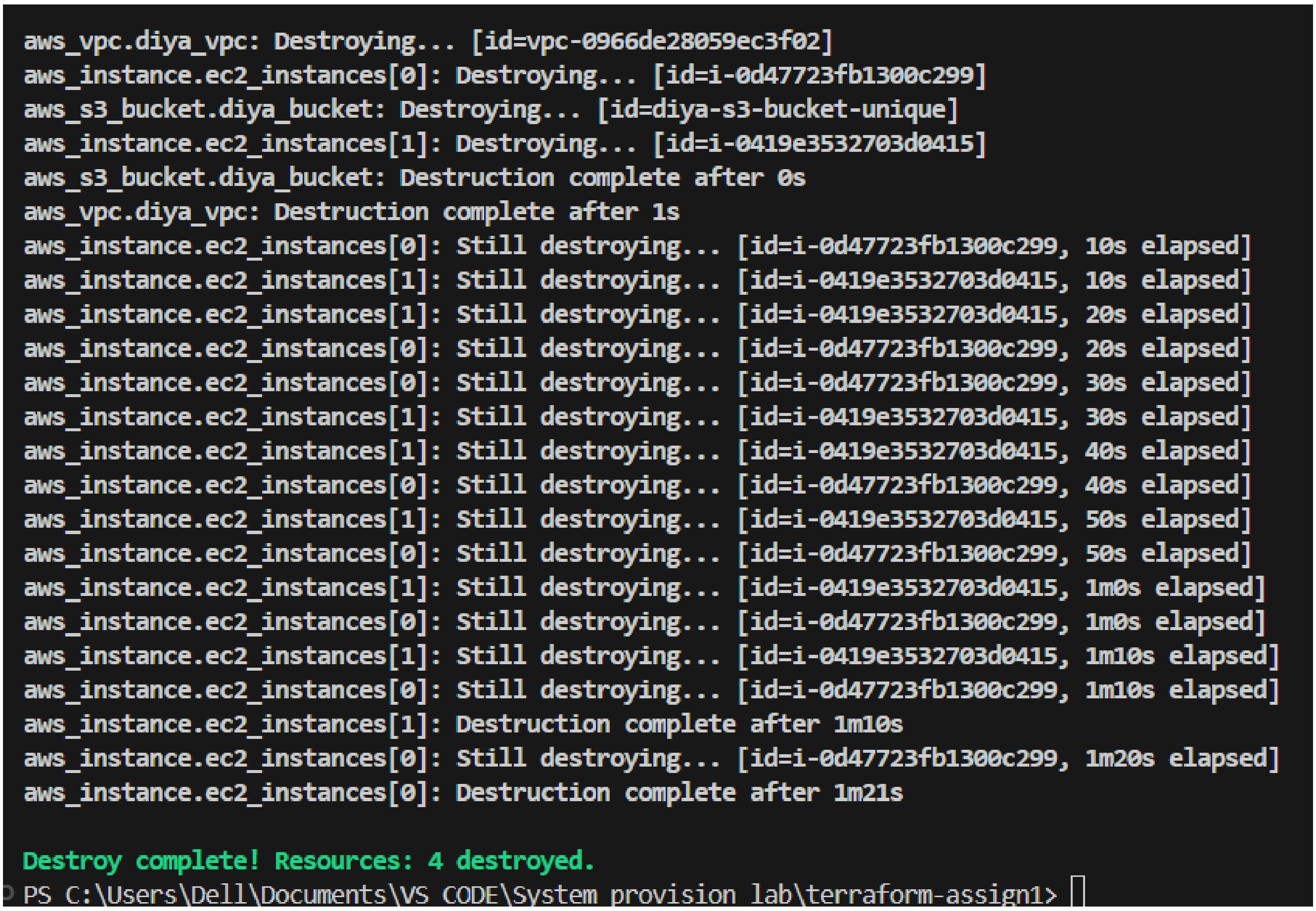
```
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
PS Cterraform apply^Cments\VS_CODE\System provision lab\terraform-assign1>
PS C:\Users\Dell\Documents\VS_CODE\System provision lab\terraform-assign1> terraform destroy
aws_vpc.diya_vpc: Refreshing state... [id=vpc-0966de28059ec3f02]
aws_s3_bucket.diya_bucket: Refreshing state... [id=diya-s3-bucket-unique]
aws_instance.ec2_instances[1]: Refreshing state... [id=i-0419e3532703d0415]
aws_instance.ec2_instances[0]: Refreshing state... [id=i-0d47723fb1300c299]

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.ec2_instances[0] will be destroyed
  - resource "aws_instance" "ec2_instances" {
      - ami                                  = "ami-0d682f26195e9ec0f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:970859610180:instance/i-
0d47723fb1300c299" -> null
```

It will ask for approval before destroying, enter “yes” to continue.



```
Plan: 0 to add, 0 to change, 9 to destroy.

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
```

After approval, it will start destroying.



```
aws_vpc.diya_vpc: Destroying... [id=vpc-0966de28059ec3f02]
aws_instance.ec2_instances[0]: Destroying... [id=i-0d47723fb1300c299]
aws_s3_bucket.diya_bucket: Destroying... [id=diya-s3-bucket-unique]
aws_instance.ec2_instances[1]: Destroying... [id=i-0419e3532703d0415]
aws_s3_bucket.diya_bucket: Destruction complete after 0s
aws_vpc.diya_vpc: Destruction complete after 1s
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 10s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 10s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 20s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 20s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 30s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 30s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 40s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 40s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 50s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 50s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 1m0s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 1m0s elapsed]
aws_instance.ec2_instances[1]: Still destroying... [id=i-0419e3532703d0415, 1m10s elapsed]
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 1m10s elapsed]
aws_instance.ec2_instances[1]: Destruction complete after 1m10s
aws_instance.ec2_instances[0]: Still destroying... [id=i-0d47723fb1300c299, 1m20s elapsed]
aws_instance.ec2_instances[0]: Destruction complete after 1m21s

Destroy complete! Resources: 4 destroyed.
PS C:\Users\Dell\Documents\VS CODE\System provision lab\terraform-assign1>
```