



System Provisioning & Configuration Management

ASSIGNMENT-1

SUBMITTED BY:

NAME: Anurag Negi

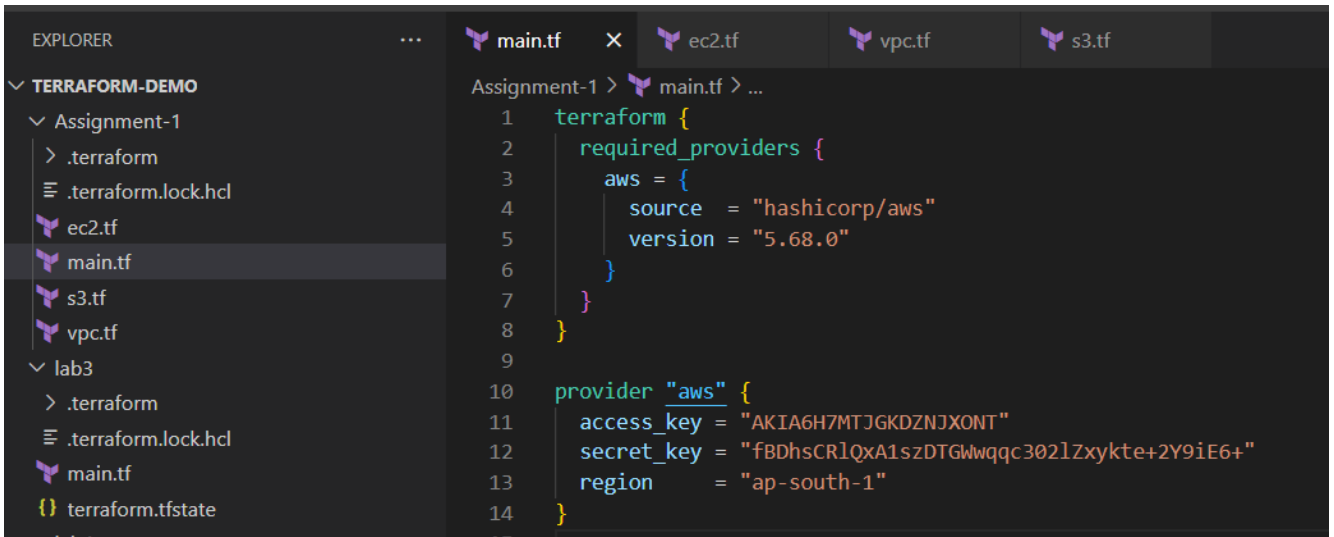
SAP ID: 500110161

BATCH: 2 (DevOps)

ROLL NO: R2142221383

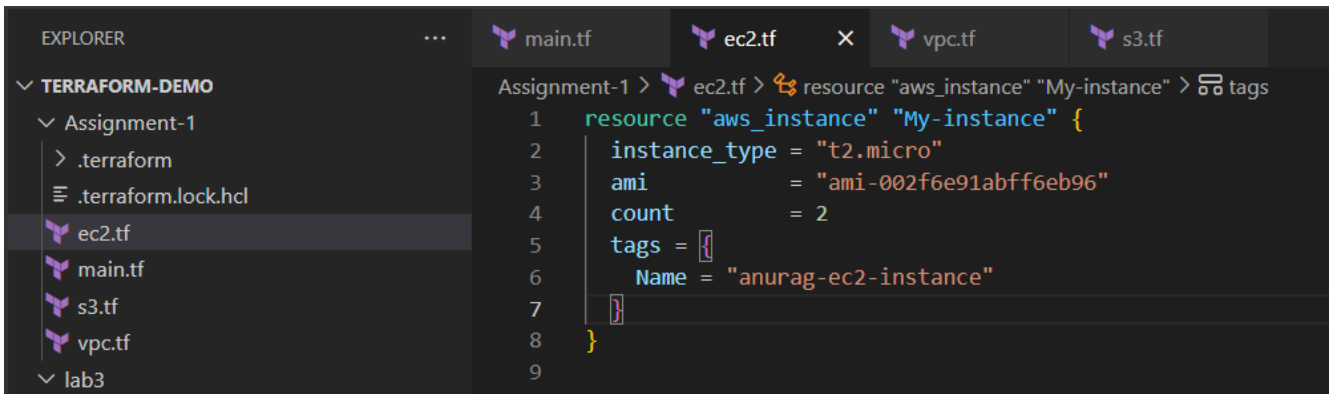
Write Terraform script to do perform following tasks on AWS cloud Platform

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



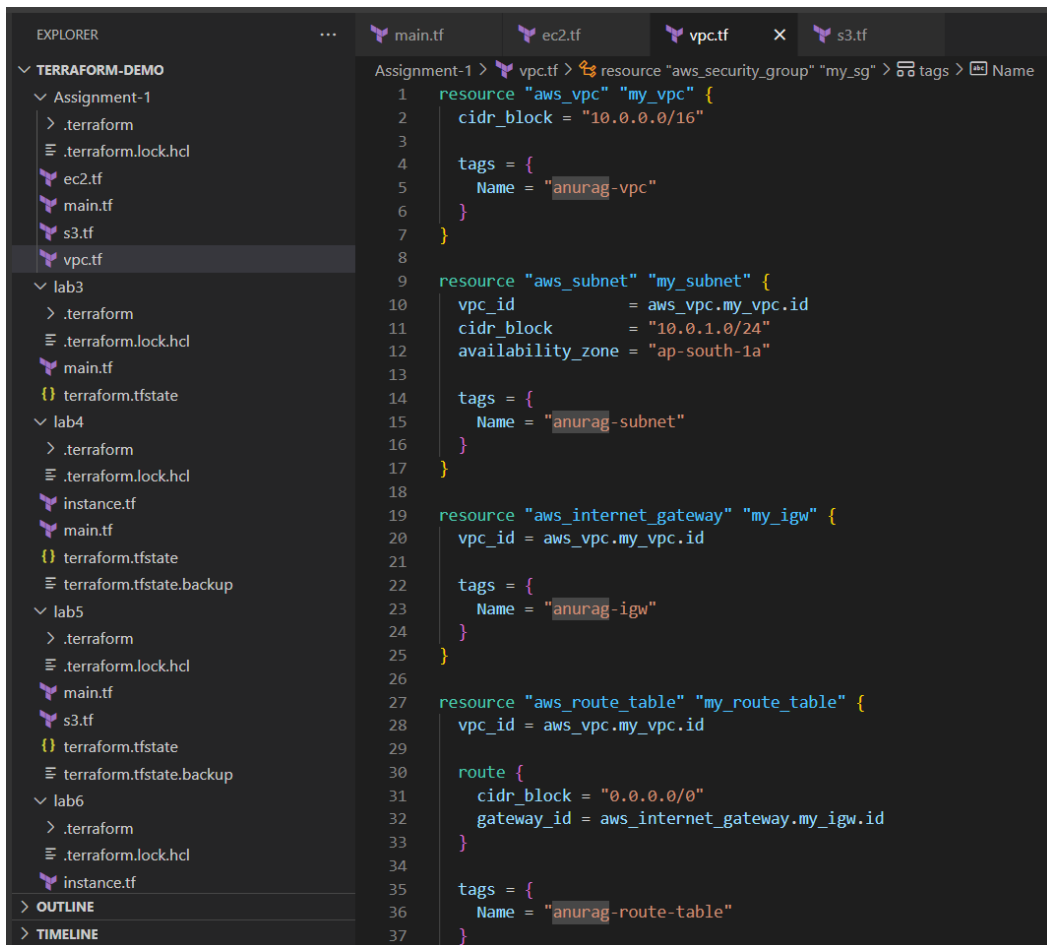
```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.68.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   access_key = "AKIA6H7MTJGKDZNXONT"
12   secret_key = "fBDhsCRlQxA1szDTGWwqqc302lZxykte+2Y9iE6+"
13   region     = "ap-south-1"
14 }
```

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



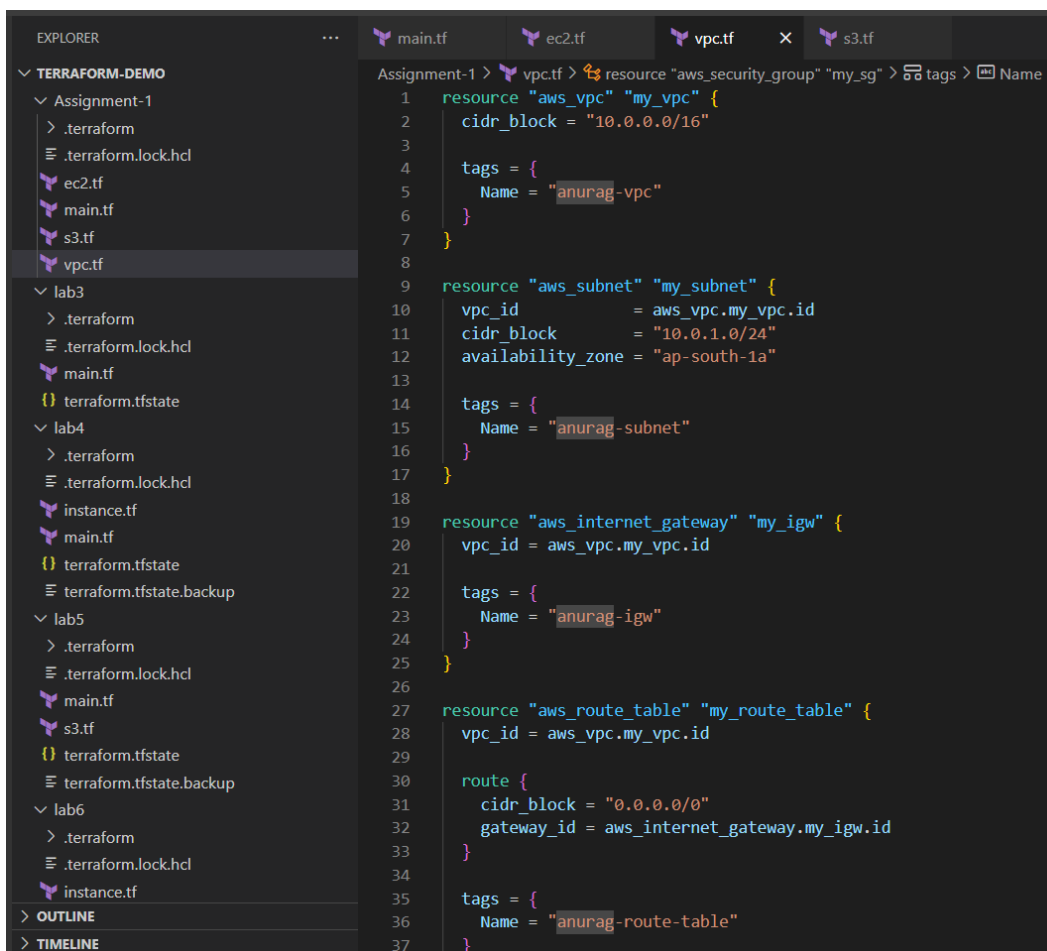
```
1 resource "aws_instance" "My-instance" {
2   instance_type = "t2.micro"
3   ami           = "ami-002f6e91abff6eb96"
4   count         = 2
5   tags = {
6     Name = "anurag-ec2-instance"
7   }
8 }
9
```

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



The screenshot shows the VS Code interface with the Terraform Explorer on the left and the main editor on the right. The Explorer shows a project structure for 'TERRAFORM-DEMO' with several labs and their associated files. The main editor displays the 'vpc.tf' file, which contains Terraform code for creating an AWS VPC, a subnet, an internet gateway, and a route table. The code is as follows:

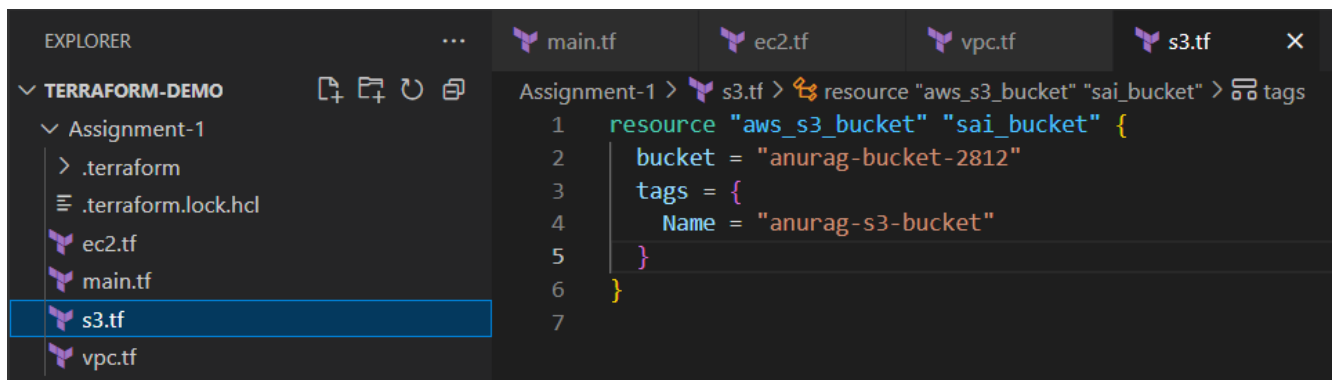
```
1 resource "aws_vpc" "my_vpc" {
2   cidr_block = "10.0.0.0/16"
3
4   tags = {
5     Name = "anurag-vpc"
6   }
7 }
8
9 resource "aws_subnet" "my_subnet" {
10  vpc_id      = aws_vpc.my_vpc.id
11  cidr_block   = "10.0.1.0/24"
12  availability_zone = "ap-south-1a"
13
14  tags = {
15    Name = "anurag-subnet"
16  }
17 }
18
19 resource "aws_internet_gateway" "my_igw" {
20  vpc_id = aws_vpc.my_vpc.id
21
22  tags = {
23    Name = "anurag-igw"
24  }
25 }
26
27 resource "aws_route_table" "my_route_table" {
28  vpc_id = aws_vpc.my_vpc.id
29
30  route {
31    cidr_block = "0.0.0.0/0"
32    gateway_id = aws_internet_gateway.my_igw.id
33  }
34
35  tags = {
36    Name = "anurag-route-table"
37  }
```



This screenshot is identical to the one above, showing the same VS Code interface and Terraform configuration for creating an AWS VPC, subnet, internet gateway, and route table. The code is repeated for clarity:

```
1 resource "aws_vpc" "my_vpc" {
2   cidr_block = "10.0.0.0/16"
3
4   tags = {
5     Name = "anurag-vpc"
6   }
7 }
8
9 resource "aws_subnet" "my_subnet" {
10  vpc_id      = aws_vpc.my_vpc.id
11  cidr_block   = "10.0.1.0/24"
12  availability_zone = "ap-south-1a"
13
14  tags = {
15    Name = "anurag-subnet"
16  }
17 }
18
19 resource "aws_internet_gateway" "my_igw" {
20  vpc_id = aws_vpc.my_vpc.id
21
22  tags = {
23    Name = "anurag-igw"
24  }
25 }
26
27 resource "aws_route_table" "my_route_table" {
28  vpc_id = aws_vpc.my_vpc.id
29
30  route {
31    cidr_block = "0.0.0.0/0"
32    gateway_id = aws_internet_gateway.my_igw.id
33  }
34
35  tags = {
36    Name = "anurag-route-table"
37  }
```

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

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named 'TERRAFORM-DEMO' with a subdirectory 'Assignment-1'. Inside 'Assignment-1', there are files: '.terraform', '.terraform.lock.hcl', 'ec2.tf', 'main.tf', 's3.tf' (which is selected and highlighted in blue), and 'vpc.tf'. The main editor area shows the content of 's3.tf'. The breadcrumb at the top of the editor reads 'Assignment-1 > s3.tf > resource "aws_s3_bucket" "sai_bucket" > tags'. The code in the editor is:

```
1 resource "aws_s3_bucket" "sai_bucket" {
2     bucket = "anurag-bucket-2812"
3     tags = {
4         Name = "anurag-s3-bucket"
5     }
6 }
7
```

Step 5: Initialize Terraform

Run the following command to initialize your Terraform working directory:

Terraform init

Terraform has been successfully initialized!

Terraform has been successfully initialized!

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 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.

Step 6: Review Plan

Run the following command to see what Terraform will do:

Terraform plan

```
PS C:\Github_Repositorios\Terraform-Demo\Assignment-1> 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-instance[0] will be created
+ resource "aws_instance" "My-instance" {
  + ami                      = "ami-002f6e91abff6eb96"
  + 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)
```

```
# aws_vpc.my_vpc will be created
+ resource "aws_vpc" "my_vpc" {
  + arn                      = (known after apply)
  + cidr_block               = "10.0.0.0/16"
  + default_network_acl_id   = (known after apply)
  + default_route_table_id   = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id          = (known after apply)
  + enable_dns_hostnames     = (known after apply)
  + enable_dns_support       = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                       = (known after apply)
  + instance_tenancy         = "default"
  + ipv6_association_id      = (known after apply)
  + ipv6_cidr_block          = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id      = (known after apply)
  + owner_id                 = (known after apply)
  + tags                     = {
    + "Name" = "anurag-vpc"
  }
  + tags_all                 = {
    + "Name" = "anurag-vpc"
  }
}
```

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

Step 7: Apply Changes:

Apply the changes to create the AWS resources:

Terraform apply

```
PS C:\Github_Repositoires\Terraform-Demo\Assignment-1> terraform apply

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

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.

```
aws_vpc.my_vpc: Creating...
aws_instance.My-instance[1]: Creating...
aws_instance.My-instance[0]: Creating...
aws_s3_bucket.anurag_bucket: Creating...
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-0d7f165e9c5fef79c]
aws_internet_gateway.my_igw: Creating...
aws_subnet.my_subnet: Creating...
aws_security_group.my_sg: Creating...
aws_s3_bucket.anurag_bucket: Creation complete after 2s [id=anurag-bucket-2812]
aws_internet_gateway.my_igw: Creation complete after 1s [id=igw-0d2ce55bd8e6249c6]
aws_route_table.my_route_table: Creating...
aws_subnet.my_subnet: Creation complete after 1s [id=subnet-03f097b216ba3db3b]
aws_route_table.my_route_table: Creation complete after 1s [id=rtb-0e6eabcf99b1c6c33]
aws_route_table_association.my_route_assoc: Creating...
aws_route_table_association.my_route_assoc: Creation complete after 0s [id=rtbassoc-08bb4914a78ce95c9]
aws_security_group.my_sg: Creation complete after 3s [id=sg-0a822a00307c64dab]
aws_instance.My-instance[1]: Still creating... [10s elapsed]
aws_instance.My-instance[0]: Still creating... [10s elapsed]
aws_instance.My-instance[1]: Creation complete after 13s [id=i-09a9a1571c537fd08]
aws_instance.My-instance[0]: Creation complete after 13s [id=i-06eef57014a7dd8ed]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
PS C:\Github_Repositoires\Terraform-Demo\Assignment-1> 
```

You can verify by logging into the AWS Console,

- Creation of ec2 instances

Instances (2) [Info](#) Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#)

[Instance state = running](#) [Clear filters](#) < 1 > [Settings](#)

<input type="checkbox"/>	Name ↗	Instance ID	Instance state ↕	Instance type ↕	Status check	Alarm status	Availability Zone ↕
<input type="checkbox"/>	anurag-ec2-instance	i-06eef57014a7dd8ed	Running 🔍 🔍	t2.micro	🕒 Initializing	View alarms +	ap-south-1b
<input type="checkbox"/>	anurag-ec2-instance	i-09a9a1571c537fd08	Running 🔍 🔍	t2.micro	🕒 Initializing	View alarms +	ap-south-1b

- Creation of a VPC

Your VPCs (2) [Info](#) Last updated less than a minute ago [Actions](#) [Create VPC](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	VPC ID	State	Block Public... ↕	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-0ff99bd2dbfcaa1f8	Available	Off	172.31.0.0/16	-
<input type="checkbox"/>	anurag-vpc	vpc-0d7f165e9c5fef79c	Available	Off	10.0.0.0/16	-

- Creation of S3 Bucket

aws [Alt+S] Asia Pacific (Mumbai) Account ID: 9792-1186-4468 anurag

Amazon S3 🔍 [Account snapshot - updated every 24 hours](#) [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (1) [Info](#) [All AWS Regions](#) [Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

< 1 > [Settings](#)

<input type="radio"/>	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	anurag-bucket-2812	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 11, 2025, 22:44:03 (UTC+05:30)

Step 8: Cleanup Resources

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

Terraform destroy

```
PS C:\Github_Repositories\Terraform-Demo\Assignment-1> terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0d7f165e9c5fef79c]
aws_s3_bucket.anurag_bucket: Refreshing state... [id=anurag-bucket-2812]
aws_instance.My-instance[1]: Refreshing state... [id=i-09a9a1571c537fd08]
aws_instance.My-instance[0]: Refreshing state... [id=i-06eef57014a7dd8ed]
aws_internet_gateway.my_igw: Refreshing state... [id=igw-0d2ce55bd8e6249c6]
aws_subnet.my_subnet: Refreshing state... [id=subnet-03f097b216ba3db3b]
aws_security_group.my_sg: Refreshing state... [id=sg-0a822a00307c64dab]
aws_route_table.my_route_table: Refreshing state... [id=rtb-0e6eabcf99b1c6c33]
aws_route_table_association.my_route_assoc: Refreshing state... [id=rtbassoc-08bb4914a78ce95c9]

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

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.