Infrastructure Automation using Terraform

Create VPC

accenture

Table of Contents

This Activity demonstrates the creation of VPC in AWS using Terraform.

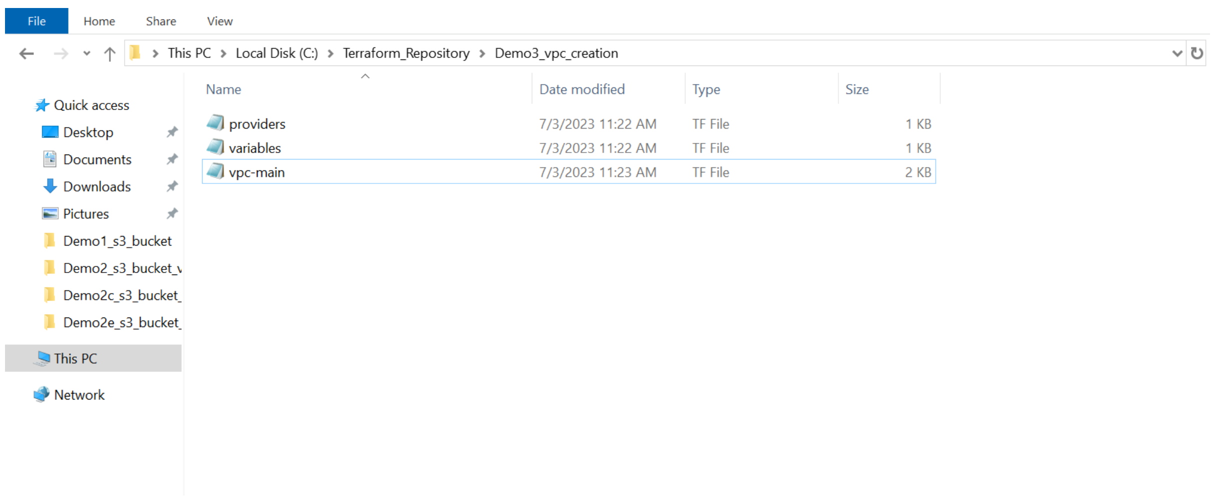
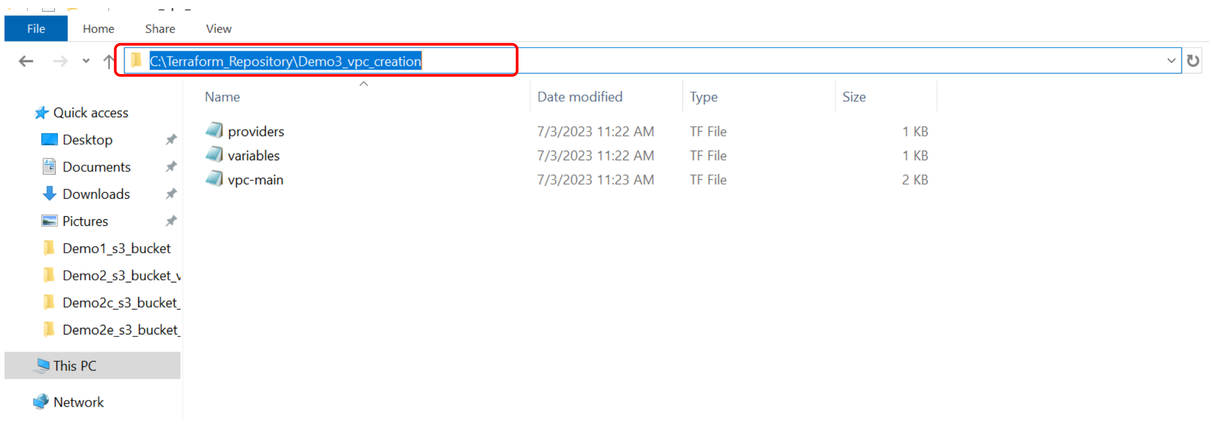
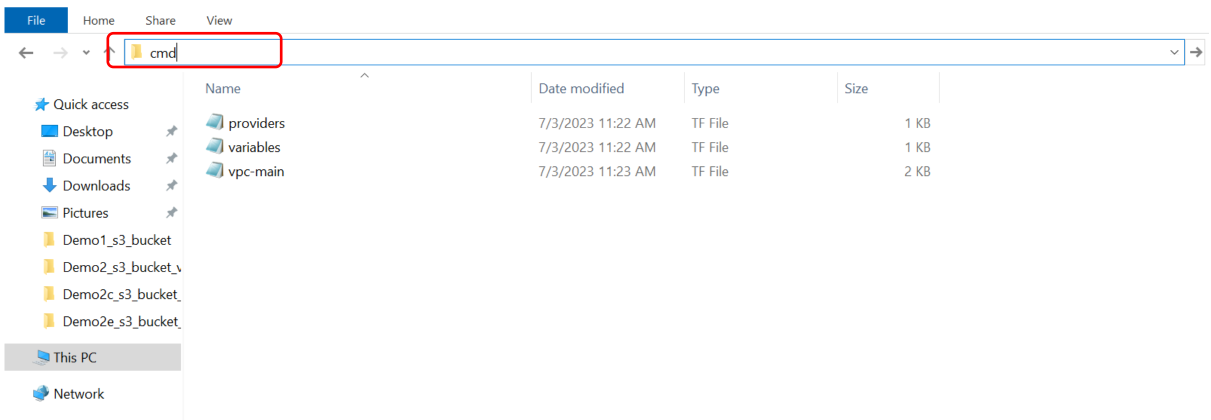
Prerequisite:

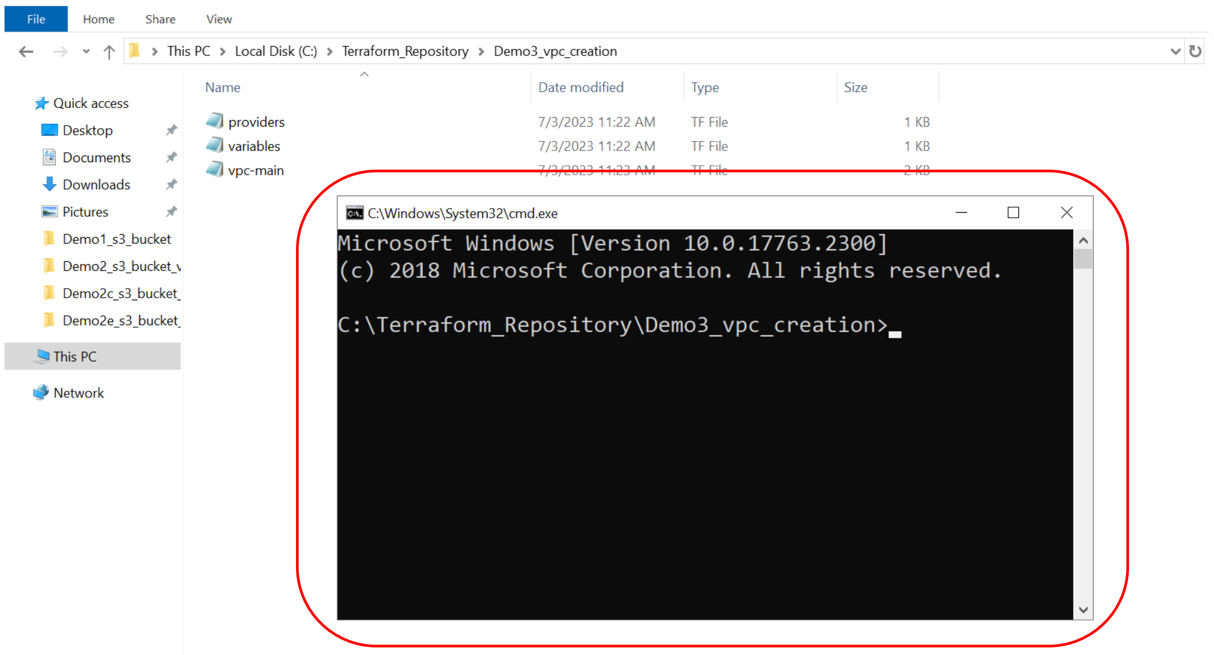
1) Download the zip file shared by the trainer and extract it.

Walkthrough:

1. Initializing Terraform Directory
2. Creating VPC
3. Destroying VPC

Part 1: Initializing Terraform Directory

1	Open the extracted folder and navigate to ".tf" files.  This PC > Local Disk (C:) > Terraform_Repository > Demo3_vpc_creation; Name / Date modified / Type / Size: providers 7/3/2023 11:22 AM TF File 1 KB; variables 7/3/2023 11:22 AM TF File 1 KB; vpc-main 7/3/2023 11:23 AM TF File 2 KB
2	Click on the Address bar and type cmd. Press Enter (It will open a command prompt from that location).  C:\Terraform_Repository\Demo3_vpc_creation  cmd

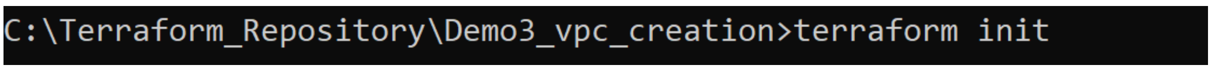


```
Name          Date modified        Type      Size
providers     7/3/2023 11:22 AM    TF File   1 KB
variables     7/3/2023 11:22 AM    TF File   1 KB
vpc-main      7/3/2023 11:23 AM    TF File   2 KB
```

```
Microsoft Windows [Version 10.0.17763.2300]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Terraform_Repository\Demo3_vpc_creation>
```

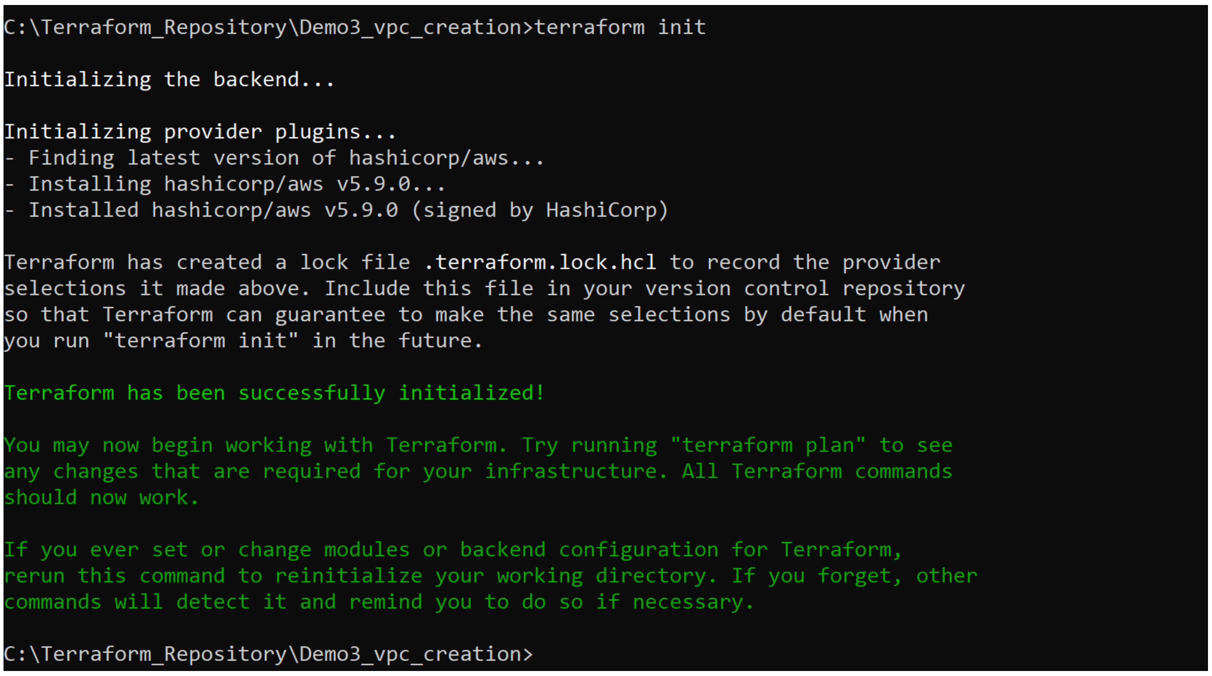
3 Execute below command to initialize the current directory as Terraform directory which enables us to run terraform commands to manage Infrastructure.

Command :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform init
```

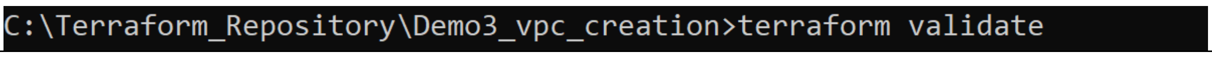
Result :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.9.0...
- Installed hashicorp/aws v5.9.0 (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:\Terraform_Repository\Demo3_vpc_creation>
```

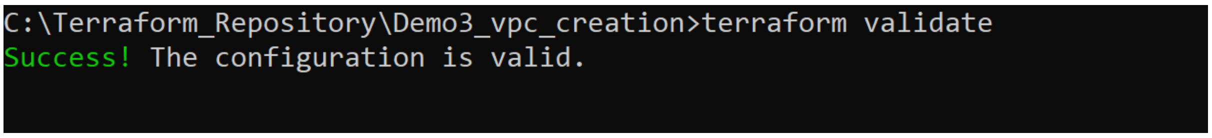
4 Next execute below command to validate syntax and configuration of terraform configuration files. If everything is proper, it will return a success message otherwise it will display the errors.

Command :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform validate
```

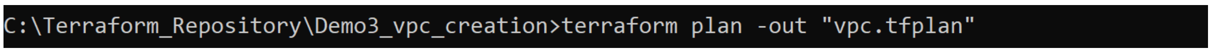
Result :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform validate
Success! The configuration is valid.
```

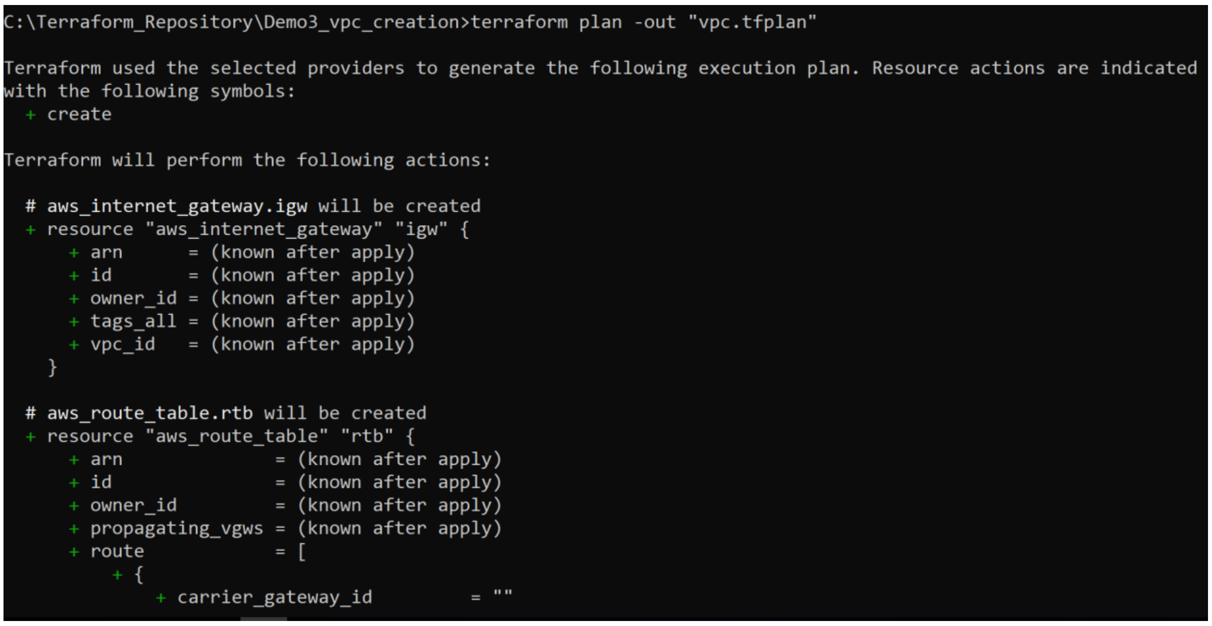
5 Next run below command and observe the output. The output contains information depicting all the changes which will happen in the AWS cloud. It is like dry-run to ensure whatever we are trying to do using terraform commands is what we want.

Command :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform plan -out "vpc.tfplan"
```

Result :

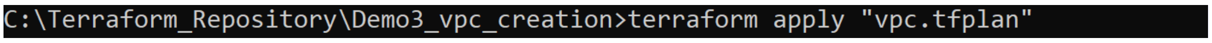


```
C:\Terraform_Repository\Demo3_vpc_creation>terraform plan -out "vpc.tfplan"

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_internet_gateway.igw will be created
  + resource "aws_internet_gateway" "igw" {
      + arn      = (known after apply)
      + id       = (known after apply)
      + owner_id = (known after apply)
      + tags_all = (known after apply)
      + vpc_id   = (known after apply)
    }

  # aws_route_table.rtb will be created
  + resource "aws_route_table" "rtb" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = [
          + {
              + carrier_gateway_id         = ""
```

Part 2: Creating VPC

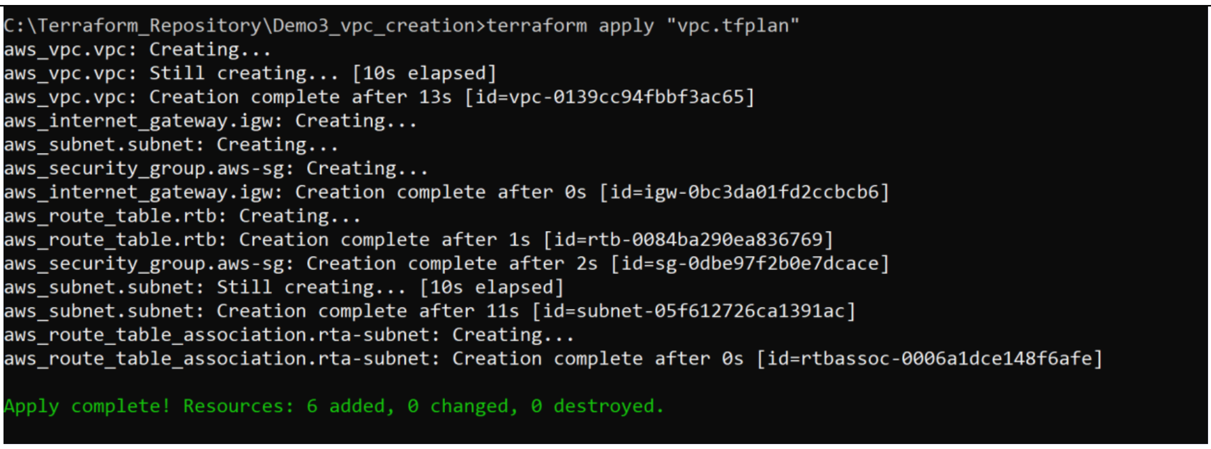
1 For creating VPC , execute below command and observe the actions performed by the command.

Command :



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform apply "vpc.tfplan"
```

Result :

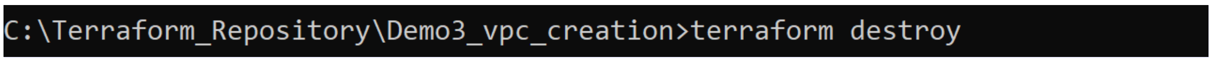


```
C:\Terraform_Repository\Demo3_vpc_creation>terraform apply "vpc.tfplan"
aws_vpc.vpc: Creating...
aws_vpc.vpc: Still creating... [10s elapsed]
aws_vpc.vpc: Creation complete after 13s [id=vpc-0139cc94fbbf3ac65]
aws_internet_gateway.igw: Creating...
aws_subnet.subnet: Creating...
aws_security_group.aws-sg: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0bc3da01fd2ccbcb6]
aws_route_table.rtb: Creating...
aws_route_table.rtb: Creation complete after 1s [id=rtb-0084ba290ea836769]
aws_security_group.aws-sg: Creation complete after 2s [id=sg-0dbe97f2b0e7dcace]
aws_subnet.subnet: Still creating... [10s elapsed]
aws_subnet.subnet: Creation complete after 11s [id=subnet-05f612726ca1391ac]
aws_route_table_association.rta-subnet: Creating...
aws_route_table_association.rta-subnet: Creation complete after 0s [id=rtbassoc-0006a1dce148f6afe]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
```

Part 3: Destroying VPC

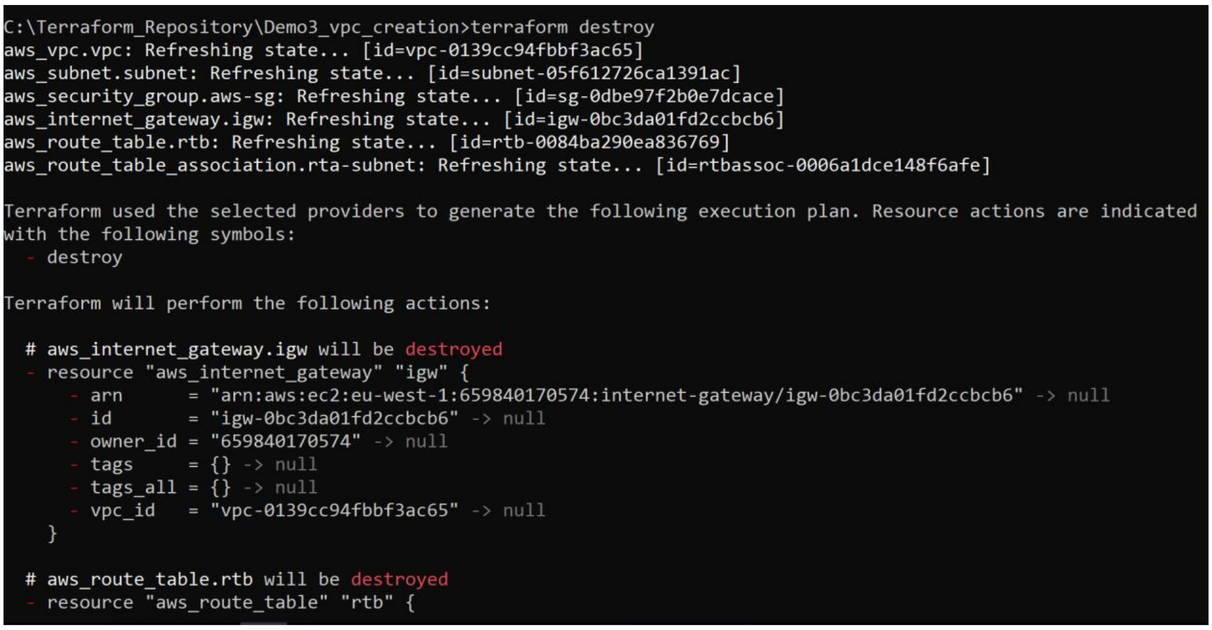
1 Execute below command to destroy the VPC which we have created in previous step. After you execute below command, it will show you what changes will be done and before doing those changes it will ask for your approval. So, if you want to proceed with destroying VPC, provide "yes".

Command:

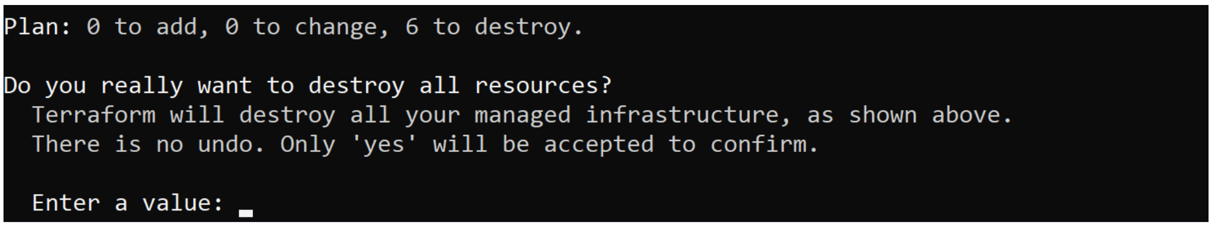


```
C:\Terraform_Repository\Demo3_vpc_creation>terraform destroy
```

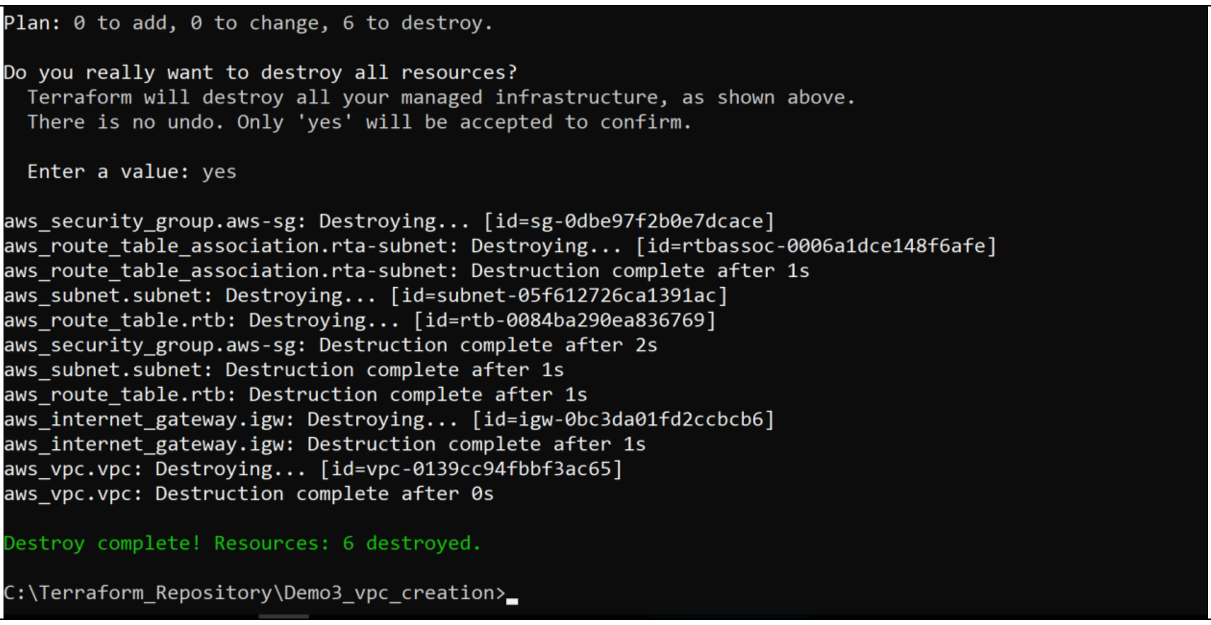
Result:



```
C:\Terraform_Repository\Demo3_vpc_creation>terraform destroy
aws_vpc.vpc: Refreshing state... [id=vpc-0139cc94fbbf3ac65]
aws_subnet.subnet: Refreshing state... [id=subnet-05f612726ca1391ac]
aws_security_group.aws-sg: Refreshing state... [id=sg-0dbe97f2b0e7dcace]
aws_internet_gateway.igw: Refreshing state... [id=igw-0bc3da01fd2ccbcb6]
aws_route_table.rtb: Refreshing state... [id=rtb-0084ba290ea836769]
aws_route_table_association.rta-subnet: Refreshing state... [id=rtbassoc-0006a1dce148f6afe]

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_internet_gateway.igw will be destroyed
  - resource "aws_internet_gateway" "igw" {
      - arn      = "arn:aws:ec2:eu-west-1:659840170574:internet-gateway/igw-0bc3da01fd2ccbcb6" -> null
      - id       = "igw-0bc3da01fd2ccbcb6" -> null
      - owner_id = "659840170574" -> null
      - tags     = {} -> null
      - tags_all = {} -> null
      - vpc_id   = "vpc-0139cc94fbbf3ac65" -> null
    }

  # aws_route_table.rtb will be destroyed
  - resource "aws_route_table" "rtb" {
```



```
Plan: 0 to add, 0 to change, 6 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:
```



```
Plan: 0 to add, 0 to change, 6 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

aws_security_group.aws-sg: Destroying... [id=sg-0dbe97f2b0e7dcace]
aws_route_table_association.rta-subnet: Destroying... [id=rtbassoc-0006a1dce148f6afe]
aws_route_table_association.rta-subnet: Destruction complete after 1s
aws_subnet.subnet: Destroying... [id=subnet-05f612726ca1391ac]
aws_route_table.rtb: Destroying... [id=rtb-0084ba290ea836769]
aws_security_group.aws-sg: Destruction complete after 2s
aws_subnet.subnet: Destruction complete after 1s
aws_route_table.rtb: Destruction complete after 1s
aws_internet_gateway.igw: Destroying... [id=igw-0bc3da01fd2ccbcb6]
aws_internet_gateway.igw: Destruction complete after 1s
aws_vpc.vpc: Destroying... [id=vpc-0139cc94fbbf3ac65]
aws_vpc.vpc: Destruction complete after 0s

Destroy complete! Resources: 6 destroyed.

C:\Terraform_Repository\Demo3_vpc_creation>
```