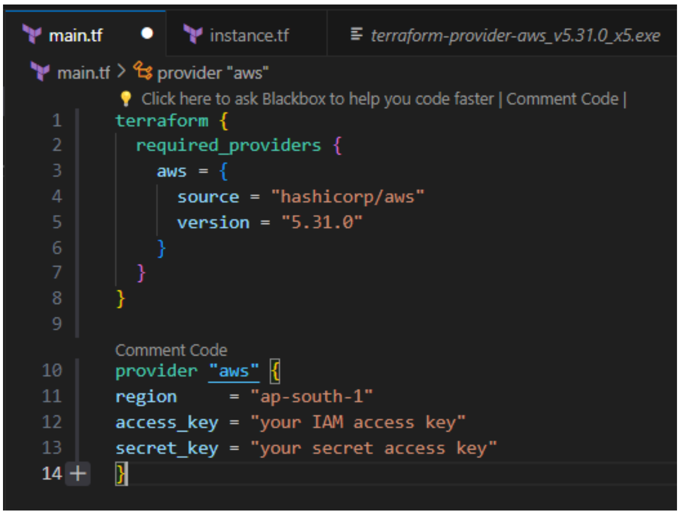
EXPERIMENT – 4

Name: - Shashwat. Dnyaneshwar Kamdi
Batch – 2 [DevOps Non-Hons]
SAP ID- 500092140
Subject – System Provisioning and Configuration Management Lab

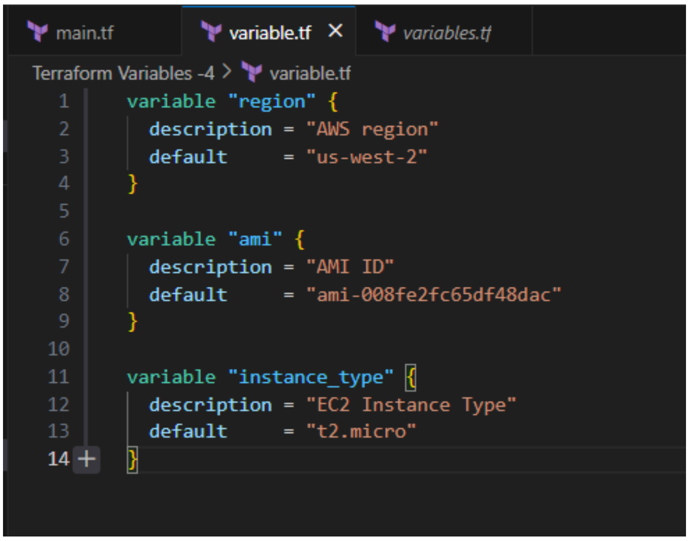
Aim: Terraform Variables.

1] Create a Terraform Configuration File (main.tf)


main.tf
instance.tf
terraform-provider-aws_v5.31.0_x5.exe
main.tf > provider "aws"
Click here to ask Blackbox to help you code faster | Comment Code |
Comment Code

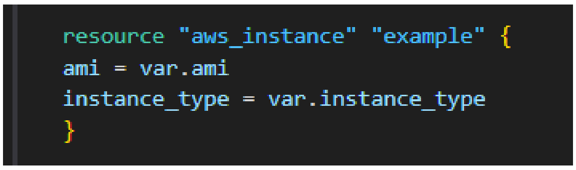
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

provider "aws" {
region      = "ap-south-1"
access_key = "your IAM access key"
secret_key = "your secret access key"
}
```

2] Create new file name as “variables.tf”



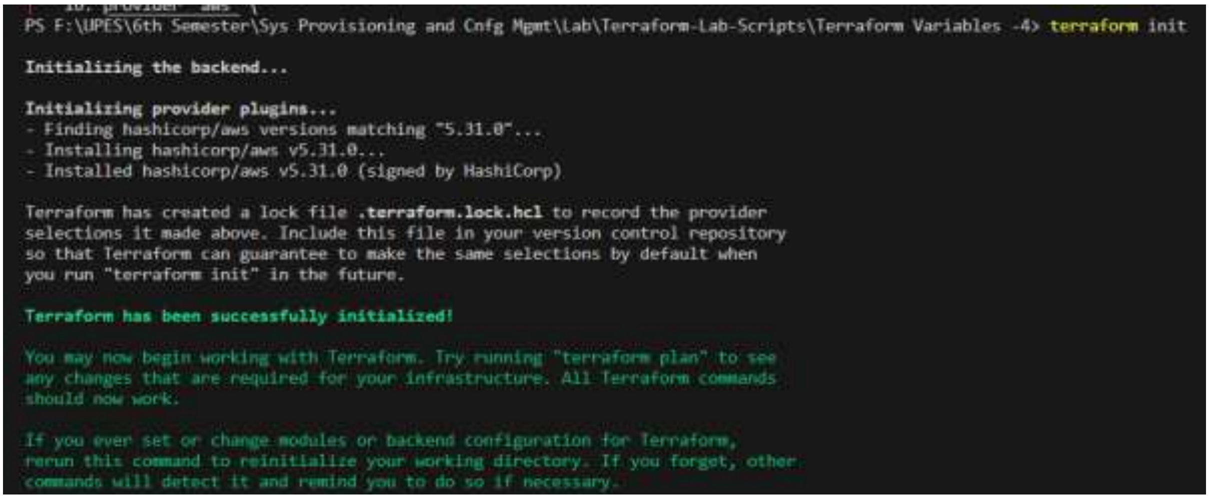
```
variable "region" {
  description = "AWS region"
  default     = "us-west-2"
}

variable "ami" {
  description = "AMI ID"
  default     = "ami-008fe2fc65df48dac"
}

variable "instance_type" {
  description = "EC2 Instance Type"
  default     = "t2.micro"
}
```

3] Use Variables in “main.tf” and update main.tf file.



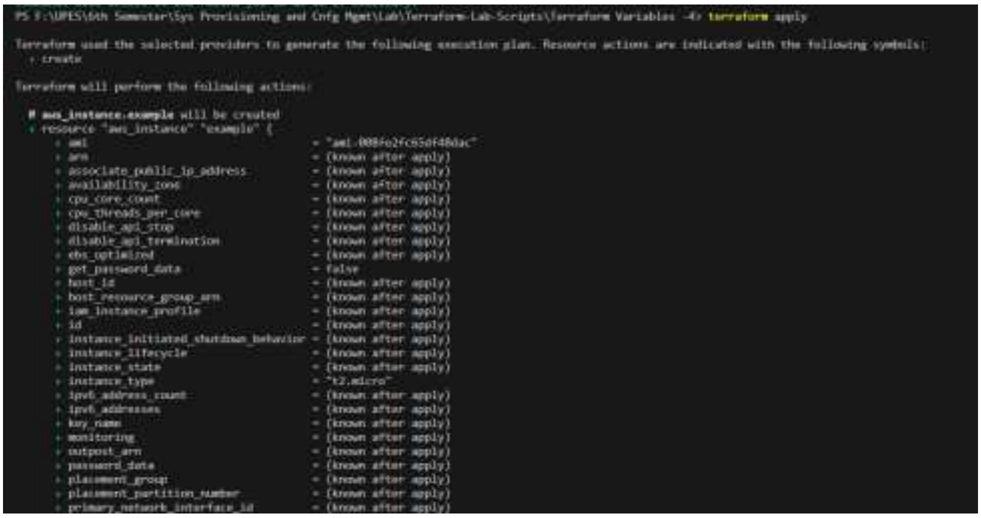
```
resource "aws_instance" "example" {
ami = var.ami
instance_type = var.instance_type
}
```

4] Initialize Terraform using command “terraform init”

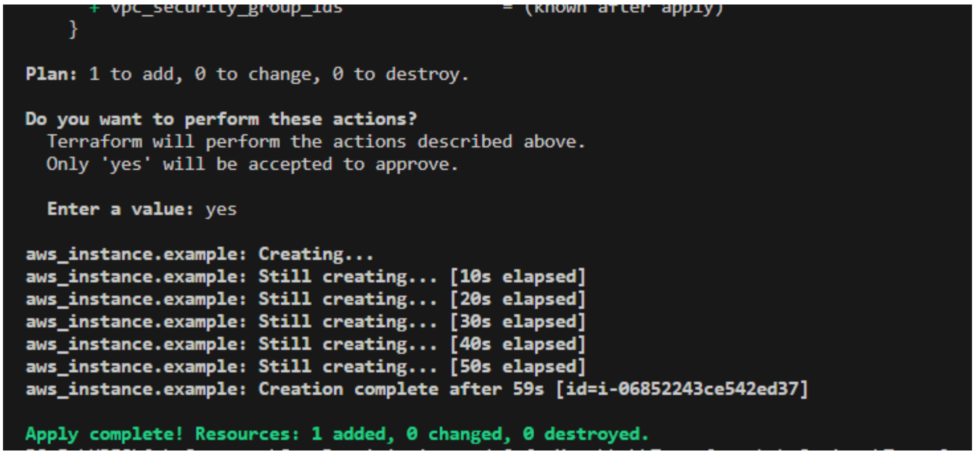


```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\Terraform Variables -4> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.
```

5] Apply it using command “Terraform apply”

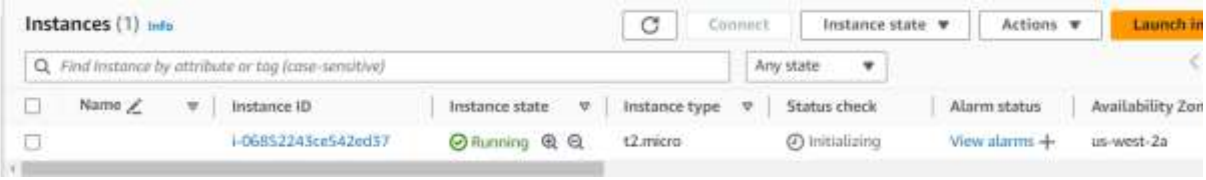


```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\Terraform Variables -4> terraform apply

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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-008fe2fc65df48dac"
      + 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)
```



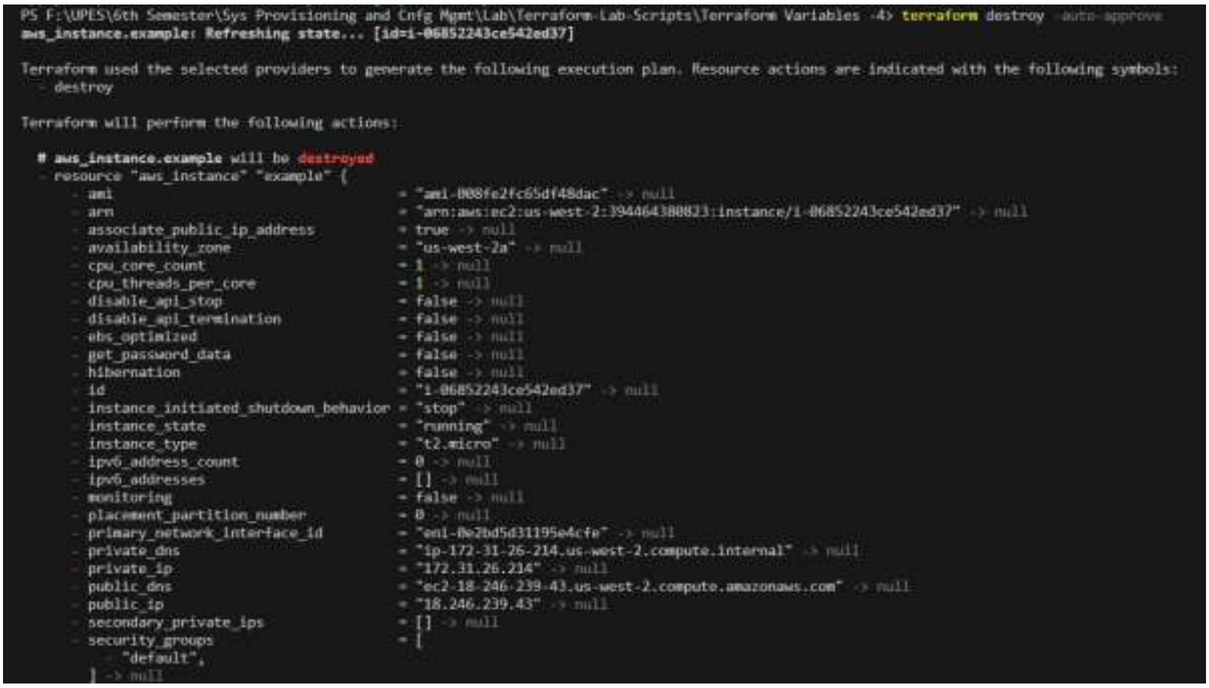
```
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 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

aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Still creating... [40s elapsed]
aws_instance.example: Still creating... [50s elapsed]
aws_instance.example: Creation complete after 59s [id=i-06852243ce542ed37]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

6] Verify Resources on AWS Management Console.

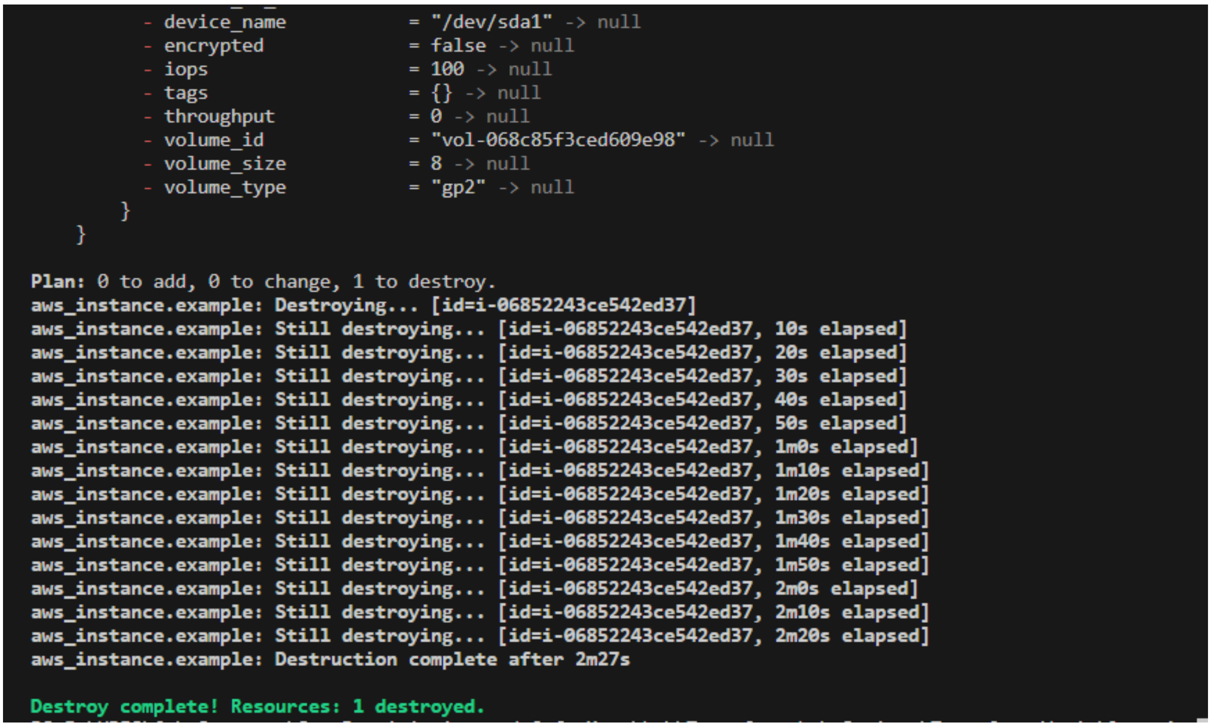


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
□		i-06852243ce542ed37	Running	t2.micro	Initializing	View alarms +	us-west-2a

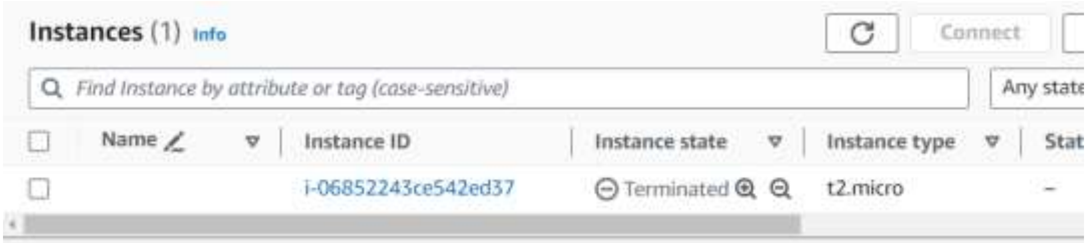
7] Cleanup Resources using command “Terraform destroy”



```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\Terraform Variables -4> terraform destroy -auto-approve
aws_instance.example: Refreshing state... [id=i-06852243ce542ed37]

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.example will be destroyed
  - resource "aws_instance" "example" {
      - ami                                  = "ami-008fe2fc65df48dac" -> null
      - arn                                  = "arn:aws:ec2:us-west-2:394464380823:instance/i-06852243ce542ed37" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-west-2a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-06852243ce542ed37" -> 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-0e2bd5d31195e4cfe" -> null
      - private_dns                          = "ip-172-31-26-214.us-west-2.compute.internal" -> null
      - private_ip                           = "172.31.26.214" -> null
      - public_dns                           = "ec2-18-246-239-43.us-west-2.compute.amazonaws.com" -> null
      - public_ip                            = "18.246.239.43" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
          - "default",
        ] -> null
```



```
          - device_name           = "/dev/sda1" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-068c85f3ced609e98" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp2" -> null
        }
    }

Plan: 0 to add, 0 to change, 1 to destroy.
aws_instance.example: Destroying... [id=i-06852243ce542ed37]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 10s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 20s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 30s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 40s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 50s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m0s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m10s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m20s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m30s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m40s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 1m50s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 2m0s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 2m10s elapsed]
aws_instance.example: Still destroying... [id=i-06852243ce542ed37, 2m20s elapsed]
aws_instance.example: Destruction complete after 2m27s

Destroy complete! Resources: 1 destroyed.
```



Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Stat
□		i-06852243ce542ed37	⊖ Terminated	t2.micro	–