SPCM LAB

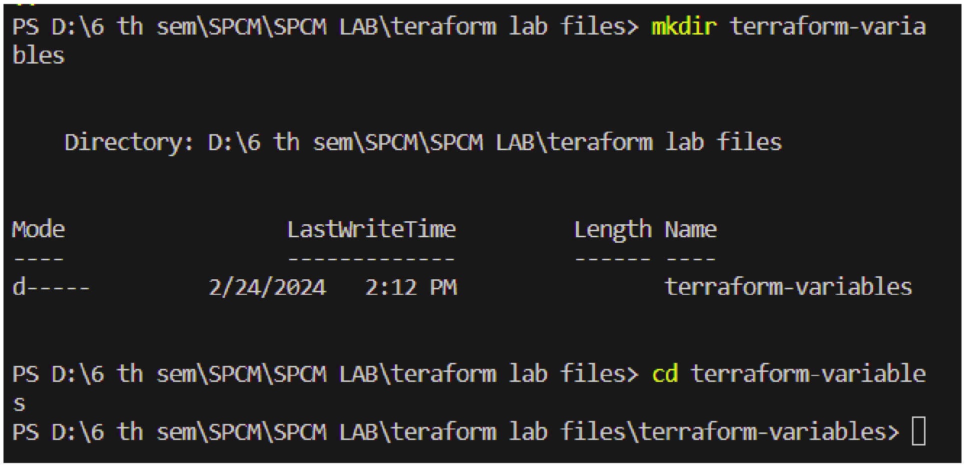
Gopika Jhanwar
500093662
R2142210318
Btech cse devops B2

Lab Exercise 4– Terraform Variables

1. Create a Terraform Directory:

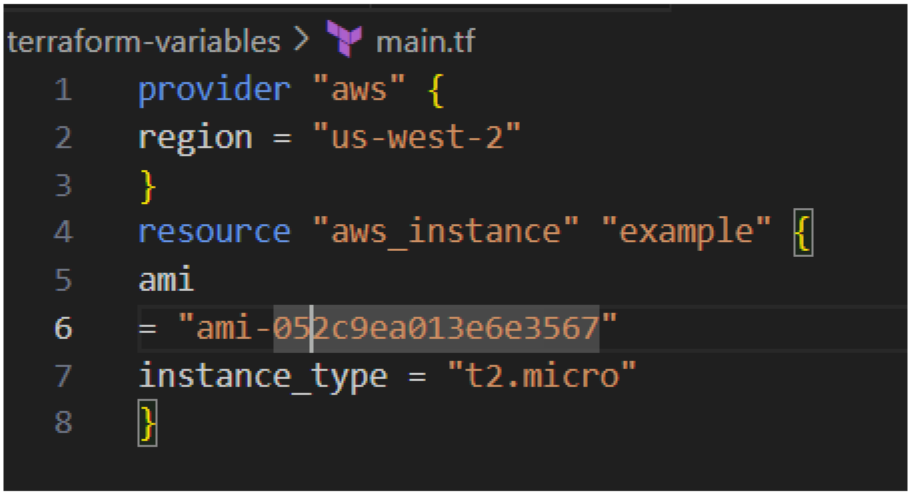
mkdir terraform-variables

cd terraform-variables



```
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files> mkdir terraform-varia
bles


    Directory: D:\6 th sem\SPCM\SPCM LAB\teraform lab files


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/24/2024   2:12 PM                terraform-variables


PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files> cd terraform-variable
s
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-variables> 
```

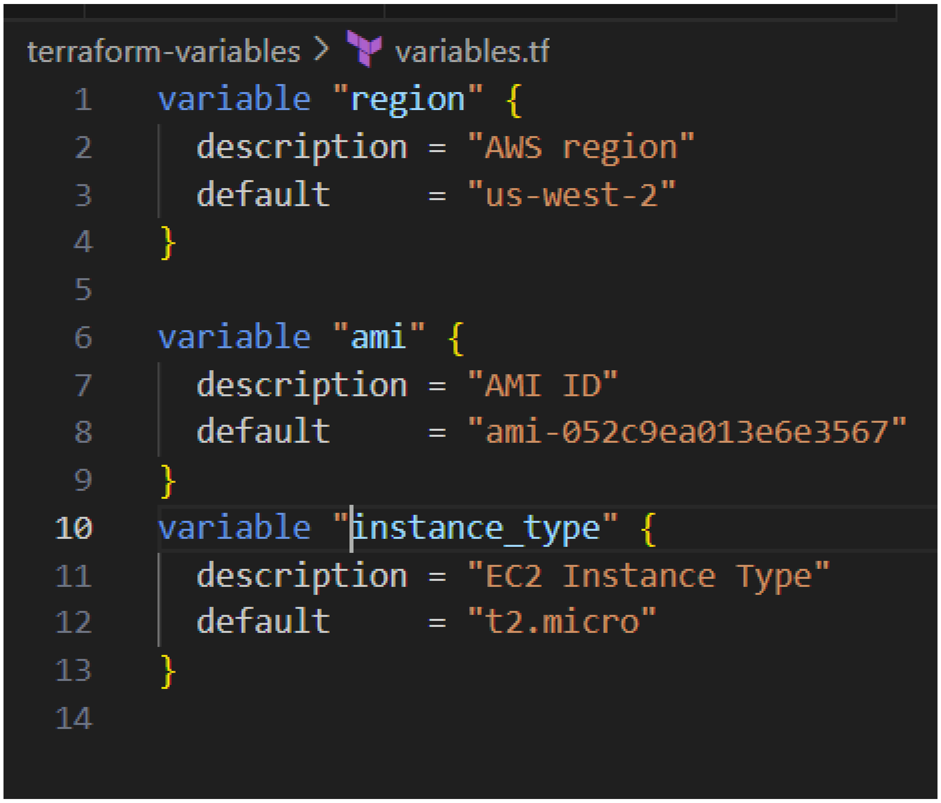
2. Create a Terraform Configuration File:



terraform-variables > main.tf

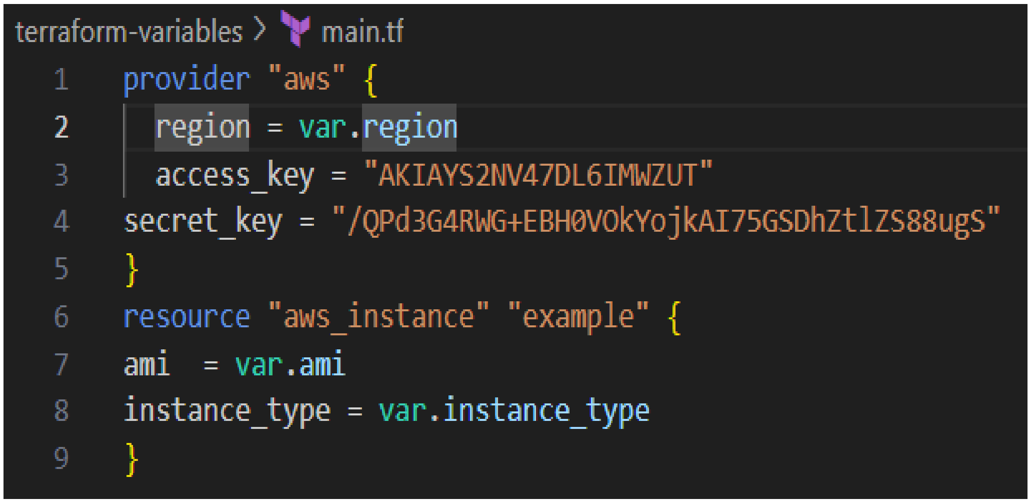
```
provider "aws" {
region = "us-west-2"
}
resource "aws_instance" "example" {
ami
= "ami-052c9ea013e6e3567"
instance_type = "t2.micro"
}
```

3. Define Variables:


terraform-variables > variables.tf

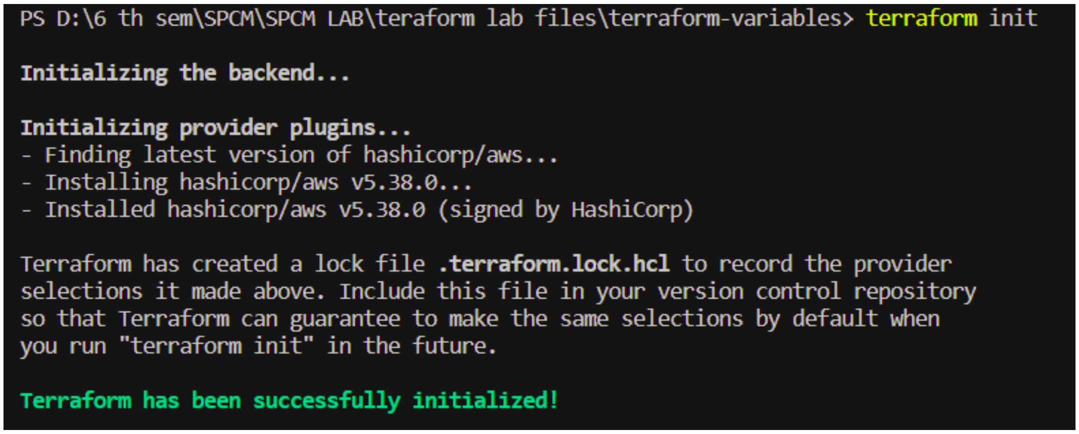
```
variable "region" {
  description = "AWS region"
  default     = "us-west-2"
}

variable "ami" {
  description = "AMI ID"
  default     = "ami-052c9ea013e6e3567"
}
variable "instance_type" {
  description = "EC2 Instance Type"
  default     = "t2.micro"
}

```

4. Use Variables in main.tf:

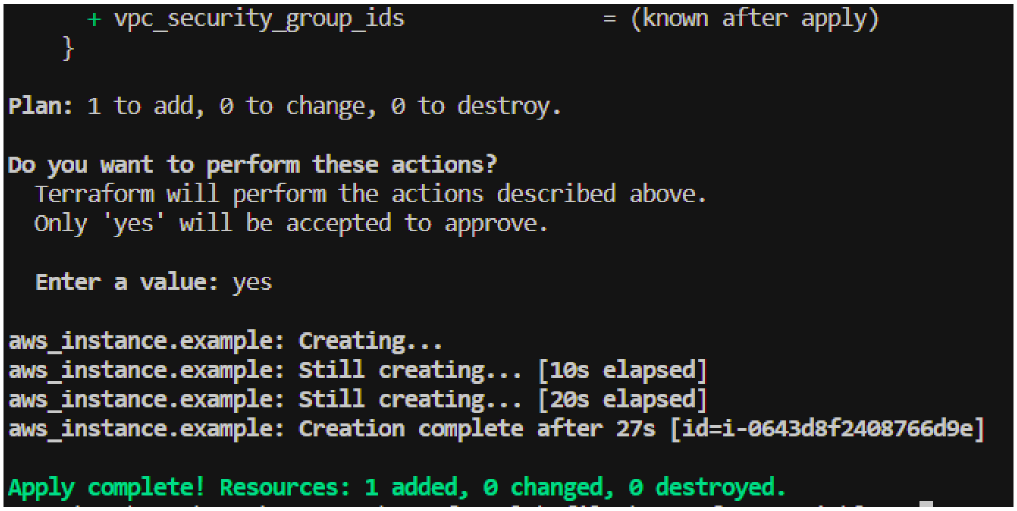

terraform-variables > main.tf

```
provider "aws" {
  region = var.region
  access_key = "AKIAYS2NV47DL6IMWZUT"
secret_key = "/QPd3G4RWG+EBH0VOkYojkAI75GSDhZtlZS88ugS"
}
resource "aws_instance" "example" {
ami  = var.ami
instance_type = var.instance_type
}
```

5. Initialize and Apply:
terraform init
terraform apply

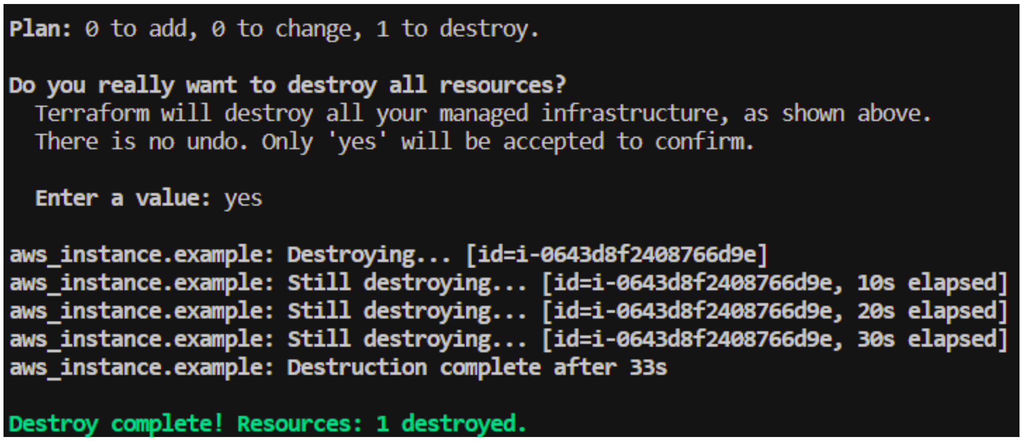


```
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-variables> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.38.0...
- Installed hashicorp/aws v5.38.0 (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!
```

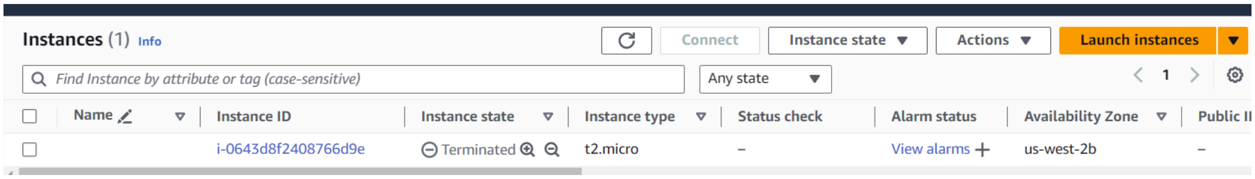


```
      + vpc_security_group_ids                = (known after apply)
    }

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: Creation complete after 27s [id=i-0643d8f2408766d9e]

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

6. Clean Up: terraform destroy



```
Plan: 0 to add, 0 to change, 1 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_instance.example: Destroying... [id=i-0643d8f2408766d9e]
aws_instance.example: Still destroying... [id=i-0643d8f2408766d9e, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0643d8f2408766d9e, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0643d8f2408766d9e, 30s elapsed]
aws_instance.example: Destruction complete after 33s

Destroy complete! Resources: 1 destroyed.
```



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
□		i-0643d8f2408766d9e	Terminated	t2.micro	–	View alarms +	us-west-2b	–