Lab Exercise 4– Terraform Variables

Objective:

Learn how to define and use variables in Terraform configuration.

Prerequisites:

- Install Terraform on your machine.

Steps:

1. Create a Terraform Directory:

- Create a new directory for your Terraform project.

```
mkdir terraform-variables

cd terraform-variables
```

2. Create a Terraform Configuration File:

- Create a file named main.tf within your project directory.

 # main.tf

```
provider "aws" {
    region = "us-west-2"
    }
resource "aws_instance" "example" {
    ami = "ami-03f4878755434977f"
     instance_type = "t2.micro"
     }

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

3. Define Variables:

- Open a new file named variables.tf. Define variables for region, ami, and instance_type.

 # variables.tf

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

```
variable "ami" {
     description = "AMI ID"
type = string
default    = "ami-03f4878755434977f"
}
```

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

4. Use Variables in main.tf:

- Modify main.tf to use the variables.

 # main.tf

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

5. Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration.

terraform apply

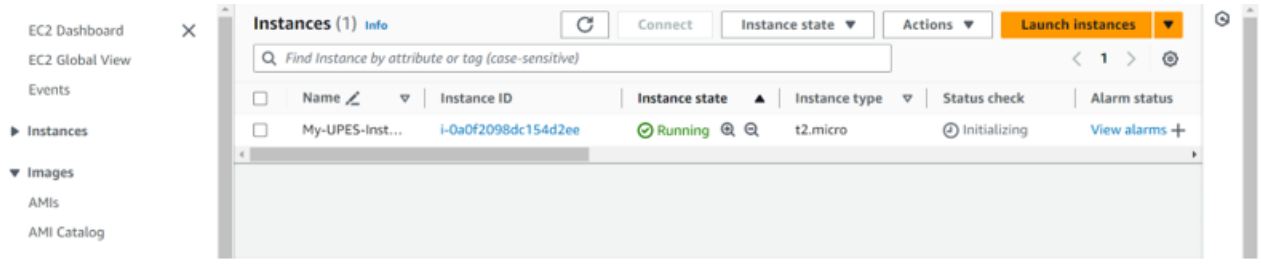
Observe how the region changes based on the variable override.

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if
you run "terraform apply" now.
PS C:\Users\anshi\OneDrive\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS> 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.My-Instnace[0] will be created
  + resource "aws_instance" "My-Instnace" {
      + 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"
```



6. Clean Up:

After testing, you can clean up resources.

terraform destroy

```
PS C:\Users\anshi\OneDrive\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS> terraform destroy
aws_instance.My-Instnace[0]: Refreshing state... [id=i-0a0f2098dc154d2ee]

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.My-Instnace[0] will be destroyed
```

Confirm the destruction by typing yes.

```
  Enter a value: yes

aws_instance.My-Instnace[0]: Destroying... [id=i-0a0f2098dc154d2ee]
aws_instance.My-Instnace[0]: Still destroying... [id=i-0a0f2098dc154d2ee, 10s elapsed]
aws_instance.My-Instnace[0]: Still destroying... [id=i-0a0f2098dc154d2ee, 20s elapsed]
aws_instance.My-Instnace[0]: Still destroying... [id=i-0a0f2098dc154d2ee, 30s elapsed]
aws_instance.My-Instnace[0]: Still destroying... [id=i-0a0f2098dc154d2ee, 40s elapsed]
aws_instance.My-Instnace[0]: Destruction complete after 44s

Destroy complete! Resources: 1 destroyed.
```