Lab Exercise 6– 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

```
resource "aws_instance" "myinstance-1" {
  ami = var.myami
  instance_type = var.my_instance_type
  count = var.mycount
  tags = {
    Name= "My Instance"
  }
}
```

3. Define Variables:

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

 # variables.tf

```
variable "myami" {
  type = string
  default = "ami-08718895af4dfa033"
}

variable "mycount" {

  type = number
  default = 5
}

variable "my_instance_type" {
   type = string
   default = "t2.micro"
}
```

4. Initialize and Apply:

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

```
terraform init

terraform plan

terraform apply -auto-approve
```

```
aws_instance.myinstance-1[4]: Creating...
aws_instance.myinstance-1[2]: Creating...
aws_instance.myinstance-1[3]: Creating...
aws_instance.myinstance-1[0]: Creating...
aws_instance.myinstance-1[1]: Creating...
aws_instance.myinstance-1[3]: Still creating... [00m10s elapsed]
aws_instance.myinstance-1[1]: Still creating... [00m10s elapsed]
aws_instance.myinstance-1[4]: Still creating... [00m10s elapsed]
aws_instance.myinstance-1[2]: Still creating... [00m10s elapsed]
aws_instance.myinstance-1[0]: Still creating... [00m10s elapsed]
aws_instance.myinstance-1[0]: Creation complete after 15s [id=i-03cd29f8e4855c1c5]
aws_instance.myinstance-1[1]: Creation complete after 16s [id=i-06bf8e67f6ca69f01]
aws_instance.myinstance-1[4]: Creation complete after 16s [id=i-0f40442f56cedf429]
aws_instance.myinstance-1[2]: Creation complete after 16s [id=i-0c423071042f7bc96]
aws_instance.myinstance-1[3]: Creation complete after 16s [id=i-089a6dbed23b46a5e]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
```

Observe how the region changes based on the variable override.

5. Clean Up:

After testing, you can clean up resources.

terraform destroy

```
aws_instance.myinstance-1[0]: Still destroying... [id=i-03cd29f8e4855c1c5, 00m50s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-0c423071042f7bc96, 00m50s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0f40442f56cedf429, 00m50s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-089a6dbed23b46a5e, 00m50s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-06bf8e67f6ca69f01, 00m50s elapsed]
aws_instance.myinstance-1[2]: Destruction complete after 52s
aws_instance.myinstance-1[4]: Still destroying... [id=i-0f40442f56cedf429, 01m00s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-03cd29f8e4855c1c5, 01m00s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-06bf8e67f6ca69f01, 01m00s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-089a6dbed23b46a5e, 01m00s elapsed]
aws_instance.myinstance-1[0]: Destruction complete after 1m2s
aws_instance.myinstance-1[3]: Destruction complete after 1m2s
aws_instance.myinstance-1[1]: Destruction complete after 1m2s
aws_instance.myinstance-1[4]: Still destroying... [id=i-0f40442f56cedf429, 01m10s elapsed]
aws_instance.myinstance-1[4]: Destruction complete after 1m12s

Destroy complete! Resources: 5 destroyed.
mohdanas@Mohds-MacBook-Air terraform-variables %
```

Confirm the destruction by typing yes.

6. Conclusion:

This lab exercise introduces you to Terraform variables and demonstrates how to use them in your configurations. Experiment with different variable values and overrides to understand their impact on the infrastructure provisioning process.