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.**

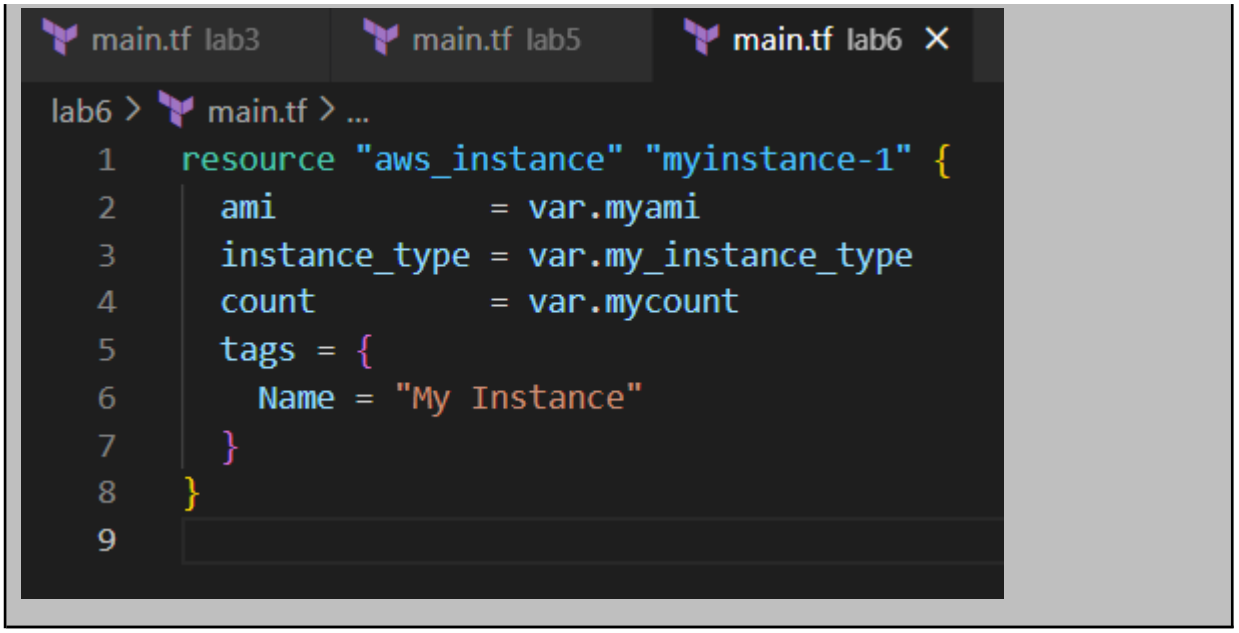
```
mkdir terraform-variables

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"
```

}

main.tf lab3 | main.tf lab5 | main.tf lab6 | va

lab6 > 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

```
PS C:\Github Repositores\Terraform-Demo\lab5> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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.
PS C:\Github Repositores\Terraform-Demo\lab5>
```

terraform plan

```
      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

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.
```

terraform apply -auto-approve

```
Plan: 5 to add, 0 to change, 0 to destroy.
aws_instance.myinstance-1[0]: Creating...
aws_instance.myinstance-1[4]: Creating...
aws_instance.myinstance-1[3]: Creating...
aws_instance.myinstance-1[1]: Creating...
aws_instance.myinstance-1[2]: Creating...
aws_instance.myinstance-1[0]: Still creating... [10s elapsed]
aws_instance.myinstance-1[4]: Still creating... [10s elapsed]
aws_instance.myinstance-1[3]: Still creating... [10s elapsed]
aws_instance.myinstance-1[1]: Still creating... [10s elapsed]
aws_instance.myinstance-1[2]: Still creating... [10s elapsed]
aws_instance.myinstance-1[3]: Creation complete after 19s [id=i-0143a3d306b3c17b9]
aws_instance.myinstance-1[1]: Creation complete after 19s [id=i-010c9deca7902eb0e]
aws_instance.myinstance-1[4]: Creation complete after 19s [id=i-068581e56455ce396]
aws_instance.myinstance-1[0]: Creation complete after 19s [id=i-059935081ef15c298]
aws_instance.myinstance-1[2]: Creation complete after 19s [id=i-09273b9739ffeb156]

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[1]: Destroying... [id=i-010c9deca7902eb0e]
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 10s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-0143a3d306b3c17b9, 10s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-059935081ef15c298, 10s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-09273b9739ffeb156, 10s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-010c9deca7902eb0e, 10s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-0143a3d306b3c17b9, 20s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 20s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-059935081ef15c298, 20s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-09273b9739ffeb156, 20s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-010c9deca7902eb0e, 20s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 30s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-0143a3d306b3c17b9, 30s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-059935081ef15c298, 30s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-09273b9739ffeb156, 30s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-010c9deca7902eb0e, 30s elapsed]
aws_instance.myinstance-1[2]: Destruction complete after 35s
aws_instance.myinstance-1[1]: Destruction complete after 35s
aws_instance.myinstance-1[0]: Still destroying... [id=i-059935081ef15c298, 40s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 40s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-0143a3d306b3c17b9, 40s elapsed]
aws_instance.myinstance-1[0]: Destruction complete after 45s
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 50s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-0143a3d306b3c17b9, 50s elapsed]
aws_instance.myinstance-1[3]: Destruction complete after 55s
aws_instance.myinstance-1[4]: Still destroying... [id=i-068581e56455ce396, 1m0s elapsed]
aws_instance.myinstance-1[4]: Destruction complete after 1m6s

Destroy complete! Resources: 5 destroyed.
```

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.