Lab Exercise 6– Terraform Variables

Objective:

Learn how to define and use variables in Terraform configuration.

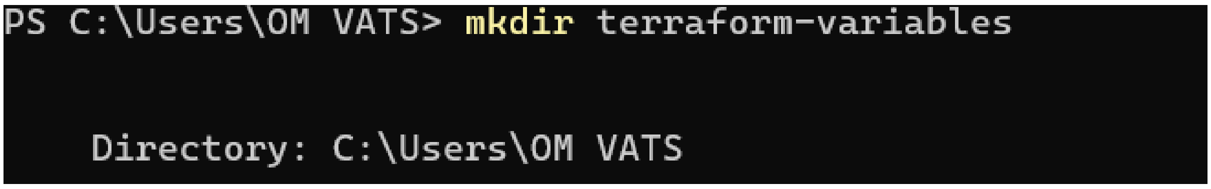
Prerequisites:

- Install Terraform on your machine.

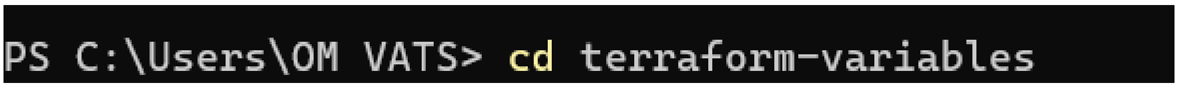
Steps:

1. Create a Terraform Directory:

- Create a new directory for your Terraform project.



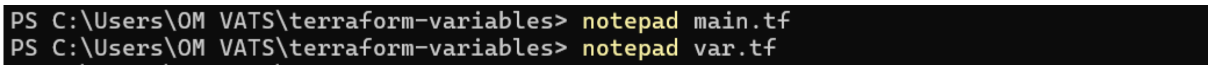
```
PS C:\Users\OM VATS> mkdir terraform-variables


    Directory: C:\Users\OM VATS
```



```
PS C:\Users\OM VATS> cd terraform-variables
```

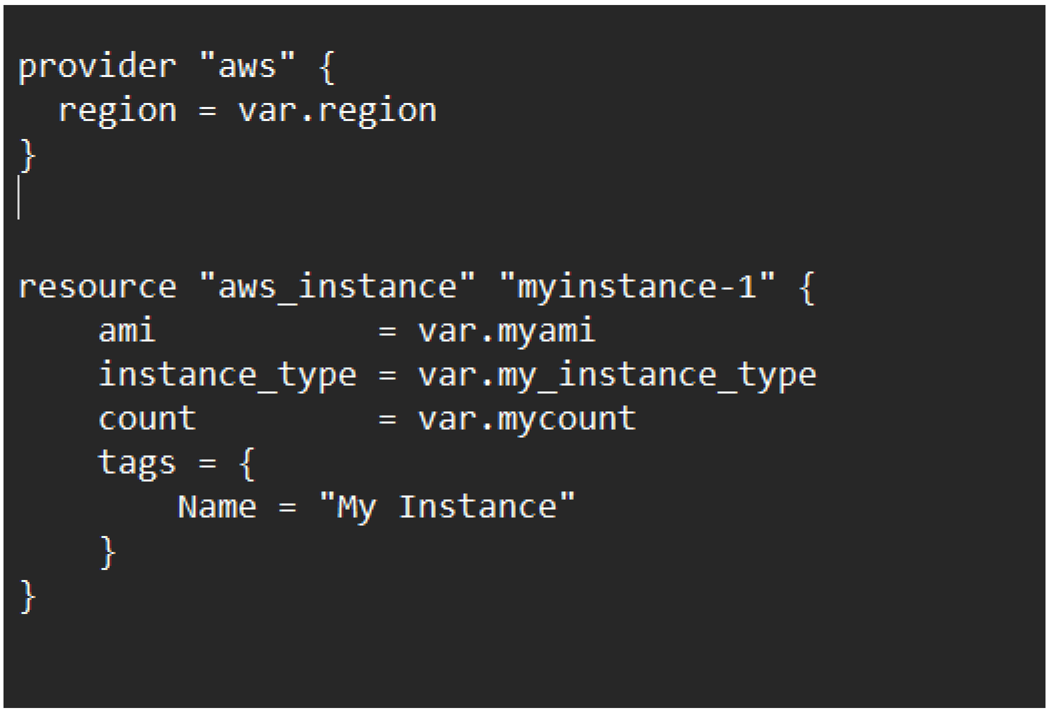
```
mkdir terraform-variables

cd terraform-variables
```

2. Create a Terraform Configuration File:

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

```
PS C:\Users\OM VATS\terraform-variables> notepad main.tf
PS C:\Users\OM VATS\terraform-variables> notepad var.tf
```

# main.tf



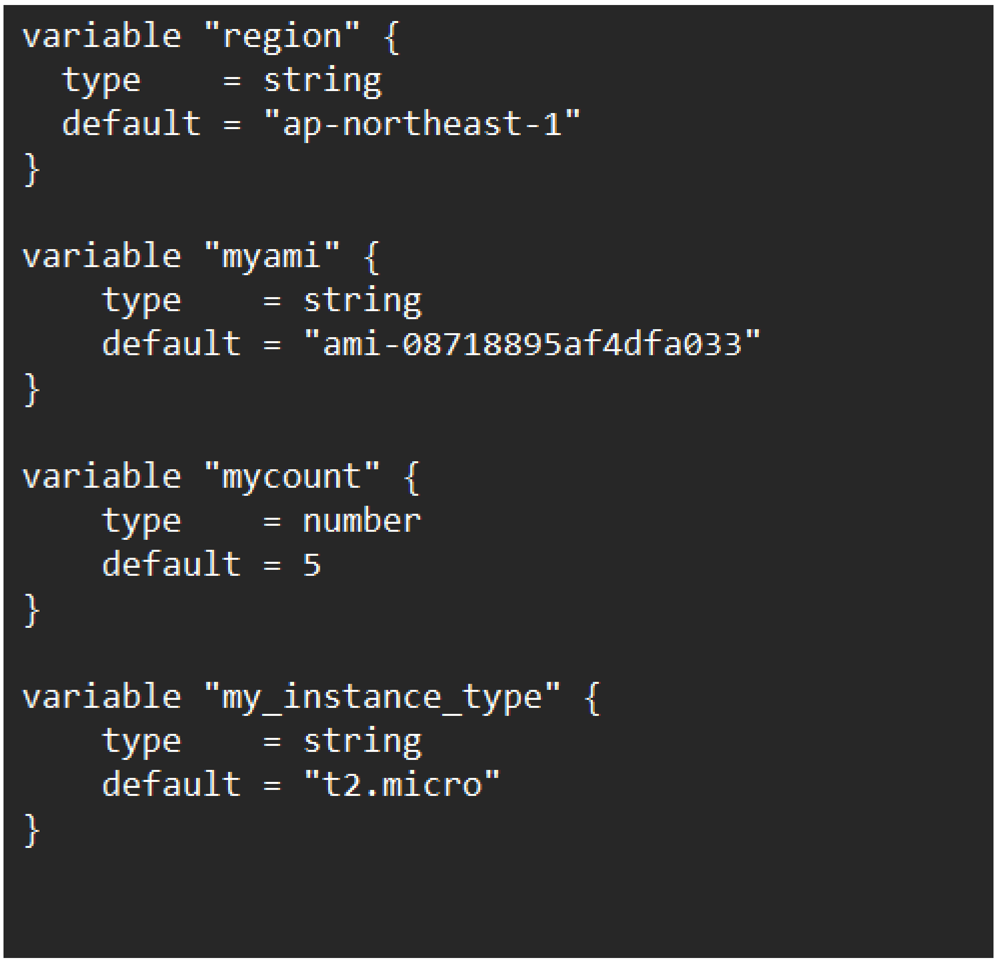
```
provider "aws" {
  region = var.region
}


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

```
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

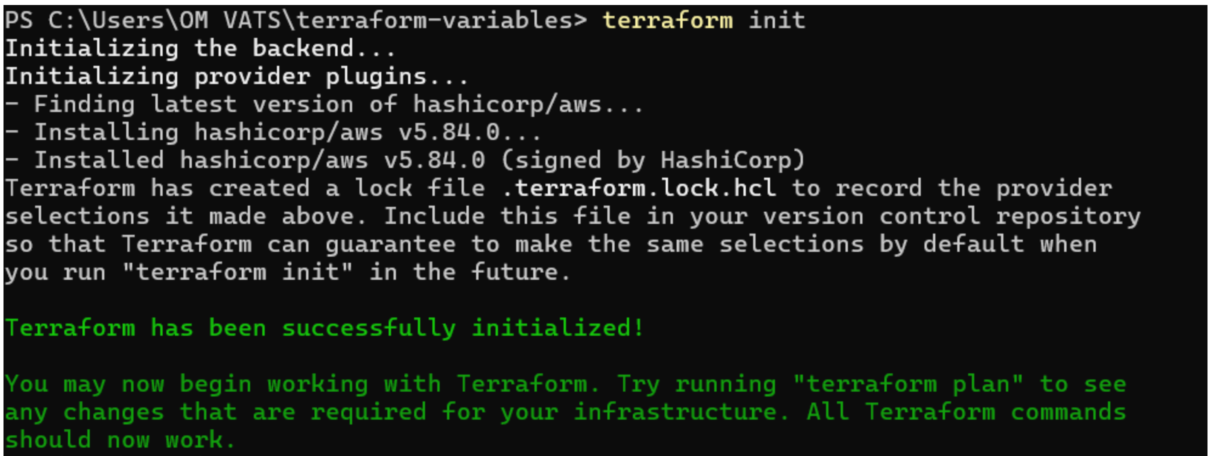


```
variable "region" {
  type    = string
  default = "ap-northeast-1"
}

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

variable "mycount" {
    type    = number
    default = 5
}

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

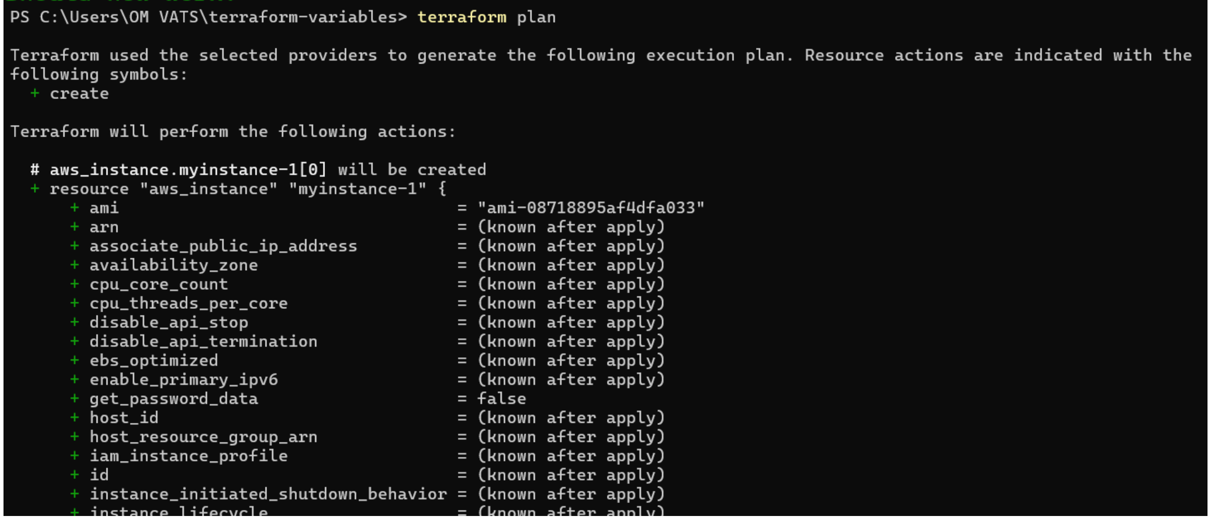
```
variable "myami" {
  type = string
  default = "ami-08718895af4dfa033"
}

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.

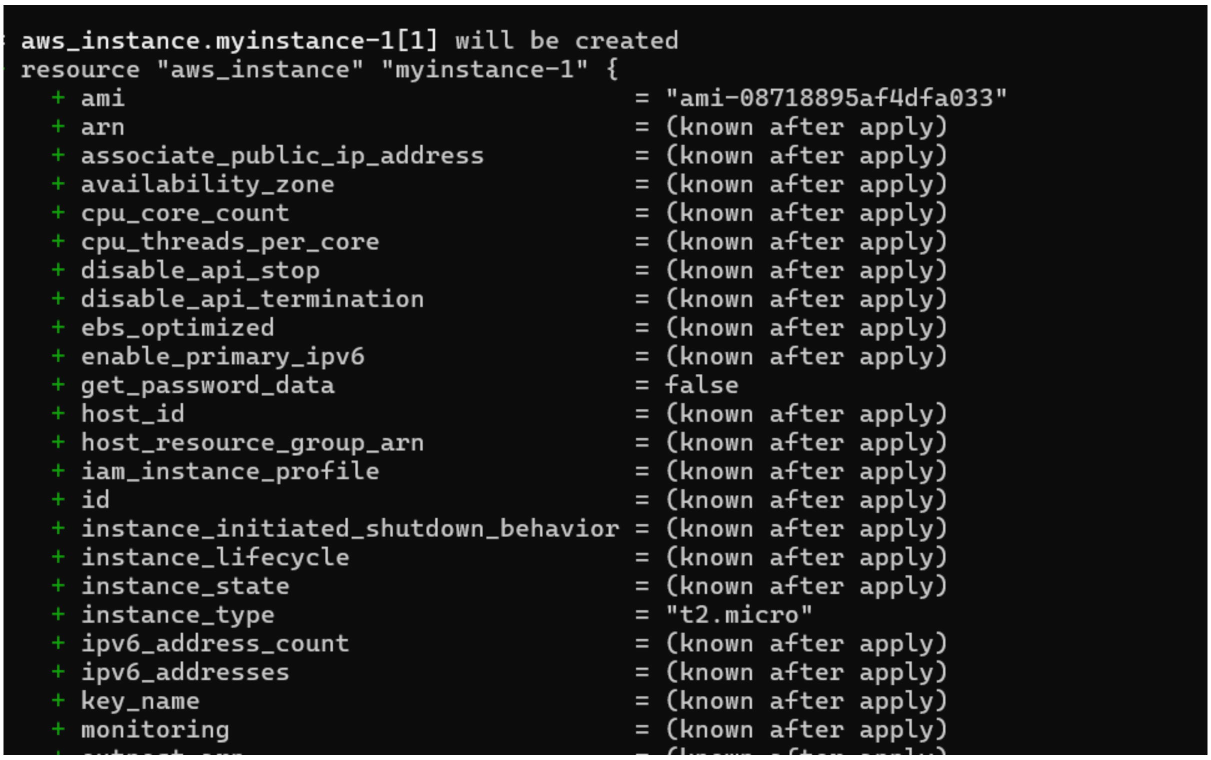


```
PS C:\Users\OM VATS\terraform-variables> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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.
```

- 

```
PS C:\Users\OM VATS\terraform-variables> terraform plan

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.myinstance-1[0] will be created
  + resource "aws_instance" "myinstance-1" {
      + ami                                  = "ami-08718895af4dfa033"
      + 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)
      + enable_primary_ipv6                  = (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)
```

-



```
aws_instance.myinstance-1[1] will be created
resource "aws_instance" "myinstance-1" {
  + ami                                  = "ami-08718895af4dfa033"
  + 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)
  + enable_primary_ipv6                  = (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)
```

-
-

```
terraform init

terraform plan

terraform apply -auto-approve
```

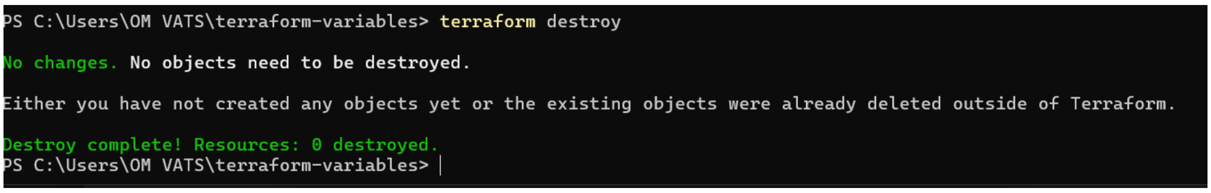
Observe how the region changes based on the variable override.

5. Clean Up:

After testing, you can clean up resources.

```
terraform destroy
```

Confirm the destruction by typing yes.



```
PS C:\Users\OM VATS\terraform-variables> terraform destroy

No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.

Destroy complete! Resources: 0 destroyed.
PS C:\Users\OM VATS\terraform-variables> 
```

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.