

Lab Exercise 6– Terraform Variables

Objective:

Learn how to define and use variables in Terraform configuration.

Prerequisites:

- Install Terraform on your machine.

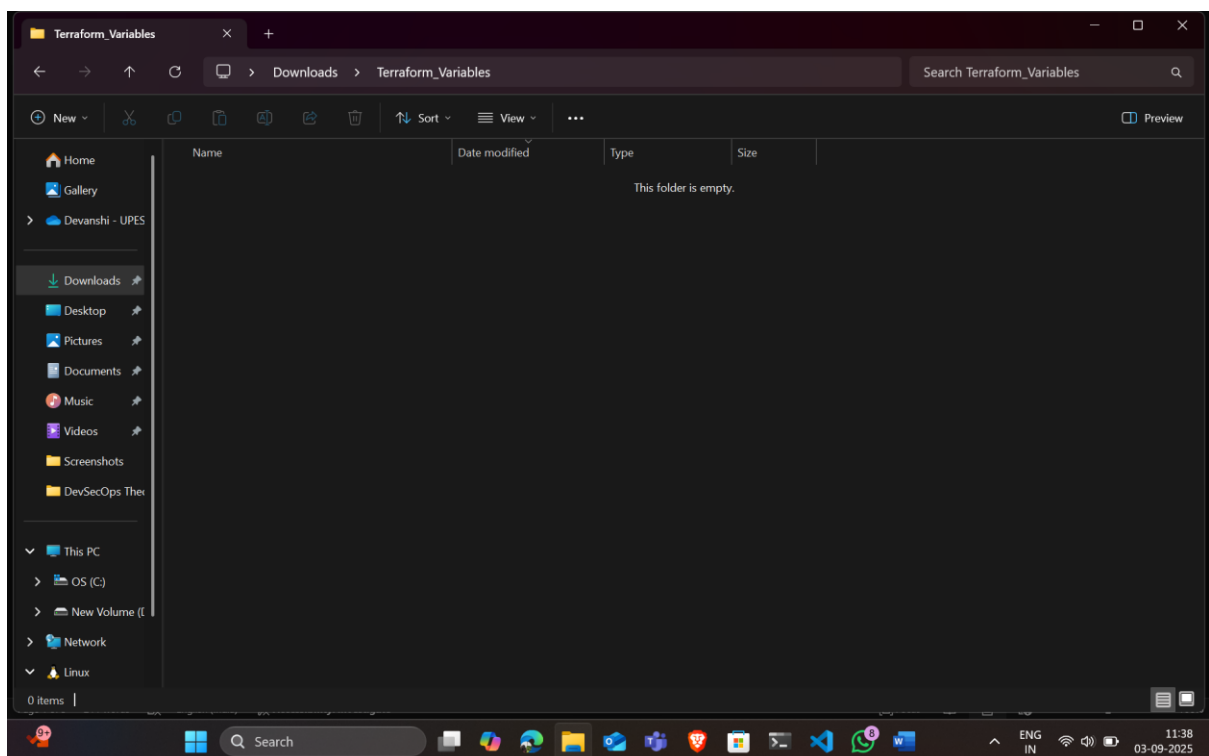
Steps:

1. Create a Terraform Directory:

- Create a new directory for your Terraform project.

```
mkdir terraform-variables
```

```
cd terraform-variables
```

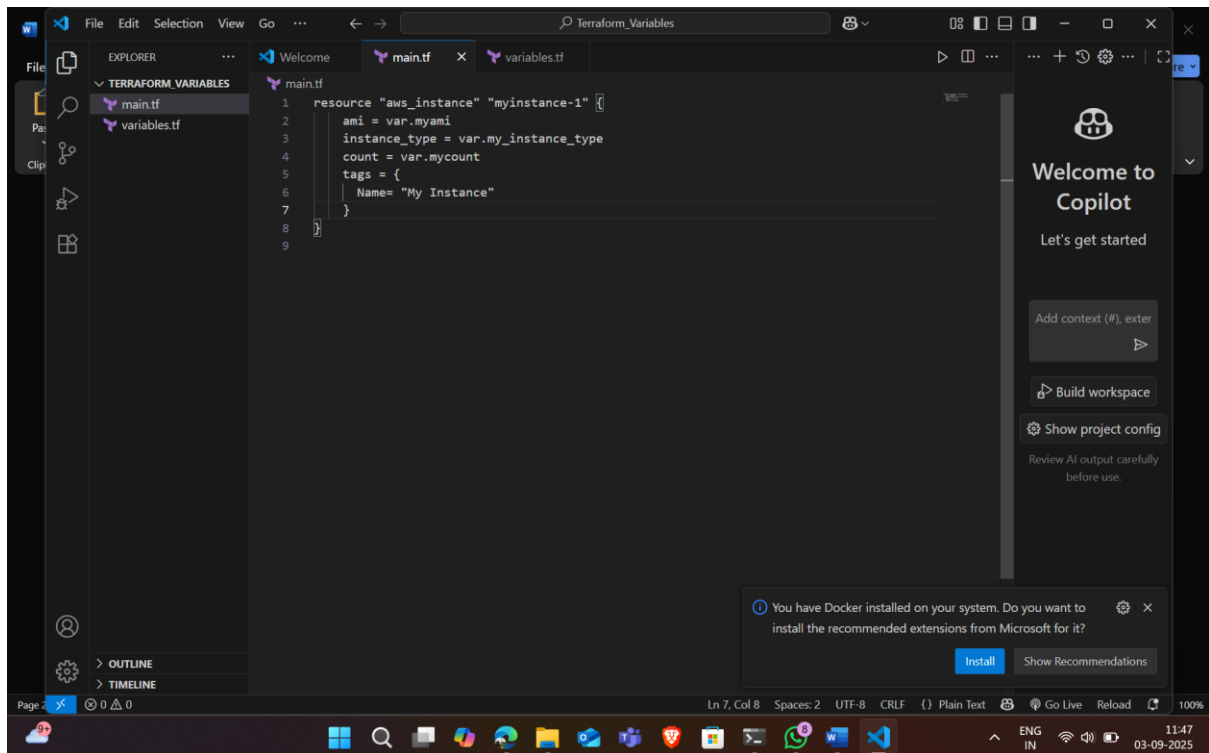


2. Create a Terraform Configuration File:

- 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"  
  }  
}
```

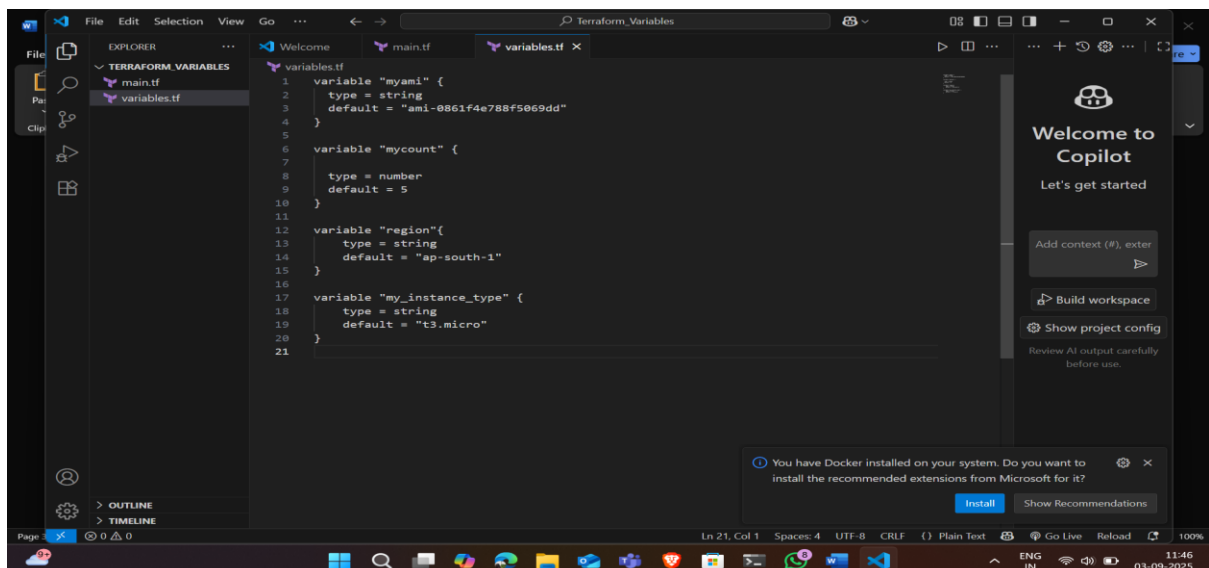


3. Define Variables:

- 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

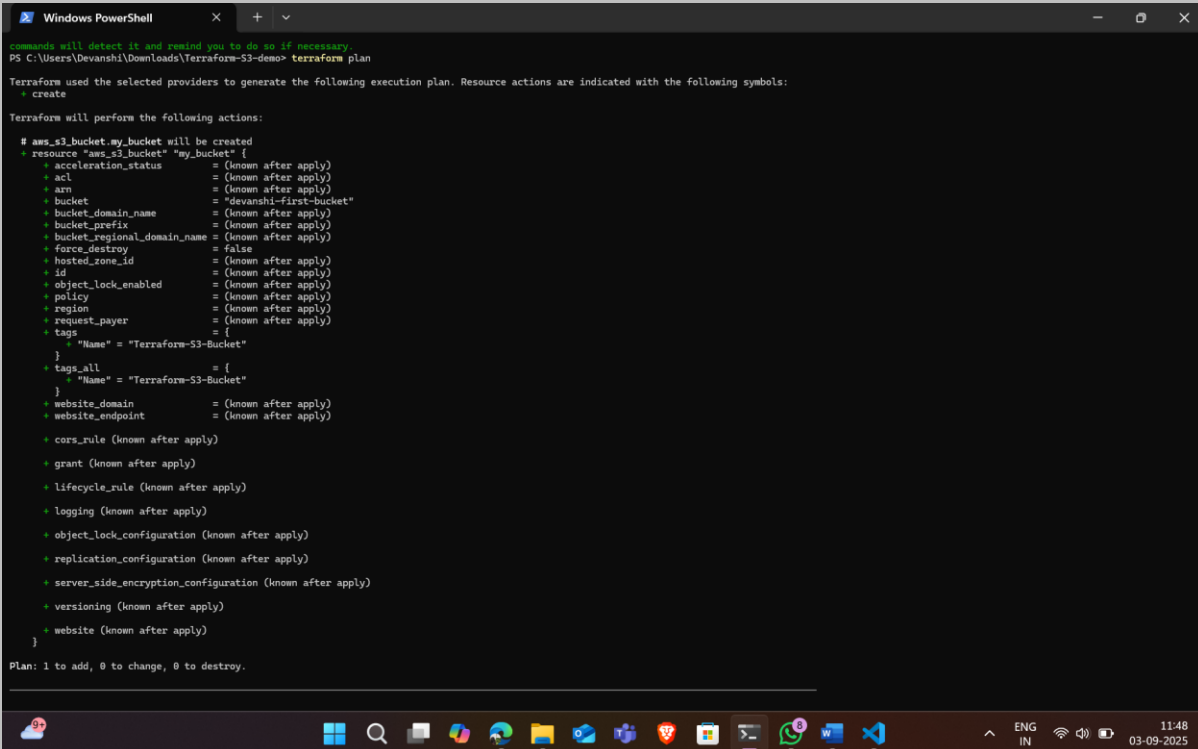
```
PS C:\Users\Devanshi\Downloads\Terraform-S3-demo> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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:\Users\Devanshi\Downloads\Terraform-S3-demo>
```

terraform plan



```
Windows PowerShell
PS C:\Users\Devanshi\Downloads\Terraform-S3-demo> 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_s3_bucket.my_bucket will be created
+ resource "aws_s3_bucket" "my_bucket" {
+   acceleration_status = (known after apply)
+   acl                 = (known after apply)
+   arn                 = (known after apply)
+   bucket              = "devanshi-first-bucket"
+   bucket_domain_name = (known after apply)
+   bucket_prefix       = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy       = false
+   hosted_zone_id      = (known after apply)
+   id                  = (known after apply)
+   object_lock_enabled = (known after apply)
+   policy              = (known after apply)
+   region              = (known after apply)
+   request_payer       = (known after apply)
+   tags                = {
+     "Name" = "Terraform-S3-Bucket"
+   }
+   tags_all            = {
+     "Name" = "Terraform-S3-Bucket"
+   }
+   website_domain      = (known after apply)
+   website_endpoint    = (known after apply)
+   cors_rule           = (known after apply)
+   grant               = (known after apply)
+   lifecycle_rule      = (known after apply)
+   logging              = (known after apply)
+   object_lock_configuration = (known after apply)
+   replication_configuration = (known after apply)
+   server_side_encryption_configuration = (known after apply)
+   versioning           = (known after apply)
+   website              = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

terraform apply -auto-approve

```

Windows PowerShell
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_s3_bucket.my_bucket will be created
+ resource "aws_s3_bucket" "my_bucket" {
+   acceleration_status = (known after apply)
+   acl                 = (known after apply)
+   ann                 = (known after apply)
+   bucket              = "devanshi-first-bucket"
+   bucket_domain_name = (known after apply)
+   bucket_prefix       = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy       = false
+   hosted_zone_id      = (known after apply)
+   id                  = (known after apply)
+   object_lock_enabled = (known after apply)
+   policy              = (known after apply)
+   region              = (known after apply)
+   request_payer       = (known after apply)
+   tags                = {
+     "Name" = "Terraform-S3-Bucket"
+   }
+   tags_all          = {
+     "Name" = "Terraform-S3-Bucket"
+   }
+   website_domain     = (known after apply)
+   website_endpoint   = (known after apply)
+   cors_rule (known after apply)
+   grant (known after apply)
+   lifecycle_rule (known after apply)
+   logging (known after apply)
+   object_lock_configuration (known after apply)
+   replication_configuration (known after apply)
+   server_side_encryption_configuration (known after apply)
+   versioning (known after apply)
+   website (known after apply)
+ }

Plan: 1 to add, 0 to change, 0 to destroy.
aws_s3_bucket.my_bucket: Creating...
aws_s3_bucket.my_bucket: Creation complete after 3s [id=devanshi-first-bucket]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Devanshi\Downloads\Terraform-S3-demo>

```

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.

```

Windows PowerShell
} -> null
- tags_all = {
-   "Name" = "Terraform-S3-Bucket"
- } -> null
# (3 unchanged attributes hidden)

- grant {
-   id = "bd9c9e20676a876f749b640b9b941470b1a40fe2cc693834b83f51f7be145d06" -> null
-   permissions = [
-     "FULL_CONTROL",
-   ] -> null
-   type = "CanonicalUser" -> null
-   # (1 unchanged attribute hidden)
- }

- server_side_encryption_configuration {
-   rule {
-     bucket_key_enabled = false -> null
-     apply_server_side_encryption_by_default {
-       sse_algorithm = "AES256" -> null
-       # (1 unchanged attribute hidden)
-     }
-   }
- }

- versioning {
-   enabled = false -> null
-   mfa_delete = false -> null
- }
}

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_s3_bucket.my_bucket: Destroying... [id=devanshi-first-bucket]
aws_s3_bucket.my_bucket: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\Devanshi\Downloads\Terraform-S3-demo>

```

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.