

# Lab Exercise 6– Terraform V

## Objective:

Learn how to define and use variables in Terraform

## Prerequisites:

- Install Terraform on your machine.

## Steps:

### 1. Create a Terraform Directory:

- Create a new directory for your Terraform i

```
[ujjwal@ujjwalmacbook Terraform-S3-demo % mkdir Terraform-variables  
[ujjwal@ujjwalmacbook Terraform-S3-demo % cd Terraform-variables
```

## 2. Create a Terraform Configuration

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

**# main.tf**

```
resource "aws_instance" "myinstance-1" {
```

```
ujjwal@ujjwalmacbook Terraform-variables % touch variables.tf
[ujjwal@ujjwalmacbook Terraform-variables % vim variables.tf
ujjwal@ujjwalmacbook Terraform-variables % █
```

```
resource "aws_instance" "myinstance-1-ujjwal" {
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 instance\_type.

**# variables.tf**

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

```
iivARIABLE "myami" {  
  type = string  
  , default = "ami-00bb6a80F01 03502"*  
  variable "mycount" {  
    type = number  
    default = 5  
  }  
  variable "my_instance_type" {  
    type = string  
    default =  
  ~  
  ~  
  ~  
  ~  
  ~  
  ~  
  ~  
}
```

## 4. Initialize and Apply:

- Run the following Terraform commands to configuration.

**terraform init**

**terraform plan**

**terraform apply -auto-approve**

```
$ terraform init
Initializing the backend...
- Finding latest version of haskend...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.85.0(signed by HashiCorp)
Terraform has created a lockf file .terraform.lock.hcl to
record the provider selections it made above. Inis file in
your version control repository so that Terraform can mak
the same selections by default when you run turr
```

**Terraform has been successssfully initialized!**

You may now begin working with Terraform. Try running ter  
rarform plan' to see anyhages that are required for your  
infrastructure. All Terraform commands should now work.

```
> terraform plan
```

```
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)
+ private_ip = (known after apply)
+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
  + "Name" = "My Instance"
}
+ tags_all = {
  + "Name" = "My Instance"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)

+ instance_market_options (known after apply)

+ maintenance_options (known after apply)

+ metadata_options (known after apply)

+ network_interface (known after apply)

+ 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

```
terraform apply -auto-approve
```

```
+ spot_instance_request_id      = (known after apply)
+ subnet_id                    = (known after apply)
+ tags                          = {
  + "Name" = "My Instance"
}
+ tags_all                      = {
  + "Name" = "My Instance"
}
+ tenancy                       = (known after apply)
+ user_data                     = (known after apply)
+ user_data_base64              = (known after apply)
+ user_data_replace_on_change   = false
+ vpc_security_group_ids        = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)

+ instance_market_options (known after apply)

+ maintenance_options (known after apply)

+ metadata_options (known after apply)

+ network_interface (known after apply)

+ private_dns_name_options (known after apply)

+ root_block_device (known after apply)
}
```

