



# **System Provisioning and Configuration Management LAB**

SUBMITTED TO

Dr. Hitesh Kumar Sharma

SUBMITTED BY

Siddharth Agarwal

500107594

R2142220663

Btech CSE DevOps B1

# 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"  
  }  
}
```

```
main.tf  login.tf  variables.tf
lab6 > terraform-variables > main.tf
1  resource "aws_instance" "myinstance-1" {
2      ami = var.myami
3      instance_type = var.my_instance_type
4      count = var.mycount
5      tags = {
6          Name= "My Instance"
7      }
8  }
9
```

### 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 login.tf variables.tf X
lab6 > terraform-variables > variables.tf
1  variable "myami" {
2    type = string
3    default = "ami-0e2c8caa4b6378d8c"
4  }
5
6  variable "mycount" {
7
8    type = number
9    default = 5
10 }
11
12 variable "my_instance_type" {
13   type = string
14   default = "t2.micro"
15 }
16 |
```

```
main.tf login.tf X variables.tf
lab6 > terraform-variables > login.tf
1  terraform {
2    required_providers {
3      aws = {
4        source = "hashicorp/aws"
5        version = "5.31.0"
6      }
7    }
8  }
9
10 provider "aws" {
11   region      = "ap-south-1" # Replace with your preferred region
12   access_key  = "AKIA2BRNT5GDKSJHCHAQ" # Replace with your Access K
13   secret_key  = "oL5Yo3P1b7MJfV15eJebkI4sm2AfmwQl20DjeDw/" # Replac
14 }
15 |
```

## 4. Initialize and Apply:

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

```
terraform init
```

```
terraform plan
```

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

Observe how the region changes based on the variable override.

```
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables> terraform i
nit
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.83.1...
- Installed hashicorp/aws v5.83.1 (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.

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:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables> |
```

```
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables> terraform p
lan

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-0e2c8caa4b6378d8c"
  + 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)
  + get_password_data     = false
  + host_id               = (known after apply)
  + host_resource_group_arn = (known after apply)
```

```

+ root_block_device (known after apply)
}

# aws_instance.myinstance-1[1] will be created
+ resource "aws_instance" "myinstance-1" {
+   ami                        = "ami-0e2c8caa4b6378d8c"
+   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)
+   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)

```

```

+ 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.

PS C:\SID\_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables>

PS C:\SID\_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables> terraform apply -auto-approve

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-00bb6a80f01f03502"
+   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)
+   get_password_data          = false
+   host_id                   = (known after apply)
+   host_resource_group_arn    = (known after apply)
+   iam_instance_profile       = (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.
aws_instance.myinstance-1[0]: Creating...
aws_instance.myinstance-1[3]: Creating...
aws_instance.myinstance-1[2]: Creating...
aws_instance.myinstance-1[1]: Creating...
aws_instance.myinstance-1[4]: Creating...
aws_instance.myinstance-1[3]: Still creating... [10s elapsed]
aws_instance.myinstance-1[4]: Still creating... [10s elapsed]
aws_instance.myinstance-1[1]: Still creating... [10s elapsed]
aws_instance.myinstance-1[0]: Still creating... [10s elapsed]
aws_instance.myinstance-1[2]: Still creating... [10s elapsed]
aws_instance.myinstance-1[3]: Creation complete after 13s [id=i-063e1704f5fd394cd]
aws_instance.myinstance-1[1]: Creation complete after 13s [id=i-038bff8535167ba8c]
aws_instance.myinstance-1[0]: Creation complete after 13s [id=i-09a7f3f4d24f77a06]
aws_instance.myinstance-1[2]: Creation complete after 13s [id=i-07490f9592872db0a]
aws_instance.myinstance-1[4]: Creation complete after 13s [id=i-0d5d660dd4b671835]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables>

```

## 5. Clean Up:

After testing, you can clean up resources.

### terraform destroy

Confirm the destruction by typing yes.

```

PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\terraform-variables> terraform d
estroy
aws_instance.myinstance-1[1]: Refreshing state... [id=i-038bff8535167ba8c]
aws_instance.myinstance-1[3]: Refreshing state... [id=i-063e1704f5fd394cd]
aws_instance.myinstance-1[0]: Refreshing state... [id=i-09a7f3f4d24f77a06]
aws_instance.myinstance-1[4]: Refreshing state... [id=i-0d5d660dd4b671835]
aws_instance.myinstance-1[2]: Refreshing state... [id=i-07490f9592872db0a]

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.myinstance-1[0] will be destroyed
- resource "aws_instance" "myinstance-1" {
  - ami                                = "ami-00bb6a80f01f03502" -> null
  - arn                                = "arn:aws:ec2:ap-south-1:690511669638:instance/i-09a7f3f4d
24f77a06" -> null
  - associate_public_ip_address       = true -> null
  - availability_zone                  = "ap-south-1b" -> null
  - cpu_core_count                     = 1 -> null
  - cpu_threads_per_core               = 1 -> null
  - disable_api_stop                   = false -> null
  - disable_api_termination            = false -> null

```

```
Windows PowerShell
aws_instance.myinstance-1[2]: Destroying... [id=i-07490f9592872db0a]
aws_instance.myinstance-1[4]: Destroying... [id=i-0d5d660dd4b671835]
aws_instance.myinstance-1[3]: Destroying... [id=i-063e1704f5fd394cd]
aws_instance.myinstance-1[0]: Destroying... [id=i-09a7f3f4d24f77a06]
aws_instance.myinstance-1[1]: Destroying... [id=i-038bff8535167ba8c]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 10s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 10s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 10s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 10s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 10s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 20s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 20s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 20s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 20s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 20s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 30s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 30s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 30s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 30s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 30s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 40s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 40s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 40s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 40s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 40s elapsed]
```

```
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 30s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 30s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 40s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 40s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 40s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 40s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 40s elapsed]
aws_instance.myinstance-1[2]: Still destroying... [id=i-07490f9592872db0a, 50s elapsed]
aws_instance.myinstance-1[4]: Still destroying... [id=i-0d5d660dd4b671835, 50s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 50s elapsed]
aws_instance.myinstance-1[3]: Still destroying... [id=i-063e1704f5fd394cd, 50s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 50s elapsed]
aws_instance.myinstance-1[2]: Destruction complete after 50s
aws_instance.myinstance-1[4]: Destruction complete after 50s
aws_instance.myinstance-1[3]: Destruction complete after 50s
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 1m0s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 1m0s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 1m10s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 1m10s elapsed]
aws_instance.myinstance-1[0]: Still destroying... [id=i-09a7f3f4d24f77a06, 1m20s elapsed]
aws_instance.myinstance-1[1]: Still destroying... [id=i-038bff8535167ba8c, 1m20s elapsed]
aws_instance.myinstance-1[1]: Destruction complete after 1m21s
aws_instance.myinstance-1[0]: Destruction complete after 1m21s

Destroy complete! Resources: 5 destroyed.
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\lab6\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.