

Name – Utkarsh Rawat

Sap Id – 500124586

Batch – 3 DevOps

Lab Exercise 15– 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

```
terraform {  
  required_providers {  
    aws  
  }  
}  
  
provider "aws" {  
  region = "ap-  
south-1"  
  access_key = "your IAM  
access key"  
  secret_key = "your  
secret access key"  
}
```

instance.tf

```
resource "aws_instance" "myinstance-1" {  
  ami = var.myami  
  instance_type = var.my_instance_type  
  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 "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 terraform  
plan terraform apply -  
auto-approve
```

```
C:\Users\Rachit\Terraform-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.

```
C:\Users\Rachit\Terraform-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_instance.myinstance-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)
+   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)
+   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: 1 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.

```
C:\Users\Rachit\Terraform-Demo>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 will be created

```
C:\Users\Rachit\Terraform-Demo>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:

[illegible]

```
    + "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: 1 to add, 0 to change, 0 to destroy.
aws_instance.myinstance-1: Creating...
aws_instance.myinstance-1: Still creating... [00m10s elapsed]
aws_instance.myinstance-1: Creation complete after 13s [id=i-0b225ab1aa93df002]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Users\Rachit\Terraform-Demo>terraform destroy
aws_instance.myinstance-1: Refreshing state... [id=i-0b225ab1aa93df002]

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:
```

Observe how the region changes based on the variable override.

5. Clean Up:

After testing, you can clean up resources.

terraform destroy

```
C:\Users\Rachit\Terraform-Demo>terraform destroy
aws_instance.myinstance-1: Refreshing state... [id=i-0b225ab1aa93df002]

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 will be destroyed
- resource "aws_instance" "myinstance-1" {
  - ami                        = "ami-08718895af4dfa033" -> null
  - arn                      = "arn:aws:ec2:ap-south-1:236032563811:instance/i-0b225ab1aa93df002"
  - associate_public_ip_address = true -> null
  - availability_zone         = "ap-south-1a" -> null
  - cpu_core_count            = 1 -> null
  - cpu_threads_per_core      = 2 -> null
  - disable_api_stop          = false -> null
  - disable_api_termination   = false -> null
  - ebs_optimized              = false -> null
  - get_password_data         = false -> null
  - hibernation                = false -> null
  - id                        = "i-0b225ab1aa93df002" -> null
  - instance_initiated_shutdown_behavior = "stop" -> null
  - instance_state            = "running" -> null
  - instance_type              = "t3.micro" -> null
  - ipv6_address_count         = 0 -> null
  - ipv6_addresses            = [] -> null
  - monitoring                 = false -> null
  - placement_partition_number = 0 -> null
  - primary_network_interface_id = "eni-03852b8e7bf73aed4" -> null
  - private_dns                = "ip-172-31-36-43.ap-south-1.compute.internal" -> null
  - private_ip                 = "172.31.36.43" -> null
  - public_dns                 = "ec2-65-1-136-238.ap-south-1.compute.amazonaws.com" -> null
  - public_ip                  = "65.1.136.238" -> null
  - secondary_private_ips      = [] -> null
  - security_groups            = [
    - "default",
  ] -> null
  - source_dest_check          = true -> null
  - subnet_id                  = "subnet-0d8e2b31046160c71" -> null
  - tags                       = {
    - "Name" = "My Instance"
  } -> null
  - tags_all                   = {
    - "Name" = "My Instance"
  } -> null
  - tenancy                    = "default" -> null
```



```
    } -> null
- tenancy = "default" -> null
- user_data_replace_on_change = false -> null
- vpc_security_group_ids = [
  - "sg-058a24a110867ee92",
] -> null
# (8 unchanged attributes hidden)

- capacity_reservation_specification {
  - capacity_reservation_preference = "open" -> null
}

- cpu_options {
  - core_count = 1 -> null
  - threads_per_core = 2 -> null
  # (1 unchanged attribute hidden)
}

- credit_specification {
  - cpu_credits = "unlimited" -> null
}

- enclave_options {
  - enabled = false -> null
}

- maintenance_options {
  - auto_recovery = "default" -> null
}

- metadata_options {
  - http_endpoint = "enabled" -> null
  - http_protocol_ipv6 = "disabled" -> null
  - http_put_response_hop_limit = 2 -> null
  - http_tokens = "required" -> null
  - instance_metadata_tags = "disabled" -> null
}

- private_dns_name_options {
  - enable_resource_name_dns_a_record = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type = "ip-name" -> null
}

- root_block_device {
  - delete_on_termination = true -> null
  - device_name = "/dev/xvda" -> null
  - encrypted = false -> null
  - iops = 3000 -> null
}
```

```
- root_block_device {
  - delete_on_termination = true -> null
  - device_name           = "/dev/xvda" -> null
  - encrypted             = false -> null
  - iops                  = 3000 -> null
  - tags                  = {} -> null
  - throughput            = 125 -> null
  - volume_id             = "vol-0be9d1b2f76d25eb8" -> null
  - volume_size           = 8 -> null
  - volume_type           = "gp3" -> null
  # (1 unchanged attribute hidden)
}
}
```

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_instance.myinstance-1: Destroying... [id=i-0b225ab1aa93df002]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 00m10s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 00m20s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 00m30s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 00m40s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 00m50s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 01m00s elapsed]
aws_instance.myinstance-1: Still destroying... [id=i-0b225ab1aa93df002, 01m10s elapsed]
aws_instance.myinstance-1: Destruction complete after 1m11s

Destroy complete! Resources: 1 destroyed.

C:\Users\Rachit\Terraform-Demo>.terraform\

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.