

---

## LAB EXERCISE 8 – TERRAFORM MULTIPLE TFVARS FILES

### OBJECTIVE:

LEARN HOW TO USE MULTIPLE TFVARS FILES IN TERRAFORM FOR DIFFERENT ENVIRONMENTS.

### PREREQUISITES:

- TERRAFORM INSTALLED ON YOUR MACHINE.
- BASIC KNOWLEDGE OF TERRAFORM CONFIGURATION AND VARIABLES.

### STEPS:

#### 1. CREATE A TERRAFORM DIRECTORY:

```
mkdir terraform-multiple-tfvars
cd terraform-multiple-tfvars
```

- CREATE TERRAFORM CONFIGURATION FILES:
- CREATE A FILE NAMED MAIN.TF:

# MAIN.TF

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

resource "aws_instance" "example" {
  ami          = var.ami
  instance_type = var.instance_type
}
```

- CREATE A FILE NAMED VARIABLES.TF:

# VARIABLES.TF

```
variable "ami" {
```

```
    TYPE = STRING
}

VARIABLE "INSTANCE_TY" {

    TYPE = STRING
}
```

```
var.tf > variable "instance_type"
1   variable "ami" {
2   |     type = string
3   | }
4
5   variable "instance_type" {
6   |     type = string
7   | }
```

## 2. CREATE MULTIPLE TFVARS FILES:

- CREATE A FILE NAMED DEV.TFVARS:

# DEV.TFVARS

```
AMI          = "AMI-0123456789ABCDEFO"
INSTANCE_TYPE = "T2.MICRO"
```

- CREATE A FILE NAMED PROD.TFVARS:

# PROD.TFVARS

```
AMI          = "AMI-9876543210FEDCBAO"
INSTANCE_TYPE = "T2.LARGE"
```

```
dev.tfvars > instance_type
1   ami = "ami-02d26659fd82cf299"
2   instance_type = "t3.micro"
```

- IN THESE FILES, PROVIDE VALUES FOR THE VARIABLES BASED ON THE ENVIRONMENTS.

### 3. INITIALIZE AND APPLY FOR DEV ENVIRONMENT:

- RUN THE FOLLOWING TERRAFORM COMMANDS TO INITIALIZE AND APPLY THE CONFIGURATION FOR THE DEV ENVIRONMENT:

**TERRAFORM INIT**

**TERRAFORM APPLY -VAR-FILE=DEV.TFVARS**

```
[reckless@Arch Lab 17]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.30.0

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.
[reckless@Arch Lab 17]$ terraform apply -var-file=dev.tfvars

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.My-instance will be created
+ resource "aws_instance" "My-instance" {
  + ami                    = "ami-02d26659fd82cf299"
  + 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          = "t3.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)
```

```

+ 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_all                  = (known after apply)
+ 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.

Do you want to perform these actions?

Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

aws\_instance.My-instance: Creating...

aws\_instance.My-instance: Still creating... [00m10s elapsed]

aws\_instance.My-instance: Creation complete after 14s [id=i-00dc74b5cd1360efc]

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

[reckless@Arch Lab 17]\$

Instances (2) Info

Last updated  
1 minute ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

All states ▾

< 1 > ⚙

<input type="checkbox"/>	Name <a href="#">🔗</a> ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS ▾	Public IPv
<input type="checkbox"/>		i-06b2a0fa137c395f	⏻ Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	-	<a href="#">View alarms +</a>	ap-south-1a	-	-
<input type="checkbox"/>		i-00dc74b5cd1360efc	🟢 Running <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	🔄 Initializing	<a href="#">View alarms +</a>	ap-south-1a	ec2-13-235-0-102.ap-s...	13.235.0.1

#### 4. INITIALIZE AND APPLY FOR PROD ENVIRONMENT:

- RUN THE FOLLOWING TERRAFORM COMMANDS TO INITIALIZE AND APPLY THE CONFIGURATION FOR THE PROD ENVIRONMENT:

## TERRAFORM INIT

## TERRAFORM APPLY -VAR-FILE=PROD.TFVARS

```
[reckless@Arch Lab 17]$ terraform apply -var-file=prod.tfvars
aws_instance.My-instance: Refreshing state... [id=i-00dc74b5cd1360efc]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

# aws_instance.My-instance will be updated in-place
~ resource "aws_instance" "My-instance" {
  id              = "i-00dc74b5cd1360efc"
  ~ instance_type = "t3.micro" -> "t3.small"
  tags            = {}
  # (38 unchanged attributes hidden)

  # (8 unchanged blocks hidden)
}

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

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.My-instance: Modifying... [id=i-00dc74b5cd1360efc]
aws_instance.My-instance: Still modifying... [id=i-00dc74b5cd1360efc, 00m10s elapsed]
aws_instance.My-instance: Still modifying... [id=i-00dc74b5cd1360efc, 00m20s elapsed]
aws_instance.My-instance: Still modifying... [id=i-00dc74b5cd1360efc, 00m30s elapsed]
aws_instance.My-instance: Modifications complete after 34s [id=i-00dc74b5cd1360efc]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
[reckless@Arch Lab 17]$
```

Instances (2) <a href="#">Info</a>									
Find Instance by attribute or tag (case-sensitive)				All states					
<input type="checkbox"/>	Name <a href="#">↗</a>	Instance ID	Instance state <a href="#">↕</a>	Instance type <a href="#">↕</a>	Status check	Alarm status	Availability Zone <a href="#">↕</a>	Public IPv4 DNS <a href="#">↕</a>	Public IPv...
<input type="checkbox"/>		i-06b2a0f0a137c395f	Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	–	<a href="#">View alarms +</a>	ap-south-1a	–	–
<input type="checkbox"/>		i-00dc74b5cd1360efc	Running <a href="#">🔍</a> <a href="#">🔍</a>	t3.small	🕒 Initializing	<a href="#">View alarms +</a>	ap-south-1a	ec2-3-109-139-80.ap-s...	3.109.139.

## 5. TEST AND VERIFY:

- OBSERVE HOW DIFFERENT TFVARS FILES ARE USED TO SET VARIABLE VALUES FOR DIFFERENT ENVIRONMENTS DURING THE APPLY PROCESS.
- ACCESS THE AWS MANAGEMENT CONSOLE OR USE THE AWS CLI TO VERIFY THE CREATION OF RESOURCES IN THE SPECIFIED REGIONS AND INSTANCE TYPES.

## 6. CLEAN UP:

- AFTER TESTING, YOU CAN CLEAN UP RESOURCES:

```
TERRAFORM DESTROY -VAR-FILE=DEV.TFVARS
TERRAFORM DESTROY -VAR-FILE=PROD.TFVARS
```

- CONFIRM THE DESTRUCTION BY TYPING YES.

```
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.My-instance: Destroying... [id=i-00dc74b5cd1360efc]
aws_instance.My-instance: Still destroying... [id=i-00dc74b5cd1360efc, 00m10s elapsed]
aws_instance.My-instance: Still destroying... [id=i-00dc74b5cd1360efc, 00m20s elapsed]
aws_instance.My-instance: Still destroying... [id=i-00dc74b5cd1360efc, 00m30s elapsed]
aws_instance.My-instance: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.
```

7.

```
[reckless@Arch Lab 17]$ terraform destroy -var-file=prod.tfvars

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.
```

Instances (2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv6
<input type="checkbox"/>		i-06b2a0f0a137c395f	Terminated	t3.micro	-	View alarms +	ap-south-1a	-	-
<input type="checkbox"/>		i-00dc74b5cd1360efc	Terminated	t3.small	-	View alarms +	ap-south-1a	-	-

## CONCLUSION:

THIS LAB EXERCISE DEMONSTRATES HOW TO USE MULTIPLE TFVARS FILES IN TERRAFORM TO MANAGE VARIABLE VALUES FOR DIFFERENT ENVIRONMENTS. IT ALLOWS YOU TO MAINTAIN SEPARATE CONFIGURATION FILES FOR DIFFERENT ENVIRONMENTS, MAKING IT EASIER TO MANAGE AND MAINTAIN YOUR INFRASTRUCTURE CODE. EXPERIMENT WITH DIFFERENT VALUES IN THE DEV.TFVARS AND PROD.TFVARS FILES TO OBSERVE HOW THEY IMPACT THE INFRASTRUCTURE PROVISIONING PROCESS FOR EACH ENVIRONMENT.