

LAB-6

Terraform Multiple tfvars Files

Step 1: Create dev.tfvars and prod.tfvars

```
main.tf x instance.tf var.tf

main.tf > provider "aws"
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.32.1"
6     }
7   }
8 }
9 provider "aws" {
10   region = "ap-south-1"
11   access_key = "
12   secret_key = "
13 }
```

```
main.tf instance.tf x var.tf prod.tfvars dev.tfvars

instance.tf > resource "aws_instance" "lab4-1" > tags > Name
1 resource "aws_instance" "lab4-1" {
2   instance_type = var.instance_typ
3   ami = var.ami_id
4   count = 1
5   tags = {
6     Name = "lab4-b3-2"
7   }
8 }
```

```
main.tf instance.tf var.tf prod.tfvars dev.tfvars x

dev.tfvars > ami_id
1 instance_typ = "t2.micro"
2 ami_id = "
```

```
main.tf  instance.tf  var.tf  dev.tfvars X
dev.tfvars > ami_id
1  instance_type = "t2.micro"
2  ami_id = "
```

Step 2: Now run terraform cycle

```
arnim_taliyan@device:~/Desktop/terraform$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.32.1

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.

```
arnim_taliyan@device:~/Desktop/terraform$
```

```
arnim_taliyan@device:~/Desktop/terraform$ terraform validate
Success! The configuration is valid.
```

Step 3: To run terraform plan we need to use -var-file=dev.tfvars or -var-file=prod.tfvars

```
arnim_taliyan@device:~/Desktop/terraform$ terraform plan -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.lab4-1[0] will be created
+ resource "aws_instance" "lab4-1" {
  + ami                    = "ami-03f4878755434977f"
  + 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)
```

```

arnim_taliyan@device:~/Desktop/terraform$ terraform plan -var-file=prod.tfvars
aws_instance.lab4-1[0]: Refreshing state... [id=i-0aab56cc725a3d034]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

# aws_instance.lab4-1[0] must be replaced
-/+ resource "aws_instance" "lab4-1" {
  ~ ami                    = "ami-03f4878755434977f" -> "ami-05a5bb48beb785bf1" # forces replacement
  ~ arn                    = "arn:aws:ec2:ap-south-1:533266967718:instance/i-0aab56cc725a3d034" -> (known after apply)
  ~ associate_public_ip_address = true -> (known after apply)
  ~ availability_zone         = "ap-south-1a" -> (known after apply)
  ~ cpu_core_count            = 1 -> (known after apply)
  ~ cpu_threads_per_core      = 1 -> (known after apply)
  ~ disable_api_stop          = false -> (known after apply)
  ~ disable_api_termination    = false -> (known after apply)
  ~ ebs_optimized             = false -> (known after apply)
  ~ hibernation                = false -> null
  + host_id                  = (known after apply)
  + host_resource_group_arn   = (known after apply)
  + iam_instance_profile      = (known after apply)
  ~ id                       = "i-0aab56cc725a3d034" -> (known after apply)
  ~ instance_initiated_shutdown_behavior = "stop" -> (known after apply)
  ~ instance_lifecycle        = (known after apply)
  ~ instance_state            = "running" -> (known after apply)
  ~ ipv6_address_count         = 0 -> (known after apply)
  ~ ipv6_addresses            = [] -> (known after apply)
  + key_name                  = (known after apply)
  ~ monitoring                = false -> (known after apply)
  + outpost_arn               = (known after apply)
  + password_data              = (known after apply)
  + placement_group           = (known after apply)
  ~ placement_partition_number = 0 -> (known after apply)
  ~ primary_network_interface_id = "eni-0a551d62455cf7a55" -> (known after apply)
  ~ private_dns                = "ip-172-31-41-163.ap-south-1.compute.internal" -> (known after apply)

```

Step 4: To run terraform apply and destroy we need to use -var-file=dev.tfvars or -var-file=prod.tfvars

```

arnim_taliyan@device:~/Desktop/terraform$ 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.lab4-1[0] will be created
+ resource "aws_instance" "lab4-1" {
  + ami                    = "ami-03f4878755434977f"
  + 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)

```

The screenshot shows the AWS Management Console interface. On the left, the 'Instances' section is expanded. The main area displays the 'Instances (1)' page. A search bar is present with the text 'Find Instance by attribute or tag (case-sensitive)'. Below the search bar, a table lists the instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One instance is listed: 'lab4-b3-1' with ID 'i-0aab56cc725a3d034', state 'Running', type 't2.micro', and availability zone 'ap-south-1a'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
lab4-b3-1	i-0aab56cc725a3d034	Running	t2.micro	ap-south-1a

```
arnim_taliyan@device:~/Desktop/terraform$ terraform apply -var-file=prod.tfvars
aws_instance.lab4-1[0]: Refreshing state... [id=i-0aab56cc725a3d034]
```

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

- /+ destroy and then create replacement

Terraform will perform the following actions:

```
aws_instance.lab4-1[0] must be replaced
/+ resource "aws_instance" "lab4-1" {
  - ami
  - arn
  - associate_public_ip_address
  - availability_zone
  - cpu_core_count
  - cpu_threads_per_core
  - disable_api_stop
  - disable_api_termination
  - ebs_optimized
  - hibernation
+ host_id
+ host_resource_group_arn
+ iam_instance_profile
- id
- instance_initiated_shutdown_behavior
+ instance_lifecycle
- instance_state
- ipv6_address_count
- ipv6_addresses
+ key_name
+ monitoring
+ outpost_arn
+ password_data
+ placement_group
+ placement_partition_number
+ primary_network_interface_id
+ private_dns
  = "ami-03f4878755434977f" -> "ami-05a5bb48beb785bf1" # forces replacement
  = "arn:aws:ec2:ap-south-1:533266967718:instance/i-0aab56cc725a3d034" -> (known after apply)
  = true -> (known after apply)
  = "ap-south-1a" -> (known after apply)
  = 1 -> (known after apply)
  = 1 -> (known after apply)
  = false -> (known after apply)
  = false -> (known after apply)
  = false -> (known after apply)
  = false -> null
  = (known after apply)
  = (known after apply)
  = (known after apply)
  = "i-0aab56cc725a3d034" -> (known after apply)
  = "stop" -> (known after apply)
  = (known after apply)
  = "running" -> (known after apply)
  = 0 -> (known after apply)
  = [] -> (known after apply)
  = (known after apply)
  = (known after apply)
  = false -> (known after apply)
  = (known after apply)
  = (known after apply)
  = (known after apply)
  = 0 -> (known after apply)
  = "eni-0a551d62455cf7a55" -> (known after apply)
  = "ip-172-31-41-163.ap-south-1.compute.internal" -> (known after apply)
```

The screenshot shows the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, 'Services' menu, a search bar, and regional information (Mumbai, ArnimTaliyan). The left sidebar shows the 'Instances' section selected. The main content area is titled 'Instances (2)' and includes a search bar, a filter dropdown set to 'Any state', and a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. Two instances are listed: 'lab4-b3-2' (Running) and 'lab4-b3-1' (Terminated). The 'lab4-b3-2' instance is highlighted in blue.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
lab4-b3-2	I-Oa206e4fd9ce7bb26	Running	t2.micro	...	View alarms	ap-south-1
lab4-b3-1	I-Oaab56cc725a3d034	Terminated	t2.micro	...	View alarms	ap-south-1

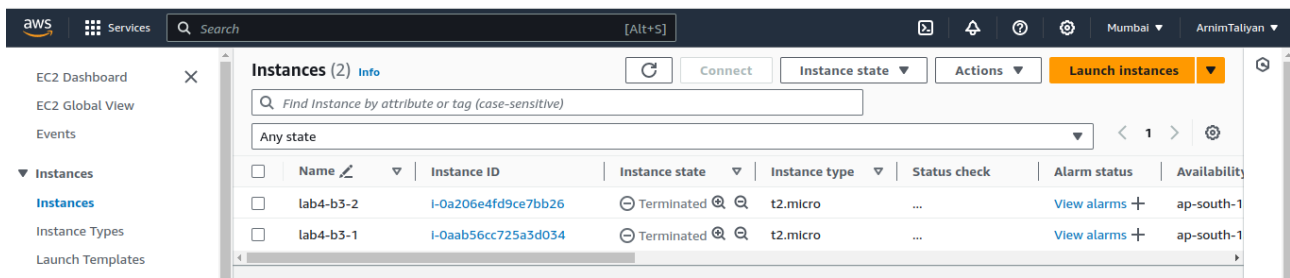
```
arnim_taliyan@device:~/Desktop/terraform$ terraform destroy -var-file=prod.tfvars
aws_instance.lab4-1[0]: Refreshing state... [id=i-0a206e4fd9ce7bb26]
```

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.lab4-1[0] will be destroyed
- resource "aws_instance" "lab4-1" {
  - ami                      = "ami-05a5bb48beb785bf1" -> null
  - arn                     = "arn:aws:ec2:ap-south-1:533266967718:instance/i-0a206e4fd9ce7bb26" -> null
  - associate_public_ip_address = true -> null
  - availability_zone        = "ap-south-1a" -> null
  - cpu_core_count           = 1 -> null
  - cpu_threads_per_core     = 1 -> 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-0a206e4fd9ce7bb26" -> null
  - instance_initiated_shutdown_behavior = "stop" -> null
  - instance_state           = "running" -> null
  - instance_type            = "t2.micro" -> null
  - ipv6_address_count        = 0 -> null
  - ipv6_addresses           = [] -> null
  - monitoring               = false -> null
  - placement_partition_number = 0 -> null
  - primary_network_interface_id = "eni-0fcdbabc3cf17a8d8" -> null
  - private_dns              = "ip-172-31-33-94.ap-south-1.compute.internal" -> null
  - private_ip               = "172.31.33.94" -> null
  - public_dns               = "ec2-13-233-216-184.ap-south-1.compute.amazonaws.com" -> null
  - public_ip                = "13.233.216.184" -> null
  - secondary_private_ips     = [] -> null
  - security_groups          = [
    - "default",
  ] -> null
}
```



When we run `terraform apply -var-file=prod.tfvars` previously created `terraform apply -var-file=dev.tfvars` automatically destroy.