

Lab Exercise 6– Terraform Multiple tfvars Files

Objective:

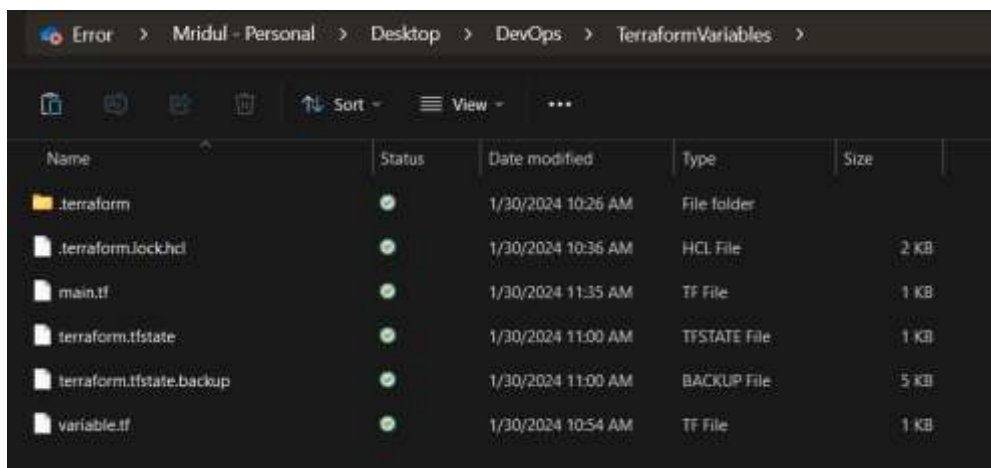
Learn how to use multiple tfvars files in Terraform for different environments.

Prerequisites:

- Terraform installed on your machine.
- Basic knowledge of Terraform configuration and variables.

Steps:

1. Create a Terraform Directory:



Name	Status	Date modified	Type	Size
.terraform	●	1/30/2024 10:26 AM	File folder	
.terraform.lock.hcl	●	1/30/2024 10:36 AM	HCL File	2 KB
main.tf	●	1/30/2024 11:35 AM	TF File	1 KB
terraform.tfstate	●	1/30/2024 11:00 AM	TFSTATE File	1 KB
terraform.tfstate.backup	●	1/30/2024 11:00 AM	BACKUP File	5 KB
variable.tf	●	1/30/2024 10:54 AM	TF File	1 KB

Create Terraform Configuration Files:

- Create a file named main.tf:

main.tf

```

main.tf X
main.tf > provider "aws"
1 resource "aws_instance" "example" {
2   ami = var.ami
3   instance_type = var.instance_ty
4 }
5
6 terraform {
7   required_providers {
8     aws = {
9       source = "hashicorp/aws"
10      version = "5.31.0"
11    }
12  }
13 }
14
15 provider "aws" {
16   region = var.region_ec2
17   access_key = "AKIAII2LIAJGSHG999HP"
18   secret_key = "FgSojIkGskuNVG1hPhu4Ky413zX1/XG/6zeQr6k/"
19 }

```

- Create a file named variables.tf:

variables.tf

```

variable.tf X
variable.tf > variable "region_ec2"
1 variable "ami" {
2   description = "AMI ID"
3   default = "ami-03f4878755434977f"
4 }
5
6 variable "instance_ty" {
7   description = "ec2-instance"
8   default = "t2.micro"
9 }
10
11 variable "region_ec2" {
12   description = "ec2-region"
13   default = "ap-south-1"
14 }

```

2. Create Multiple tfvars Files:

- Create a file named dev.tfvars:

dev.tfvars

```

variable.tf  dev.tfvars X
dev.tfvars > ...
1 region = "ap-south-1"
2 ami = "ami-03f4878755434977f"
3 instance_type = "t2.micro"

```

- Create a file named prod.tfvars: #

prod.tfvars

```

variable.tf  dev.tfvars  prod.tfvars X

prod.tfvars > ...
1   region = "ap-south-1"
2   ami = "ami-0d63de463e6604d0a"
3   instance_type = "t2.large" |

```

3. Initialize and Apply for Dev Environment:

```

PS C:\Users\Devi\OneDrive\Desktop\DevOps\TerraformVariables> 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.example will be created
+ resource "aws_instance" "example" {
  + ami                    = "ami-03f4678755434977f"
  + 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)
}

```

```

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

Warning: Value for undeclared variable
The root module does not declare a variable named "region" but a value was found in file "dev.tfvars". If you meant to use this value, add a "variable" block to the configuration.

To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organisation. To reduce the verbosity of these warnings, use the -compact-warnings option.

Warning: Value for undeclared variable
The root module does not declare a variable named "instance_type" but a value was found in file "dev.tfvars". If you meant to use this value, add a "variable" block to the configuration.

To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organisation. To reduce the verbosity of these warnings, use the -compact-warnings option.

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.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 32s [id=i-0890c38676732884]

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

```

4. Initialize and Apply for Prod Environment:

```
PS C:\Users\Delil\OneDrive\Desktop\DevOps\Terraform\Variables> terraform apply -var-file="prod.tfvars"
aws_instance.example: Refreshing state... [id=i-089e2c386767328b6]

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.example must be replaced
  /- resource "aws_instance" "example" {
    ~ ami                      = "ami-09f4878755434977f" -> "ami-0d61de463e669408e" # forces replacement
    ~ arm
    ~ 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_arm    = (known after apply)
    ~ iam_instance_profile       = (known after apply)
    ~ id                         = "i-089e2c386767328b6" -> (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-030bcbf8r3ierfb09" -> (known after apply)
  }
```

```
Warning: Value for undeclared variable

The root module does not declare a variable named "instance_type" but a value was found in file "prod.tfvars". If you meant to use this
value, add a "variable" block to the configuration.

To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your
organization. To reduce the verbosity of these warnings, use the -compact-warnings option.

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.example: Destroying... [id=i-089e2c386767328b6]
aws_instance.example: Still destroying... [id=i-089e2c386767328b6, 10s elapsed]
aws_instance.example: Still destroying... [id=i-089e2c386767328b6, 20s elapsed]
aws_instance.example: Still destroying... [id=i-089e2c386767328b6, 30s elapsed]
aws_instance.example: Destruction complete after 32s
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 33s [id=i-4aa580ad613d4917a]

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

5. Test and Verify:



6. Clean Up:

```
PS C:\Users\ DELL\OneDrive\Desktop\DevOps\TerraformVariables> terraform destroy -var-file="dev.tfvars"
aws_instance.example: Refreshing state... [id=i-8aa180a013d0917a]

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.

Warning: Value for undeclared variable
The root module does not declare a variable named "instance_type" but a value was found in file "dev.tfvars". If you meant to use this value, add a "variable" block to the configuration.

To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organization. To reduce the verbosity of these warnings, use the -compact-warnings option.

Warning: Value for undeclared variable
The root module does not declare a variable named "region" but a value was found in file "dev.tfvars". If you meant to use this value, add a "variable" block to the configuration.

To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organization. To reduce the verbosity of these warnings, use the -compact-warnings option.

Destroy complete! Resources: 0 destroyed.
```

```
PS C:\Users\ DELL\OneDrive\Desktop\DevOps\TerraformVariables> terraform destroy -var-file="prod.tfvars"
aws_instance.example: Refreshing state... [id=i-8c0ba8a0947e35d4]

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.example will be destroyed
- resource "aws_instance" "example" {
  - ami                               = "ami-8051de461ec084d8a" -> null
  - arn                               = "arn:aws:ec2:ap-south-1:557425583821:instance/i-8c0ba8a0947e35d4" -> 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-8c0ba8a0947e35d4" -> null
  - instance_initiated_shutdown_behavior = "stop" -> null
  - instance_state                    = "running" -> null
  - instance_type                     = "t2.micro" -> null
  - ipv4_addresses_count              = 0 -> null
  - ipv4_addresses                    = [] -> null
  - monitoring                        = false -> null
  - placement_partition_number        = 0 -> null
  - primary_network_interface_id      = "eni-824ac1cabf6125281" -> null
  - private_dns                       = "ip-172-31-37-36.ap-south-1.compute.internal" -> null
  - private_ip                        = "172.31.37.36" -> null
  - public_dns                        = "ec2-13-233-125-46.ap-south-1.compute.amazonaws.com" -> null
  - public_ip                         = "13.233.125.46" -> null
```