

Lab Exercise 6

Terraform Multiple tfvars Files

1. Create a Terraform Directory:

```
PS E:\> mkdir terraform-multiple-tfvars

Directory: E:\

Mode                LastWriteTime         Length Name
----                -
d-----          06-02-2024    21:57          terraform-multiple-tfvars

PS E:\> cd .\terraform-multiple-tfvars\
PS E:\terraform-multiple-tfvars> |
```

2. Create a Terraform Configuration File:

```
variables.tf  main.tf  instance.tf

main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.34.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = var.region
12   access_key = "AKIAVRUVV37F66GBPTT4"
13   secret_key = "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"
14 }
```

```
instances.tf
1 resource "aws_instance" "My-Instance" {
2   instance_type = var.instance_type
3   ami = var.ami
4   count = 1
5   tags = {
6     Name = "UPES-EC2-INSTANCE"
7   }
8 }
```

```

variables.tf
1  variable "ami" {
2      description = "AWS ami"
3      default = "ami-05fb0b8c1424f266b"
4  }
5  variable "region" {
6      description = "AWS region"
7      default = "us-east-2"
8  }
9  variable "instance_type" {
10     description = "AWS instance Type"
11     default = "t2.micro"
12 }

```

3. Create Multiple tfvars Files:

```

dev.tfvars
1  //Windows
2  region = "us-east-2"
3  ami = "ami-094aa6728b151e05a"
4  instance_type = "t2.micro"

```

```

prod.tfvars
1  //Amazon Linux
2  region = "us-east-2"
3  ami = "ami-0866a04d72a1f5479"
4  instance_type = "t2.micro"

```

4. Initialize and Apply for Dev Environment:

```

PS E:\terraform-multiple-tfvars> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.34.0"...
- Installing hashicorp/aws v5.34.0...
- Installed hashicorp/aws v5.34.0 (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 E:\terraform-multiple-tfvars> 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[0] will be created

```
+ resource "aws_instance" "My-Instance" {
  + ami                  = "ami-094aa6728b151e05a"
  + 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)
```

Instances (1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

Instance state = running

✕

Clear filters

Any state

▼

Refresh

Connect

Instance state ▼

Actions ▼

Launch instances ▼

< 1 > ⚙

4 ... ▼	Elastic IP	IPv6 IPs ▼	Monitoring ▼	Security group name ▼	Key name ▼	Launch time ▼	Platfor... ▼
44	-	-	disabled	default	-	2024/02/06 22:15 GMT+5:30	Windows

5. Initialize and Apply for Prod Environment:

```
PS E:\terraform-multiple-tfvars> terraform apply -var-file="prod.tfvars"
aws_instance.My-Instance[0]: Refreshing state... [id=i-08bf74654ad5c5b43]
```

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.My-Instance[0] must be replaced
-/+ resource "aws_instance" "My-Instance" {
  ~ ami                  = "ami-094aa6728b151e05a" -> "ami-0866a04d72a1f5479" # forces replacement
  ~ arn                  = "arn:aws:ec2:us-east-2:381492256715:instance/i-08bf74654ad5c5b43" -> (known after apply)
  ~ associate_public_ip_address = true -> (known after apply)
  ~ availability_zone      = "us-east-2a" -> (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-08bf74654ad5c5b43" -> (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-08b77dac525e4a898" -> (known after apply)
  ~ private_dns              = "ip-172-31-12-74.us-east-2.compute.internal" -> (known after apply)
```

Instances (1)

Info

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

Any state ▾

Instance state = running

X

Clear filters

<

1

>

4 ... ▾	Elastic IP	IPv6 IPs ▾	Monitoring ▾	Security group name ▾	Key name ▾	Launch time ▾	Platfor... ▾
1.185	-	-	disabled	default	-	2024/02/06 22:20 GMT+5:30	Linux/UNIX
4							

6. Clean Up:

```

- 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
- tags                  = {} -> null
- throughput            = 125 -> null
- volume_id             = "vol-09ae41d80f546663b" -> null
- volume_size           = 8 -> null
- volume_type           = "gp3" -> null
}
}

```

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[0]: Destroying... [id=i-01d77275fbf3885d4]
aws_instance.My-Instance[0]: Still destroying... [id=i-01d77275fbf3885d4, 10s elapsed]
aws_instance.My-Instance[0]: Still destroying... [id=i-01d77275fbf3885d4, 20s elapsed]
aws_instance.My-Instance[0]: Still destroying... [id=i-01d77275fbf3885d4, 30s elapsed]
aws_instance.My-Instance[0]: Destruction complete after 33s

```

Destroy complete! Resources: 1 destroyed.

PS E:\terraform-multiple-tfvars> |