SPCM LAB

Gopika Jhanwar 500093662 R2142210318 Btech cse devops B2

Lab Exercise 6– Terraform Multiple tfvars Files

- Create a Terraform Directory: mkdir terraform-multiple-tfvars cd terraform-multiple-tfvars
 - Create a file named main.tf:

Create a file named variables.tf:

```
terraform-cli-variables >  variable.tf

1  variable "region" {
2  description = "AWS region"
3  default = "us-west-2"
4  }
5
6  variable "ami" {
7  description = "AMI ID"
8  default = "ami-052c9ea013e6e3567"
9  }
10
11  variable "instance_type" {
12  description = "EC2 Instance Type"
13  default = "t2.small"
14  }
```

- 2. Create Multiple tfvars Files:
- Create a file named dev.tfvars:

Create a file named prod.tfvars:

```
terraform-multiple-tfvars > ₩ prod.tfvars

1 region = "us-east-1"

2 ami = "ami-0440d3b780d96b29d"

3 instance_type = "t2.large"

4
```

3. Initialize and Apply for Dev Environment:

terraform init

```
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv ars> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.38.0

Terraform has been successfully initialized!
```

terraform apply -var-file=dev.tfvars

```
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.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 37s [id=i-0c2d4b04333a99
612]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv
```

4. Initialize and Apply for Prod Environment: terraform init

```
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv
ars> terraform init

Initializing the backend...

Initializing provider plugins...

- Reusing previous version of hashicorp/aws from the dependency lock file

- Using previously-installed hashicorp/aws v5.38.0

Terraform has been successfully initialized!
```

terraform apply -var-file=prod.tfvars

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

Do you want to perform these actions?

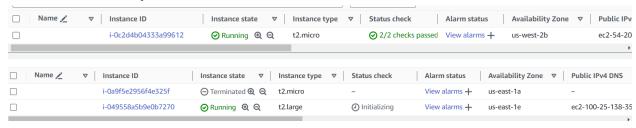
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Creation complete after 27s [id=i-049558a5b9e0b7
270]

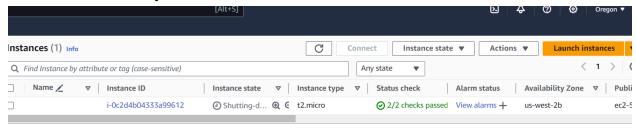
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv
```

5. Test and Verify:



6. Clean Up:

terraform destroy -var-file=dev.tfvars



```
S D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv
rs> terraform destroy -var-file="dev.tfvars"
ws_instance.example: Refreshing state... [id=i-049558a5b9e0b7270]

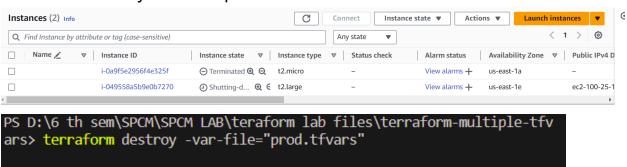
to changes. No objects need to be destroyed.

Sither you have not created any objects yet or the existing objects
were already deleted outside of Terraform.

Testroy complete! Resources: 0 destroyed.

S D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv
rs>
```

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.

Destroy complete! Resources: 0 destroyed.

PS D:\6 th sem\SPCM\SPCM LAB\teraform lab files\terraform-multiple-tfv