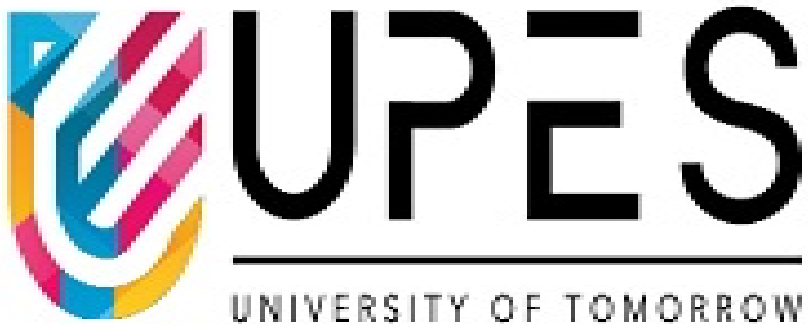


SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT

LAB FILE

NAME: SMRITI RAI
SAP ID: 500096396
BATCH: B3
SUBMITTED TO: Dr. Hitesh Kumar Sharma
SEMESTER: VI
ENROLLMENT NO.: R2142211212

EXPERIMENT 6:

Terraform Multiple tfvars Files

1. Create a main.tf file
2. Create a file named variables.tf.
3. Create a dev.tfvars file
4. Create a prod.tfvars file
5. Run the command terraform init to initialise the environment

```
D:\docss\UPES\sem 6\SPCM Lab\lab 6>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.32.1"...
- Installing hashicorp/aws v5.32.1...
- Installed hashicorp/aws v5.32.1 (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.
```

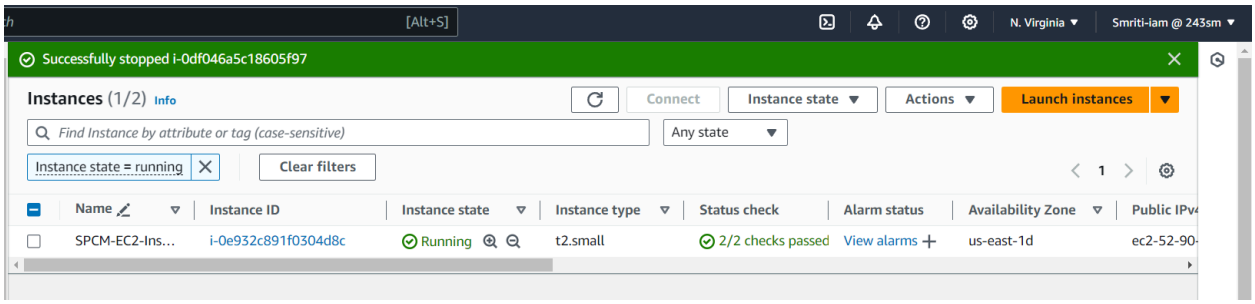
6. Then run the terraform apply command with configuration to apply the changes.
7. Run terraform apply with dev.tfvars

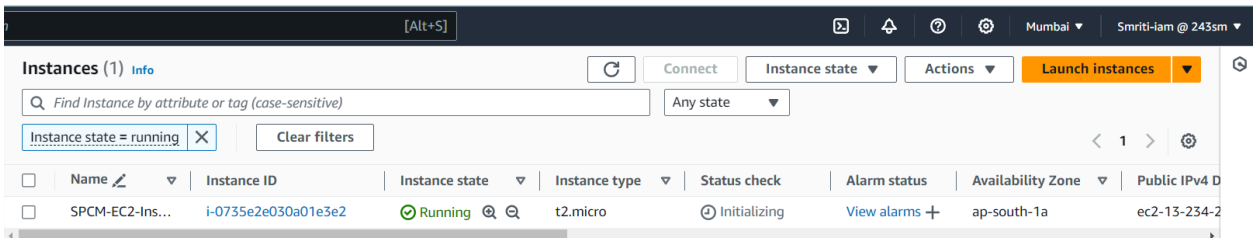
```
D:\docss\UPES\sem 6\SPCM Lab\lab 6>terraform init

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.

D:\docss\UPES\sem 6\SPCM Lab\lab 6>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:
```

```
          + "Name" = "SPCM-EC2-Instance"
        }
      + 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)
    }

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.Smriti-ec2: Creating...
aws_instance.Smriti-ec2: Still creating... [10s elapsed]
aws_instance.Smriti-ec2: Still creating... [20s elapsed]
aws_instance.Smriti-ec2: Still creating... [30s elapsed]
aws_instance.Smriti-ec2: Creation complete after 37s [id=i-0e932c891f0304d8c]

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

D:\docss\UPES\sem 6\SPCM Lab\lab 6>
```

8. Run terraform init followed by terraform apply with prod.tfvars

9. Open AWS console to verify the creation of your resources.





10. Run the terraform destroy command to shut down your resources.

```
D:\docss\UPES\sem 6\SPCM Lab\lab 6>terraform destroy -var-file=prod.tfvars
aws_instance.Smriti-ec2: Refreshing state... [id=i-0735e2e030a01e3e2]

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.Smriti-ec2 will be destroyed
  - resource "aws_instance" "Smriti-ec2" {
      - ami                                  = "ami-09b9e25b6db1d130c" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:667769287100:i
nstance/i-0735e2e030a01e3e2" -> 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-0735e2e030a01e3e2" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
```

```
  Enter a value: yes

aws_instance.Smriti-ec2: Destroying... [id=i-0735e2e030a01e3e2]
aws_instance.Smriti-ec2: Still destroying... [id=i-0735e2e030a01e3e2, 10s elapsed]
aws_instance.Smriti-ec2: Still destroying... [id=i-0735e2e030a01e3e2, 20s elapsed]
aws_instance.Smriti-ec2: Still destroying... [id=i-0735e2e030a01e3e2, 30s elapsed]
aws_instance.Smriti-ec2: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.

D:\docss\UPES\sem 6\SPCM Lab\lab 6>
```