EXPERIMENT – 5

Name: - Shashwat. Dnyaneshwar Kamdi
Batch – 2 [DevOps Non-Hons]
SAP ID- 500092140
Subject – System Provisioning and Configuration Management Lab

Aim: Terraform Variables with command Line Arguments

1] Create a Terraform Configuration File (main.tf)

2] Create new file name as “variables.tf”’

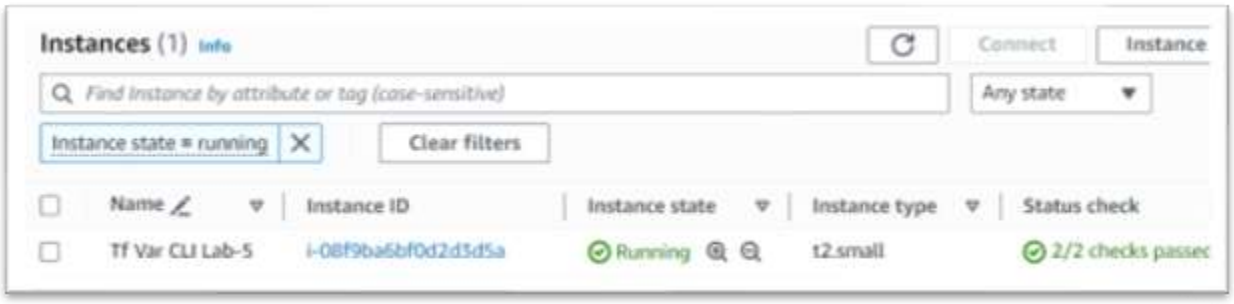
3] Initialize Terraform using command “terraform init”

```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\Terraform Variables -4> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.
```

4] Apply command with command line arguments to set variables values using command “terraform apply -var 'region=us-west-2' -var "ami=ami-008fe2fc65df48dac" -var 'instance_type=t2.small'”

```
PS F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts\Terraform Variables with CLI-5> terraform apply -var 'region=us-west-2' -var "ami=ami-008fe2fc65df48da
c" -var 'instance_type=t2.small'

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.Lab5 will be created
  + resource "aws_instance" "Lab5" {
      + ami                                  = "ami-008fe2fc65df48dac"
      + 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.small"
      + 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)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Tf Var CLI Lab-5"
        }
      + tags_all                             = {
          + "Name" = "Tf Var CLI Lab-5"
        }
      + 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.Lab5: Creating...
aws_instance.Lab5: Still creating... [10s elapsed]
aws_instance.Lab5: Still creating... [20s elapsed]
aws_instance.Lab5: Still creating... [30s elapsed]
aws_instance.Lab5: Creation complete after 37s [id=i-08f9ba6bf0d2d3d5a]

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

5] Verify Resources on AWS Management Console.



6] Cleanup Resources using command “terraform destroy”