System Provisioning and Configuration Management Lab

Siddhant Singh

500095375

Batch - 3

Experiment 4

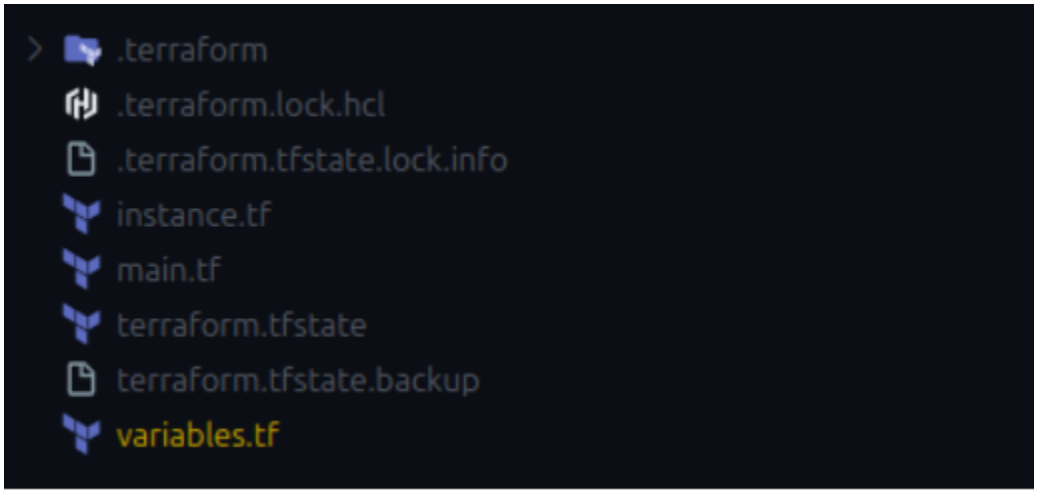
Terraform Variables

Aim

Learn how to define and use variables in Terraform configuration

Steps

1. Create a main file & terraform configuration file for EC2 Instance (instance.tf)



main.tf

```
terraform {
  required_providers {
      aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

provider "aws" {
  region = "ap-south-1"
  access_key = "AKIA4MKYQZSREOVYM3VG"
  secret_key = "+uKLPNQNjQaIufSeXioDcBCb+tUejJ5xrTmlS5HpdL"
}
```

instance.tf

```
resource "aws_instance" "Ayroid-ec2" {
    instance_type = "t2.micro"
    ami = "ami-03f4878755434977f"
    count = 1

    tags = {
        Name = "Exp4-Instance"
    }
}
```

2. Open a new file named variables.tf. Define variables for region, ami, secret_key, access_key and instance_type.

variables.tf

```
variable region {
  type        = string
  default     = "ap-south-1"
  description = "AWS Region"
}

variable "ami"{
    type = string
    default = "ami-03f4878755434977f"
    description = "AMI ID"
}

variable "instance_type"{
    type = string
    default = "t2.micro"
    description = "Instance Type"
}
```

3. Modify main.tf & instance.tf to use the variables.

instance.tf

```
resource "aws_instance" "Ayroid-ec2" {
    instance_type = var.instance_type
    ami = var.ami
    count = 1

    tags = {
        Name = "Exp4-Instance"
    }
}
```

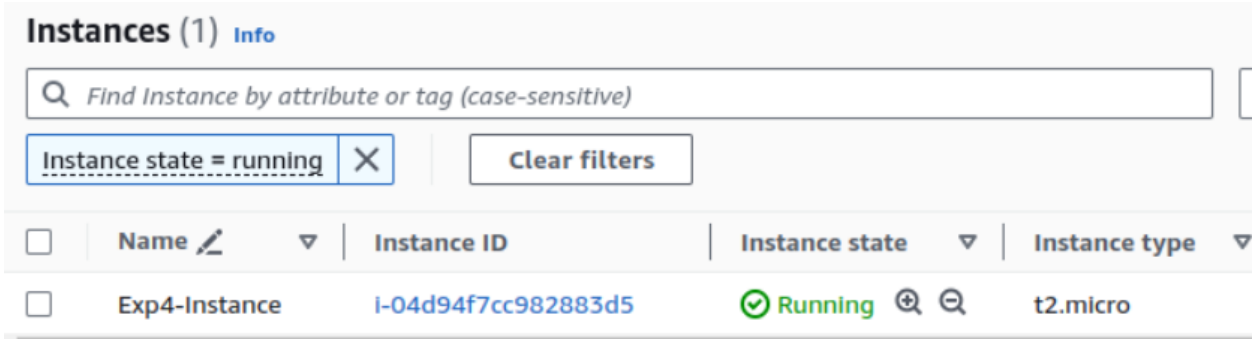
main.tf

```
terraform {
  required_providers {
    aws = {
    source = "hashicorp/aws"
    version = "5.31.0"
    }
  }
}

provider "aws" {
  region = var.region
  access_key = var.access_key
  secret_key = var.secret_key
}
```

4. Run the following Terraform commands to initialize and apply the configuration.

```
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.Ayroid-ec2[0]: Creating...
aws_instance.Ayroid-ec2[0]: Still creating... [10s elapsed]
aws_instance.Ayroid-ec2[0]: Still creating... [20s elapsed]
aws_instance.Ayroid-ec2[0]: Still creating... [30s elapsed]
aws_instance.Ayroid-ec2[0]: Creation complete after 34s [id=i-0688471057f5dbedd]

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

5. Verify Resources



6. Cleanup Resources

```
→  Exp2  terraform destroy
aws_instance.Ayroid-ec2[0]: Refreshing state... [id=i-0688471057f5dbedd]

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.Ayroid-ec2[0] will be destroyed
  - resource "aws_instance" "Ayroid-ec2" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:851931352354:
```