SIDDHARTH TIWARI 500095935 R2142210949 B.TECH CSE DEVOPS BATCH 3

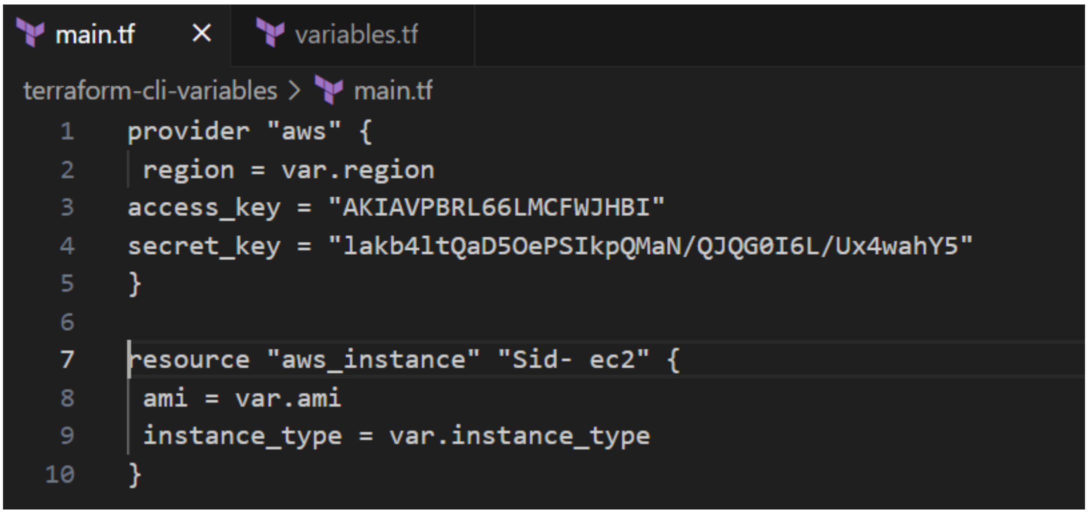
Lab Exercise 3–Provisioning an EC2 Instance on AWS

Step 1: Create a New Directory:

Create a new directory for your Terraform configuration:

```
mkdir aws-terraform-demo

cd aws-terraform-demo
```

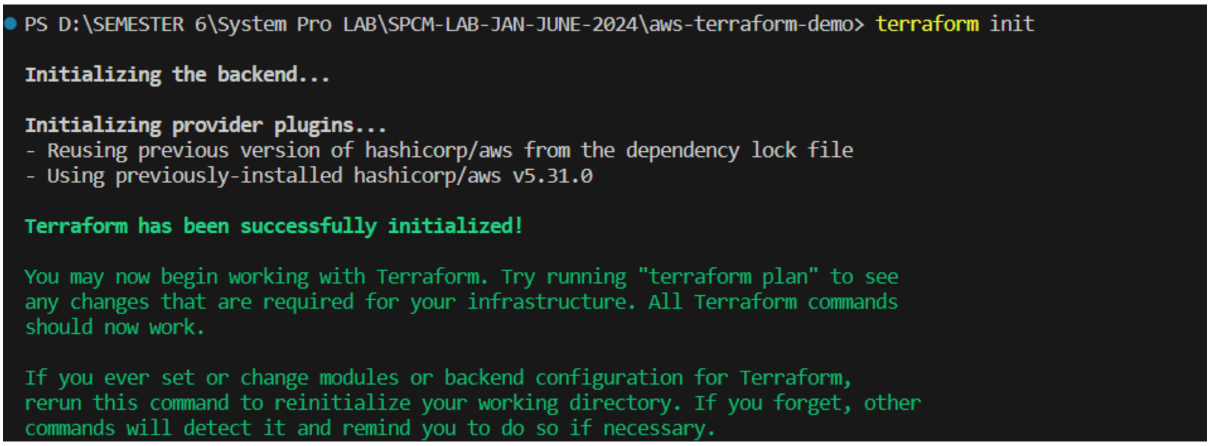
Step 2: Create Terraform Configuration File (main.tf):



```
provider "aws" {
 region = var.region
access_key = "AKIAVPBRL66LMCFWJHBI"
secret_key = "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"
}

resource "aws_instance" "Sid- ec2" {
 ami = var.ami
 instance_type = var.instance_type
}
```

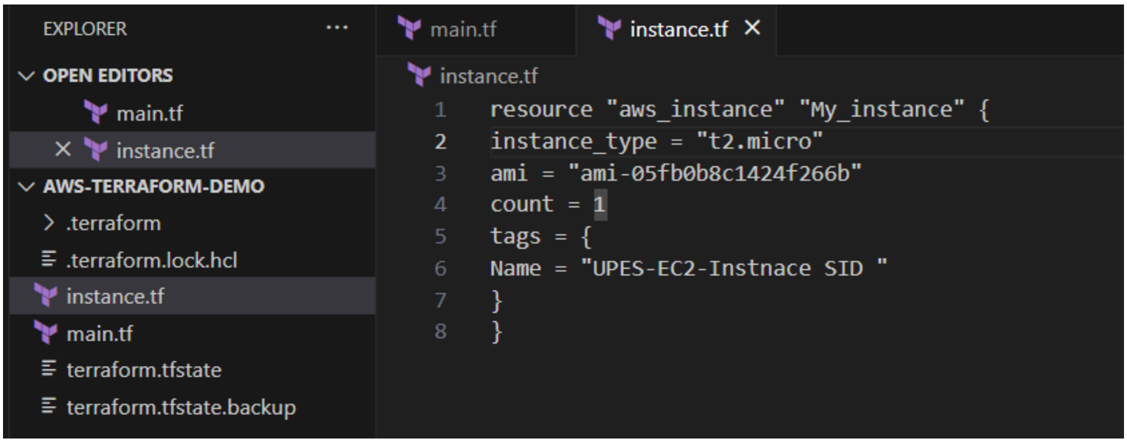
Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:



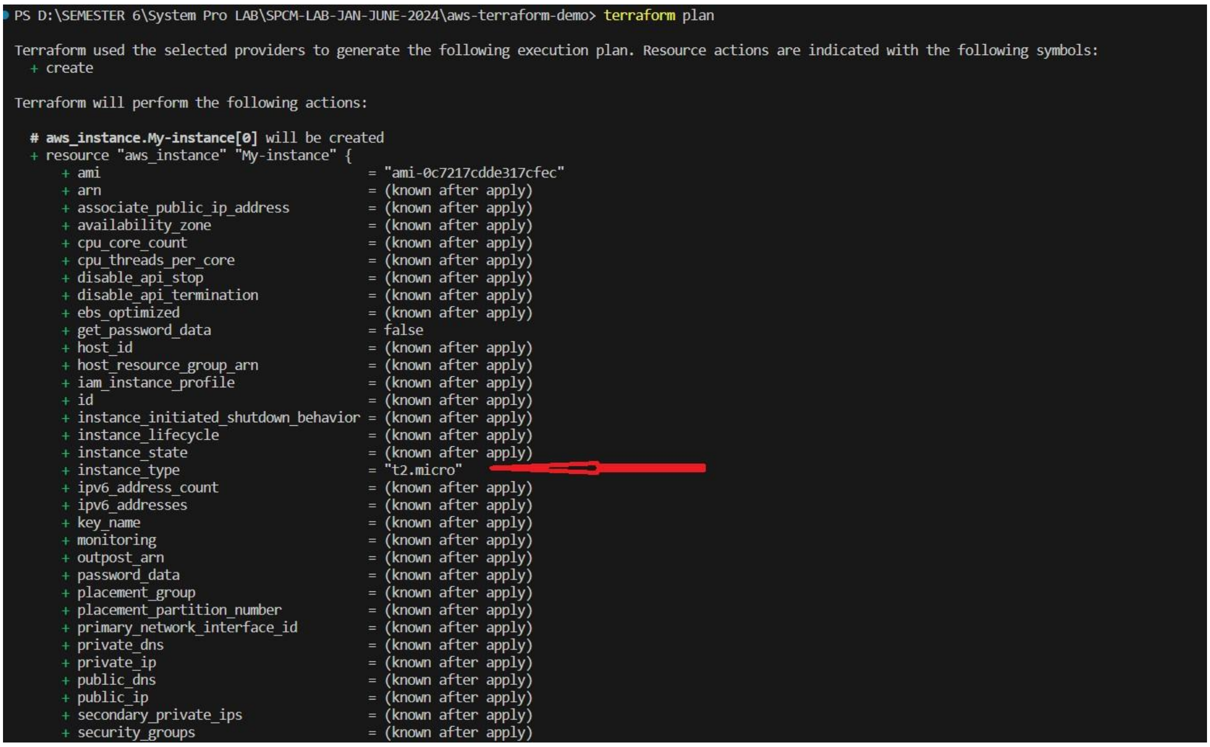
```
PS D:\SEMESTER 6\System Pro LAB\SPCM-LAB-JAN-JUNE-2024\aws-terraform-demo> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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.
```

Step 4: Create Terraform Configuration File for EC2 instance (instance.tf):



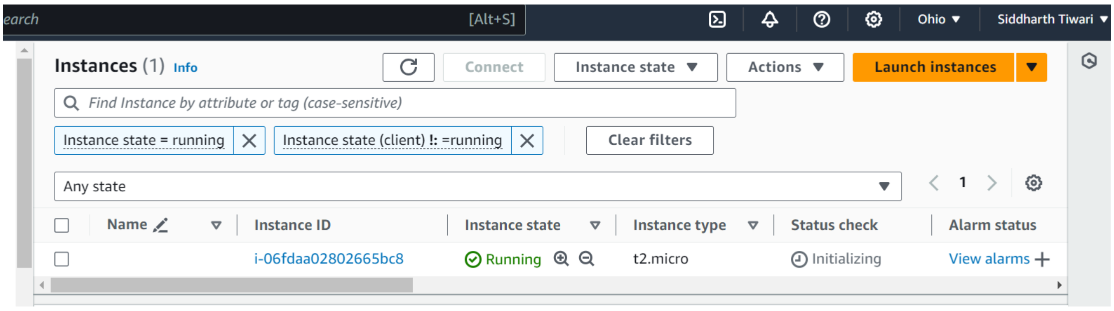
```
resource "aws_instance" "My_instance" {
instance_type = "t2.micro"
ami = "ami-05fb0b8c1424f266b"
count = 1
tags = {
Name = "UPES-EC2-Instnace SID "
}
}
```

Step 5: Review Plan: Run the following command to see what Terraform will do:



```
PS D:\SEMESTER 6\System Pro LAB\SPCM-LAB-JAN-JUNE-2024\aws-terraform-demo> terraform plan

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-0c7217cdde317cfec"
      + 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)
      + 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)
```

After the terraform apply command completes, log in to your AWS Management Console and navigate to the EC2 dashboard. Verify that the EC2 instance has been created.

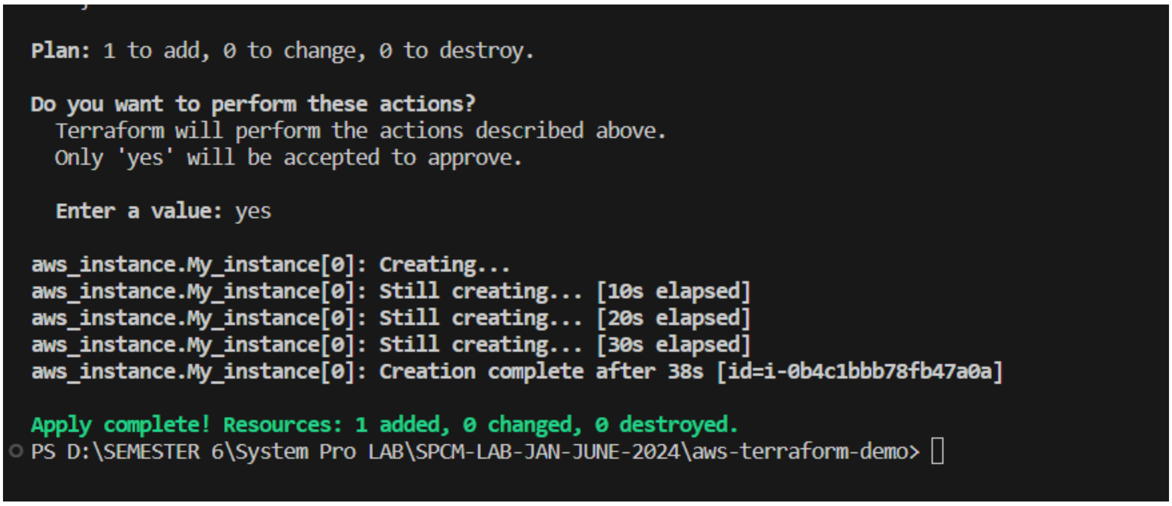


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□		i-06fdaa02802665bc8	Running	t2.micro	Initializing	View alarms +

Step 6: Cleanup Resources:

When you are done experimenting, run the following command to destroy the created resources:

terraform destroy



```
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.My_instance[0]: Creating...
aws_instance.My_instance[0]: Still creating... [10s elapsed]
aws_instance.My_instance[0]: Still creating... [20s elapsed]
aws_instance.My_instance[0]: Still creating... [30s elapsed]
aws_instance.My_instance[0]: Creation complete after 38s [id=i-0b4c1bbb78fb47a0a]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\SEMESTER 6\System Pro LAB\SPCM-LAB-JAN-JUNE-2024\aws-terraform-demo>
```