UPES

UNIVERSITY OF TOMORROW

SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT

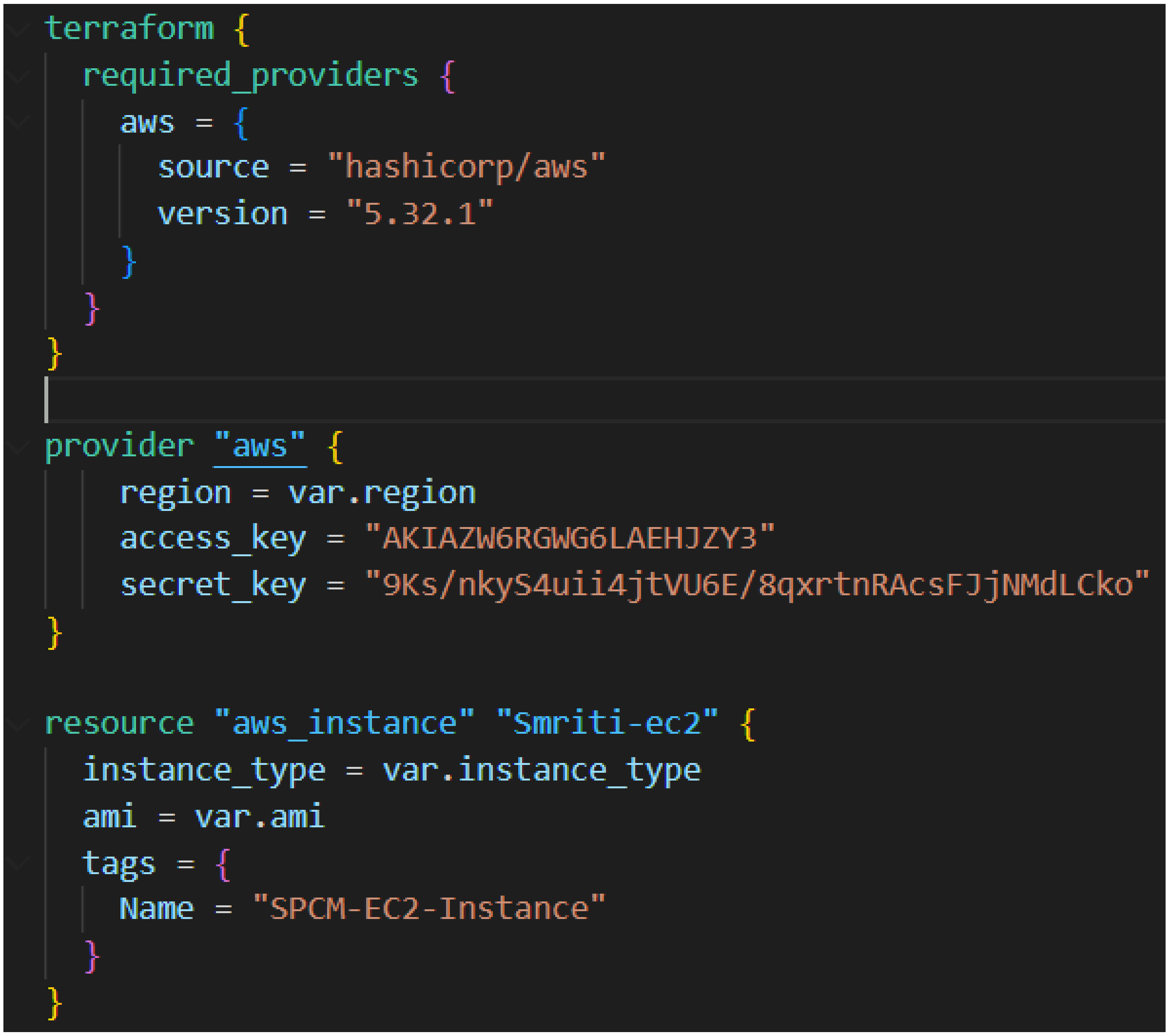
LAB FILE

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

EXPERIMENT 5:

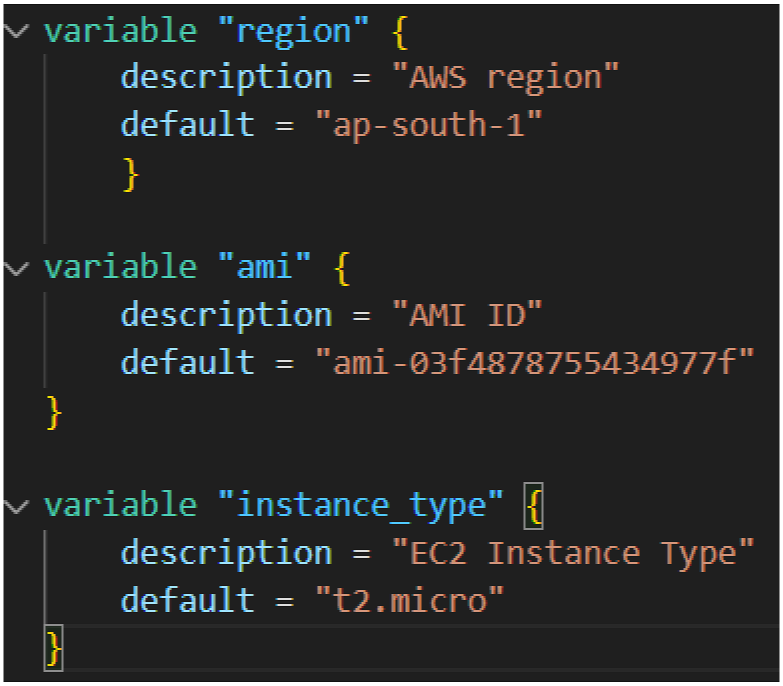
Terraform Variables with Command Line Arguments

1. Create a file named main.tf



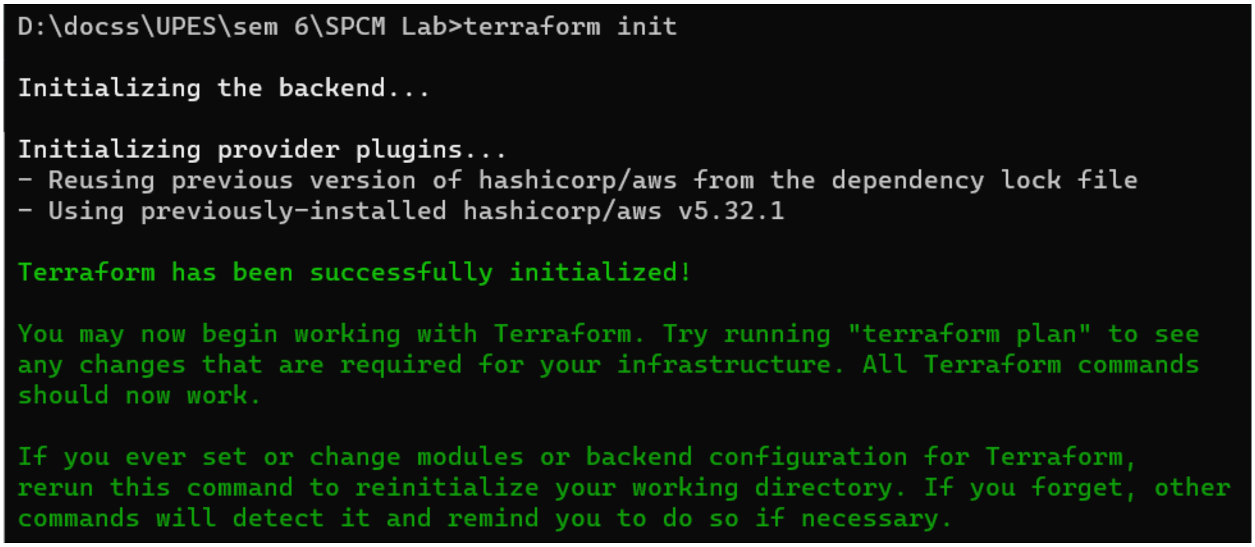
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.32.1"
    }
  }
}

provider "aws" {
    region = var.region
    access_key = "AKIAZW6RGWG6LAEHJZY3"
    secret_key = "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"
}

resource "aws_instance" "Smriti-ec2" {
  instance_type = var.instance_type
  ami = var.ami
  tags = {
    Name = "SPCM-EC2-Instance"
  }
}
```

2. Create a file named variables.tf



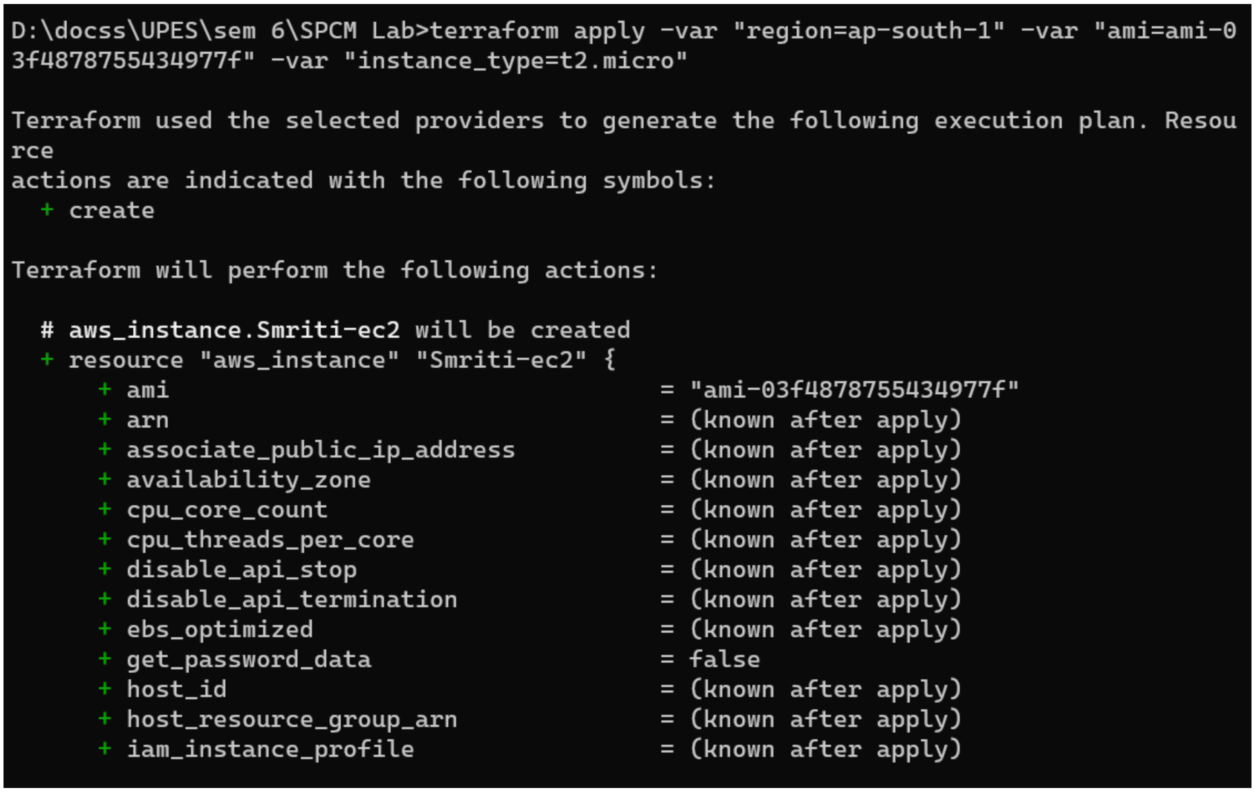
```
variable "region" {
    description = "AWS region"
    default = "ap-south-1"
    }

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

variable "instance_type" {
    description = "EC2 Instance Type"
    default = "t2.micro"
}
```

3. Open the command line and navigate to your project directory.
4. Run the terraform init command.

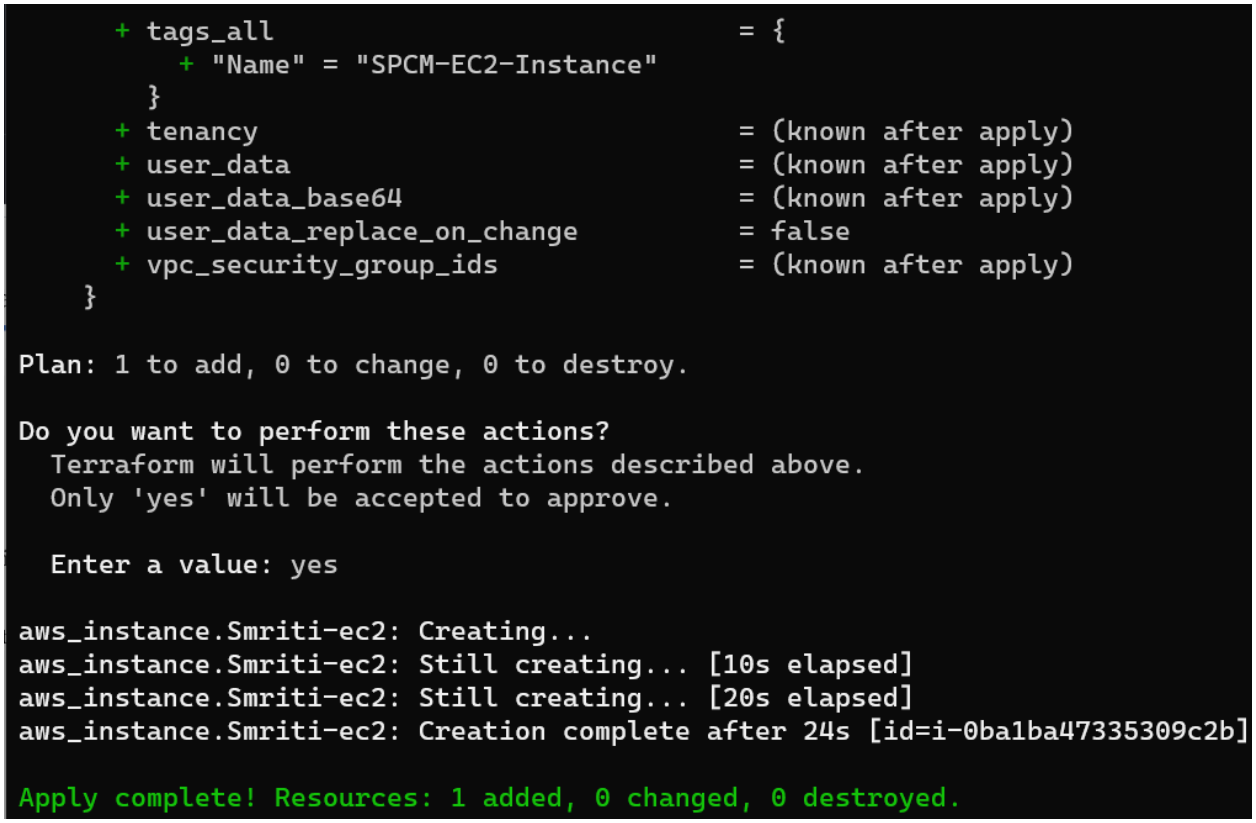


```
D:\docss\UPES\sem 6\SPCM Lab>terraform init

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.
```

5. Run the terraform apply with command line arguments to set the variable values

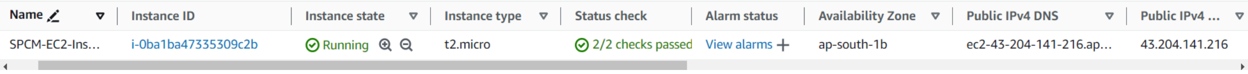


```
D:\docss\UPES\sem 6\SPCM Lab>terraform apply -var "region=ap-south-1" -var "ami=ami-0
3f4878755434977f" -var "instance_type=t2.micro"

Terraform used the selected providers to generate the following execution plan. Resou
rce
actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.Smriti-ec2 will be created
  + resource "aws_instance" "Smriti-ec2" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```



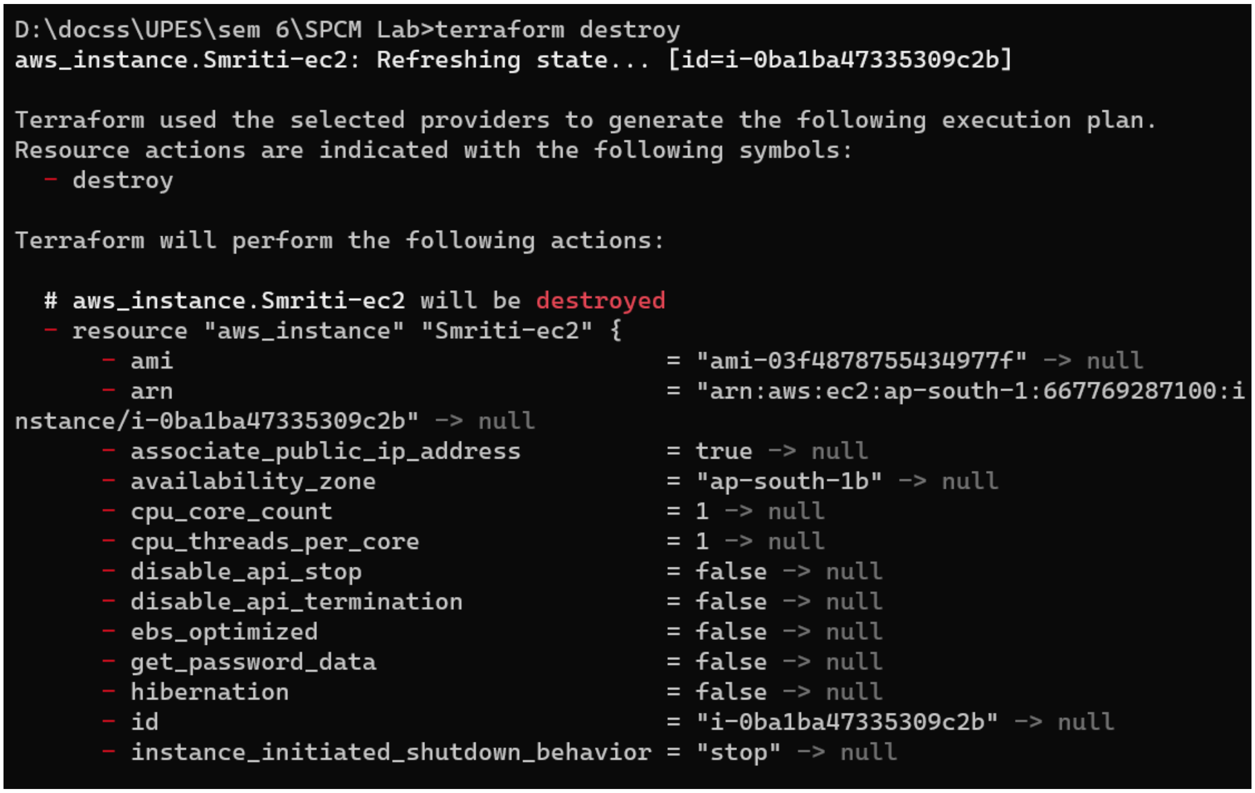
```
      + tags_all                             = {
          + "Name" = "SPCM-EC2-Instance"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + 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: Creation complete after 24s [id=i-0ba1ba47335309c2b]

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

6. Go to the AWS console to test and verify the creation of resources in the specified region.

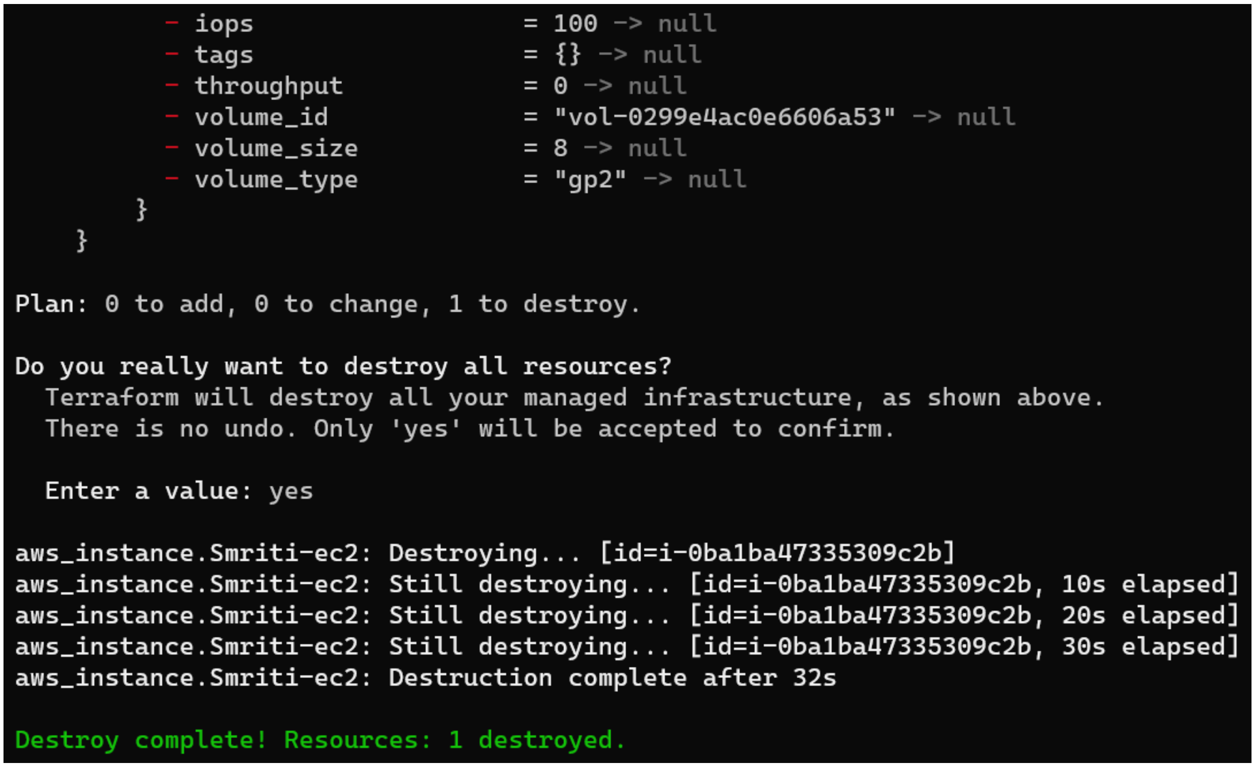


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
SPCM-EC2-Ins...	i-0ba1ba47335309c2b	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	ap-south-1b	ec2-43-204-141-216.ap...	43.204.141.216

7. Run the terraform destroy to clean up resources.



```
D:\docss\UPES\sem 6\SPCM Lab>terraform destroy
aws_instance.Smriti-ec2: Refreshing state... [id=i-0ba1ba47335309c2b]

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-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:667769287100:i
nstance/i-0ba1ba47335309c2b" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1b" -> 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-0ba1ba47335309c2b" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
```



```
          - iops                 = 100 -> null
          - tags                 = {} -> null
          - throughput           = 0 -> null
          - volume_id            = "vol-0299e4ac0e6606a53" -> null
          - volume_size          = 8 -> null
          - volume_type          = "gp2" -> null
        }
    }

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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.Smriti-ec2: Destroying... [id=i-0ba1ba47335309c2b]
aws_instance.Smriti-ec2: Still destroying... [id=i-0ba1ba47335309c2b, 10s elapsed]
aws_instance.Smriti-ec2: Still destroying... [id=i-0ba1ba47335309c2b, 20s elapsed]
aws_instance.Smriti-ec2: Still destroying... [id=i-0ba1ba47335309c2b, 30s elapsed]
aws_instance.Smriti-ec2: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.
```