

**School of Computer Science**  
**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**  
**DEHRADUN, UTTARAKHAND**



**System Provisioning and  
Configuration Management**

Submitted To:

Dr. Hitesh Kumar Sharma

Submitted By:

Mridul Vasudeva  
Batch – 1(DevOps)  
500091321  
R2142210502

## Lab Exercise 3—Provisioning an EC2 Instance on AWS

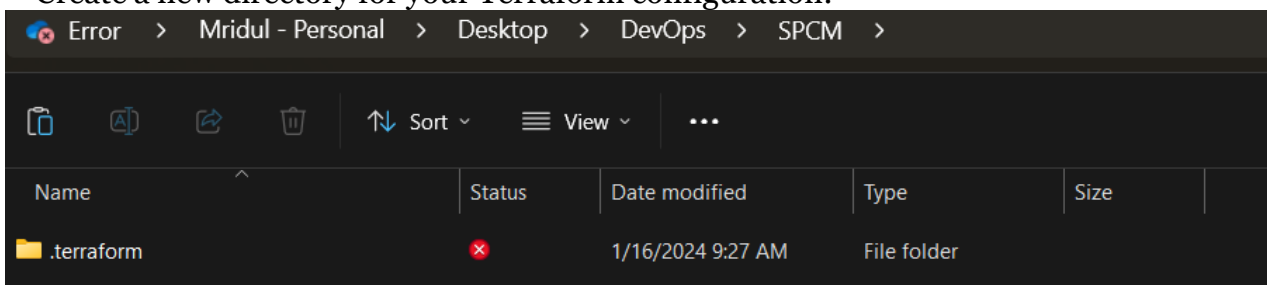
**Prerequisites: Terraform Installed: Make sure you have Terraform installed on your machine. Follow the official installation guide if needed.**

**AWS Credentials:** Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.

### Exercise Steps:

#### Step 1: Create a New Directory:

Create a new directory for your Terraform configuration:



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

Create a file named main.tf with the following content:

```
main.tf > provider "aws"
1  terraform {
2      required_providers {
3          aws = {
4              source = "hashicorp/aws"
5              version = "5.31.0"
6          }
7      }
8  }
9
10 provider "aws" {
11     region = "ap-south-1"
12     access_key = "AKIAZI2LIAJGSHGMMHP"
13     secret_key = "Fg5ojIk0skuNVG1NPhu4Kv41JzX1/XG/6zeQrGk/"
14 }
```

This script defines an AWS provider and provisions an EC2 instance.

### Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

```
C:\Users\Dell>cd C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM
C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM>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.
```

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

Create a file named instnace.tf with the following content:

```
instance.tf > ...
1  resource "aws_instance" "Automated-Instance" {
2      instance_type = "t2.micro"
3      ami = "ami-03f4878755434977f"
4      count= 1
5
6      tags = {
7          Name = "My-UPES-Instance"
8      }
9  }
```

## Step 5: Review Plan:

Run the following command to see what Terraform will do:

```
C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM>terraform validate
Success! The configuration is valid.

C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM>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.Automated-Instance[0] will be created
+ resource "aws_instance" "Automated-Instance" {
  + 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)
  + 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)
```

Review the plan to ensure it aligns with your expectations.

## Step 6: Apply Changes:

Apply the changes to create the AWS resources:

```
C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM>terraform apply

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.Automated-Instance[0] will be created
+ resource "aws_instance" "Automated-Instance" {
  + 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)
  + 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)
```

```

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.Automated-Instance[0]: Creating...
aws_instance.Automated-Instance[0]: Still creating... [10s elapsed]
aws_instance.Automated-Instance[0]: Still creating... [20s elapsed]
aws_instance.Automated-Instance[0]: Still creating... [30s elapsed]
aws_instance.Automated-Instance[0]: Creation complete after 33s [id=i-023c77b0803e03aa4]

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

```

Instances (1/1) Info								Connect	Instance state ▼	Actions ▼	Launch instances ▼
Find Instance by attribute or tag (case-sensitive)											
Instance state = running X Clear filters											
✓	Name ↗	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS			
✓	My-UPES-Inst...	i-023c77b0803e03aa4	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-110-128-2			

## Step 7: Verify Resources:

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.

## Step 8: Cleanup Resources:

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

```

C:\Users\Dell\OneDrive\Desktop\DevOps\SPCM>terraform destroy
aws_instance.Automated-Instance[0]: Refreshing state... [id=i-023c77b0803e03aa4]

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.Automated-Instance[0] will be destroyed
- resource "aws_instance" "Automated-Instance" {
  ami              = "ami-03f4878755434977f" -> null
  arn              = "arn:aws:ec2:ap-south-1:637423583821:instance/i-023c77b0803e03aa4" -> 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-023c77b0803e03aa4" -> null
  instance_initiated_shutdown_behavior = "stop" -> null
  instance_state     = "running" -> null
  instance_type      = "t2.micro" -> null
  ipv6_address_count = 0 -> null
  ipv6_addresses     = [] -> null
  monitoring         = false -> null
  placement_partition_number = 0 -> null
  primary_network_interface_id = "eni-0e13e2a2b3195837d" -> null
  private_dns        = "ip-172-31-38-43.ap-south-1.compute.internal" -> null
  private_ip         = "172.31.38.43" -> null
  public_dns         = "ec2-3-110-128-23.ap-south-1.compute.amazonaws.com" -> null
  public_ip          = "3.110.128.23" -> null
  secondary_private_ips = [] -> null
  security_groups    = [
    - "default",
  ] -> null
  source_dest_check  = true -> 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.Automated-Instance[0]: Destroying... [id=i-023c77b0803e03aa4]
aws_instance.Automated-Instance[0]: Still destroying... [id=i-023c77b0803e03aa4, 10s elapsed]
aws_instance.Automated-Instance[0]: Still destroying... [id=i-023c77b0803e03aa4, 20s elapsed]
aws_instance.Automated-Instance[0]: Still destroying... [id=i-023c77b0803e03aa4, 30s elapsed]
aws_instance.Automated-Instance[0]: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.
```

