

# Lab Exercise 13 – Provisioning an EC2 Instance on AWS

**Name:-Vansh Bhatt**

**SapId:- 500125395**

**R.No:- R2142231689**

**Batch:- DevOps B1**

**To:- Hitesh Kumar Sharma Sir**

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

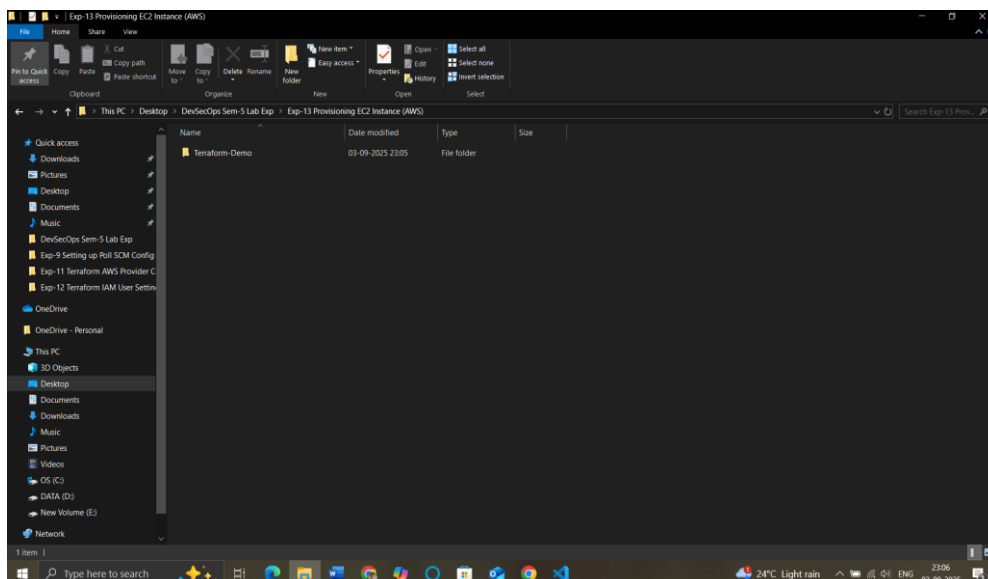
**AWS Credentials: Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.**

## **Exercise Steps:**

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

Create a new directory for your Terraform configuration:

**“Terraform-Demo”**

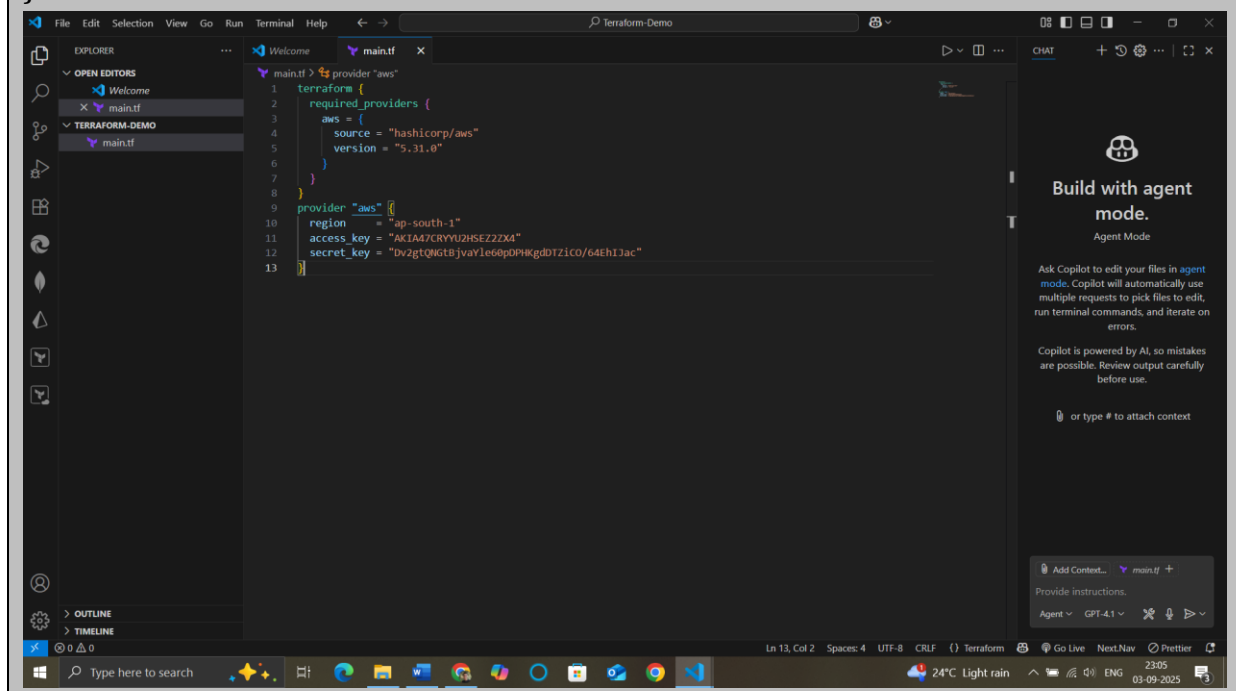


## Step 2: Create Terraform Configuration File (main.tf):

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

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

provider "aws" {
  region     = "ap-south-1"
  access_key = "your IAM access key"
  secret_key = "your secret access key"
}
```

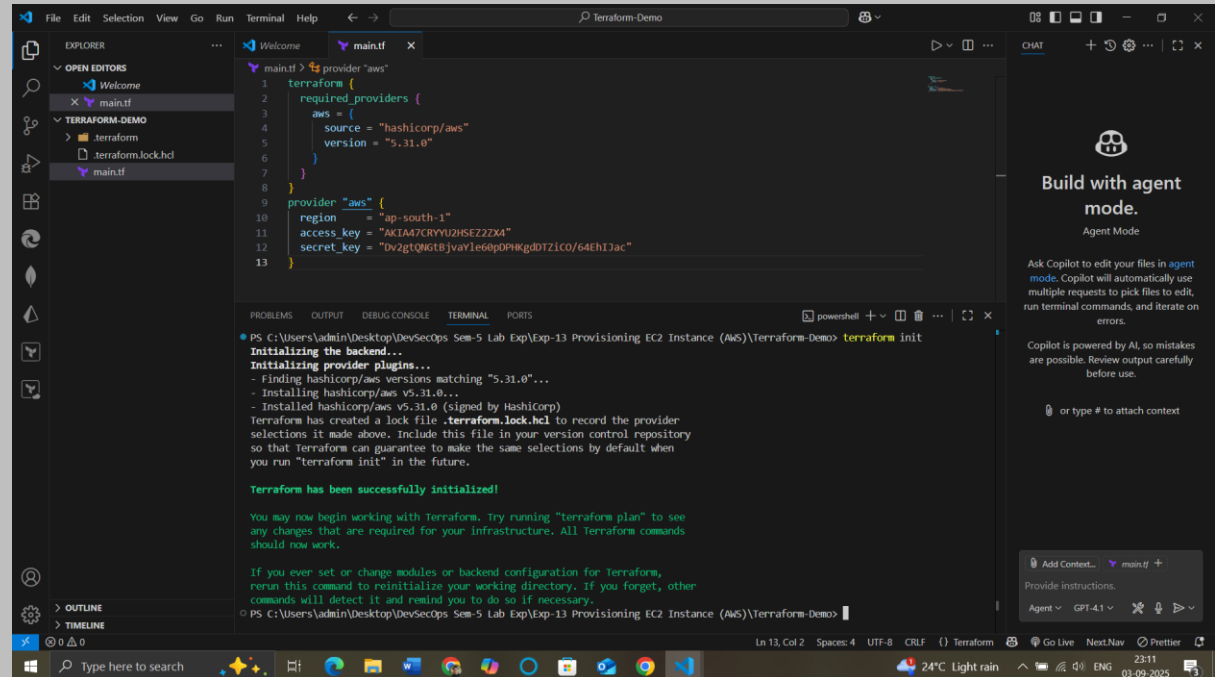


This script defines an AWS provider and provisions an EC2 instance.

## Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

## terraform init



```
1 provider "aws"
2 {
3   terraform {
4     required_providers {
5       aws = {
6         source = "hashicorp/aws"
7         version = "5.31.0"
8       }
9     }
10  }
11  provider "aws" {
12    region = "ap-south-1"
13    access_key = "AKIA47CRYVUZHSEZZXA"
14    secret_key = "Dv2gtQNGtBjvaY1e6pDRWgdbDTZ1CO/64EhIJac"
15  }
16 }
```

PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (AWS)\Terraform-Demo> 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:

```
resource "aws_instance" "My-instance" {

    ami = "ami-03f4878755434977f"

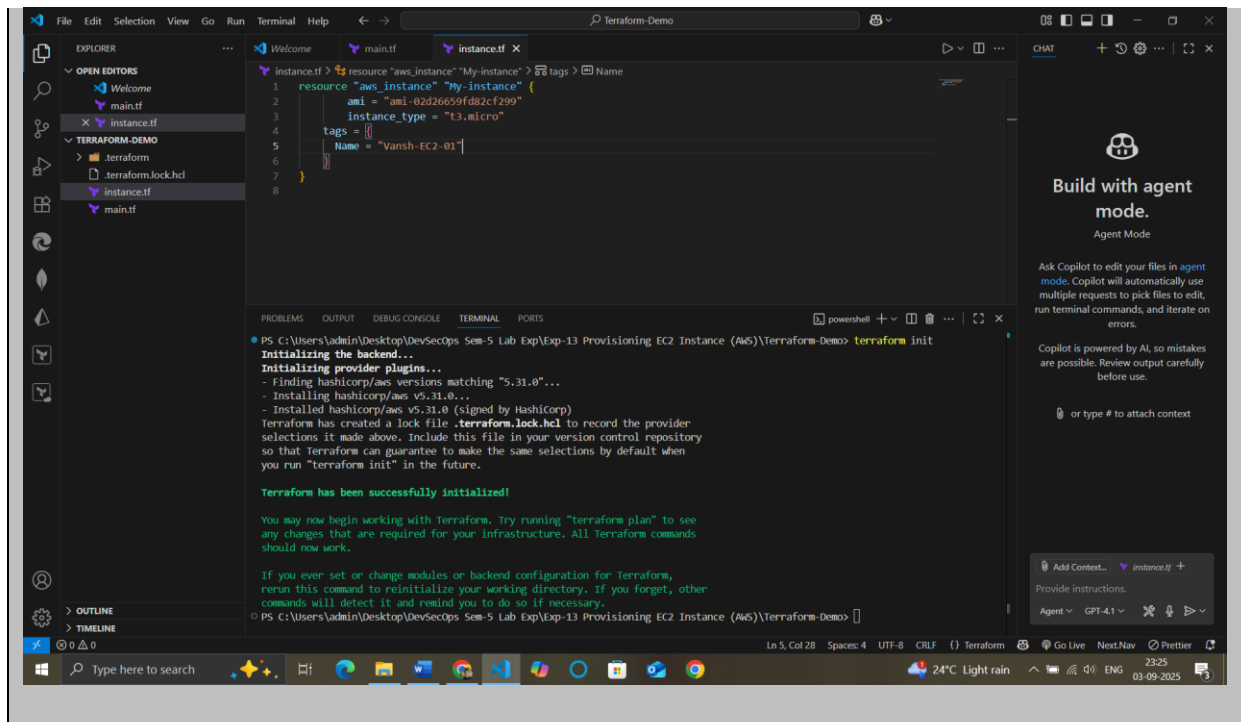
    instance_type = "t2.micro"

    tags = {

        Name = "MY-EC2-Instnace"

    }

}
```



## Step 5: Review Plan:

Run the following command to see what Terraform will do:

### terraform plan

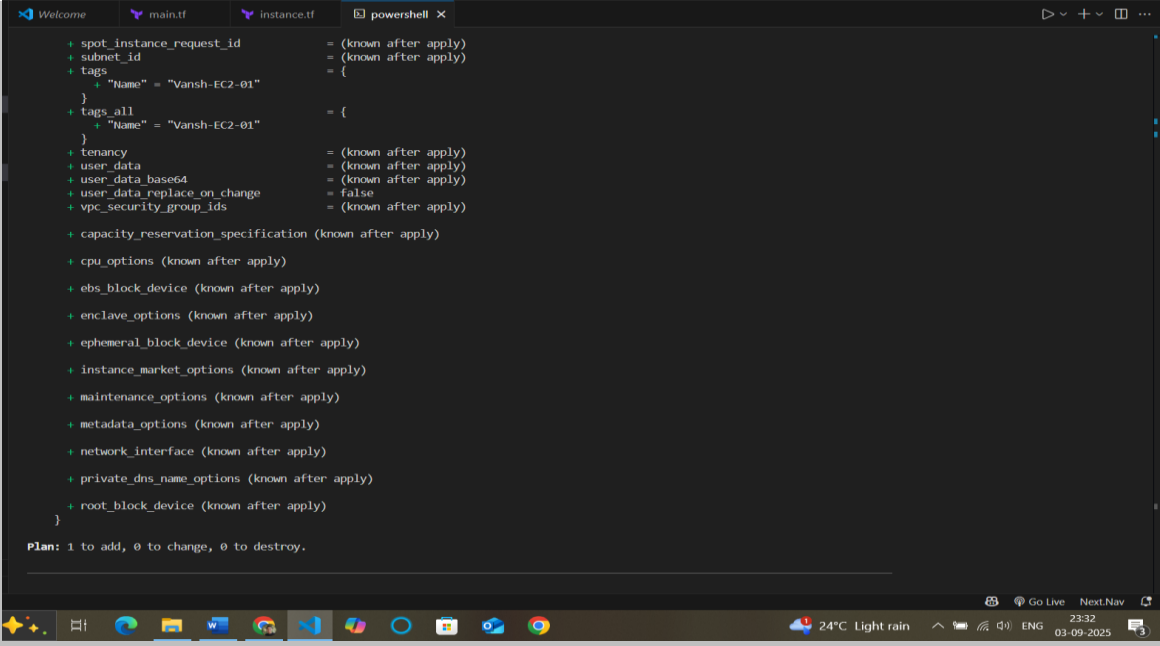
```
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (AWS)\Terraform-Demo> terraform validate
Success! The configuration is valid.

PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (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 will be created
+ resource "aws_instance" "My-instance" {
+   ami               = "ami-02d26659fd82cf299"
+   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      = "t3.micro"
+   ipv6_address_count = (known after apply)
+   ipv6_addresses     = (known after apply)
}
```



```
+ spot_instance_request_id = (known after apply)
+ subnet_id               = (known after apply)
+ tags                    = {
+   + "Name" = "Vansh-EC2-01"
+ }
+ tags_all                = {
+   + "Name" = "Vansh-EC2-01"
+ }
+ 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)
+ capacity_reservation_specification (known after apply)
+ cpu_options             (known after apply)
+ ebs_block_device        (known after apply)
+ enclave_options         (known after apply)
+ ephemeral_block_device  (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options    (known after apply)
+ metadata_options        (known after apply)
+ network_interface       (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device       (known after apply)
}

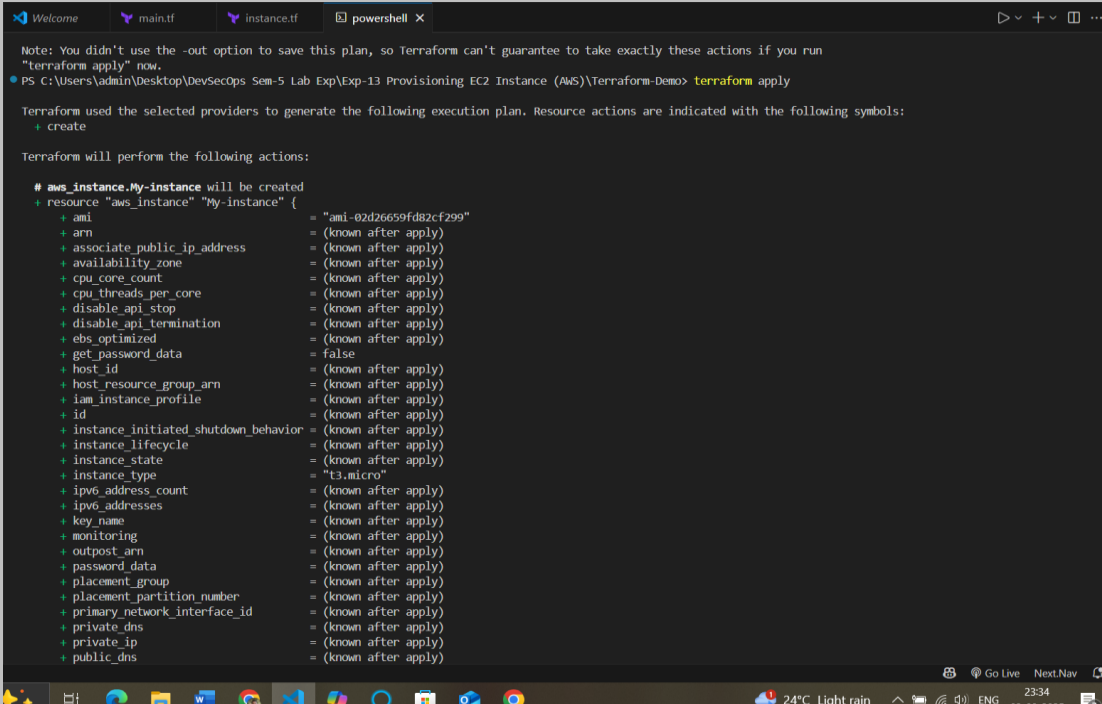
Plan: 1 to add, 0 to change, 0 to destroy.
```

Review the plan to ensure it aligns with your expectations.

## Step 6: Apply Changes:

Apply the changes to create the AWS resources:

### terraform apply



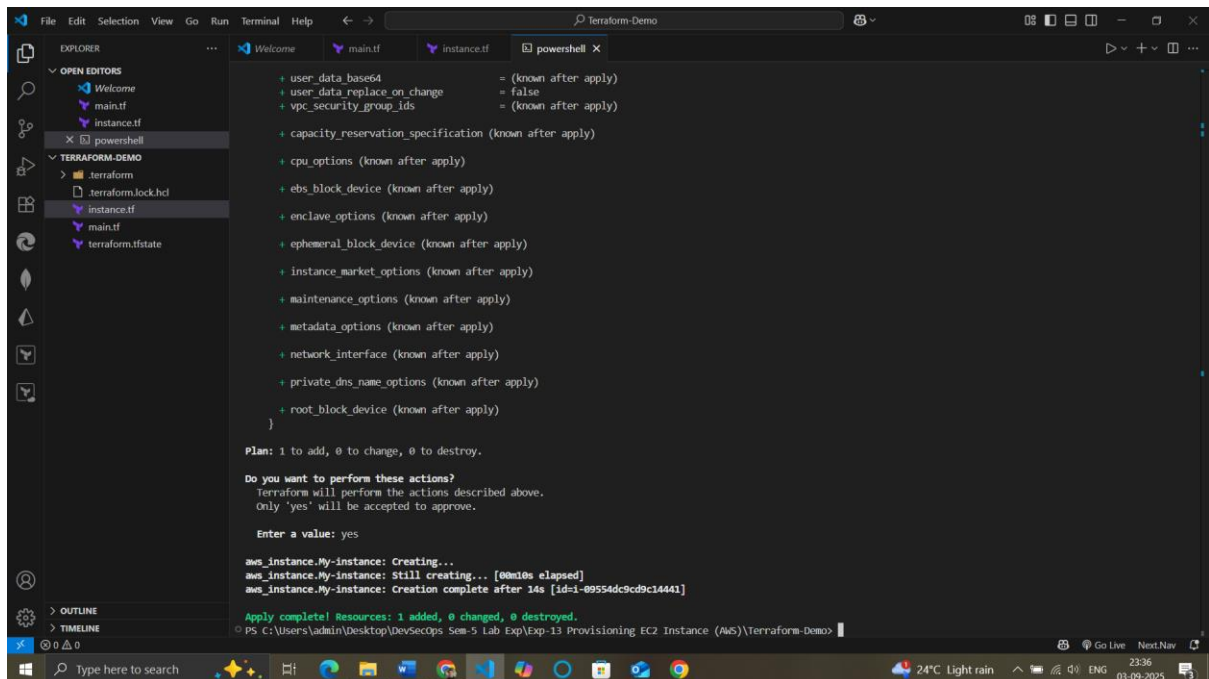
```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run
"terraform apply" now.
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (AWS)\Terraform-Demo> terraform apply

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.My-instance will be created
+ resource "aws_instance" "My-instance" {
+   ami                    = "ami-02d26659fd82cf299"
+   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          = "t3.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)
}
```

Type yes when prompted.



The screenshot shows a Windows terminal window with the Terraform CLI interface. The left sidebar displays the Explorer view with a project structure including 'TERRAFORM-DEMO' and 'instance.tf'. The main terminal area shows the output of a 'terraform apply' command. It lists various AWS resources being created, such as 'user\_data\_base64', 'user\_data\_replace\_on\_change', 'vpc\_security\_group\_ids', 'capacity\_reservation\_specification', 'cpu\_options', 'ebs\_block\_device', 'enclave\_options', 'ephemeral\_block\_device', 'instance\_market\_options', 'maintenance\_options', 'metadata\_options', 'network\_interface', 'private\_dns\_name\_options', and 'root\_block\_device'. A confirmation prompt asks 'Do you want to perform these actions?' and the user responds 'yes'. The output then shows the successful creation of an EC2 instance, 'aws\_instance.My-instance', with its ID 'i-09554dc9dc9c14441'.

```
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (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.

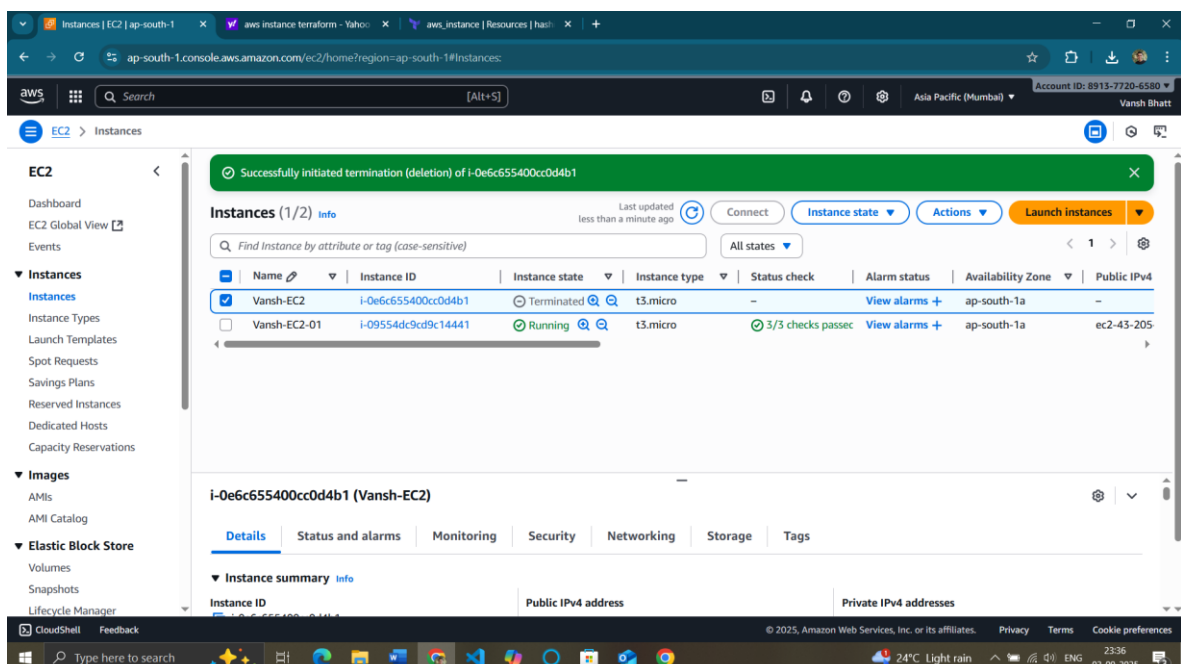
Enter a value: yes

aws_instance.My-instance: Creating...
aws_instance.My-instance: Still creating... [00m10s elapsed]
aws_instance.My-instance: Creation complete after 14s [id=i-09554dc9dc9c14441]

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

## Step 7: Verify Resources:

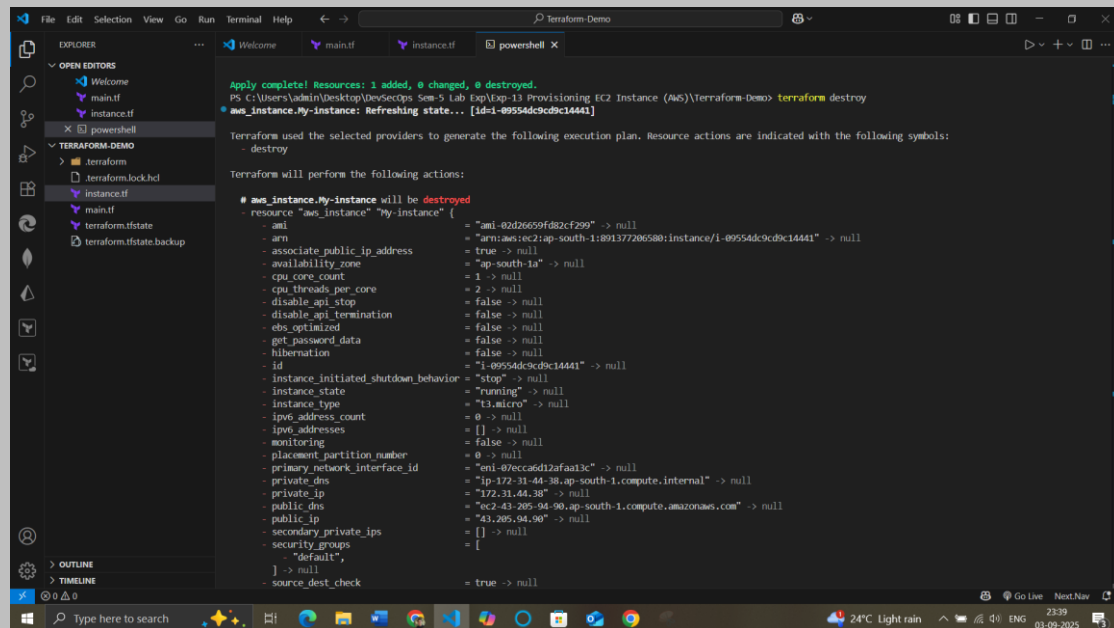
After the terraform apply command completes, log in to your AWS Management Console and navigate to the EC2 dashboard. Verify that the EC2 instance has been created.



## Step 8: Cleanup Resources:

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

terraform destroy



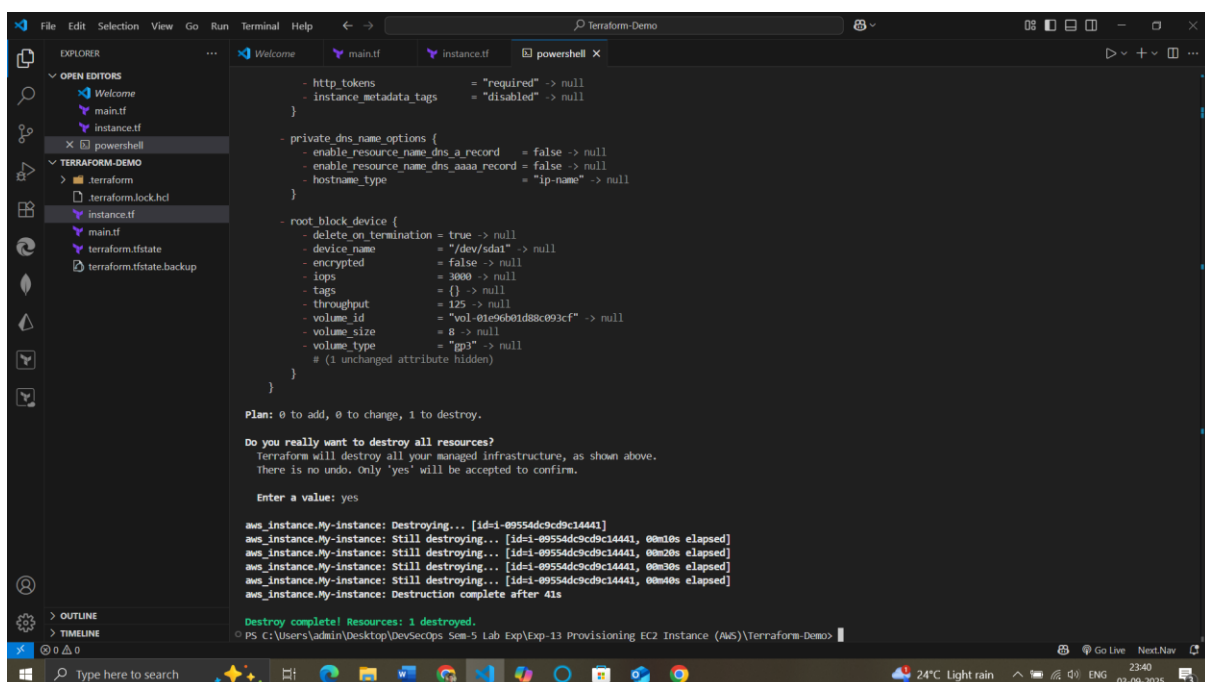
```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (AWS)\Terraform-Demo> terraform destroy
* aws_instance.My-instance: Refreshing state... [id=i-09554dc9cd9c14441]

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.My-instance will be destroyed
- resource "aws_instance" "My-instance" {
  ami           = "ami-02d26659f082cf299" -> null
  arn           = "arn:aws:ec2:ap-south-1:891377206580:instance/i-09554dc9cd9c14441" -> null
  associate_public_ip_address = true -> null
  availability_zone           = "ap-south-1a" -> null
  cpu_core_count              = 1 -> null
  cpu_threads_per_core        = 2 -> 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-09554dc9cd9c14441" -> null
  instance_initiated_shutdown_behavior = "stop" -> null
  instance_state              = "running" -> null
  instance_type               = "t3.micro" -> null
  ipv6_address_count           = 0 -> null
  ipv6_addresses               = [] -> null
  monitoring                   = false -> null
  placement_partition_number   = 0 -> null
  primary_network_interface_id = "eni-07eccad12afaal3c" -> null
  private_dns                  = "ip-172-31-44-38.ap-south-1.compute.internal" -> null
  private_ip                   = "172.31.44.38" -> null
  public_dns                   = "ec2-43-205-94-90.ap-south-1.compute.amazonaws.com" -> null
  public_ip                    = "43.205.94.90" -> null
  secondary_private_ips        = [] -> null
  security_groups              = [
    "default",
  ] -> null
  source_dest_check            = true -> null
}
```

Type yes when prompted.



```
- http_tokens           = "required" -> null
- instance_metadata_tags = "disabled" -> null
}

- private_dns_name_options {
  - enable_resource_name_dns_a_record = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type                     = "ip-name" -> null
}

- root_block_device {
  - delete_on_termination = true -> null
  - device_name           = "/dev/sda1" -> null
  - encrypted             = false -> null
  - iops                  = 3000 -> null
  - tags                  = {} -> null
  - throughput            = 125 -> null
  - volume_id             = "vol-01e9eb01d88c093cf" -> null
  - volume_size           = 8 -> null
  - volume_type           = "gp3" -> null
  # (1 unchanged attribute hidden)
}

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.My-instance: Destroying... [id=i-09554dc9cd9c14441]
aws_instance.My-instance: Still destroying... [id=i-09554dc9cd9c14441, 00m10s elapsed]
aws_instance.My-instance: Still destroying... [id=i-09554dc9cd9c14441, 00m20s elapsed]
aws_instance.My-instance: Still destroying... [id=i-09554dc9cd9c14441, 00m30s elapsed]
aws_instance.My-instance: Still destroying... [id=i-09554dc9cd9c14441, 00m40s elapsed]
aws_instance.My-instance: Destruction complete after 41s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\admin\Desktop\DevSecOps Sem-5 Lab Exp\Exp-13 Provisioning EC2 Instance (AWS)\Terraform-Demo>
```

Notes:

Customize the instance.tf file to provision different AWS resources.

Explore the Terraform AWS provider documentation for additional AWS resources and configuration options.

Always be cautious when running terraform destroy to avoid accidental resource deletion.

This exercise provides a basic introduction to using Terraform with the AWS provider. Feel free to explore more complex Terraform configurations and resources based on your needs.