# Lab Exercise 2– Terraform AWS Provider Configuration

Aditya Tomar 500106015 R2142221060 Batch-2(DevOps)

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

## **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 = "5.30.0"
  }
}
```

```
provider "aws" {
}

| Temain.tf | Sextension: HashiCorp Terraform | Sexten
```

### Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

```
terraform init
     Last login: Thu Jan 16 14:06:47 on ttys000
     [adityatomar@Adityas-MacBook-Air Terraform-Demo % terraform init
     Initializing the backend...
     Initializing provider plugins...
     - Finding hashicorp/aws versions matching "5.30.0"...
     - Installing hashicorp/aws v5.30.0...

    Installed hashicorp/aws v5.30.0 (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.
     adityatomar@Adityas-MacBook-Air Terraform-Demo %
```