

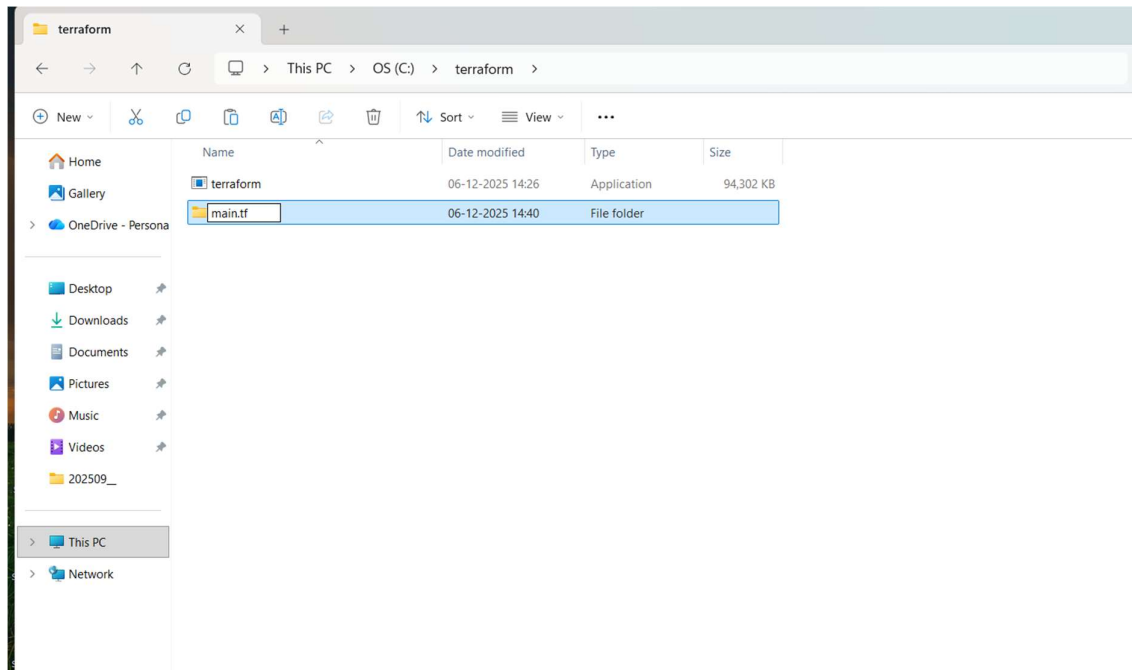
## Lab Exercise 2– Terraform AWS Provider Configuration

**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:

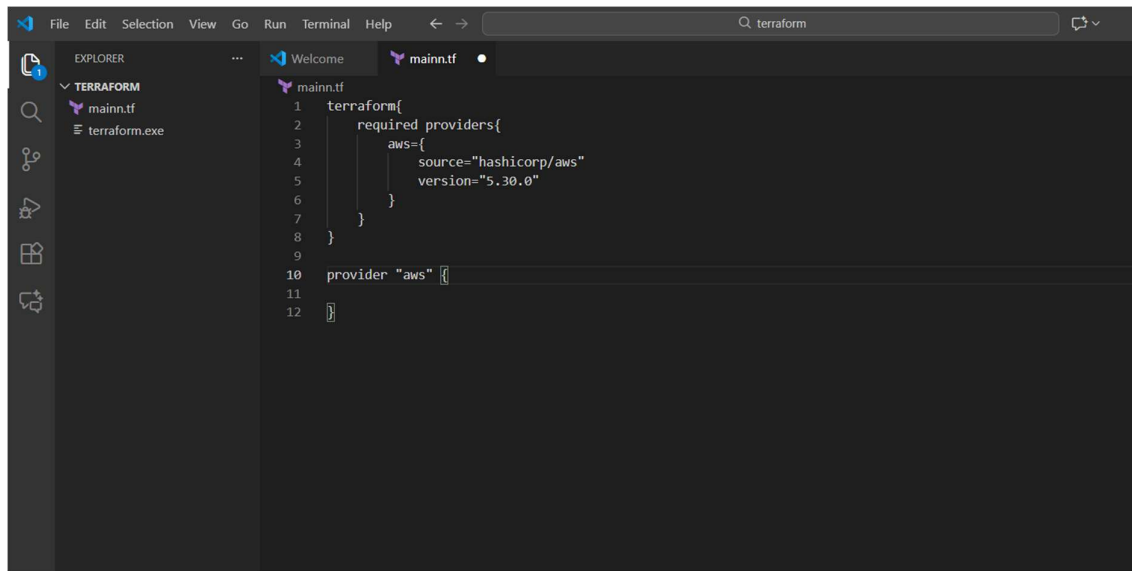
Create a new directory for your Terraform configuration:



### “Terraform-Demo”

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

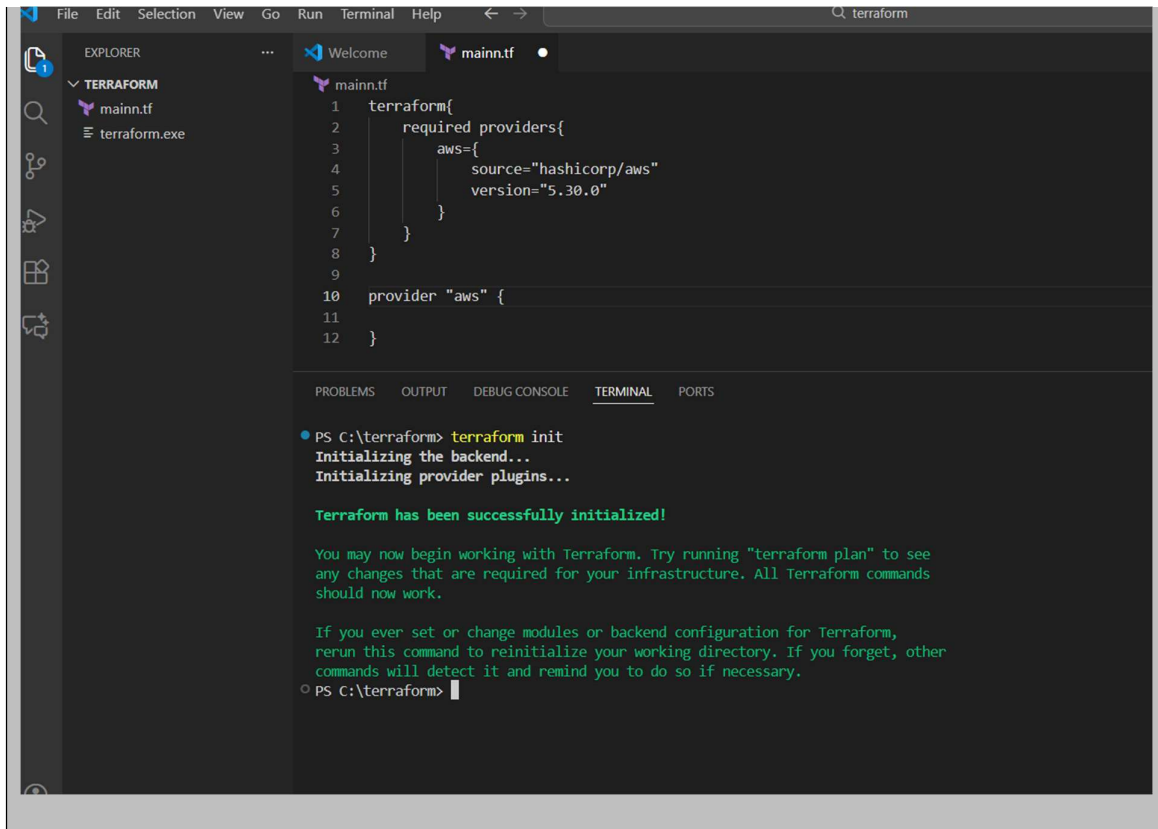
Create a file named main.tf with the following content:



### Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

```
terraform init
```



The screenshot displays the Visual Studio Code interface with a Terraform configuration file named `mainn.tf` open in the editor. The Explorer sidebar on the left shows the project structure with `TERRAFORM` as the root, containing `mainn.tf` and `terraform.exe`. The main editor window shows the following Terraform code:

```
1 terraform{
2   required_providers{
3     aws={
4       source="hashicorp/aws"
5       version="5.30.0"
6     }
7   }
8 }
9
10 provider "aws" {
11 }
12 }
```

Below the code editor, the TERMINAL panel is active, showing the output of the `terraform init` command. The output indicates that the backend and provider plugins have been successfully initialized.

```
PS C:\terraform> terraform init
Initializing the backend...
Initializing provider plugins...

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.
PS C:\terraform>
```