

Lesson 09 Demo 01

Configuring AWS and Random Providers in Terraform

Objective: To configure the AWS and random Terraform providers in the Terraform configuration file for managing cloud infrastructure efficiently

Tools required: VS Code and Linux terminal

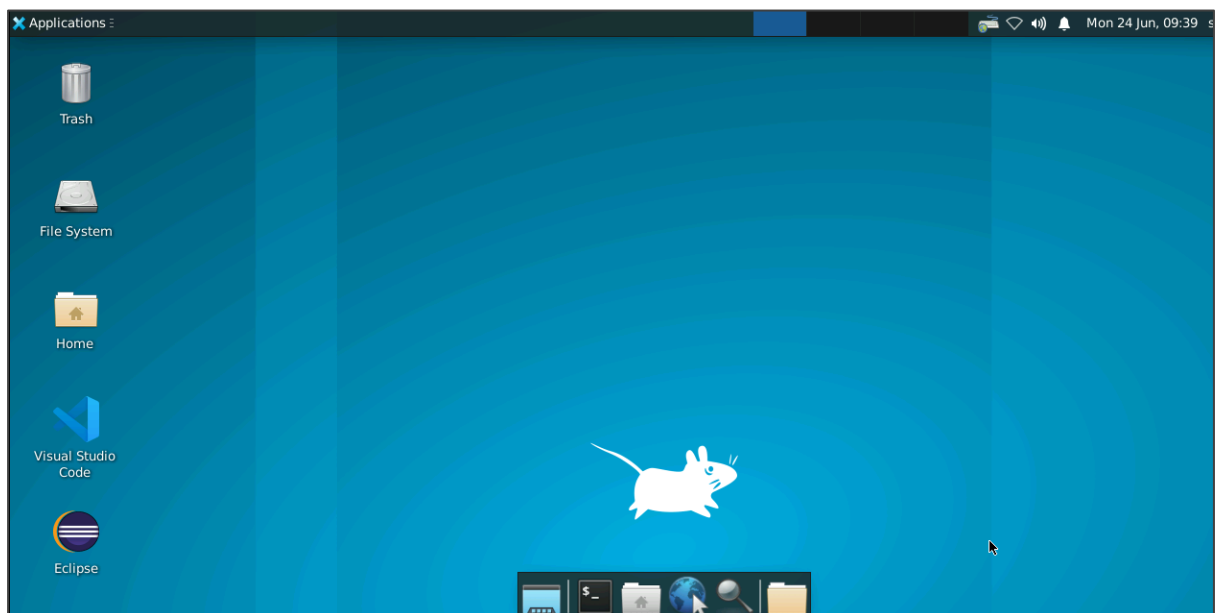
Prerequisites: None

Steps to be followed:

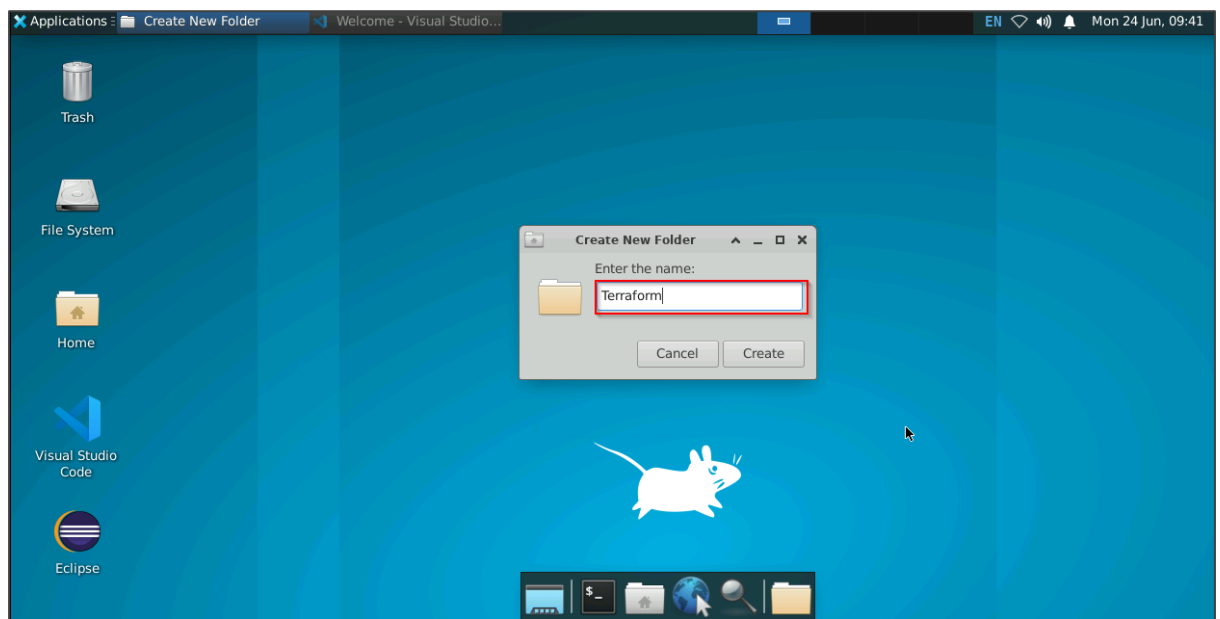
1. Install the Terraform extension in VS Code
2. Create the main Terraform configuration file
3. Configure the Terraform providers
4. Initialize the Terraform configuration file

Step 1: Install the Terraform extension in VS Code

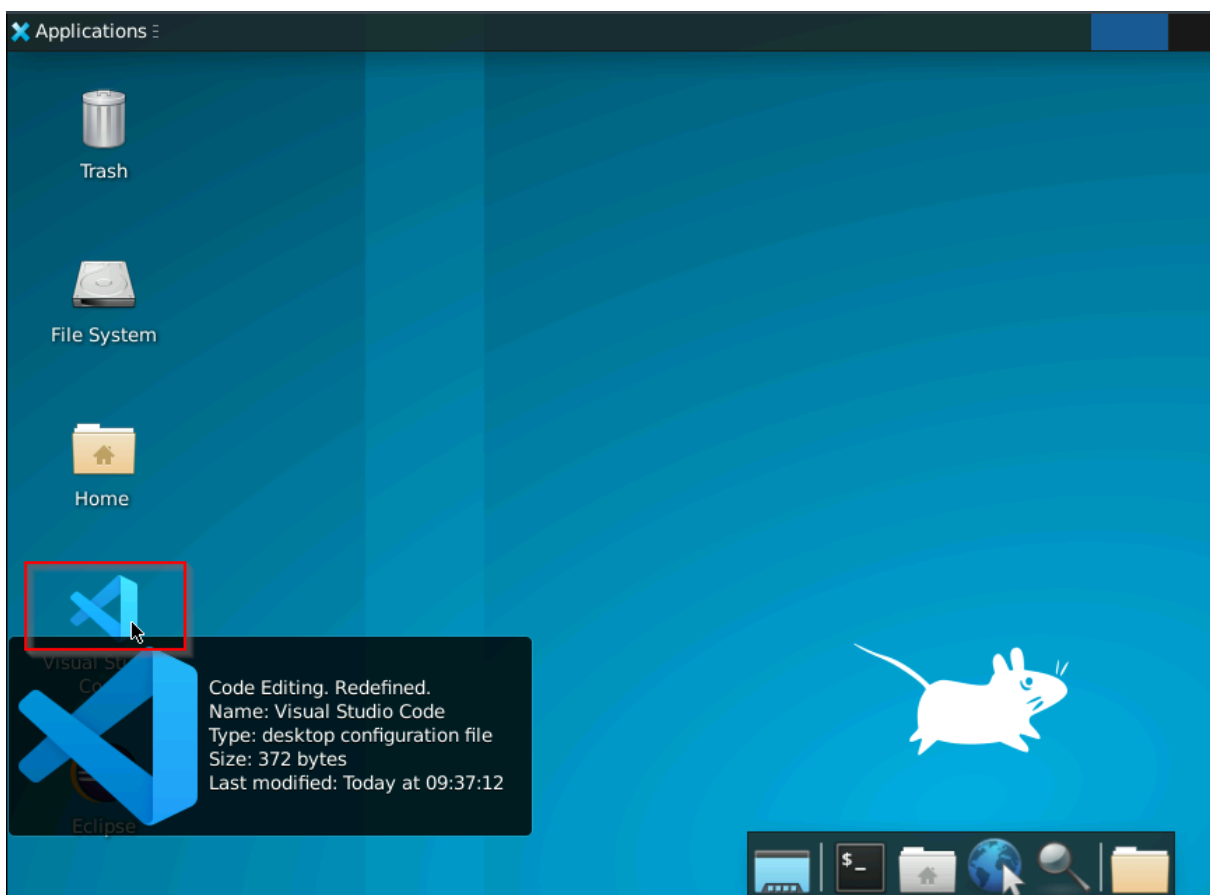
1.1 Open the provided DevOps Lab



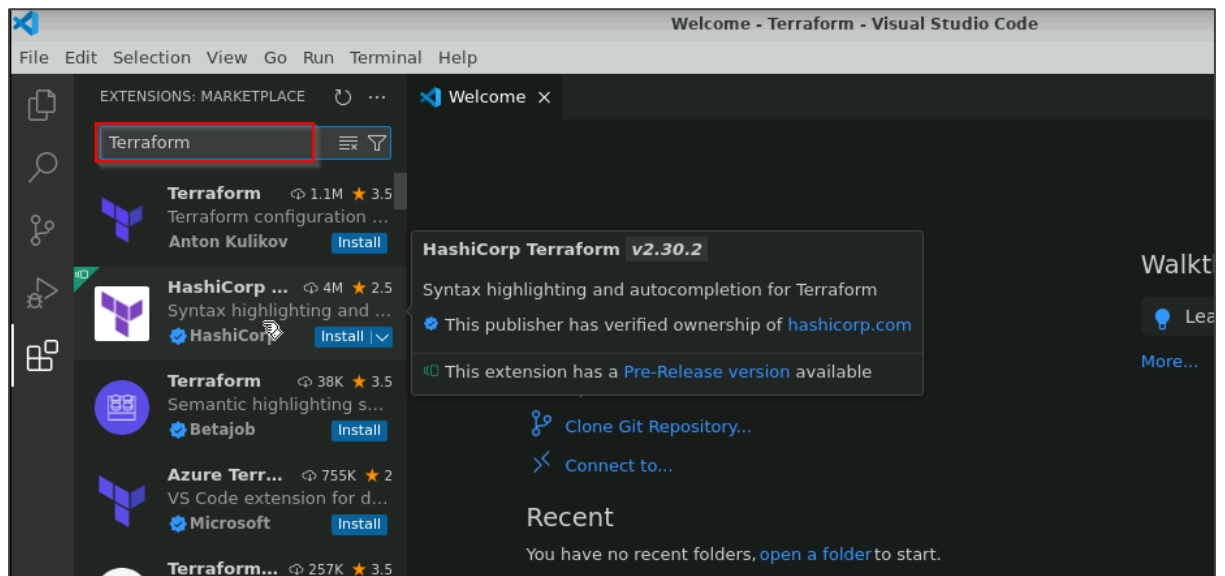
1.2 Create a folder named Terraform



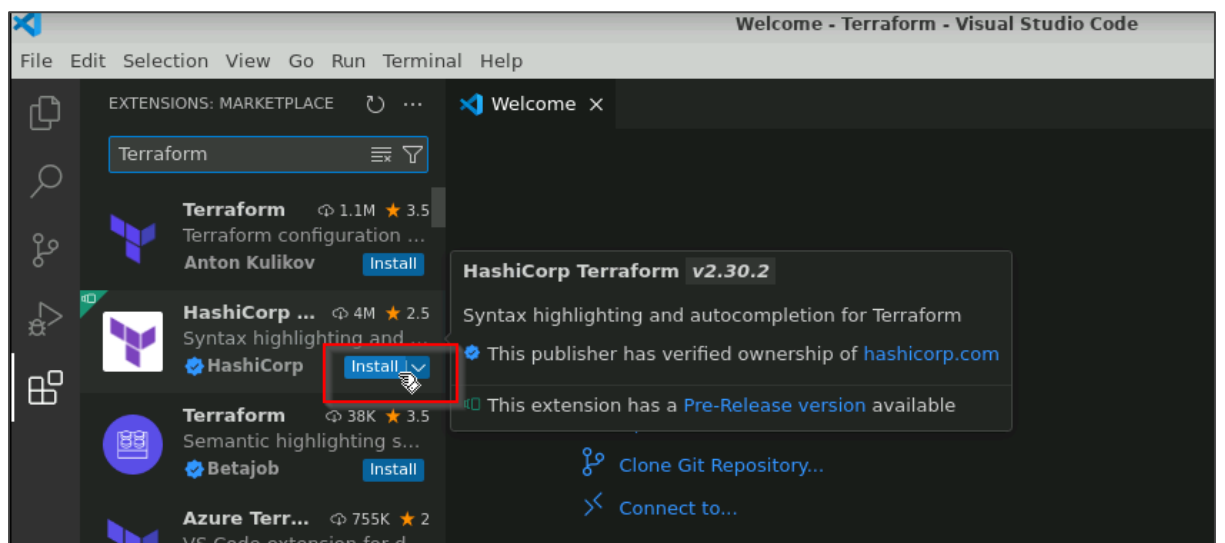
1.3 Open the VS Code editor



1.4 Go to the **Extensions** tab and search for **Terraform** in the search field



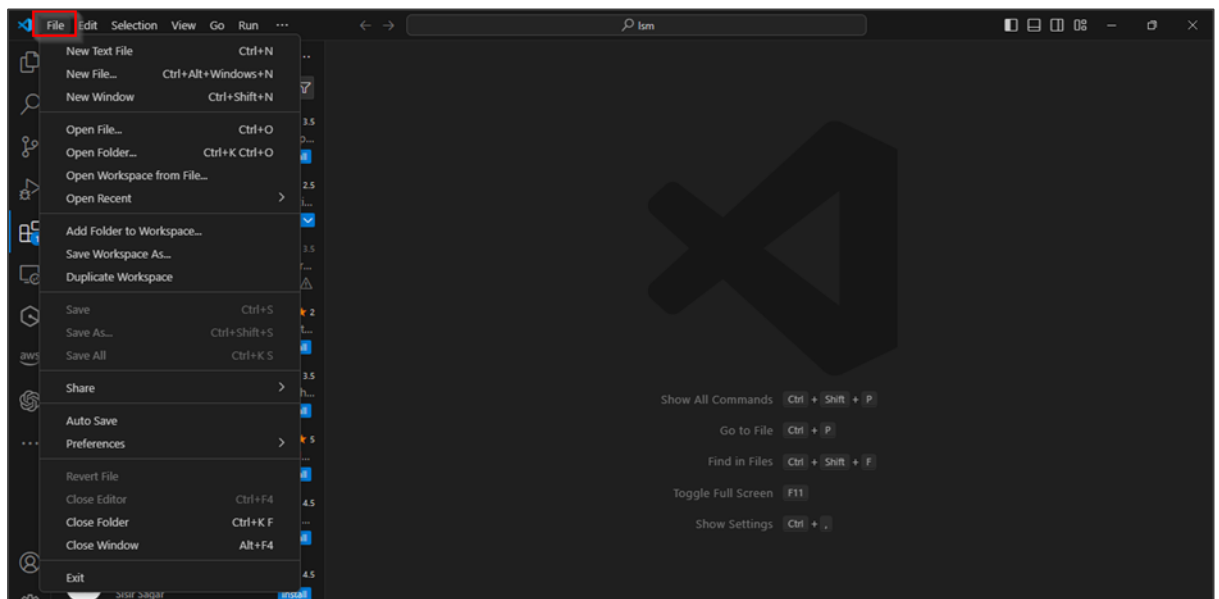
1.5 Click on the **Install** button to install the latest version of **HashiCorp Terraform V2.30.2** extension



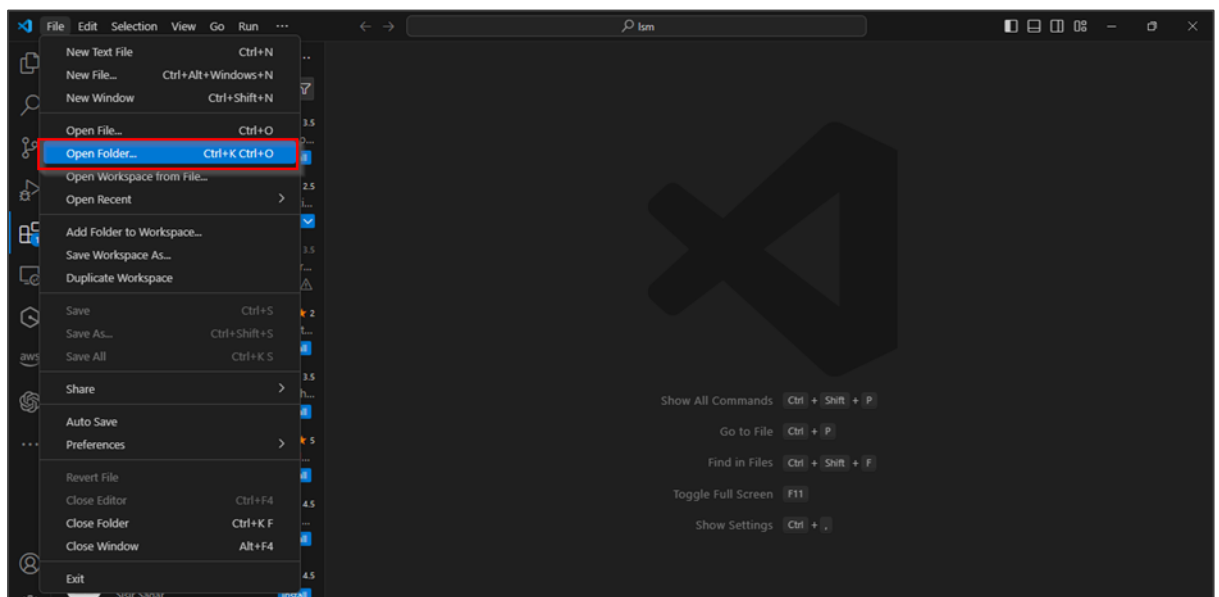
The latest version of the **HashiCorp Terraform V2.30.2** extension will be installed on your VS Code editor.

Step 2: Create the main Terraform configuration file

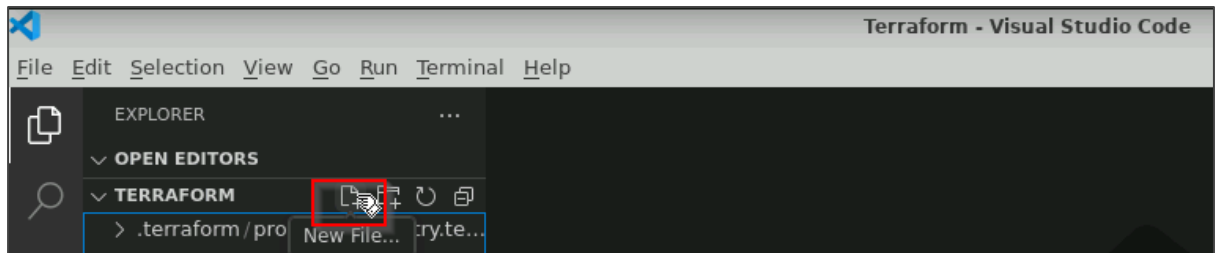
2.1 Click on the **File** option



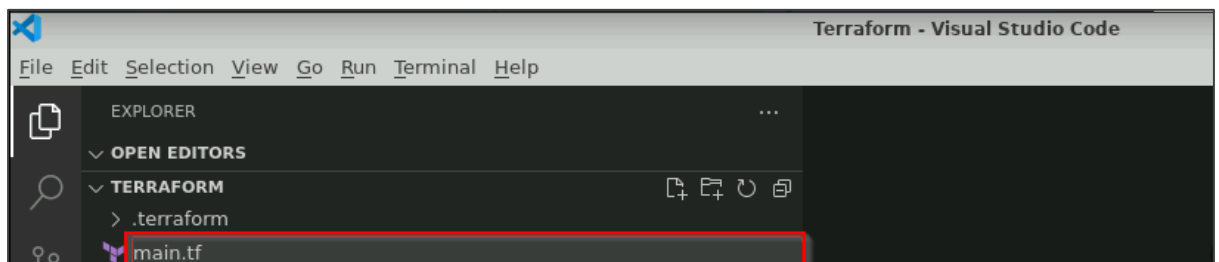
2.2 Select the **Open Folder** from the drop-down menu to open the **Terraform** folder



- 2.3 Navigate to the **Terraform** folder on the editor and click on the **New File** option to create the main file

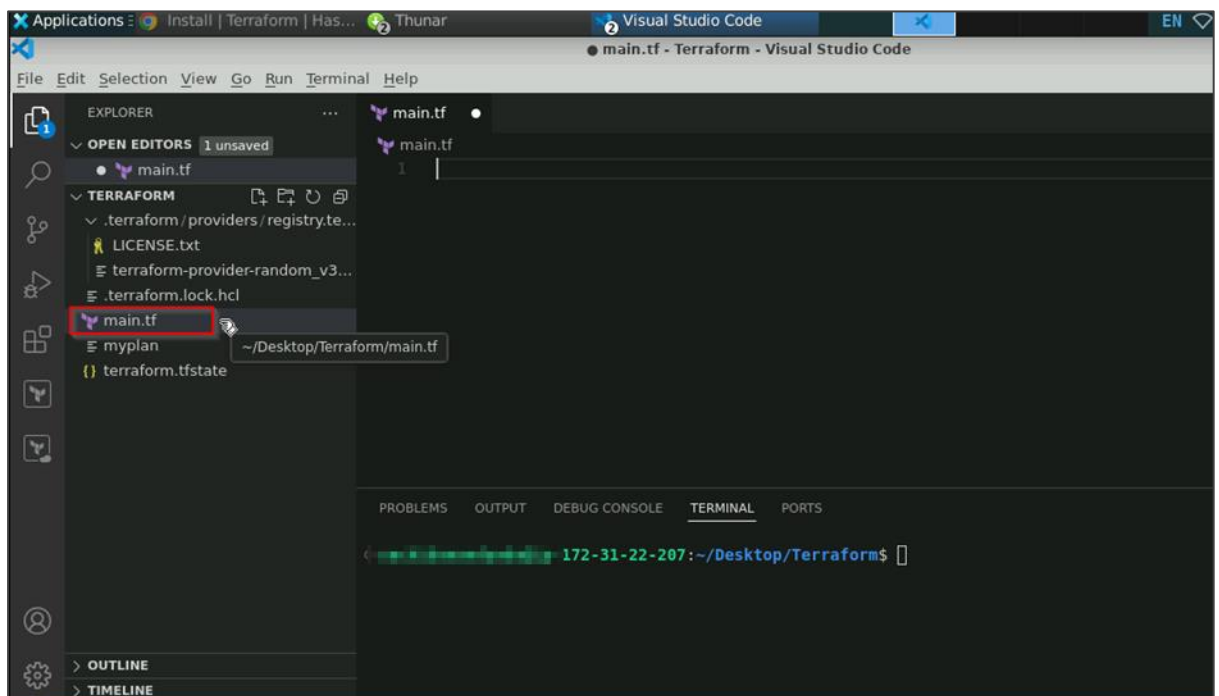


- 2.4 Enter the name of the file as **main.tf** to create the main Terraform configuration file



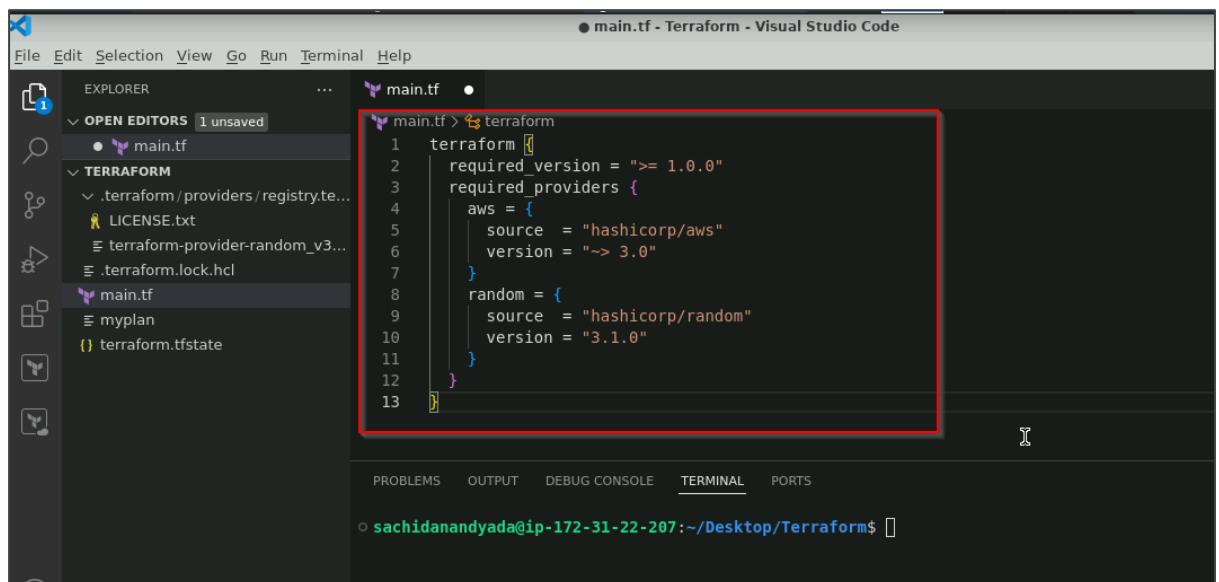
Step 3: Configure the Terraform providers

- 3.1 Open the **main.tf** file to specify the Terraform provider version configurations



3.2 Enter the following code in the **main.tf** file to specify the Terraform provider configurations and save it:

```
terraform {  
  required_version = ">= 1.0.0"  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
      version = "~> 3.0"  
    }  
    random = {  
      source = "hashicorp/random"  
      version = "3.1.0"  
    }  
  }  
}
```

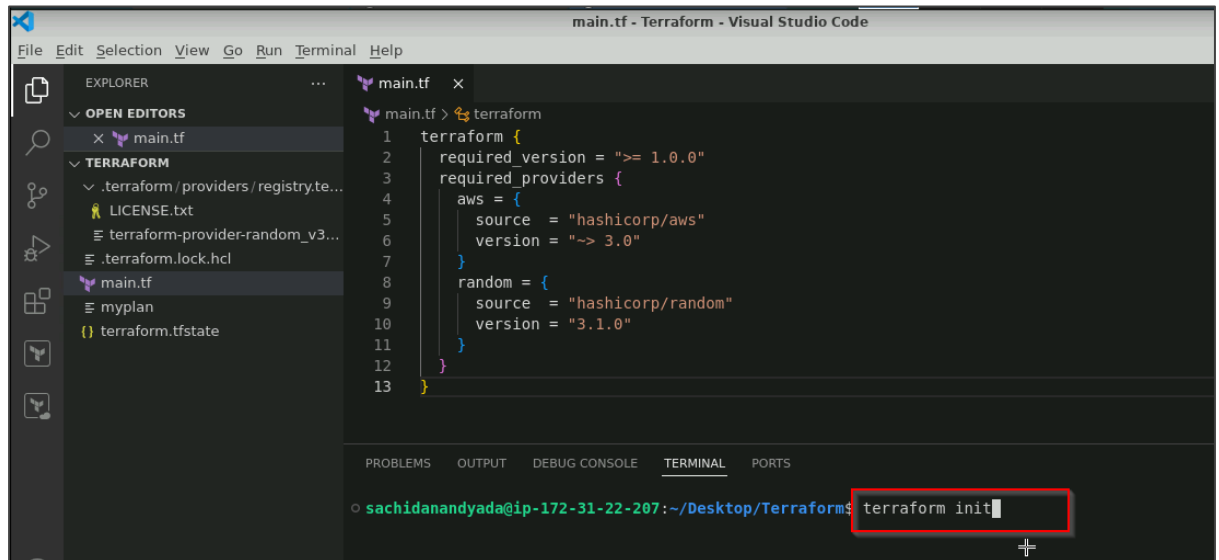


Note: You can get the configuration code of any Terraform provider from the official website of **HashiCorp Terraform** on the **Terraform Registry** page.

Step 4: Initialize the Terraform configuration file

4.1 Open the terminal and execute the following command to initialize the Terraform configurations:

terraform init

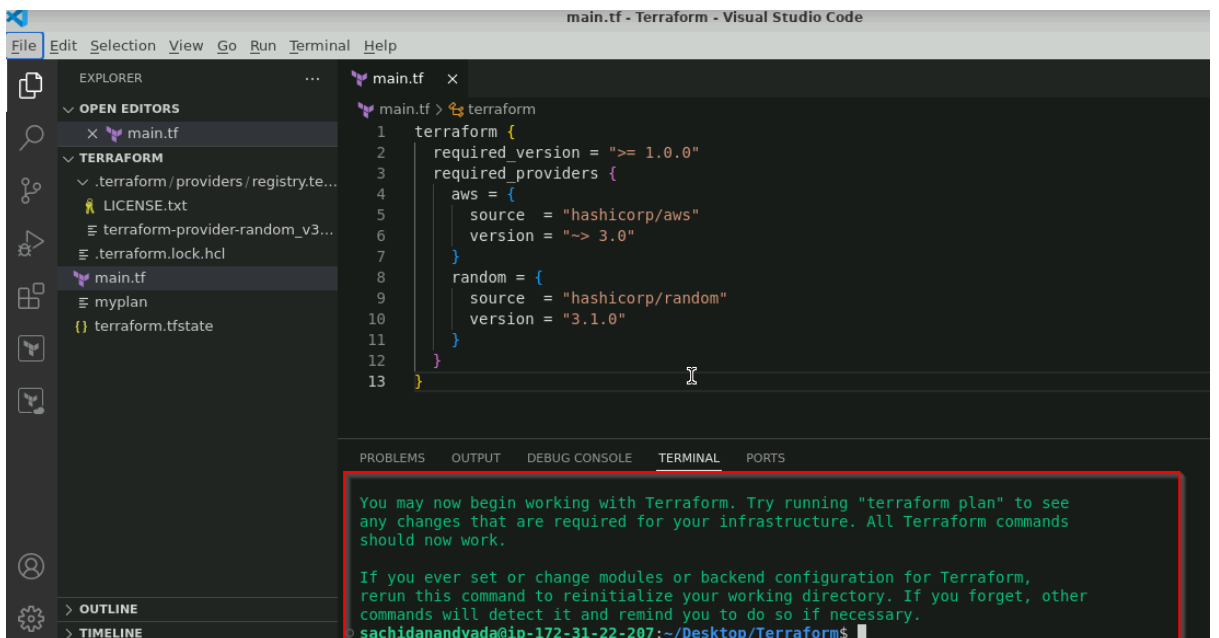


The screenshot shows the Visual Studio Code interface with the 'main.tf' file open. The file contains the following Terraform configuration:

```
1 terraform {
2   required_version = ">= 1.0.0"
3   required_providers {
4     aws = {
5       source = "hashicorp/aws"
6       version = "~> 3.0"
7     }
8     random = {
9       source = "hashicorp/random"
10      version = "3.1.0"
11    }
12  }
13 }
```

The terminal at the bottom shows the command `terraform init` being entered at the prompt `sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$`.

The Terraform configuration file is successfully initialized.



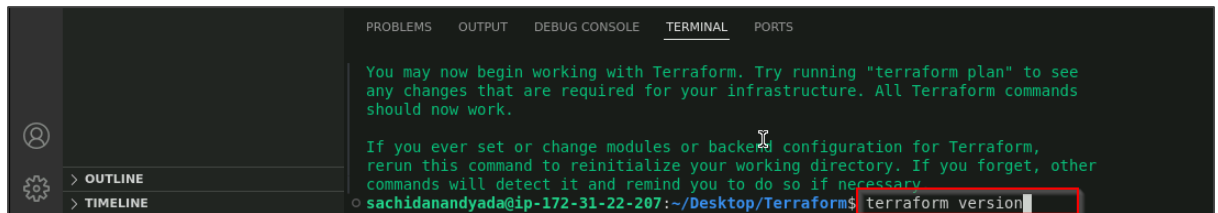
The screenshot shows the Visual Studio Code interface with the 'main.tf' file open. The terminal at the bottom displays the output of the `terraform init` command:

```
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.
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$
```

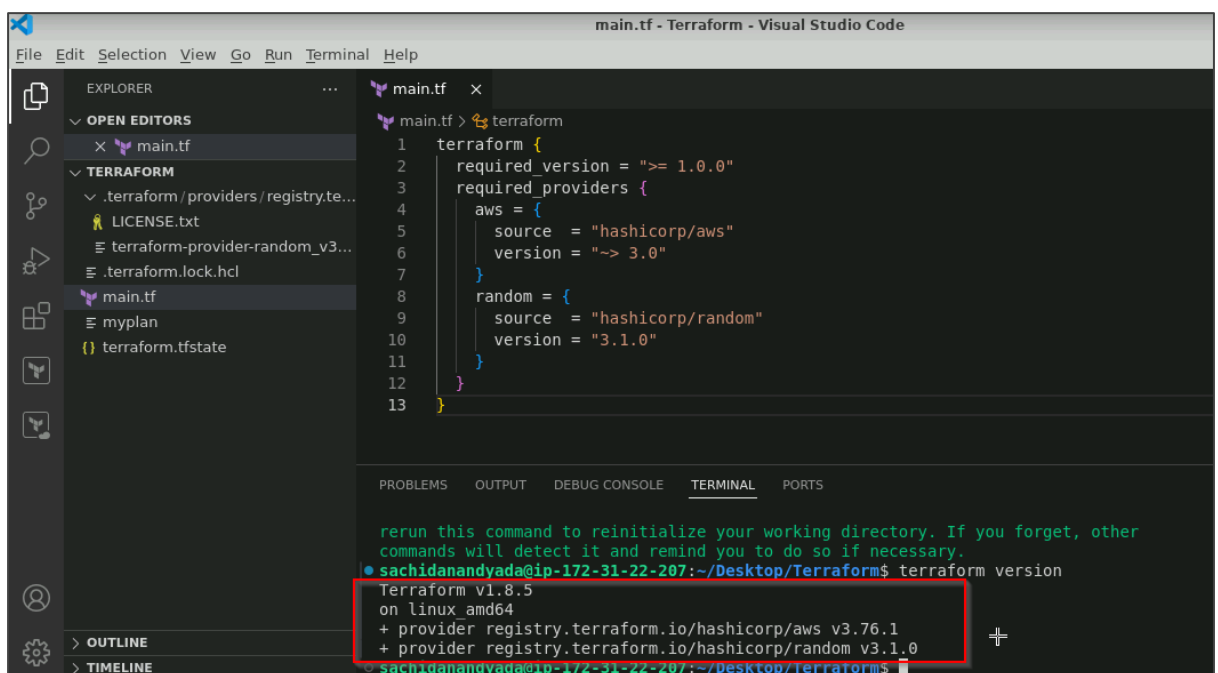
4.2 Execute the following command to verify the installed version of Terraform and its providers:

terraform version



The screenshot shows a terminal window in Visual Studio Code. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. It displays a message: "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." followed by "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." The prompt is "sachidanandyada@ip-172-31-22-207:~/Desktop/Terraforms\$". The command "terraform version" is being typed at the prompt, with the cursor at the end of the command.

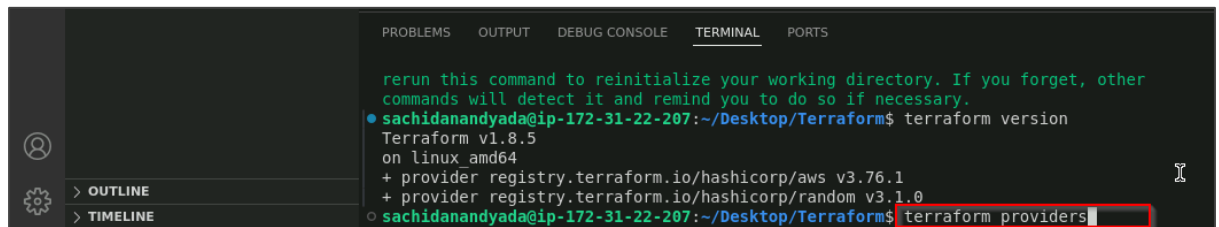
The installed terraform version and its providers is visible.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with files like LICENSE.txt, terraform-provider-random_v3..., .terraform.lock.hcl, main.tf, myplan, and terraform.tfstate. The main editor shows the content of main.tf, which is a Terraform configuration for the terraform provider. It specifies required_version = ">= 1.0.0" and required_providers for aws (source = "hashicorp/aws", version = "~> 3.0") and random (source = "hashicorp/random", version = "3.1.0"). The terminal at the bottom shows the output of the "terraform version" command: "Terraform v1.8.5 on linux_amd64 + provider registry.terraform.io/hashicorp/aws v3.76.1 + provider registry.terraform.io/hashicorp/random v3.1.0". The output is highlighted with a red box.

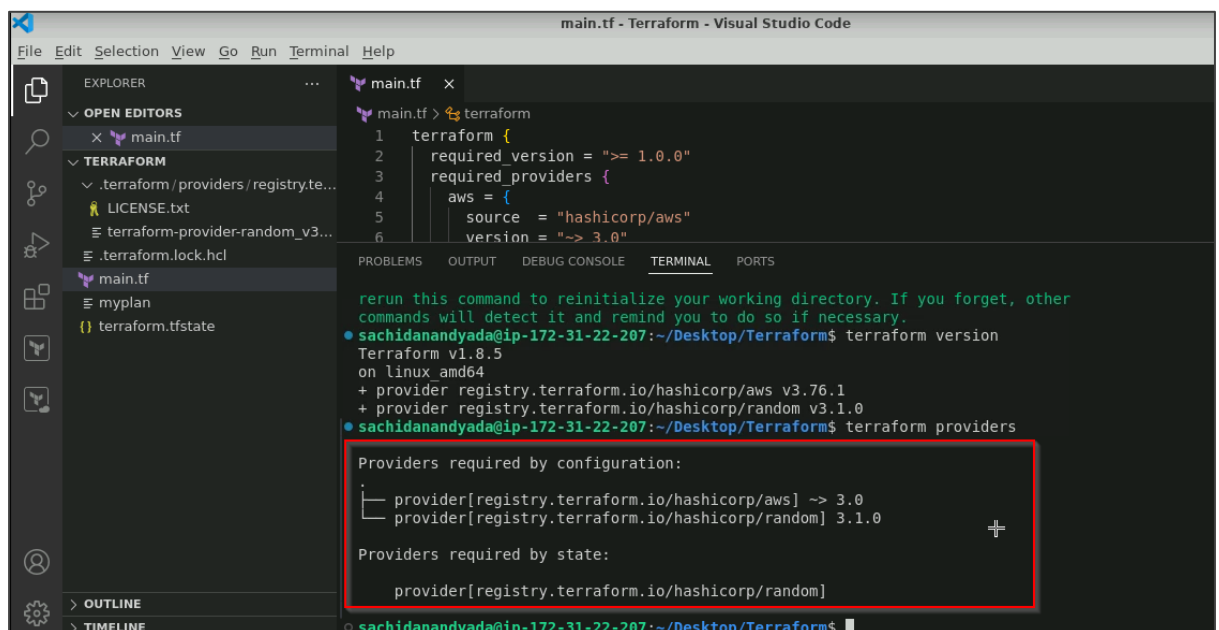
4.3 Execute the following command in the terminal to particularly check the versions of the installed Terraform providers:

terraform providers



```
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform version
Terraform v1.8.5
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v3.76.1
+ provider registry.terraform.io/hashicorp/random v3.1.0
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform providers
```

The installed Terraform provider version is visible.



```
main.tf
1 terraform {
2   required_version = ">= 1.0.0"
3   required_providers {
4     aws = {
5       source = "hashicorp/aws"
6       version = "~> 3.0"

```

```
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform version
Terraform v1.8.5
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v3.76.1
+ provider registry.terraform.io/hashicorp/random v3.1.0
sachidanandyada@ip-172-31-22-207:~/Desktop/Terraform$ terraform providers

Providers required by configuration:
└─ provider[registry.terraform.io/hashicorp/aws] ~> 3.0
   provider[registry.terraform.io/hashicorp/random] 3.1.0
+
Providers required by state:
   provider[registry.terraform.io/hashicorp/random]
```

By following these steps, you have successfully configured the AWS and the random Terraform providers in the Terraform configuration file for managing cloud infrastructure efficiently.