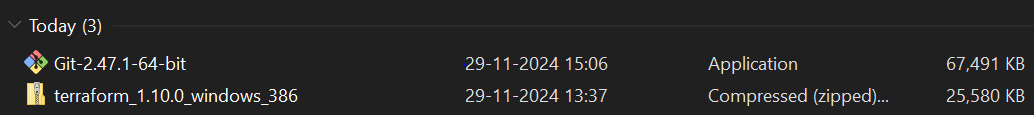
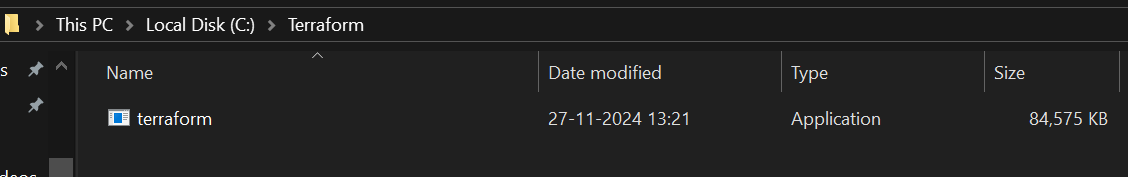
TERRAFORM[TASK1]

**1) Install Terraform on your PC**

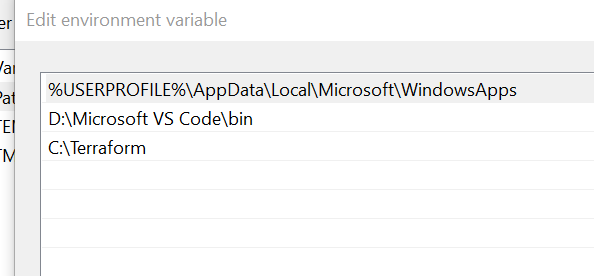
* Downloading the terraformhttps://developer.hashicorp.com/terraform/install?product\_intent=terraform

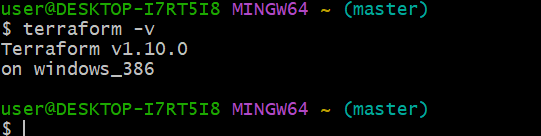


* Extracting the downloaded file,and pasted in the c:drive

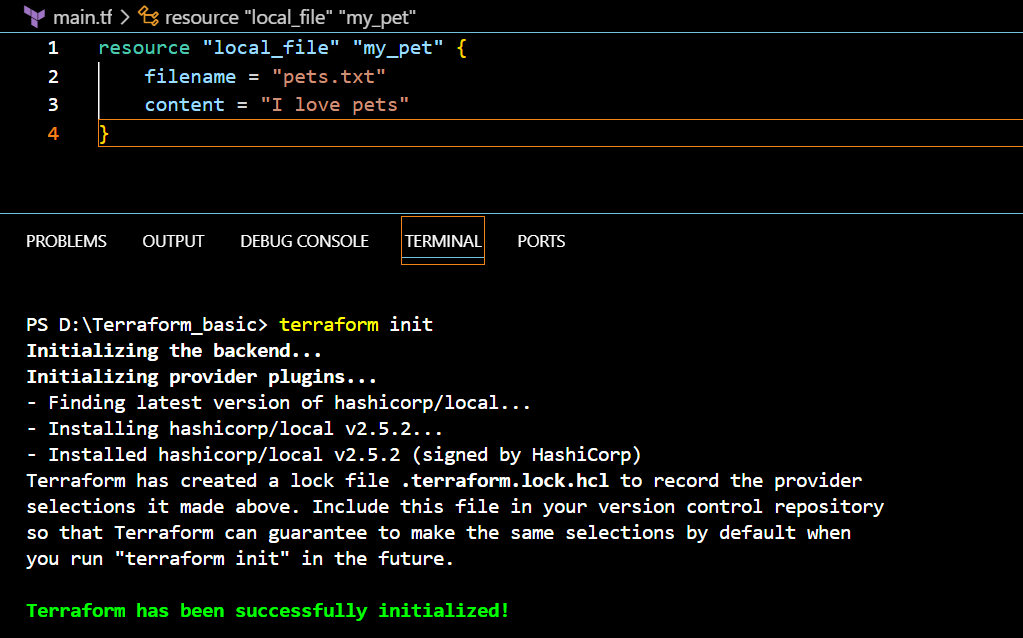


* Configure environment variables for terraform
* This PC(MyComputer)->properties ->advanced system settings-environment variables->system variable->path–edit->new paste the path



* Now we can see that terraform is installed.   
  

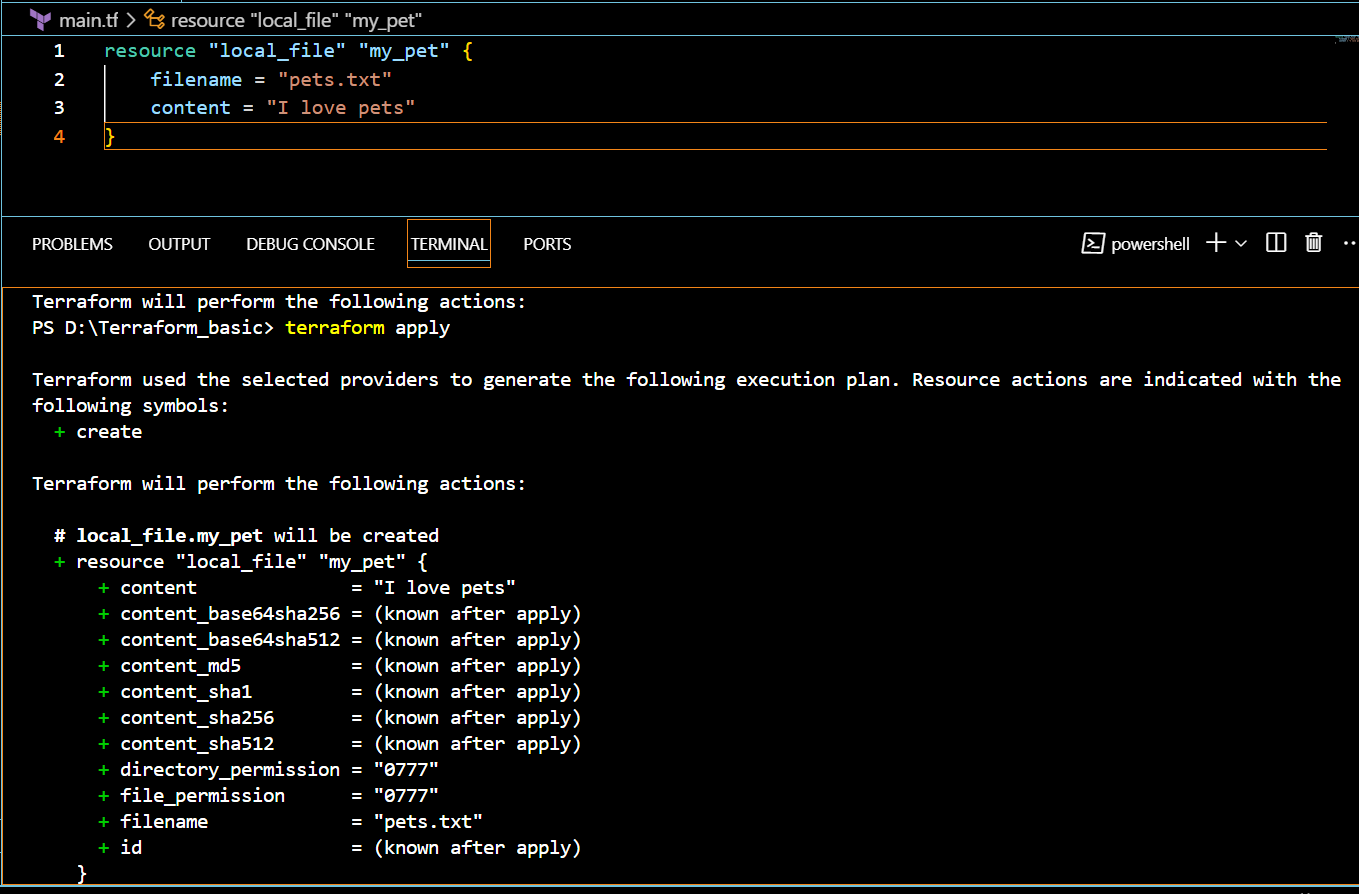
**2) Execute all the templates shown in video.**

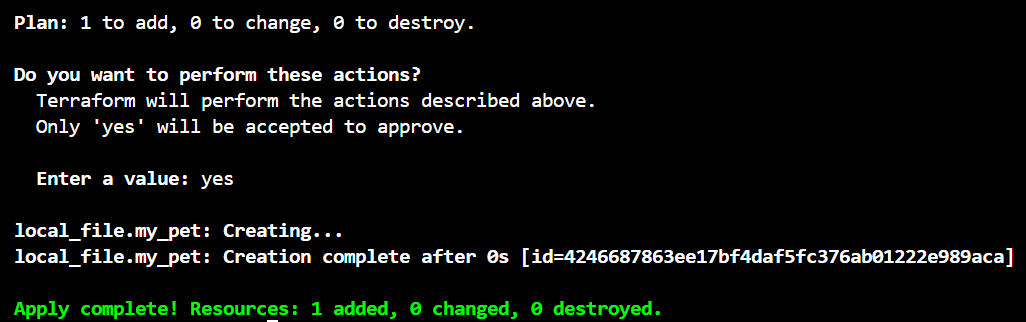
**Terraform Init**

**Terraform Plan**

It will help us to understand what changes we are going to make ,and that will not execute and will only show how the commands will execute on the source file

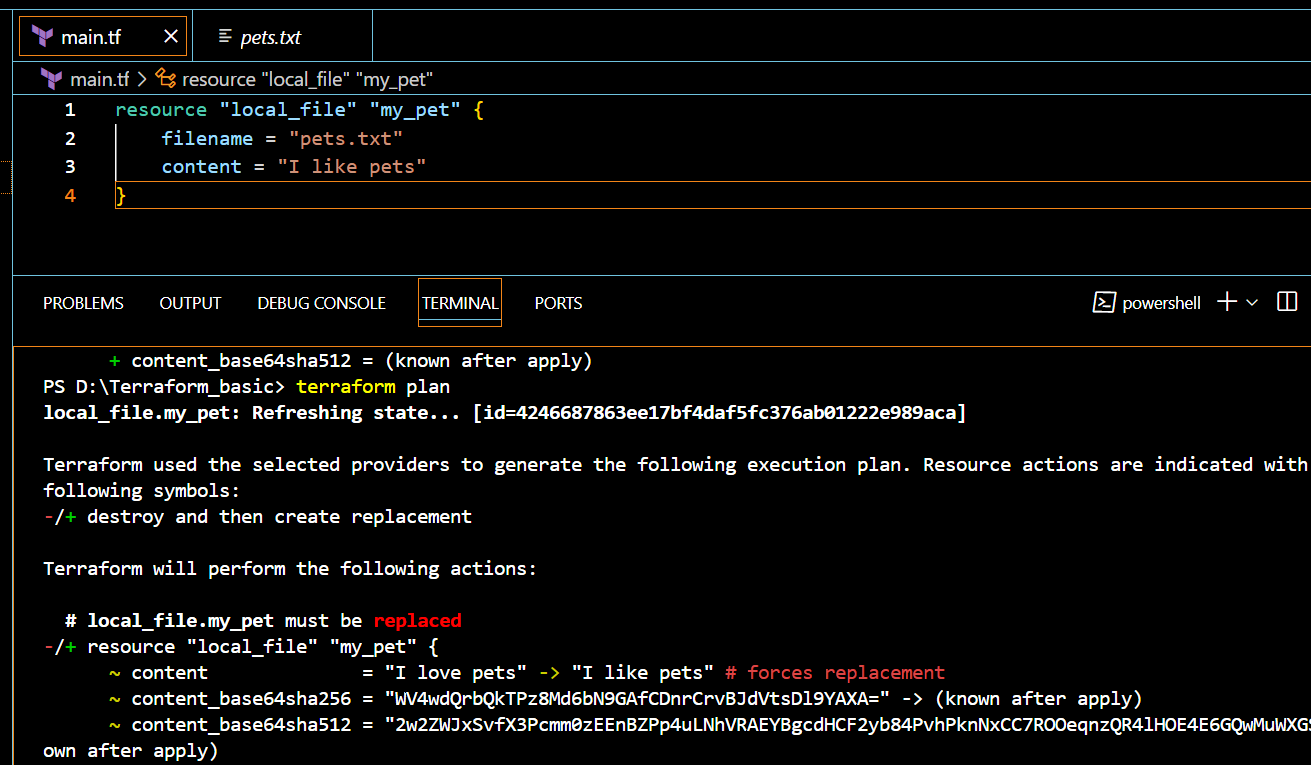


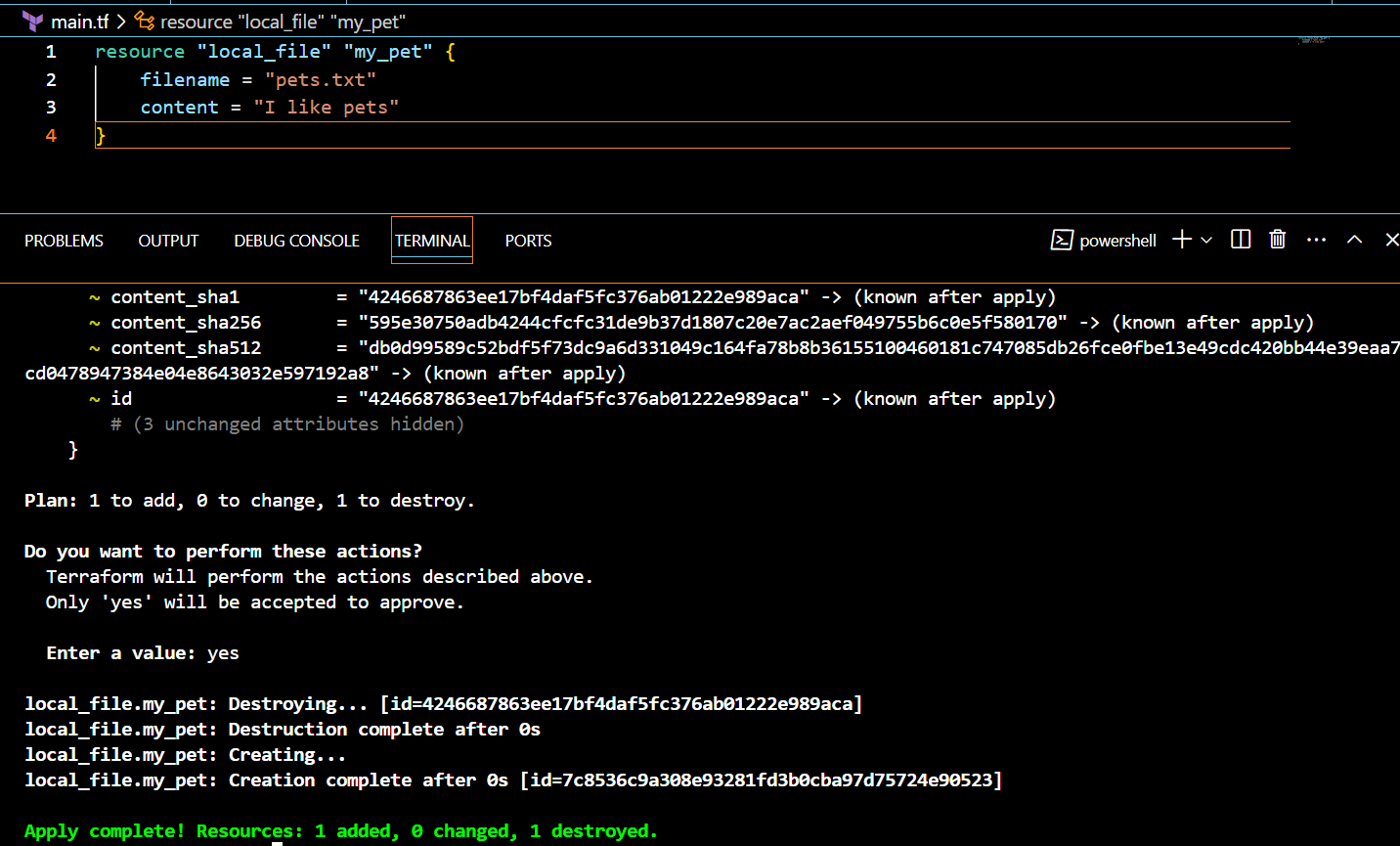
   Terraform Apply

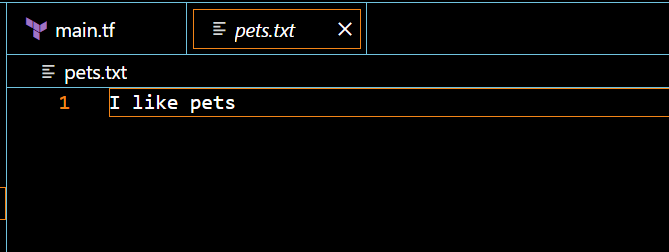




Changed the text in the main.tf



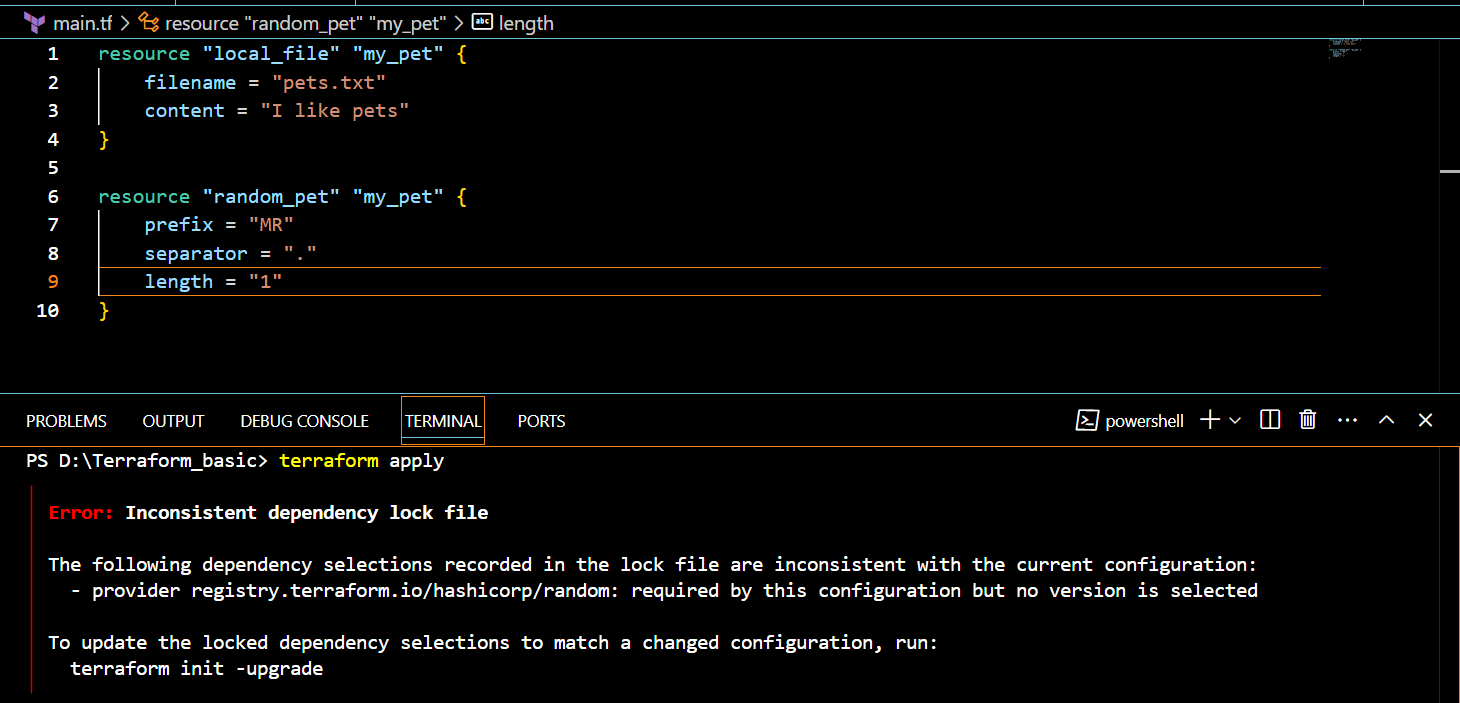
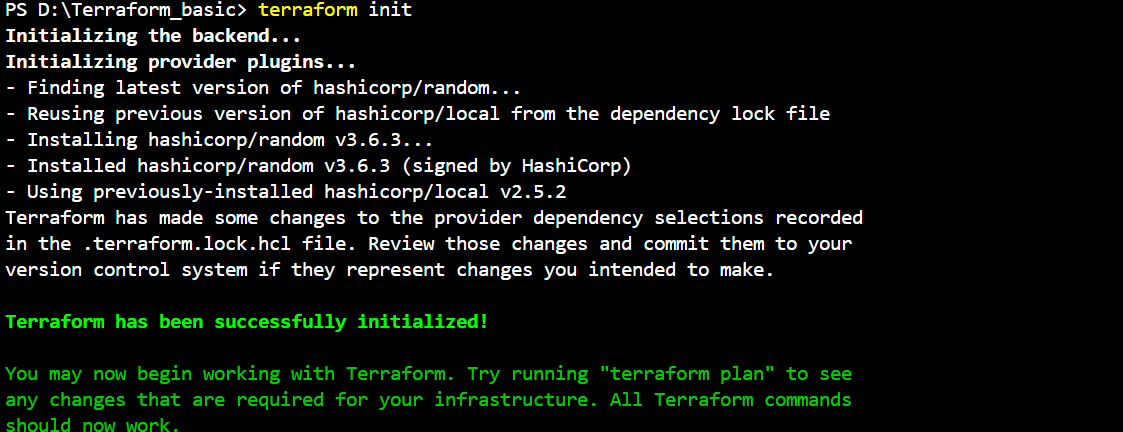


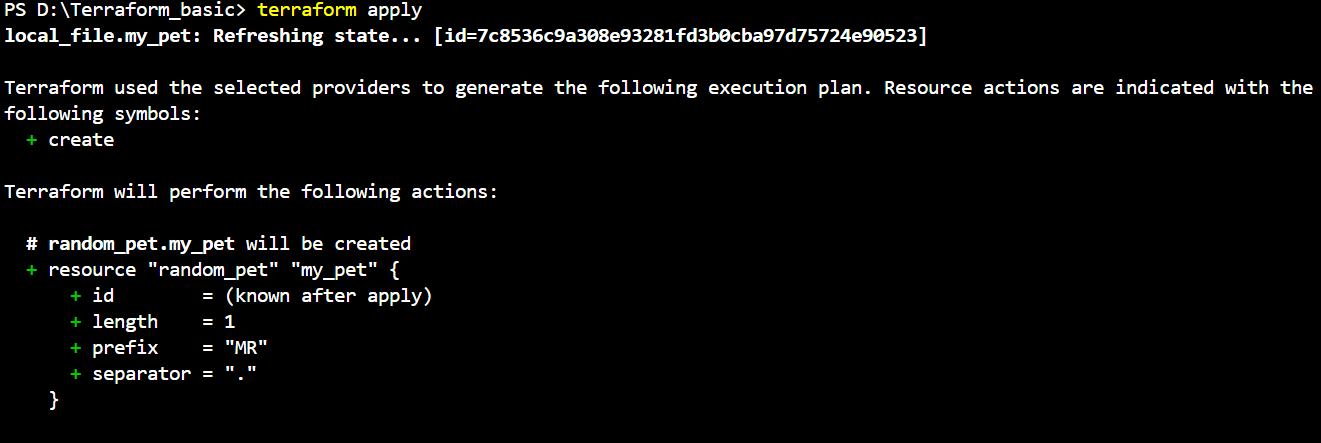


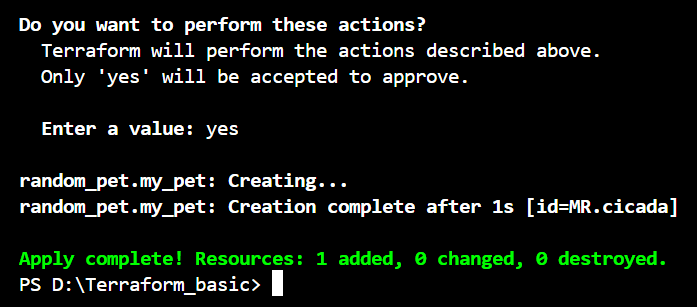
No changes in the main.tf



**Using random provider**

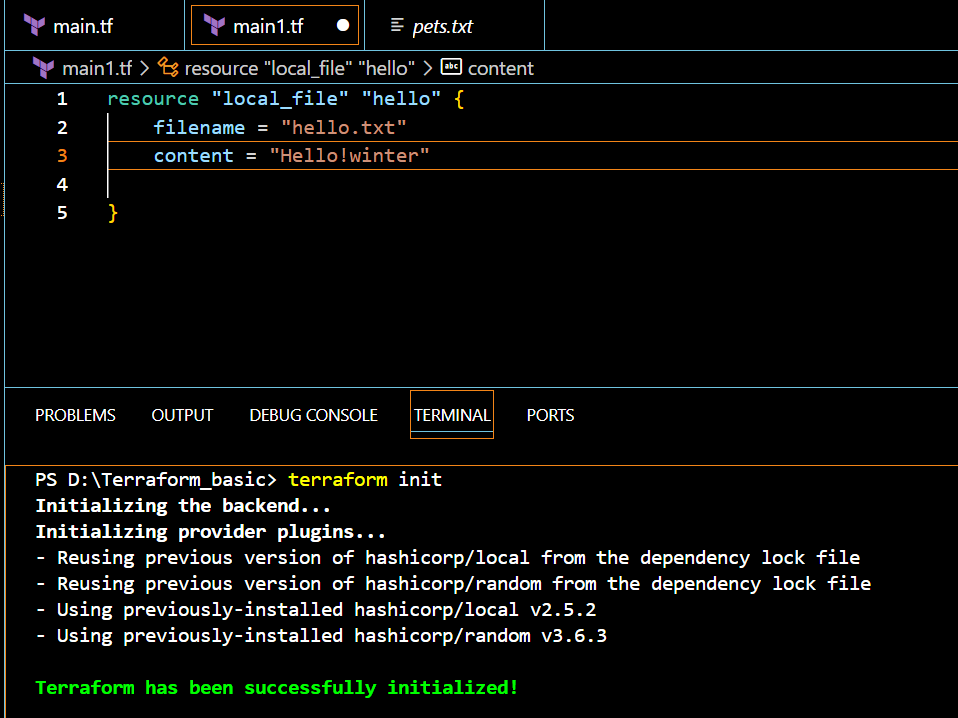




**3) Note down below points,**  Terraform init

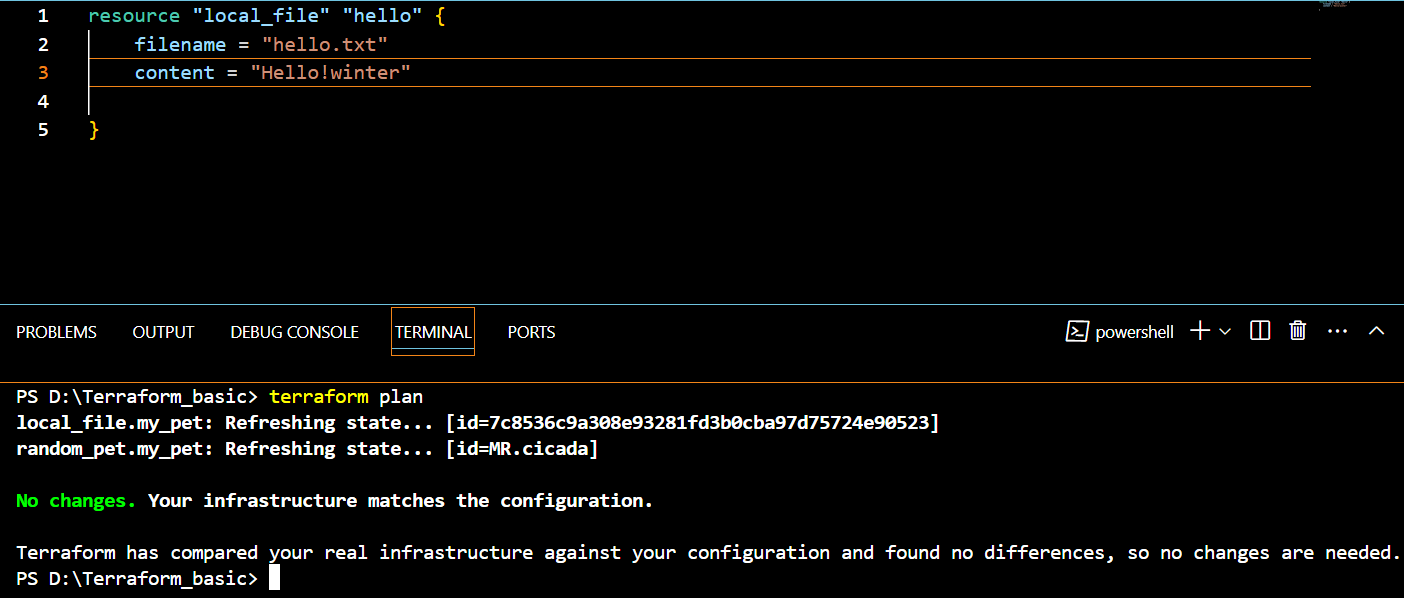
This command is like setting up a workspace.

* It prepares Terraform to work by downloading plugins/providers (e.g., AWS, Azure).
* Think of it as getting all the tools and materials ready before starting a project.



   **Terraform Plan**

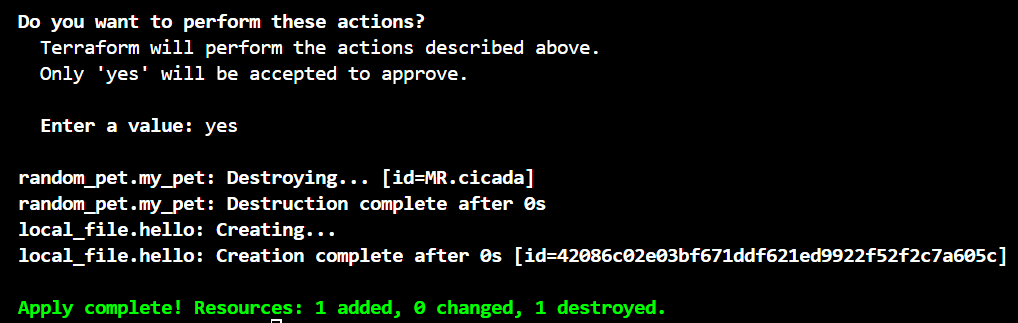
It will help us to understand what changes we are going to make ,and that will not execute and will only show how the commands will execute on the source file

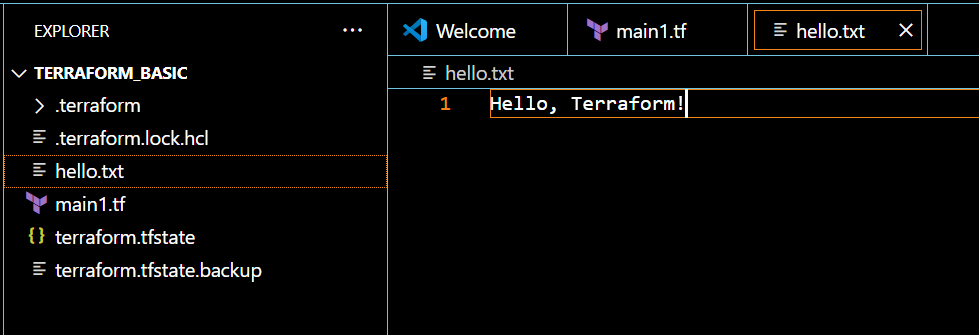


   Terraform Apply

This command is like actually building your project.

* It creates or changes resources (like servers, files, networks) based on the code you wrote.
* Think of it as hitting the "Go" button to make your infrastructure a reality





**Terraform Provider.**

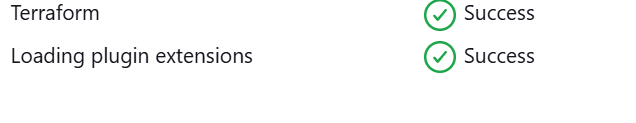
A provider in Terraform is a plugin that allows Terraform to interact with external services, APIs, and cloud providers:

**Function**

Providers act as a translation layer that allows Terraform to communicate with other services. They implement Terraform's resource type definitions and perform CRUD (Create, Read, Update, Delete) operations

**4) Integrate a sample Terraform template in jenkins**. Steps:1

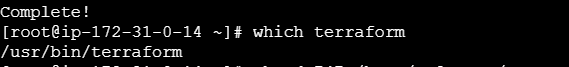
Installing terraform plugin in Jenkins



Step:2

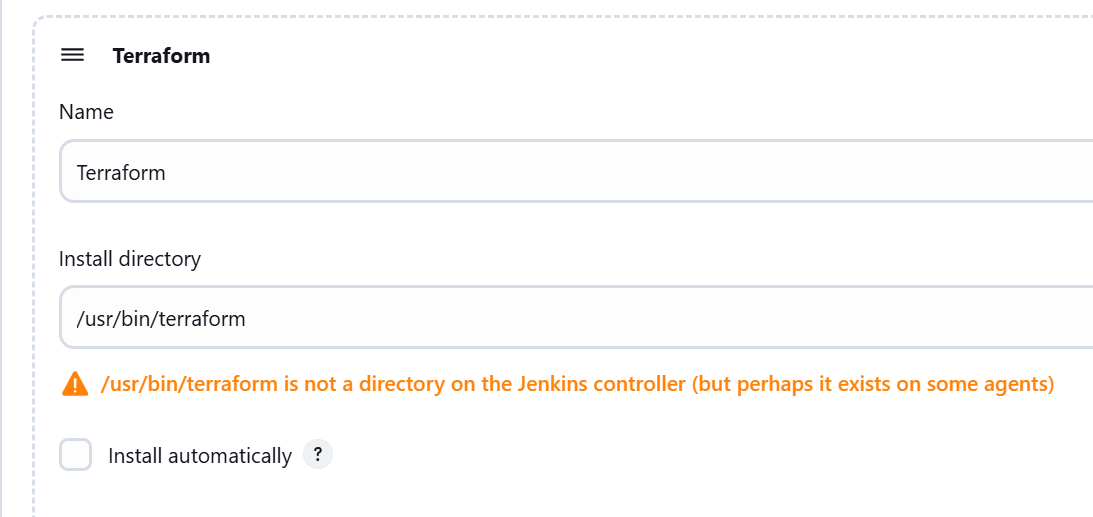
Installing terraform in the jenkins

* sudo yum install -y yum-utils shadow-utils
* sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
* sudo yum -y install terraform

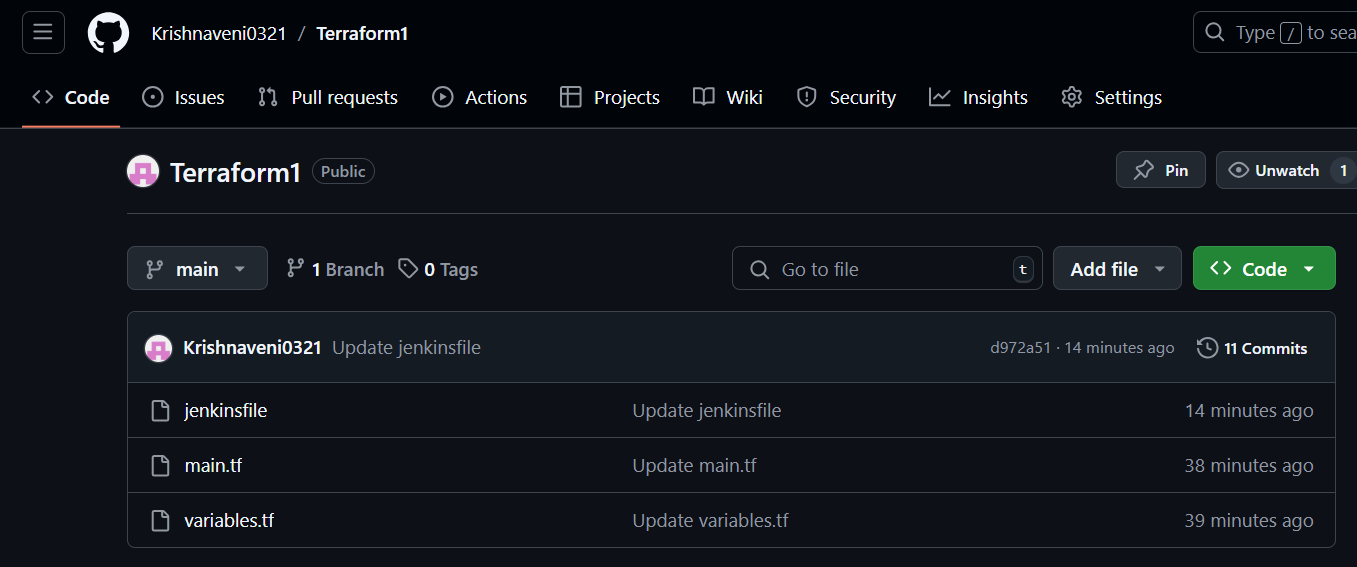


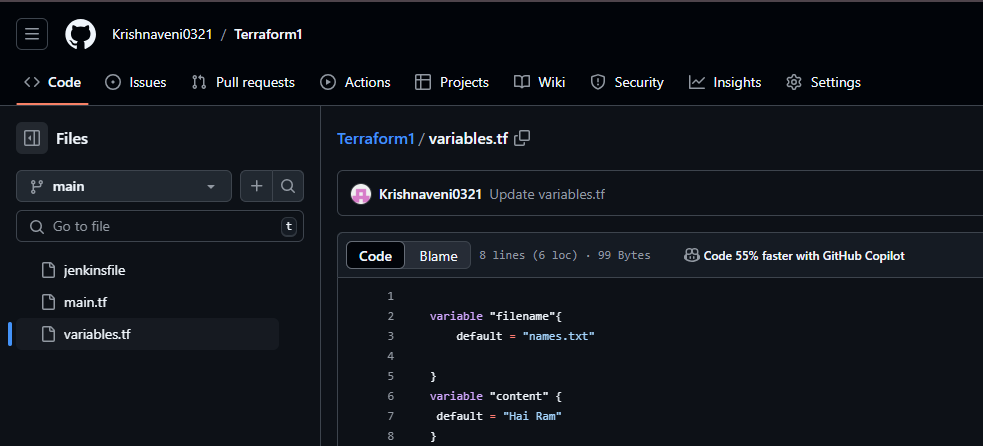
Step-3

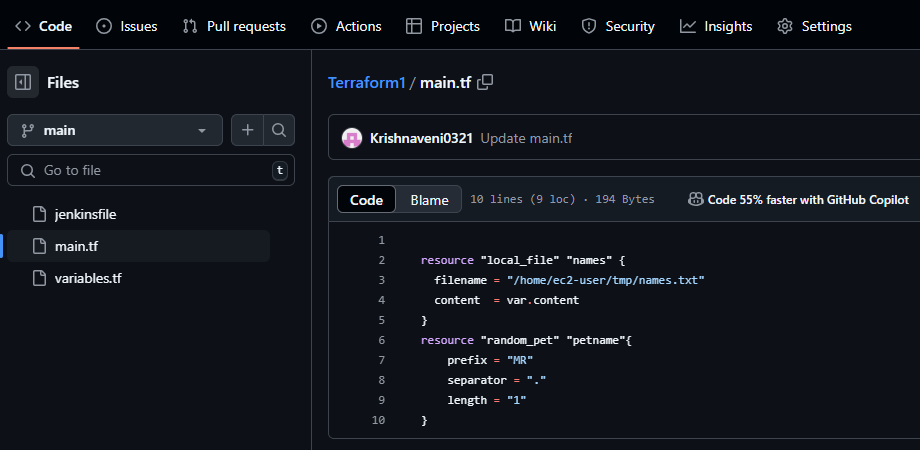
Go to manage Jenkins>tools

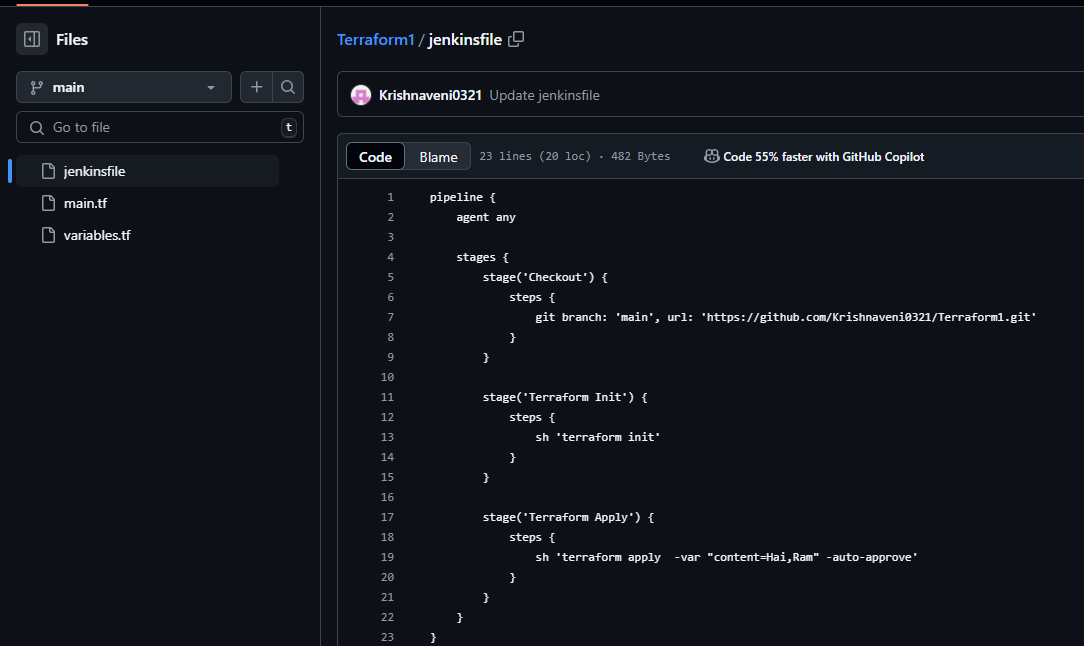


Now go to the github,

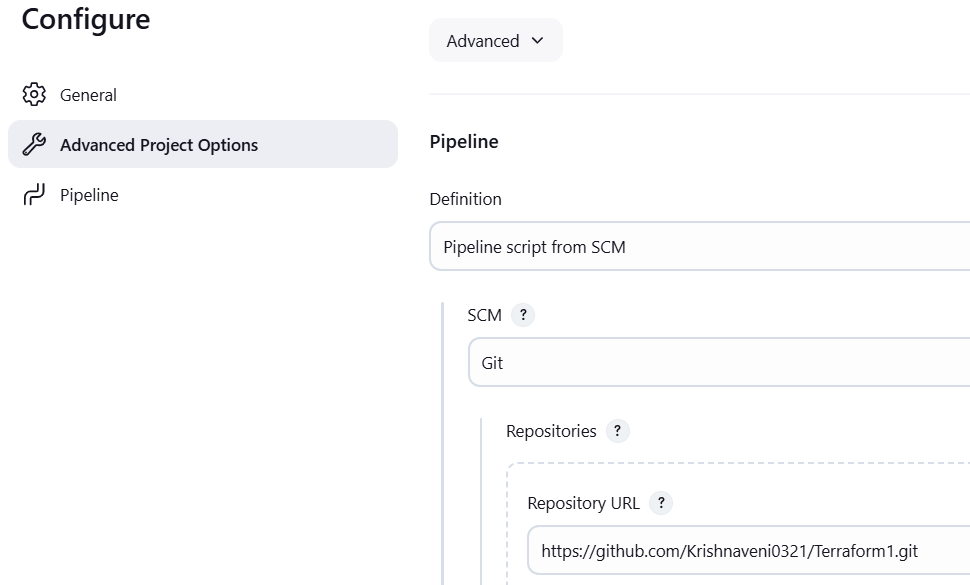




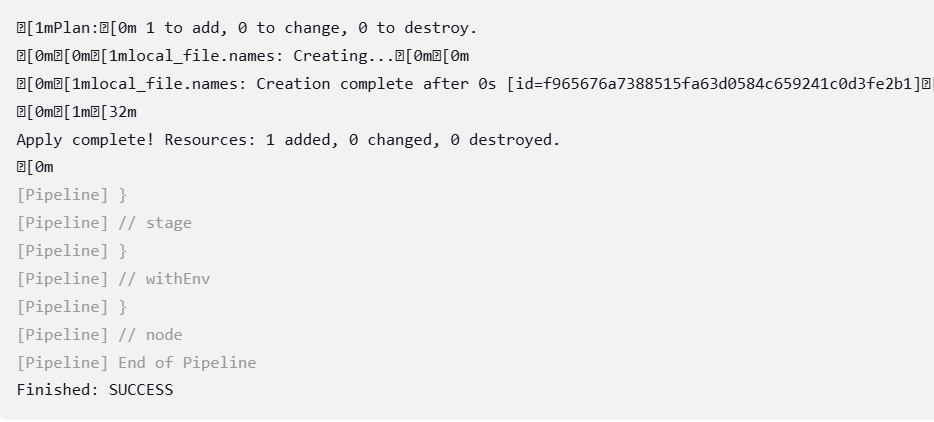




Creating a job,



Execute the job



We can also check that in Jenkins server

