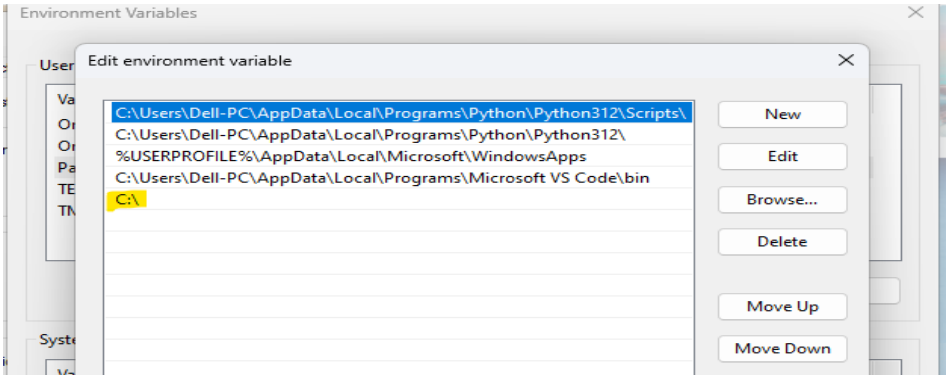
TERRAFORM

>>INSTALL TERRAFORM FROM https://developer.hashicorp.com/terraform/install

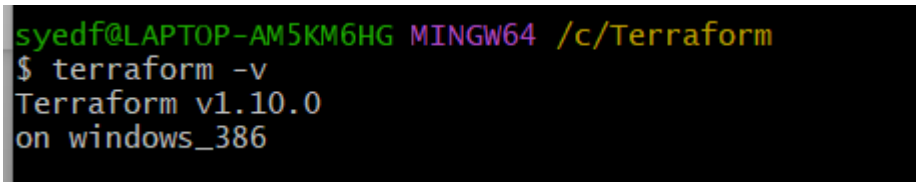
>>AFTER DOWNLOAD MOVE IT TO C:\TERRAFORM IN THE WINDOWS MACHINE

>>NOW GO TO SEARCH BAR AND SEARCH FOR EDIT SYSTEM ENVIRONMENT VARIABLES

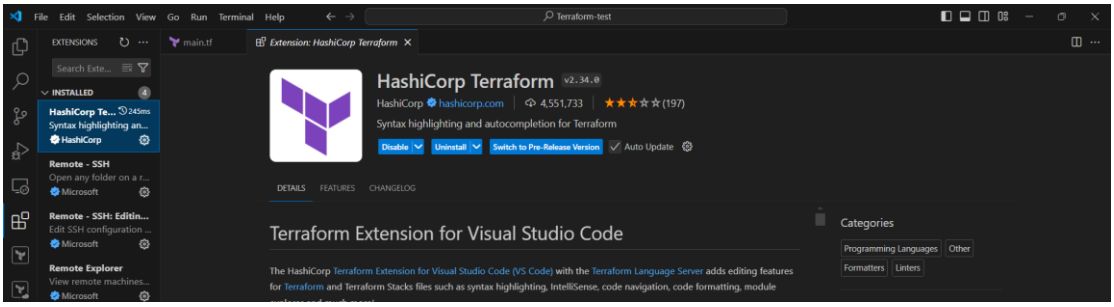
>>ADD THE EVIRONMENT VARIABLE AND SELECT THE PATH AS C:\TERRAFORM



>>OPEN GITBASH AND CHECK THE WITH COMMAND terraform -version



>>GO TO VISUAL CODE STUDIO AND ADD EXTENTION HASHI CORP BY INSTALLING IT



>>CREATE A FOLDER FOR ALL YOUR CODE

>>OPEN IT IN VISUAL CODE

>>ADD A FILE WITH .tf EXTENTION

→ resource --> Block name

→local -->provider

→file -->resource type (What needs to be created)

→fruits--> it is a logical name to identify by terraform and can be named anything.

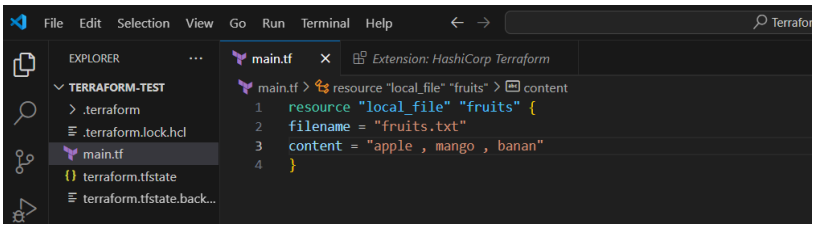
>> WRITE THE BASIC CODE

resource "local_file" "fruits" {

filename = "fruits.txt"

content = "apple , mango , banana"

}



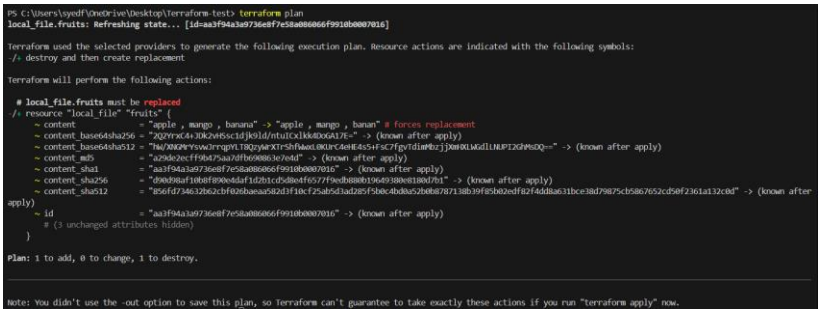
*SAVE THE CODE WITH Ctrl+S

ONCE WE HAVE THE TERRAFORM RESOURCE FILE THEN WE NEED TO PERFORM BELOW COMMANDS

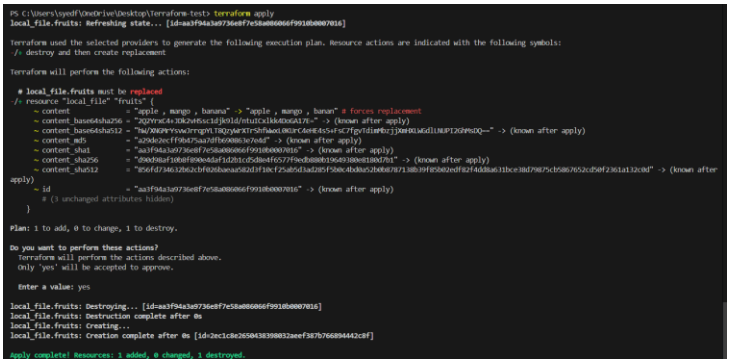
terraform init --> TO INITLIAZE THE REPOSITORY AND DOWNLOAD THE DEPENDENCIES

```
PS C:\Users\syedf\OneDrive\Desktop\Terraform-test> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.2

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.
```

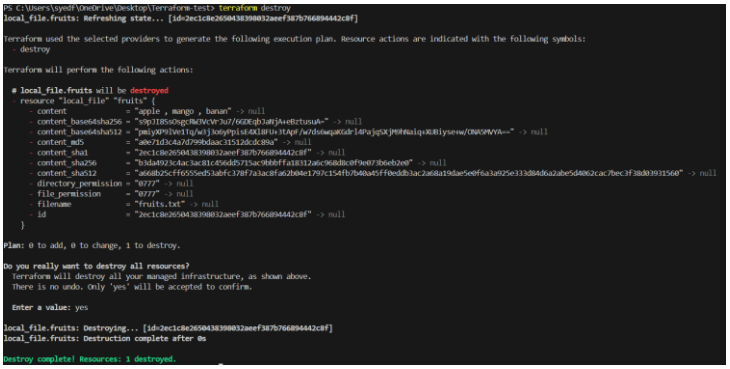
terraform plan --> DRY RUN TO CHECK HOW IT WILL BE CREATE THE RESPOURCE (IT WILL NOT ACTUALLY CREATE THEM)



terraform apply --> TO EXECUTE AND CREATE THE INFRA BASED ON THE CONFIGURATION



terraform destroy --> TO REMOVE/DELETE THE EXISTING INFRA.



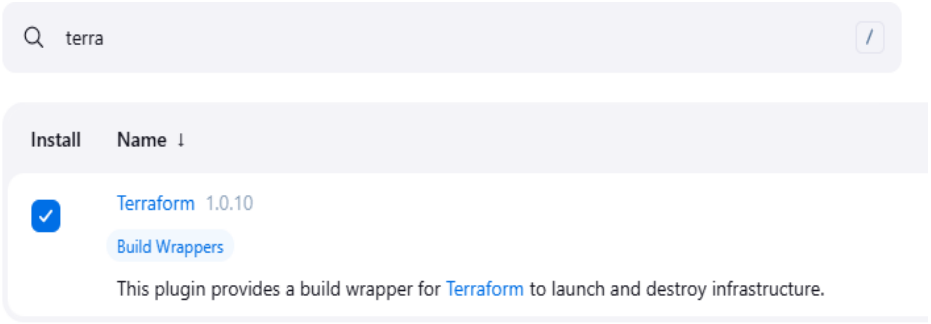
terraform show --> TO SHOW/GET THE DETAILS OF THE RESOURCES CREATED

```
PS C:\Users\syedf\OneDrive\Desktop\Terraform-test> terraform show
The state file is empty. No resources are represented.
```

>>INTEGRATING TERRAFORM WITH JENKINS

>>LAUNCH AN EC2 SERVER AND INSTALL JENKINS IN IT

>>ON JENKINS GUI INSTALL TERRAFORM PLUGIN.



>>INSTALL TERRAFORM ON JENKINS SERVER USING BELOW COMMANDS

yum install -y yum-utils shadow-utils

yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

yum -y install terraform

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : terraform-1.10.1-1.x86_64                              1/1
  Verifying        : terraform-1.10.1-1.x86_64                              1/1

Installed:
  terraform-1.10.1-1.x86_64

Complete!
[root@ip-10-0-0-19 ec2-user]# find / -name terraform
/usr/bin/terraform
/usr/share/doc/terraform
```

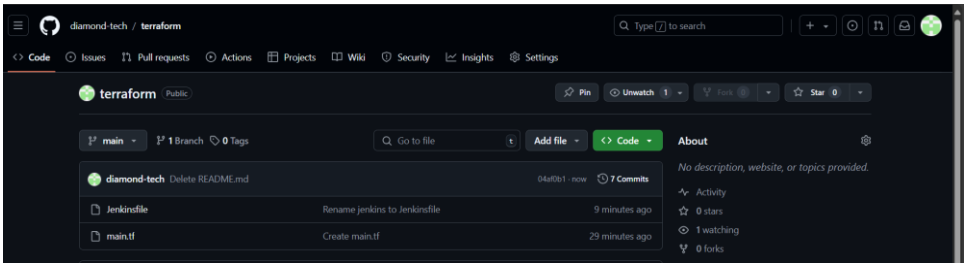
>>JENKINS GUI > MANAGE JENKINS > TOOLS > SCROLL DOWN > TERRAFORM > ADD /usr/bin/terraform

>>CREATE A GIT REPOSITORY

>>CREATE ANOTHER FILE NAMING Jenkinsfile **AND ADD THE CODE WITH THE REPOSITORY URL**

```
pipeline {
    agent any
    stages {
        stage('Clone') {
            steps {
                git branch: 'main', url: 'https://github.com/diamond-tech/terraform.git'
            }
        }
        stage('Initialize Terraform') {
            steps {
                sh 'terraform init'
            }
        }
        stage('Execute Terraform Apply') {
            steps {
                sh 'terraform apply -auto-approve'
            }
        }
    }
}
```

>>NOW PUSH THIS main.tf **AND** Jenkinsfile **TO GITHUB**



>>GO TO JENKINS AND CREATE A NEW ITEM

NEW ITEM > SELECT PIPELINE > PIPELINE WITH SCM > ADD GIT URL https://github.com/diamond-tech/terraform.git

>>BUILD

***(NOTE: MAKE SURE THE EC2-USER FILE IN JENKINS SERVER HAS** global rwx **PERMISSIONS 777)**

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[0m
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```