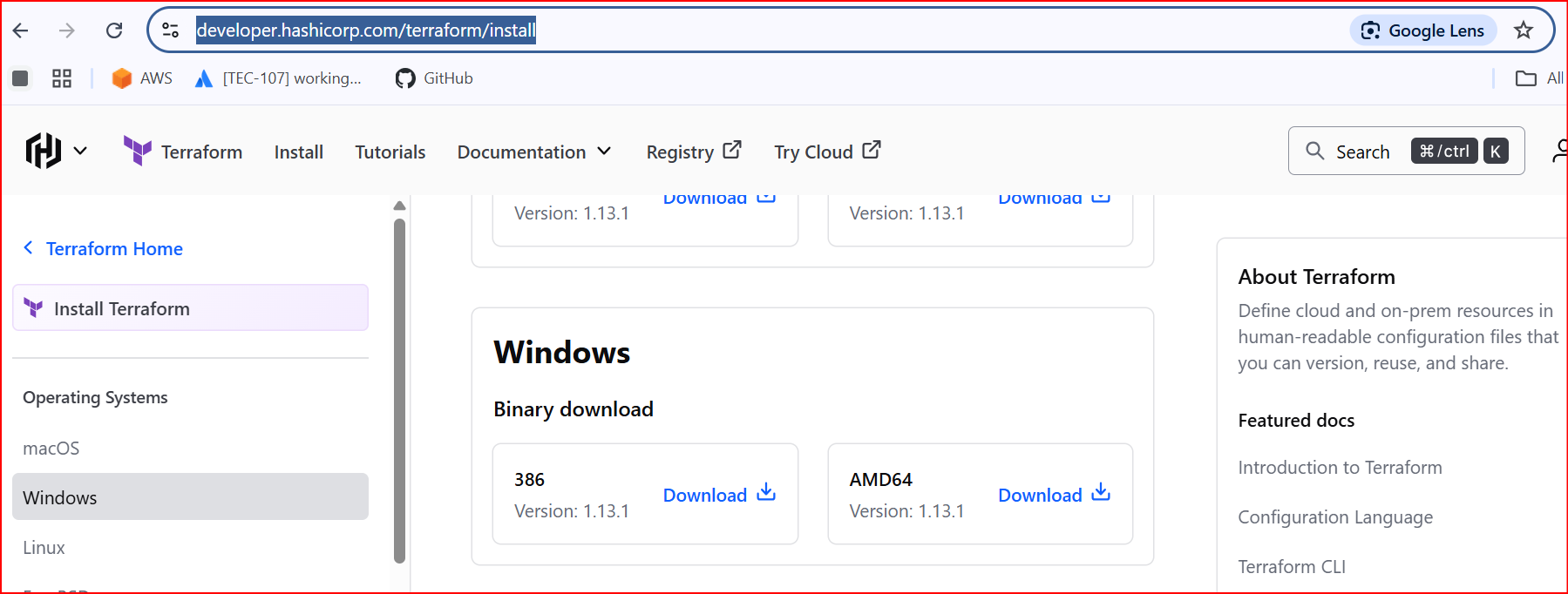
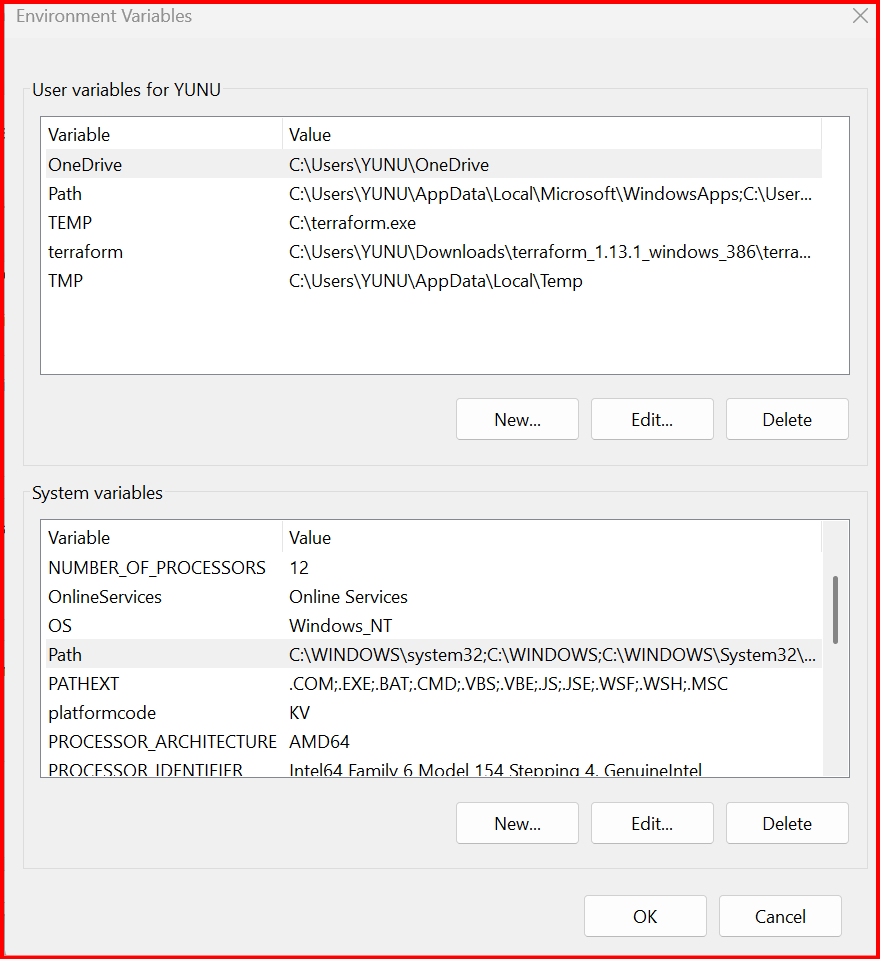
**Terraform Task-1**

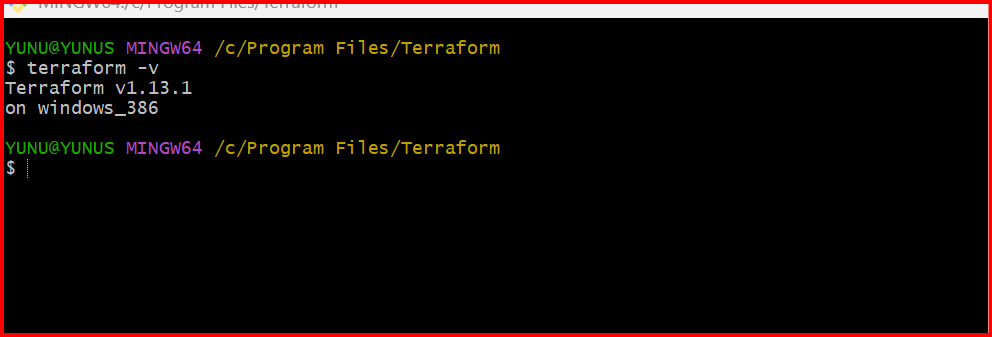
1. **Install Terraform on your PC**
2. Download the Terraform from the below link

<https://developer.hashicorp.com/terraform/install>

1. Unzip the terraform package
2. Configure environment variables for terraform
3. Verify terraform version







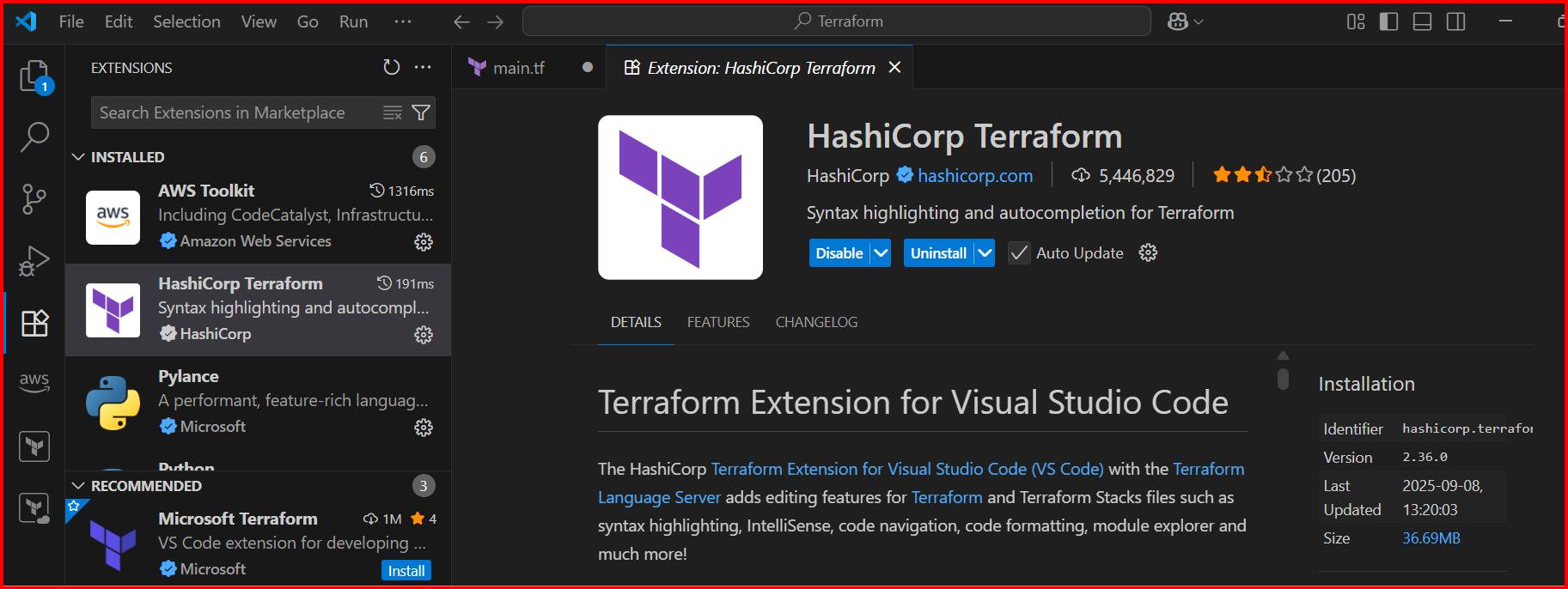
**2.Execute all the templates shown in video.**

First, we should create one folder in Desktop

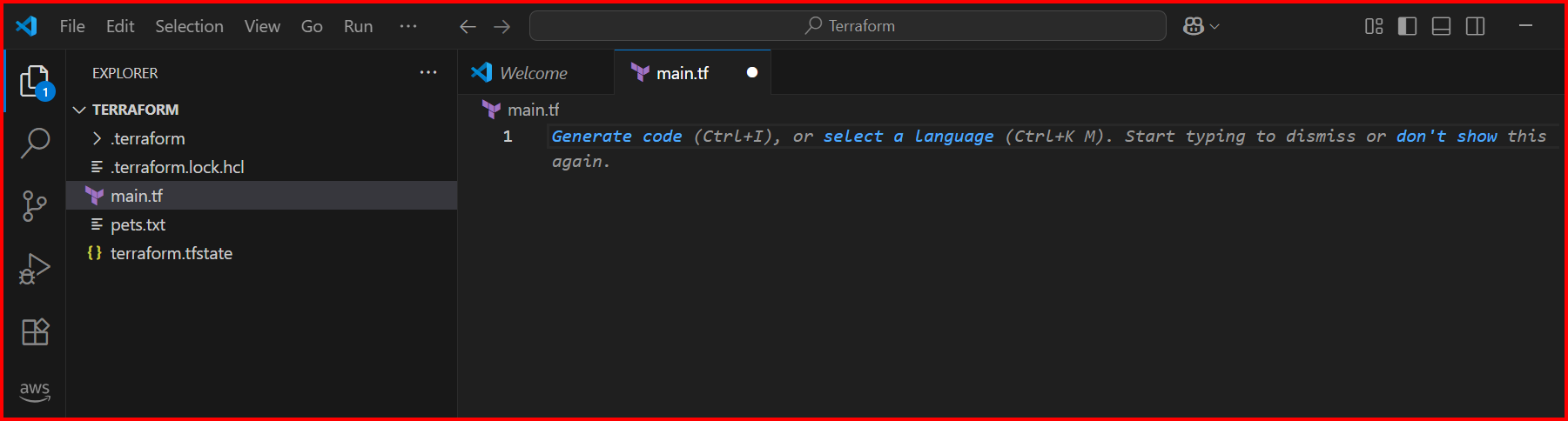
Open visual studio and select created folder

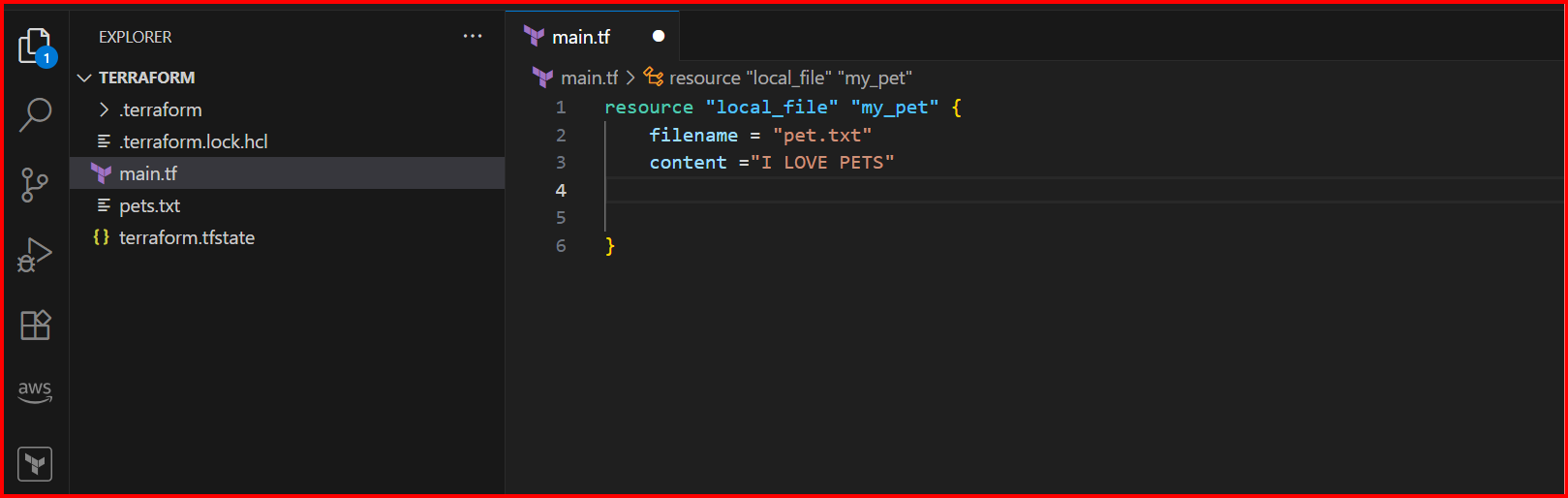
In Extensions search **Hashi Corp Terraform** and Install this Plugin

This will help us to correct the speeling mistakes spaces this will deacted automatically and it will give suggestions



Create one file extension should be .tf



****

**Resource 🡪 block**

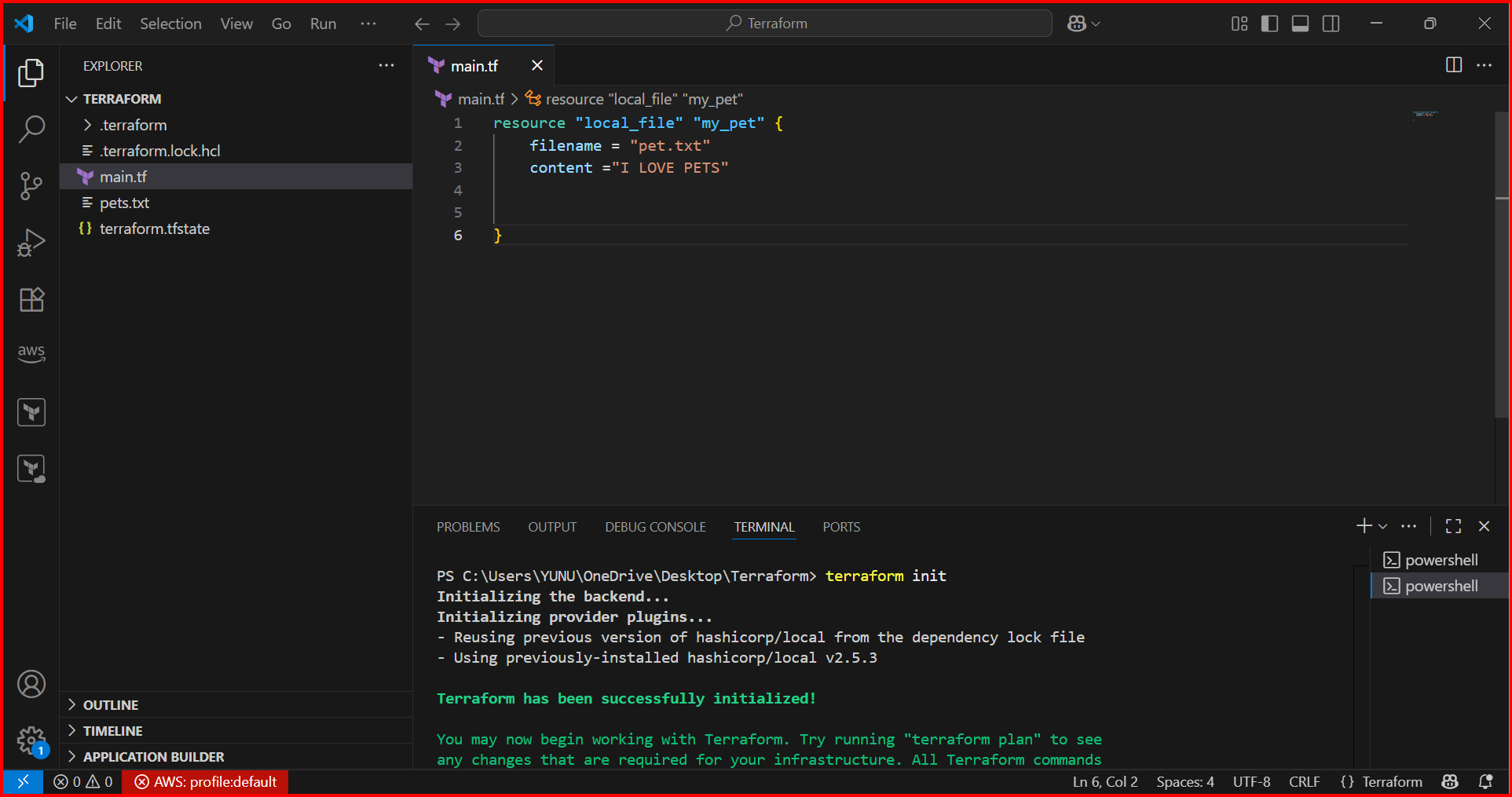
**Local 🡪 Provider**

**Type 🡪 type of resource**

**My\_pet 🡪name of terraform**

**Filename & content 🡪 attributes used for the resource**

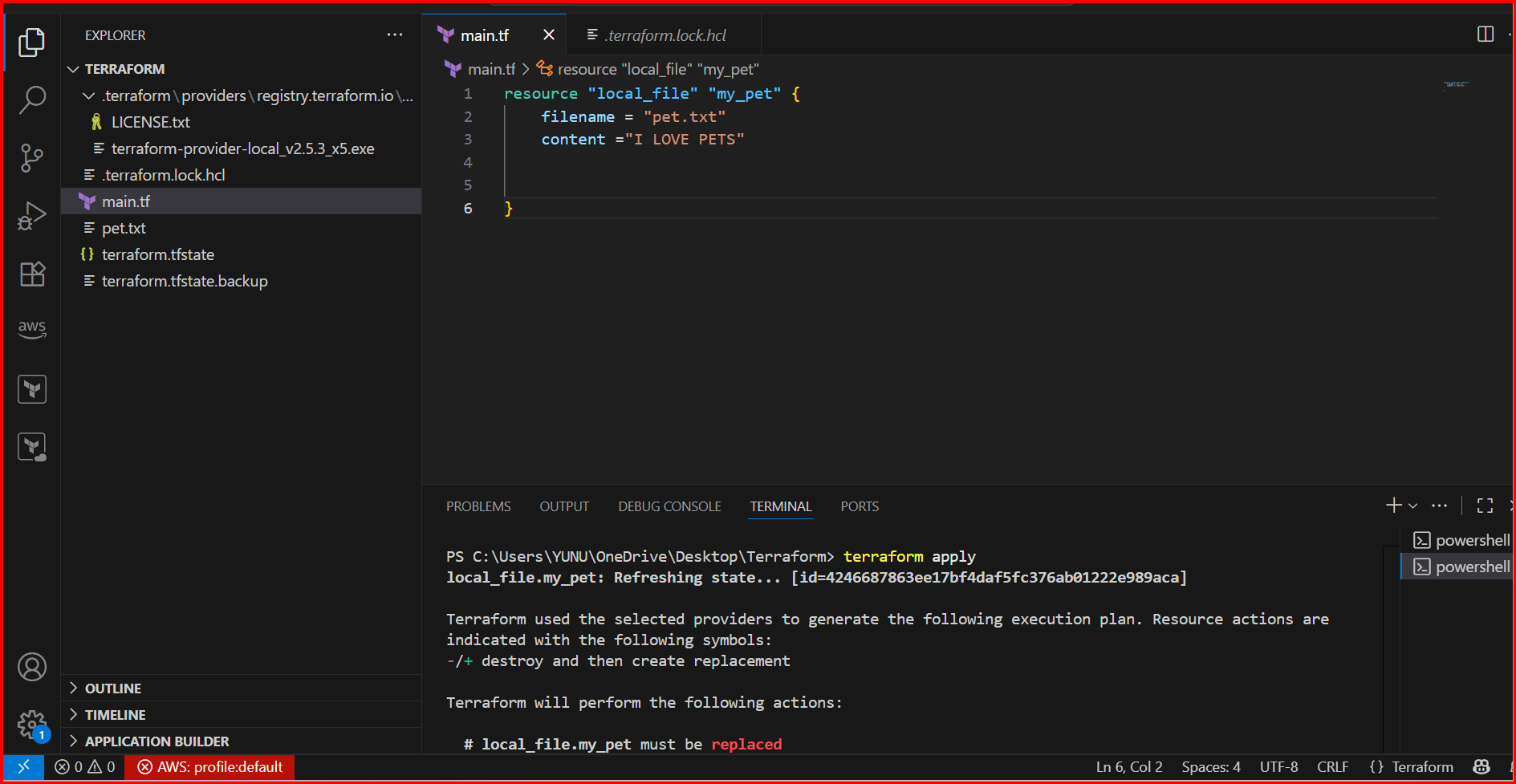
**Add the terminal**

**before executing the code first, we need to initialized**

**Terraform init 🡪 It will install all the dependencies of local file**

**One directory has been created. .terraform**

**.terraform has complete info has our provider**

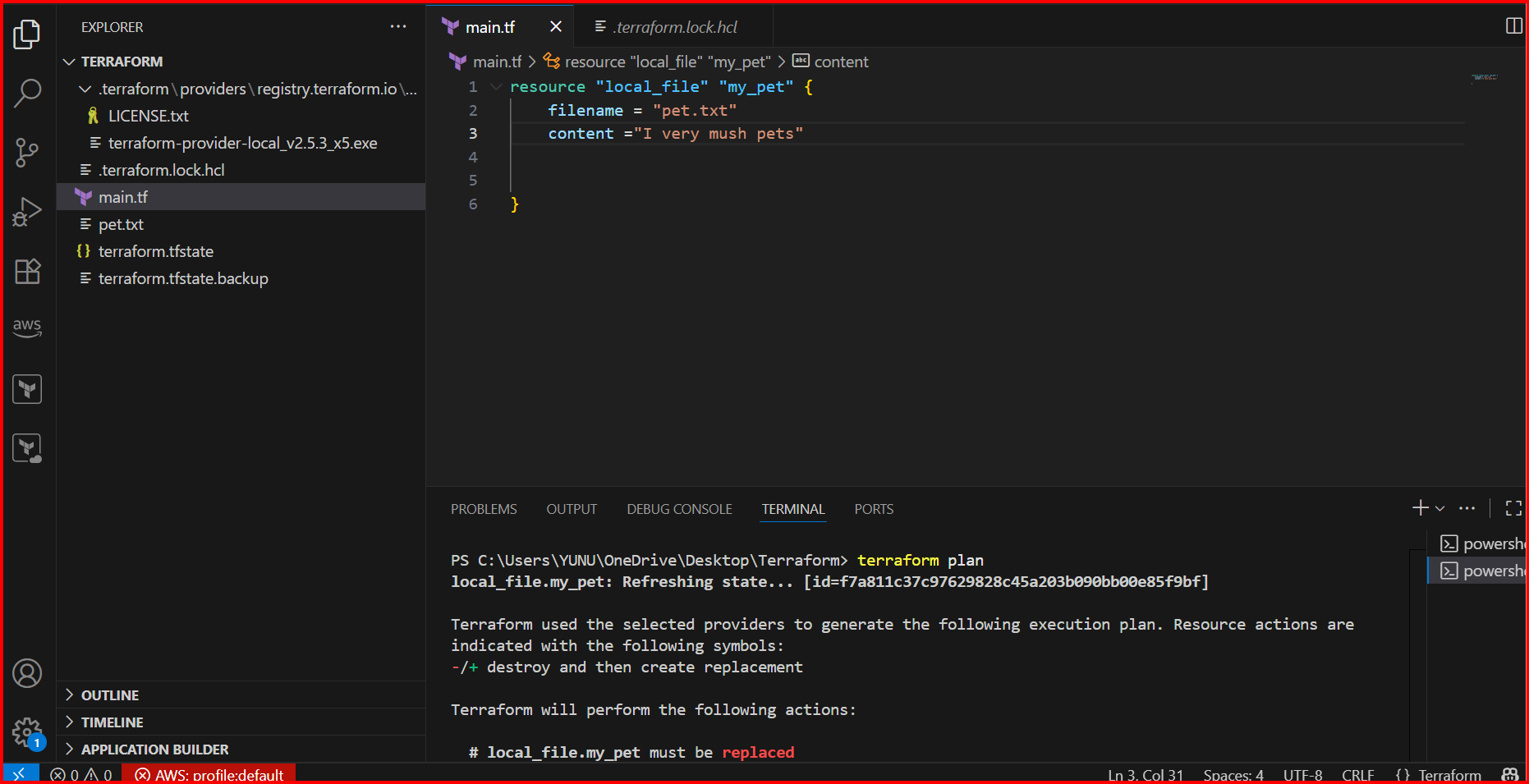
****

**Terraform apply 🡪to execute and create the infra based on the configuration**

**Here I have modified the names. this has been replaced with new names**

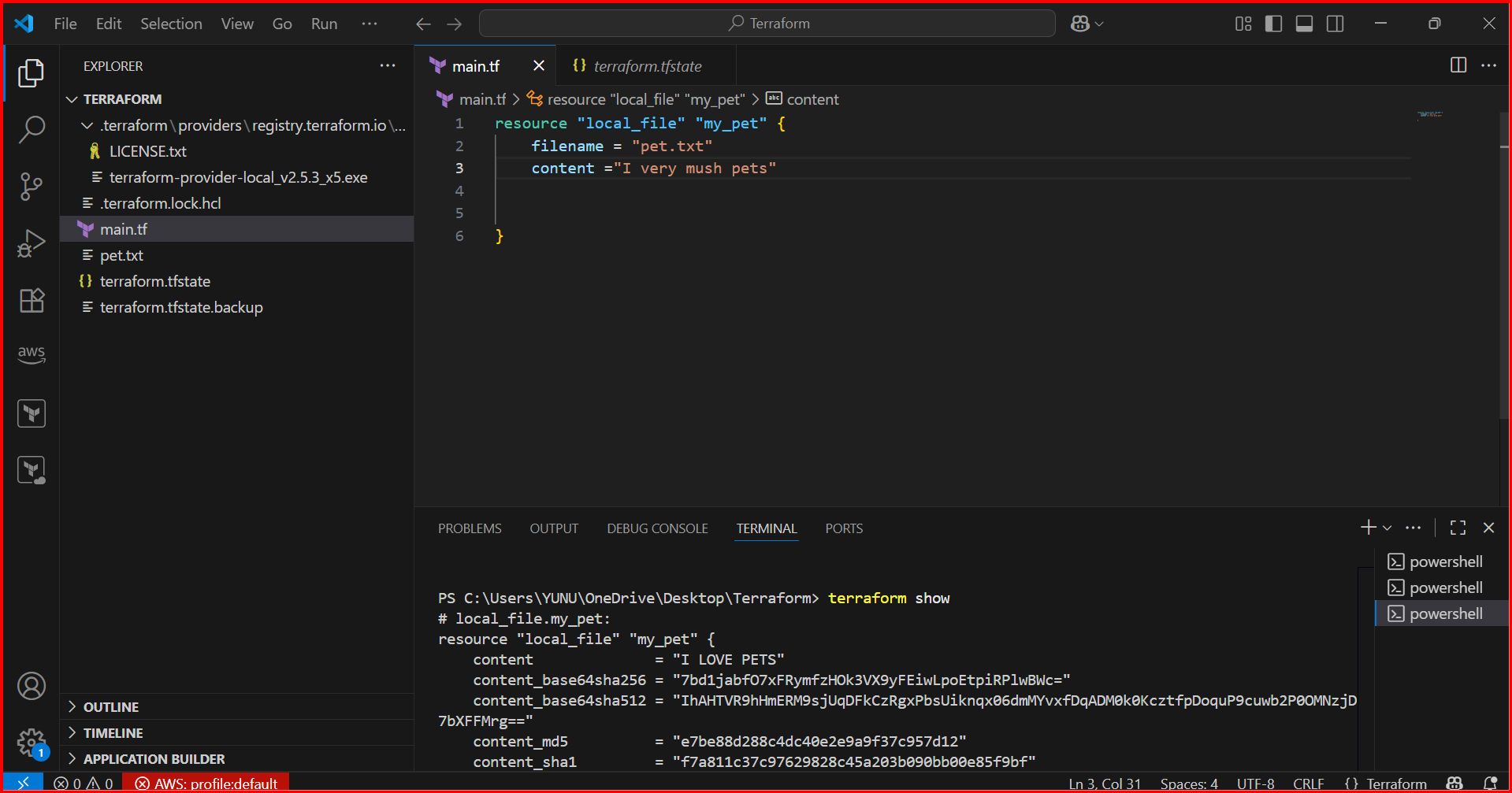
**Modifications will show once we hit yes then only changes will apply**

**And this will execute.**

****

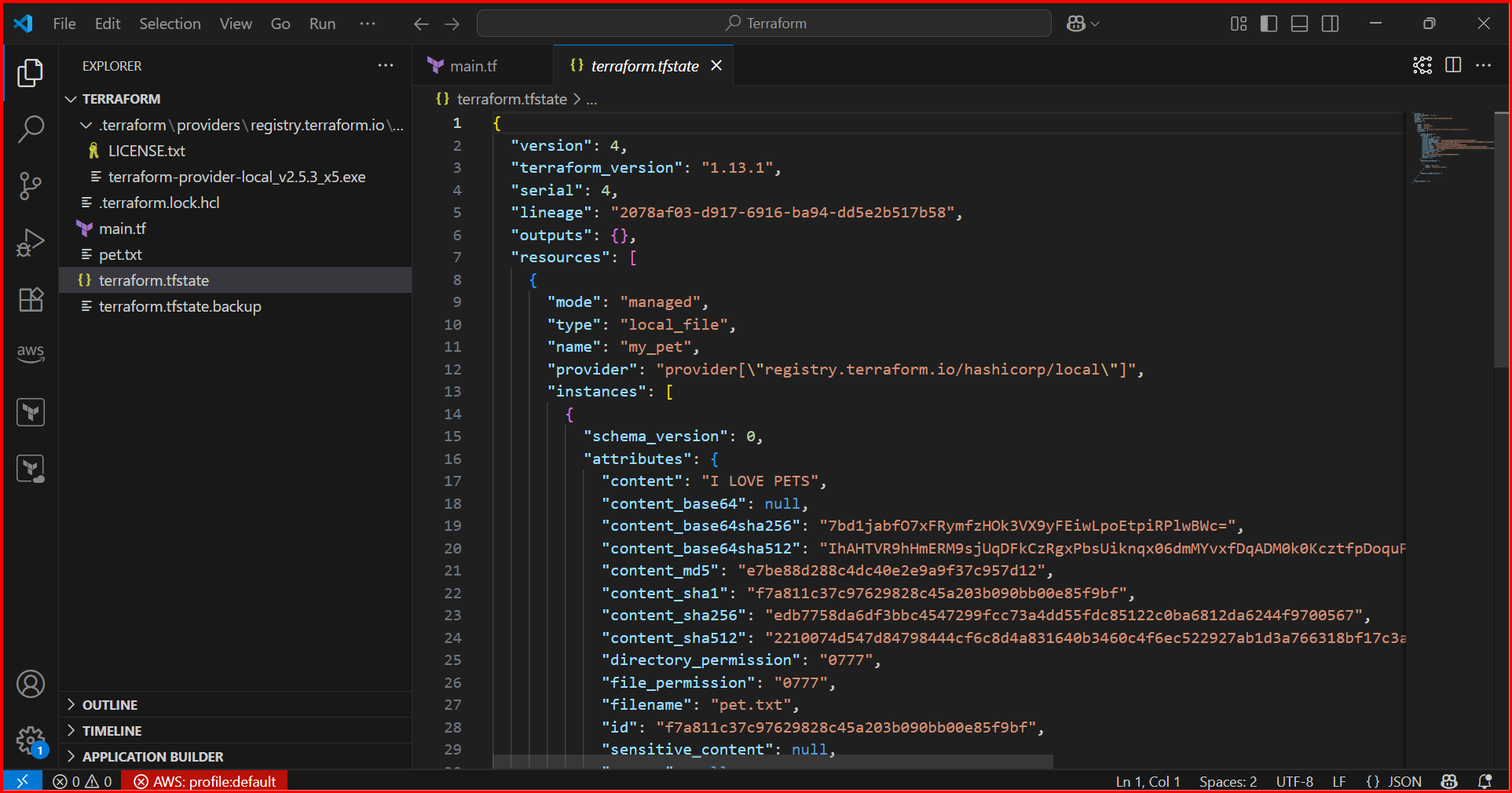
**Terraform plan 🡪The terraform plan command is used to preview the changes Terraform will make to your infrastructure before actually applying them.**

**Terraform show 🡪 The terraform show command is used to display information about Terraform state or plan files.**

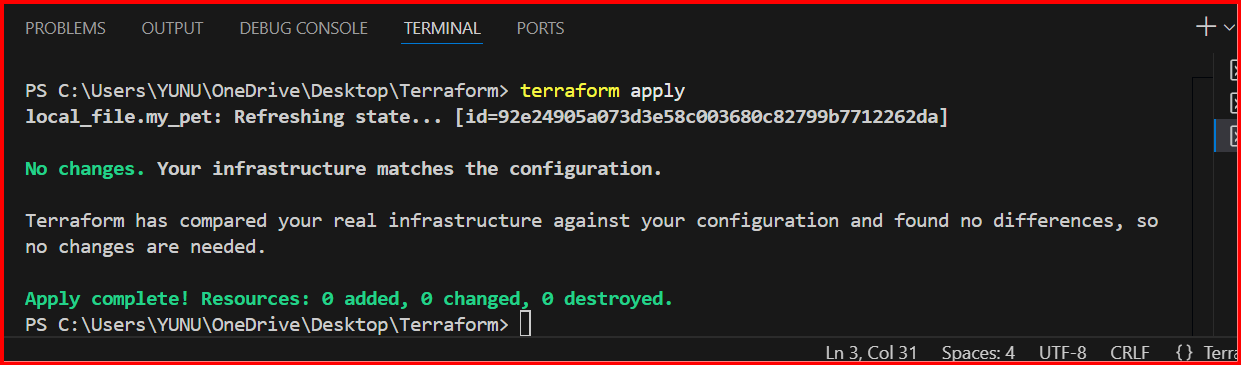
****

**Terraform destroy 🡪he terraform destroy command is used to remove all the infrastructure that Terraform created.**

****

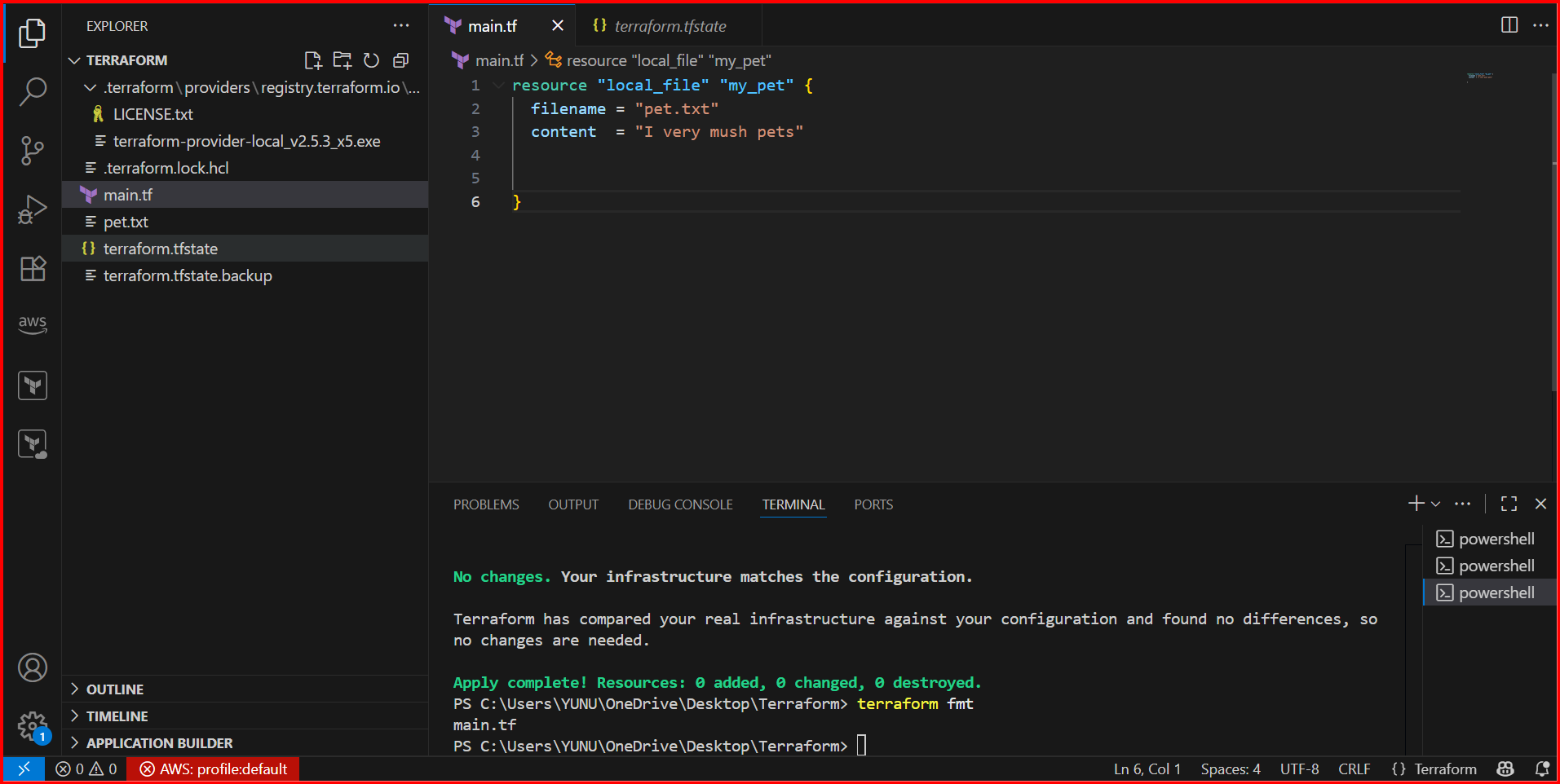
****

**Terraform.tfstate 🡪 This file have complete information about our configuration file**

****

**If you excute the terraform apply command multiple times 🡪 IDEMPOTENT**

**The terraform fmt command is used to format Terraform configuration files**

****

**3) Note down below points,  
   Terraform Init  
   Terraform Plan  
   Terraform Apply  
   Terraform Provider**

**terraform init🡪 Initializes a working directory for Terraform, 🡪 It will install all the dependencies of local file**

**terraform Plan🡪 The terraform plan command is used to preview the changes Terraform will make to your infrastructure before actually applying them.**

**terraform Apply 🡪** **Executes the plan and provisions resources.**

**terraform Provider 🡪** **It is api call which is used to communicate with the resources, it can be aws,gcp,azure or local.**

**Providers are in three categories**

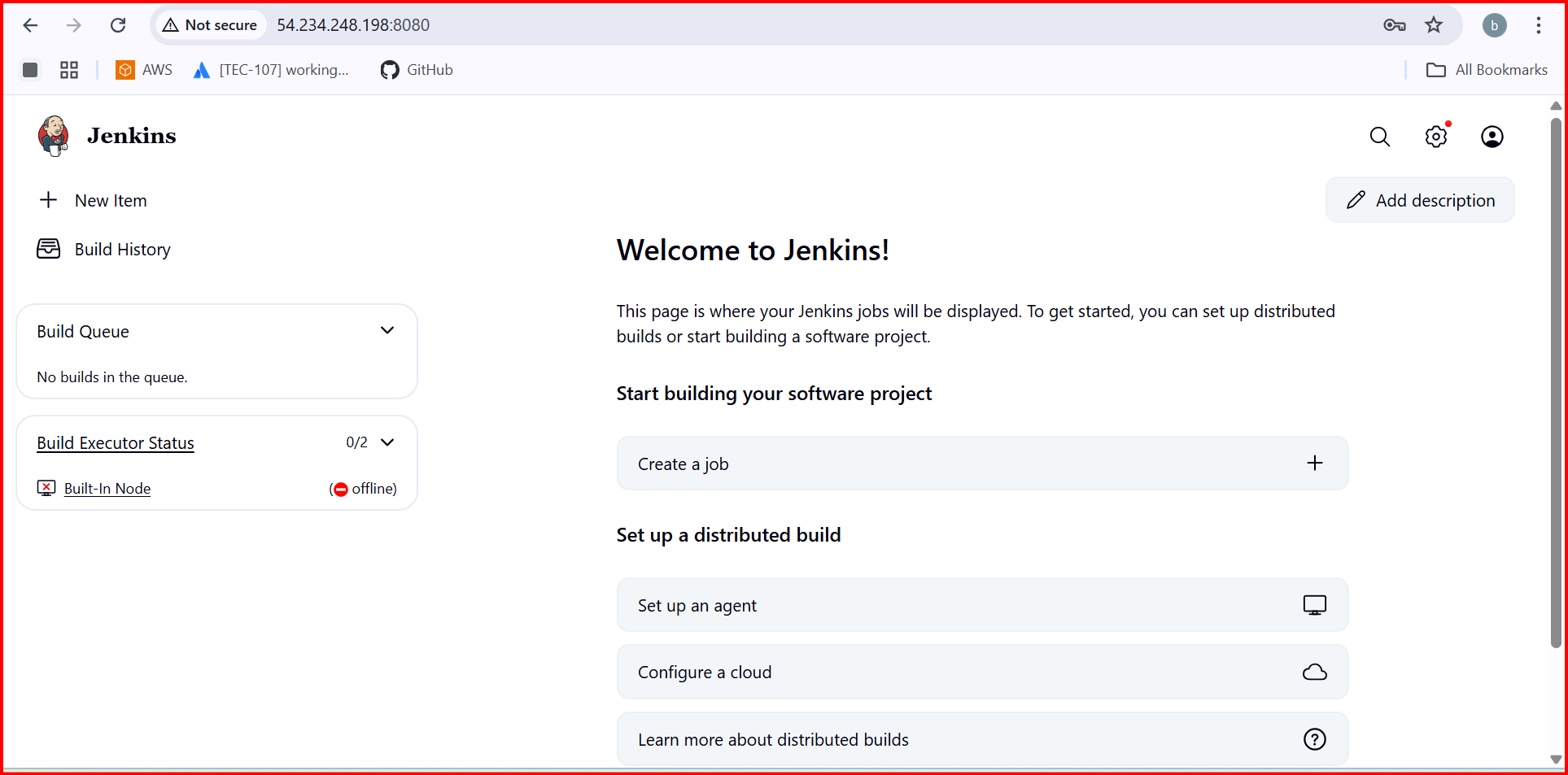
**1) Official --> provided by Hashicorp**

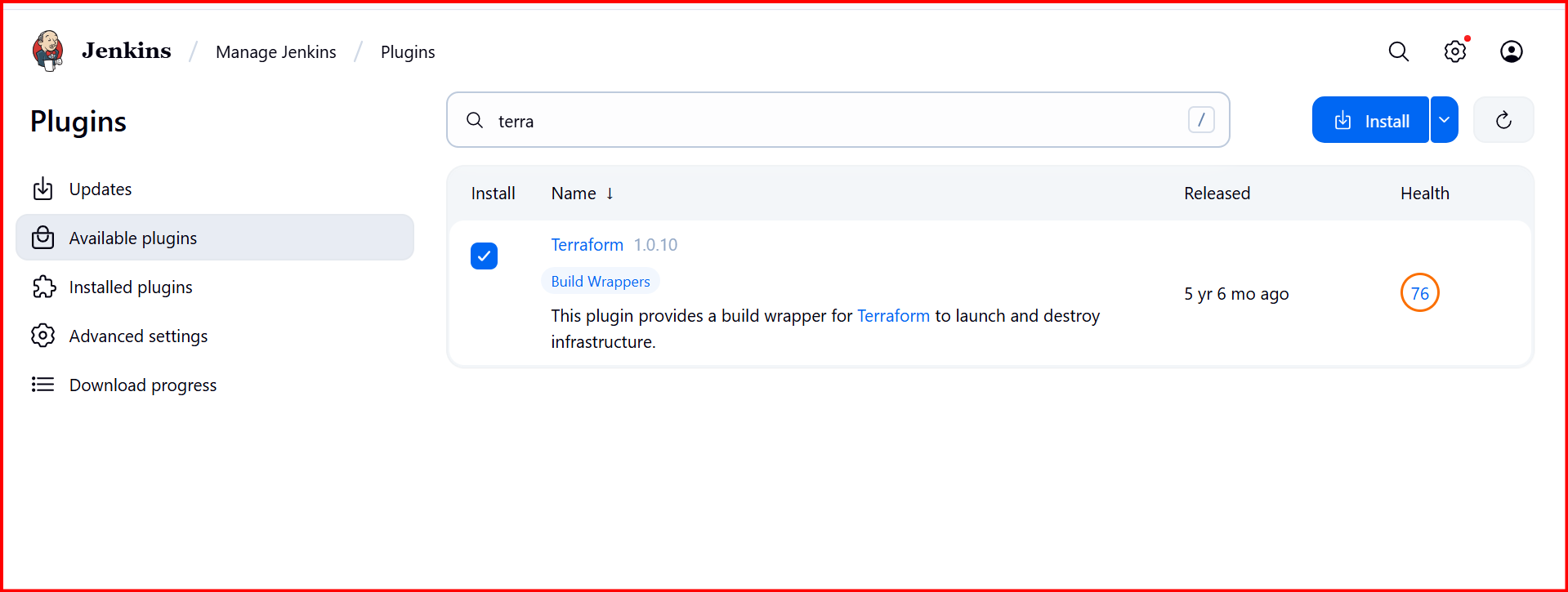
**2) Partner --> third party vendors**

**3) Community --> individual**

**4.Integrate a sample Terraform template in Jenkins**

**a Install Jenkins and check it is up and running**

****

**Install terraform plugin **

**In Jenkins server install terraform**

**wget** [**https://releases.hashicorp.com/terraform/1.9.8/terraform\_1.9.8\_linux\_amd64.zip**](https://releases.hashicorp.com/terraform/1.9.8/terraform_1.9.8_linux_amd64.zip)

**unzip terraform\_1.9.8\_linux\_amd64.zip**

**sudo mv terraform /usr/local/bin/**

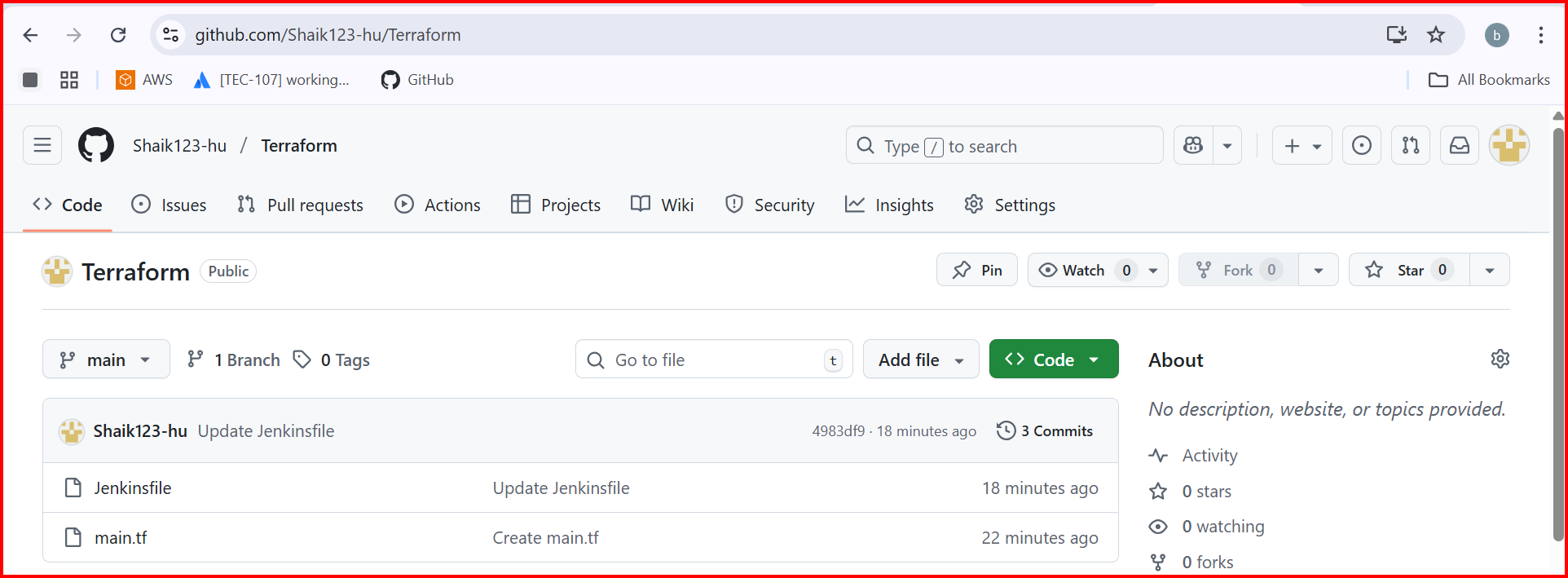
**terraform -version**

****

**In git create one new repo**

[**https://github.com/Shaik123-hu/Terraform.git**](https://github.com/Shaik123-hu/Terraform.git)

**In this create Jenkins file and main.tf files need to create**

****

pipeline {

agent any

environment {

AWS\_ACCESS\_KEY\_ID = credentials('Access\_Key')

AWS\_SECRET\_ACCESS\_KEY = credentials('Secret\_key')

}

stages {

stage('Checkout Code') {

steps {

git url: 'https://github.com/Shaik123-hu/Terraform.git', branch: 'main'

}

}

stage('Terraform Init') {

steps {

sh 'terraform init'

}

}

stage('Terraform Plan') {

steps {

sh 'terraform plan'

}

}

stage('Terraform Apply') {

steps {

sh 'terraform apply --auto-approve'

}

}

}

}

resource "local\_file" "my\_pet" {

filename = "pets.txt"

content = "I Like PETS"

}

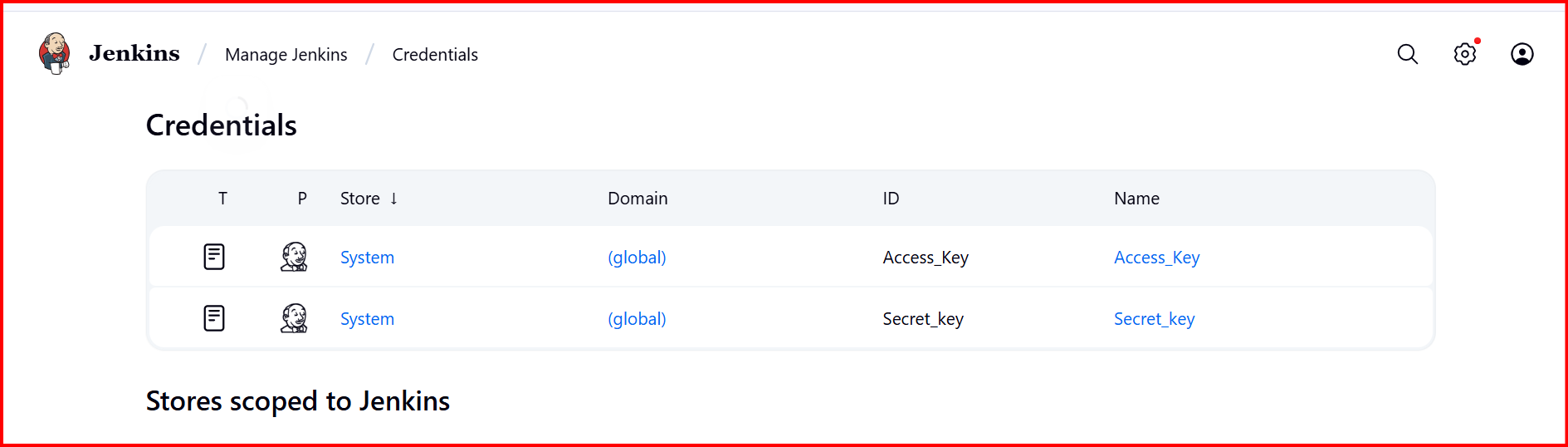
**In Aws security\_credentials**

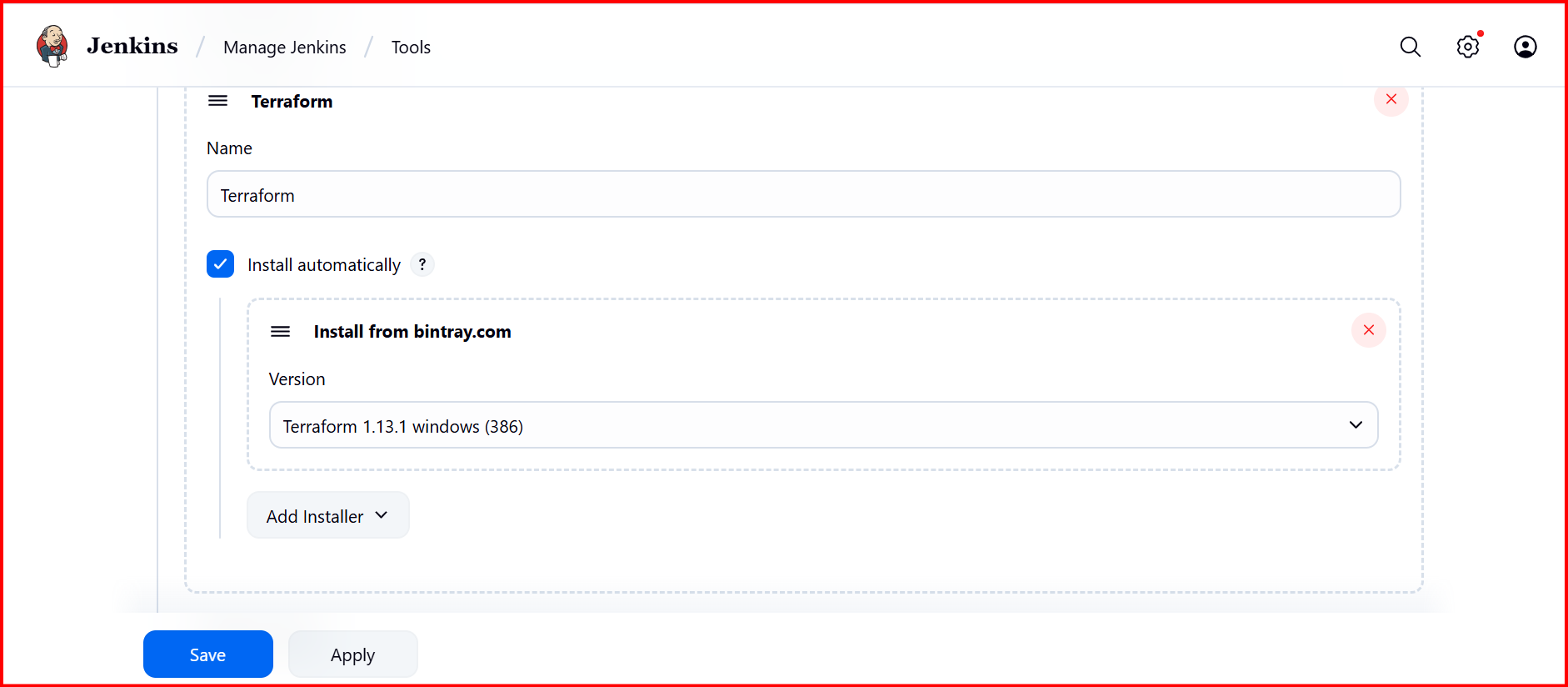
Create Access keys

1.Access key

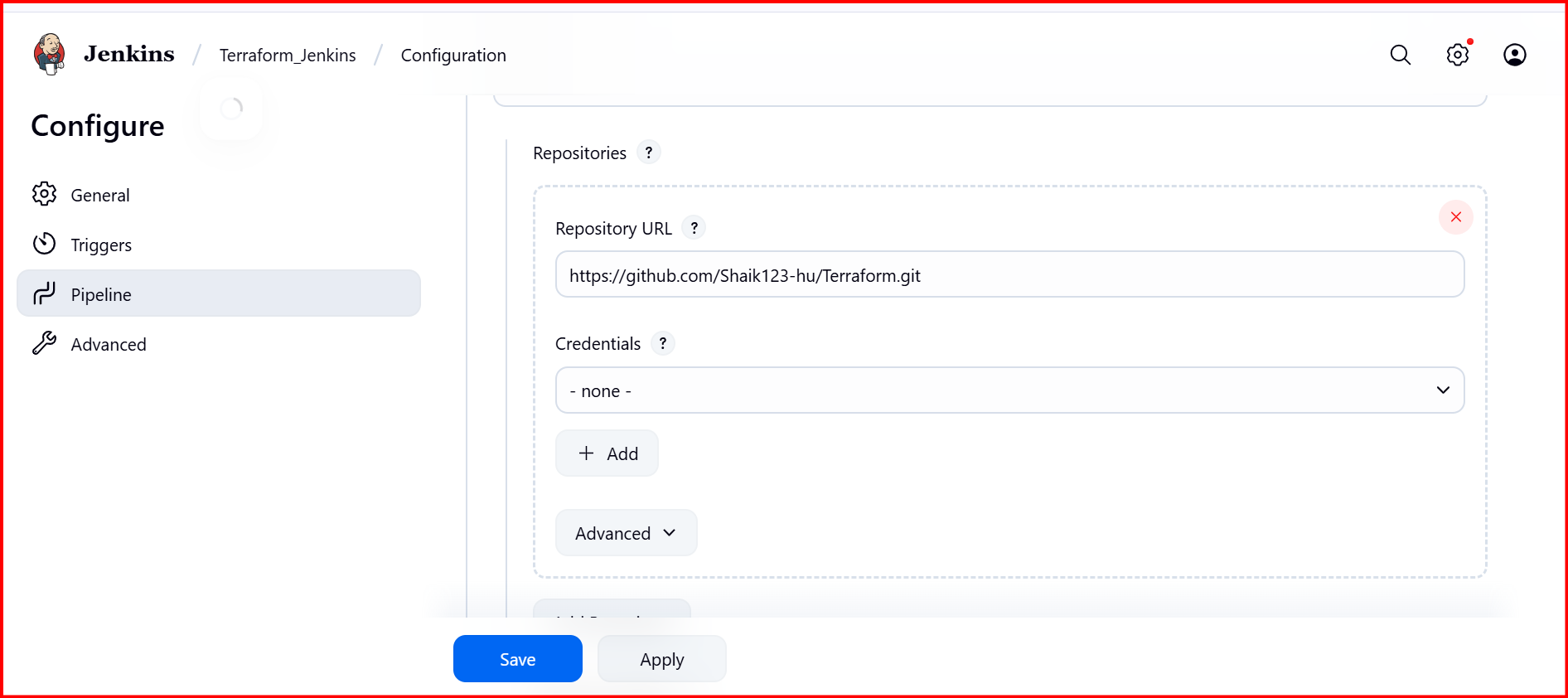
2 Secrete key

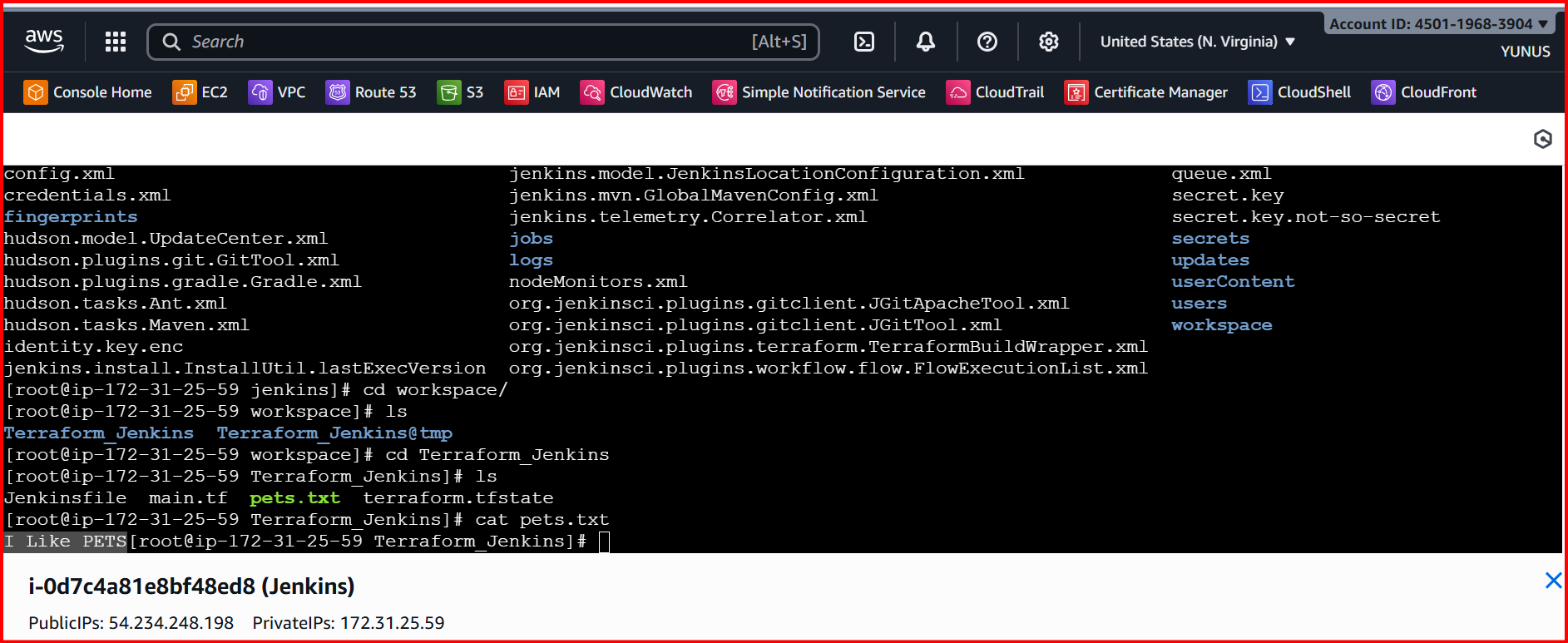
**Now go to Jenkins credentials and create two credentials**

****

****

**Jenkins Create a job**

****

****