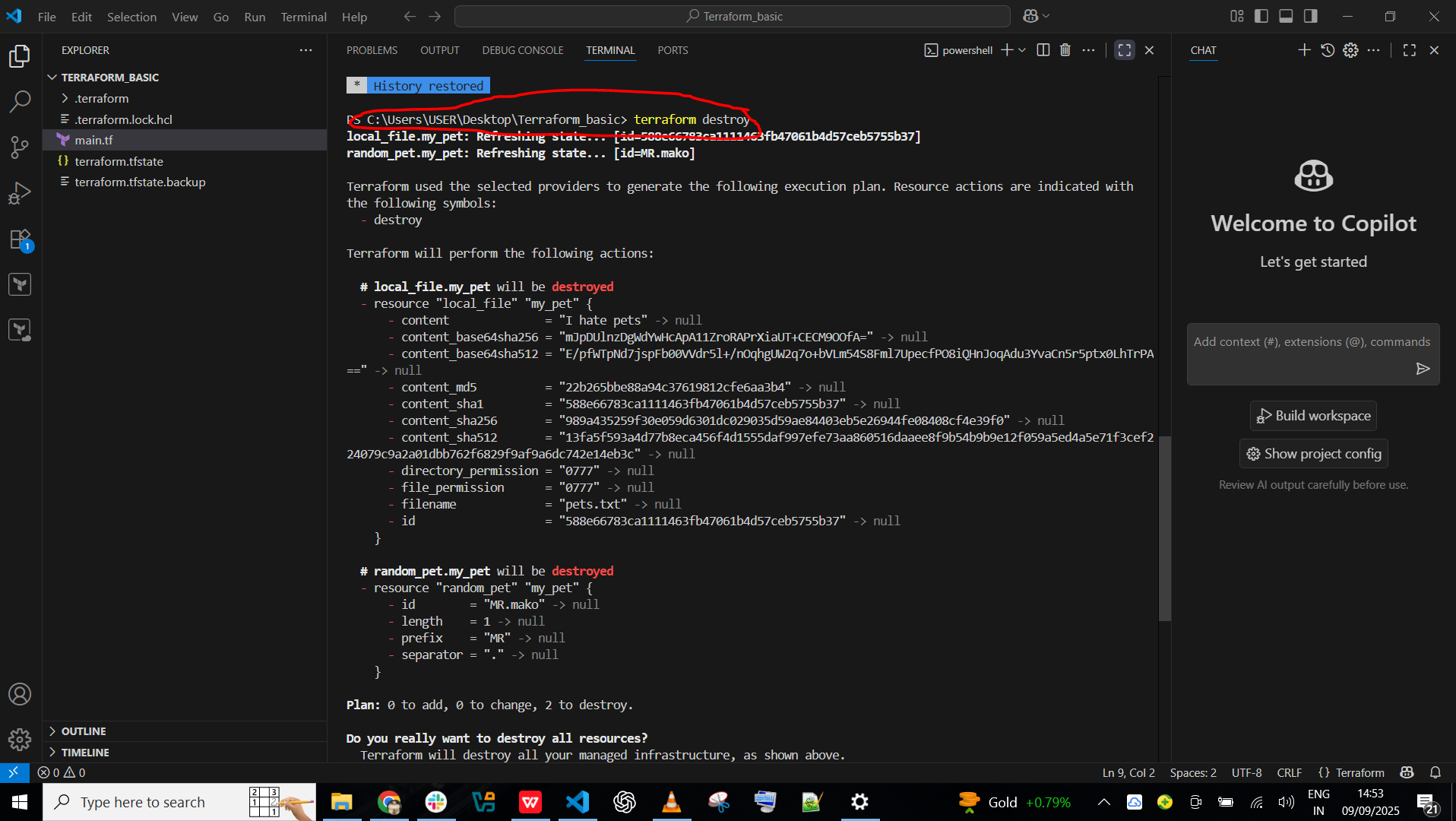
1. **Watch the terraform-02 video.**

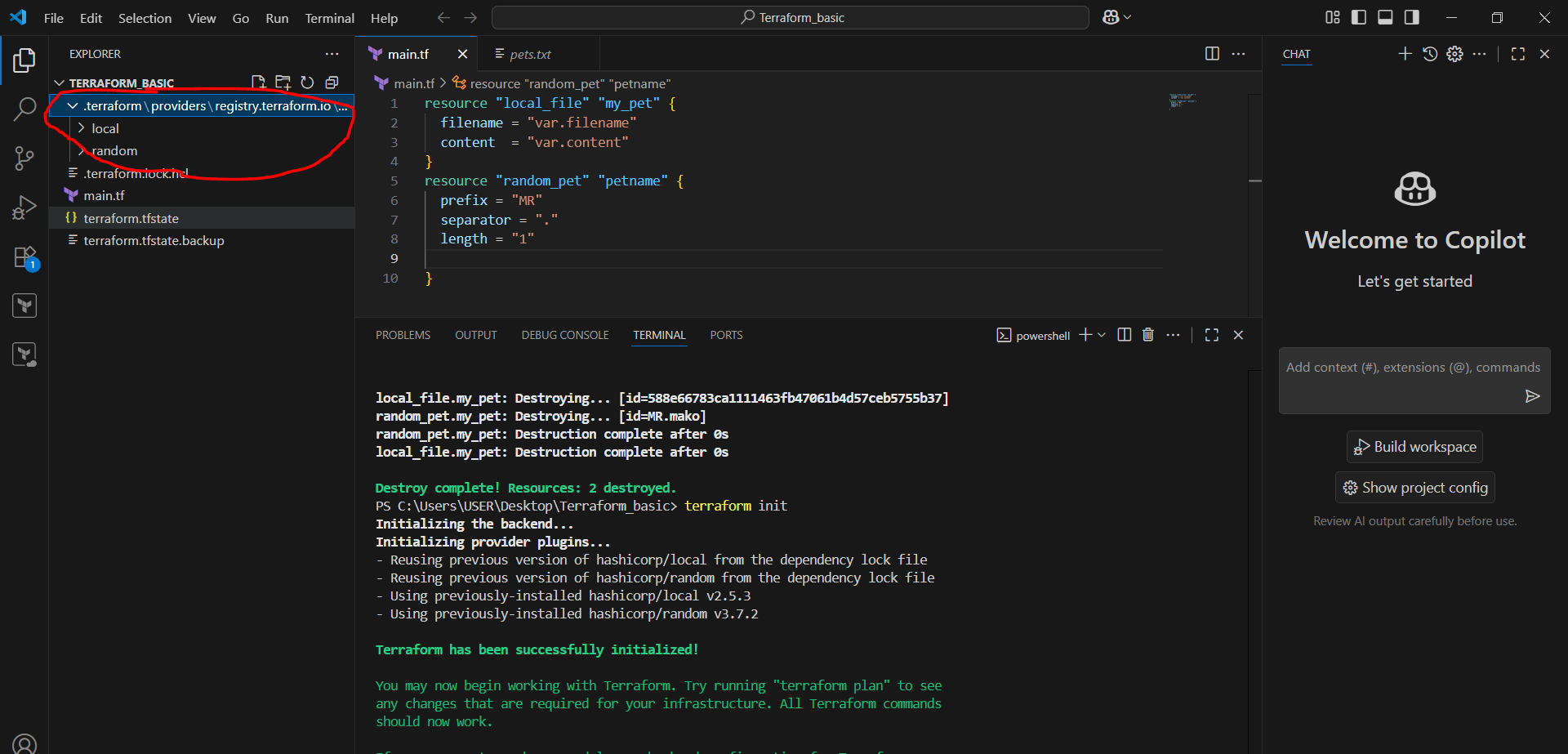
Completed the video

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

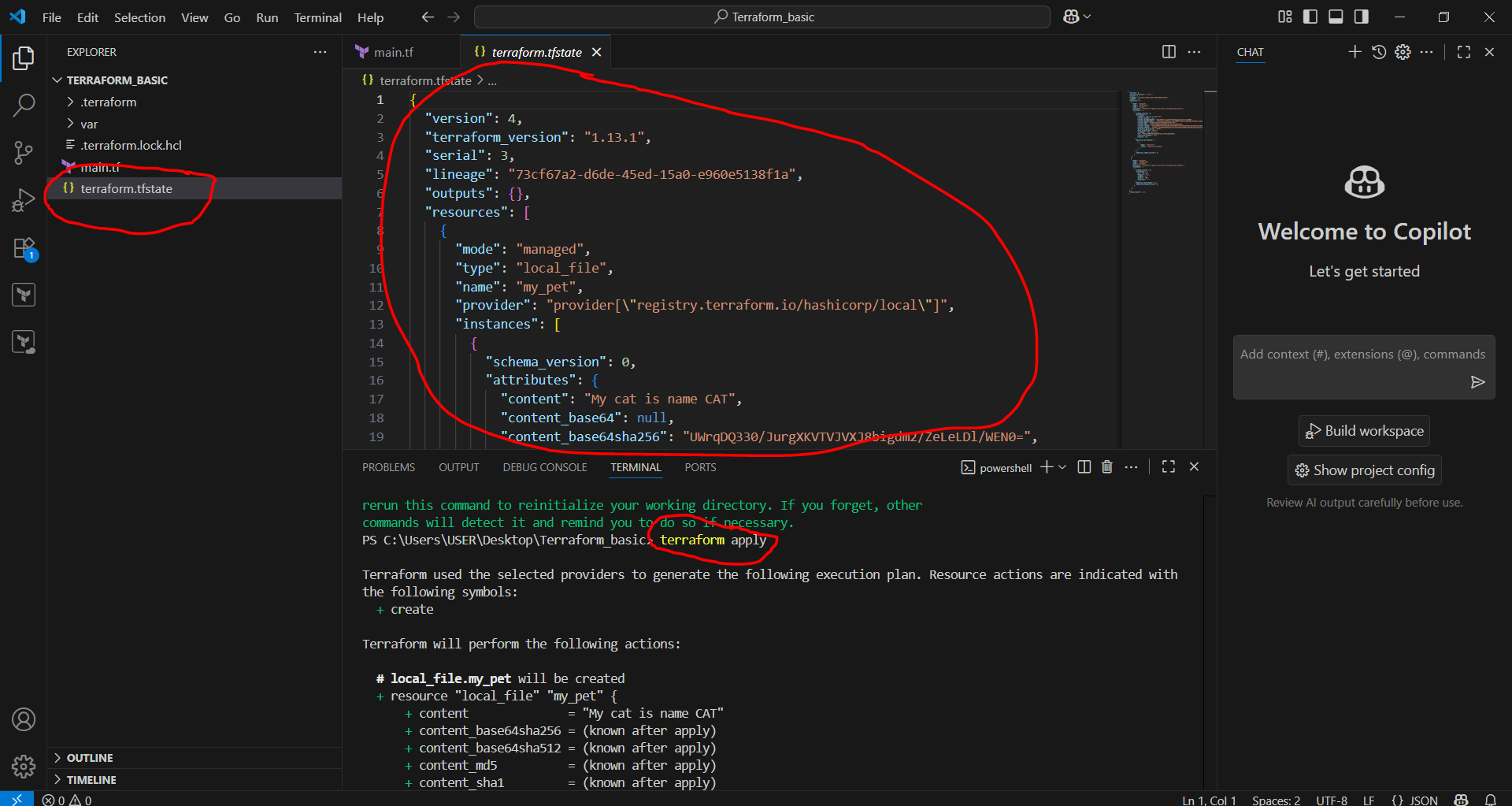
1. First we have destroyed everthing to do new executions



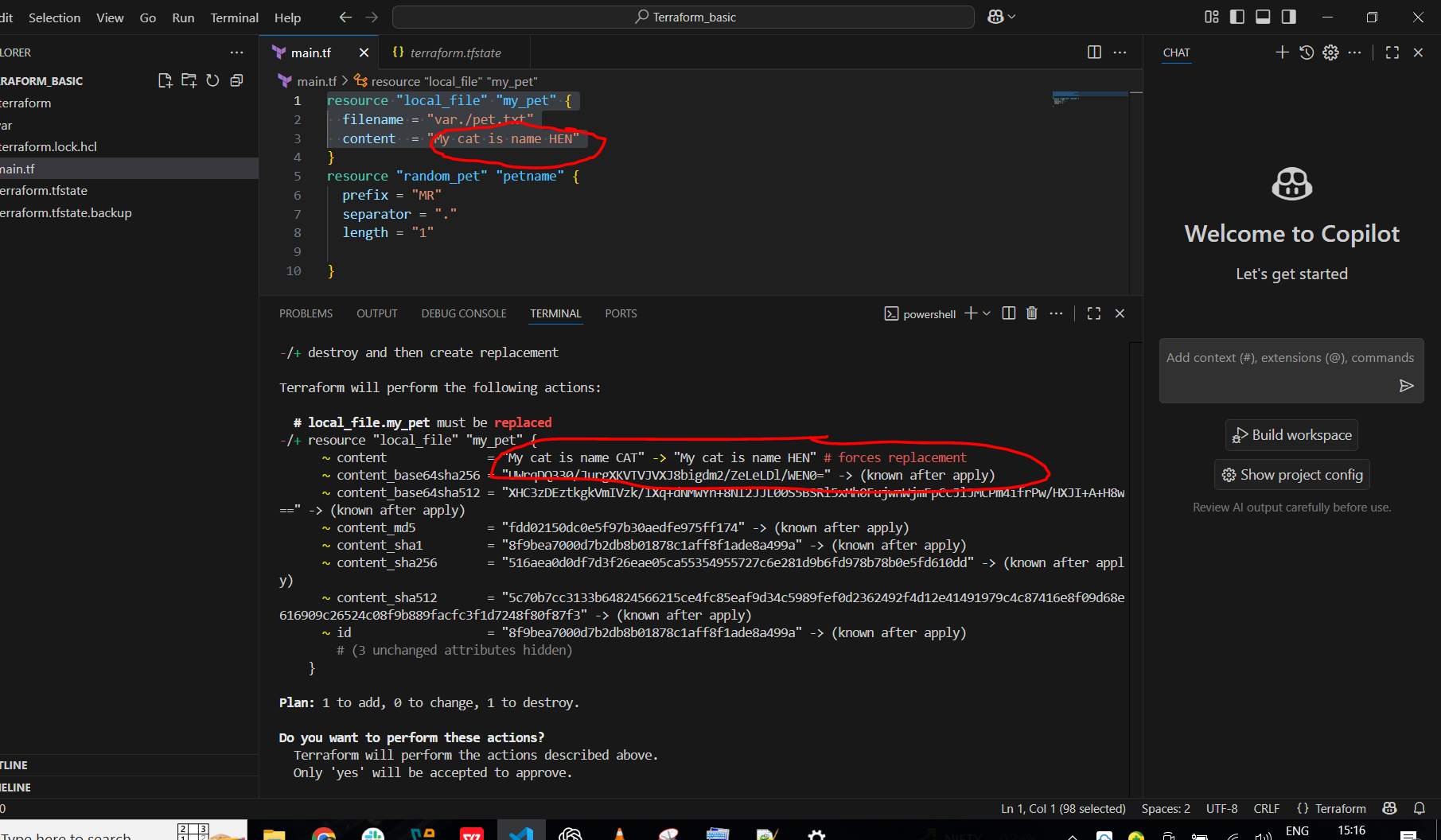
1. Now we have add the resource and gave command **terraform init** and it will get initilize the backend and provider plugin
2. Whenever we execute the command **terraform init**  it will create the **.terraform** directory and will have the providers as local and random



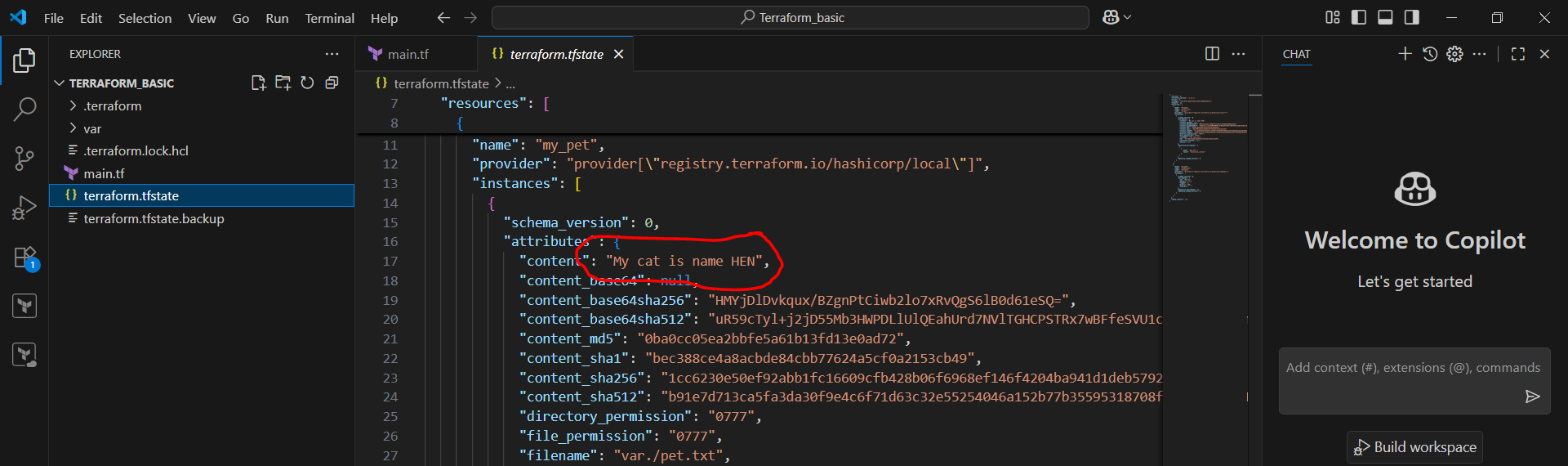
1. Whenever we execute the command **terraform apply** it will create the file as **terraform.tfstate** it will have complete metadata of our resource file like which version, what is the file name, file content etc…..!



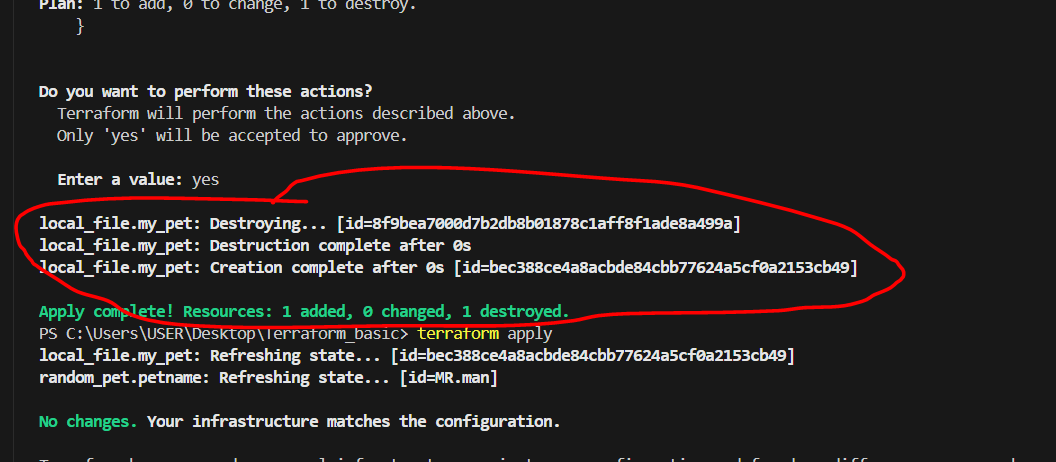
1. Here we have change the name of pet from CAT to HEN, then gave command as **terraform apply**, in below picture it shows that the name of pet change and the content will changed and get to know from the **terraform.tfstate**



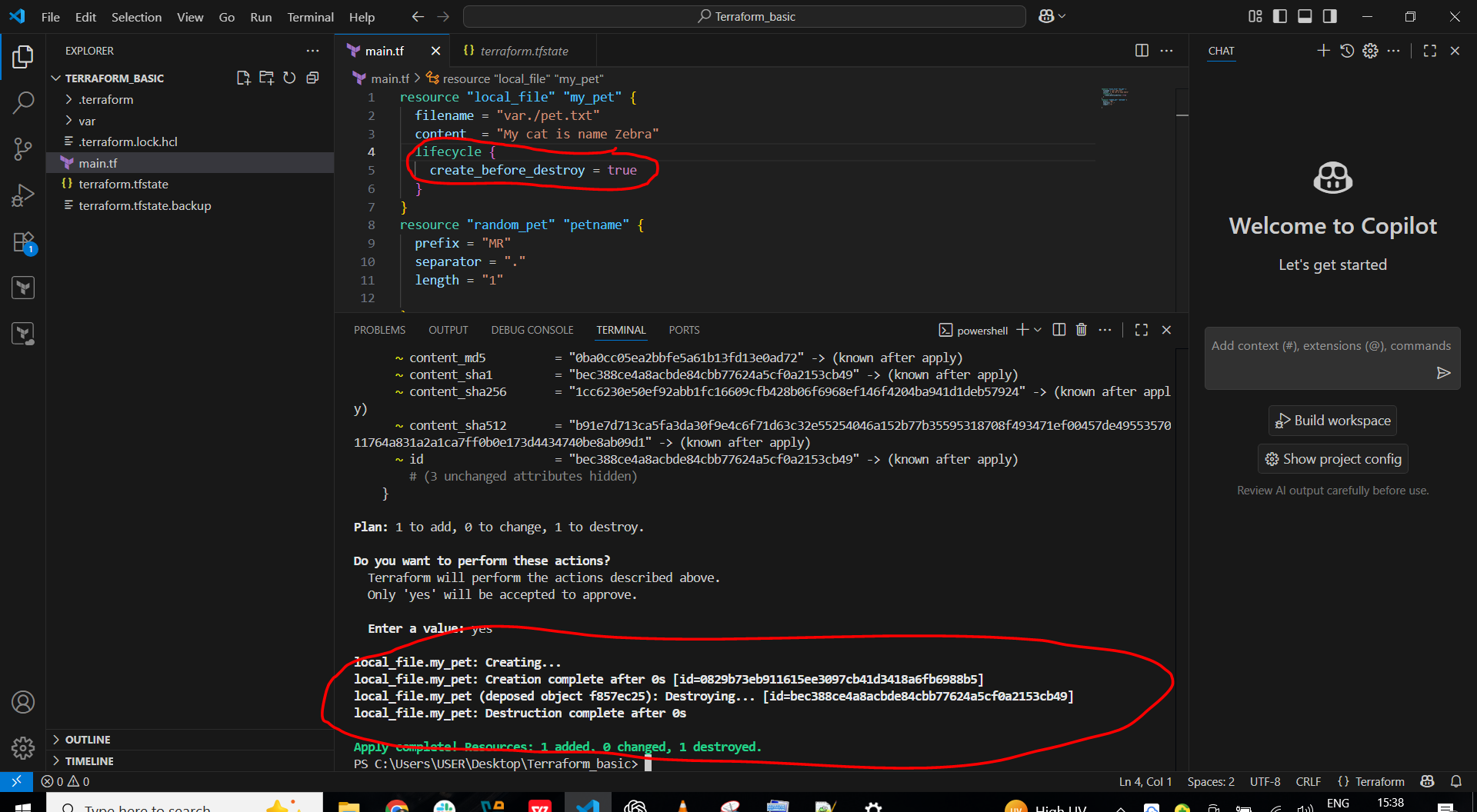
1. The **terraform.tfstate** file will also get changed the name as HEN



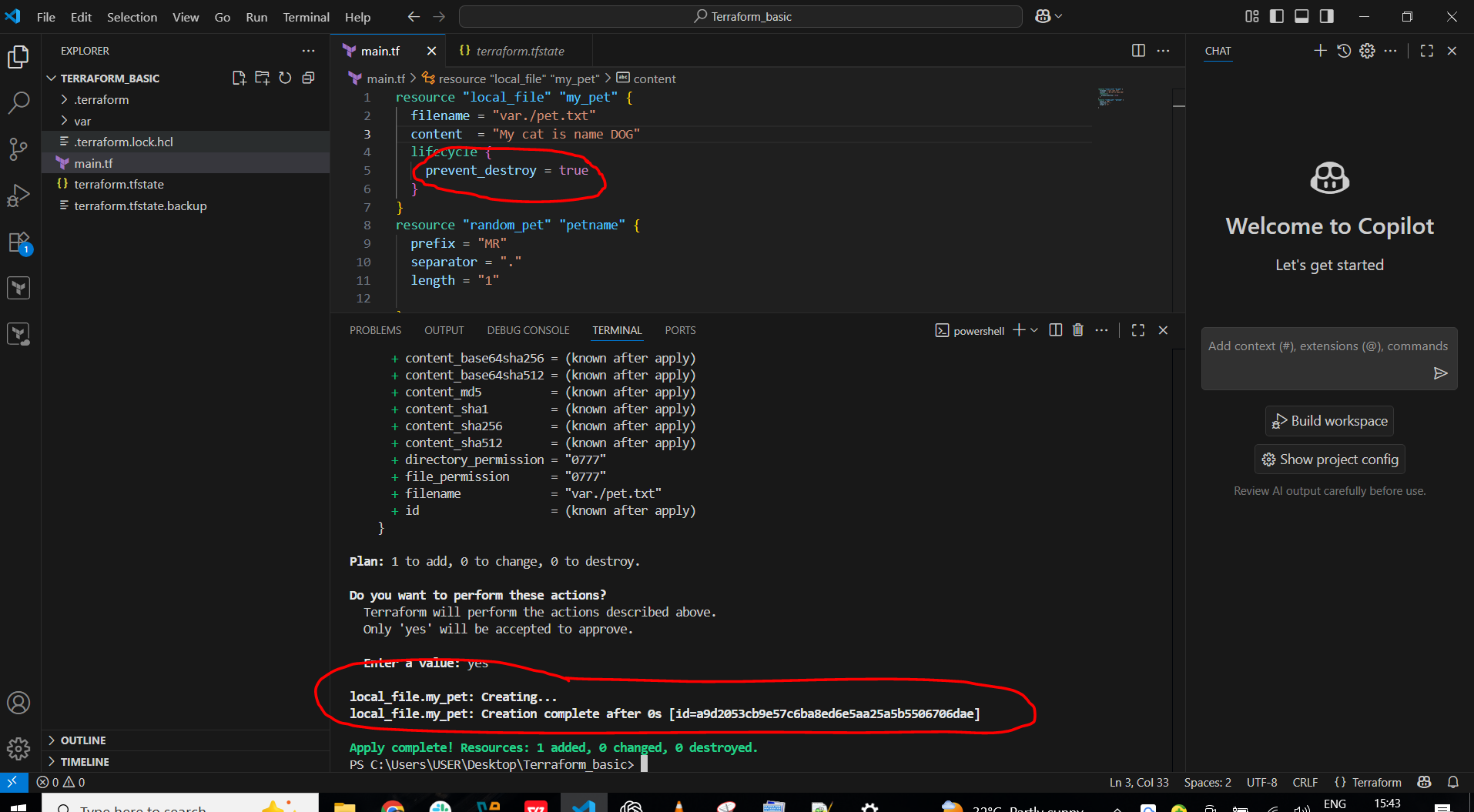
1. Here if we want to first create and then destroy we have add the **lifecycle ---- create\_before\_destroy = true** with this first it will first create and den destroy after giving the command **terraform apply**



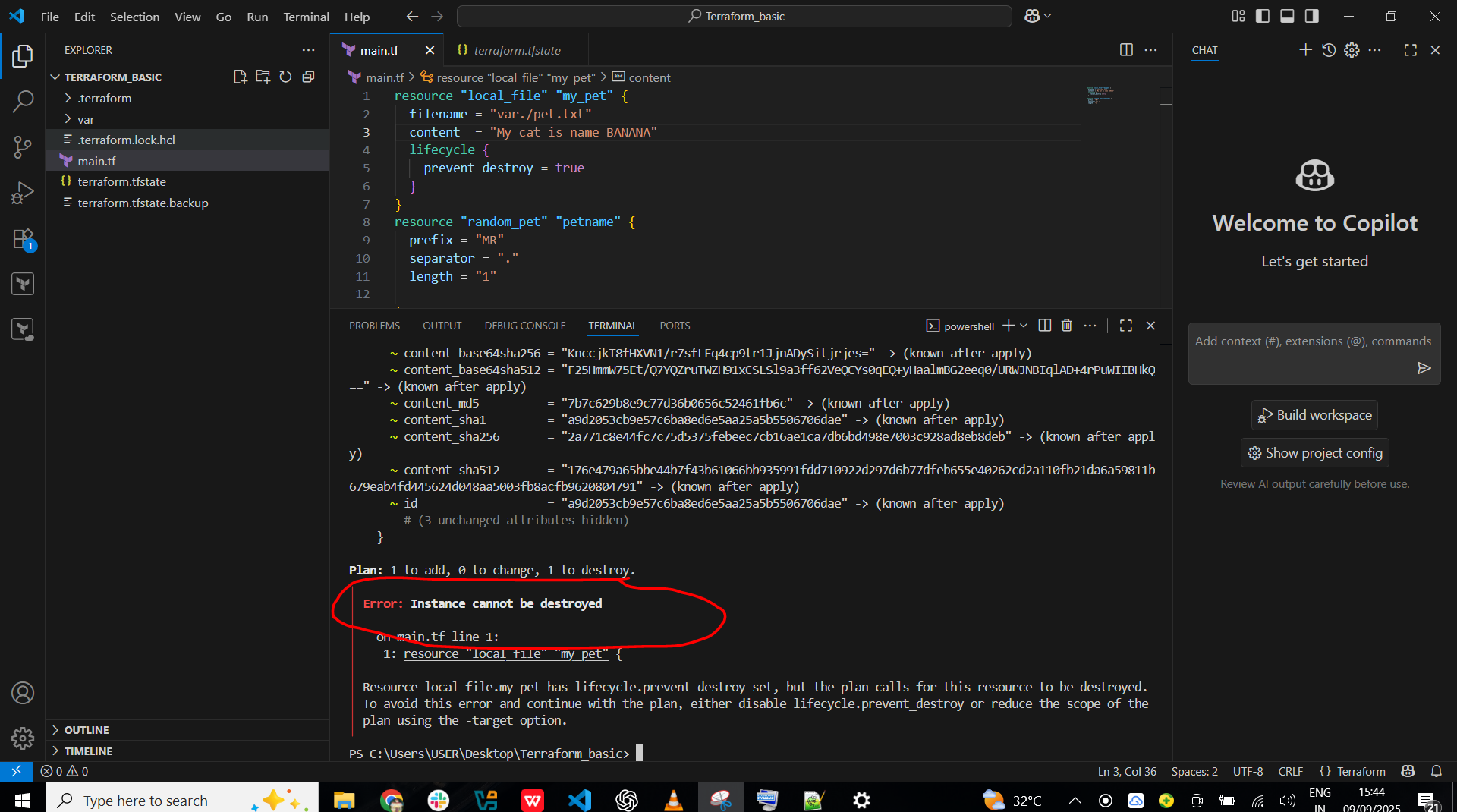
1. In the below picture it shows how it is executed



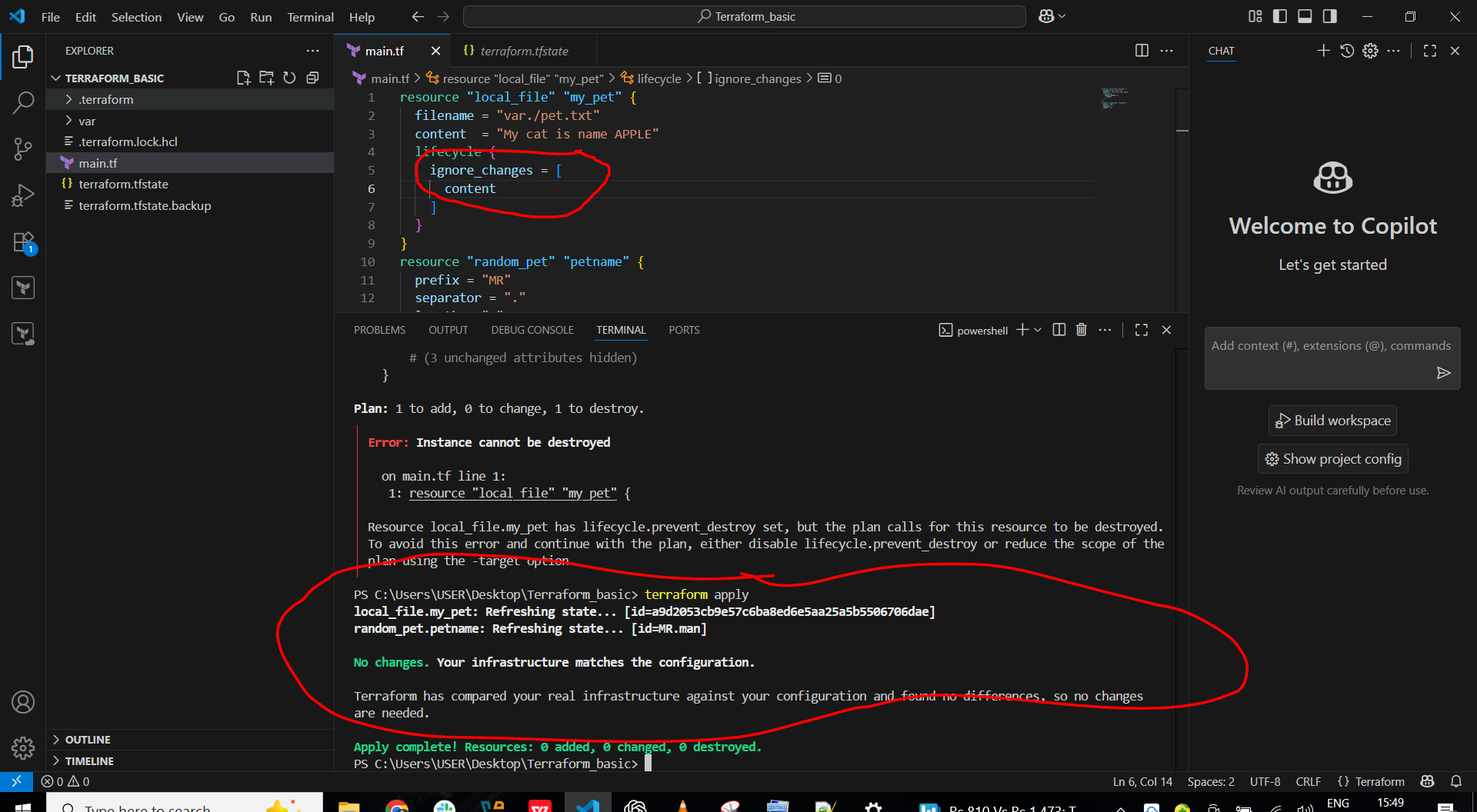
1. If we don’t want to delete existing req so we will use the **lifecycle ----- prevent\_destroy = true** and then give command **terraform apply**

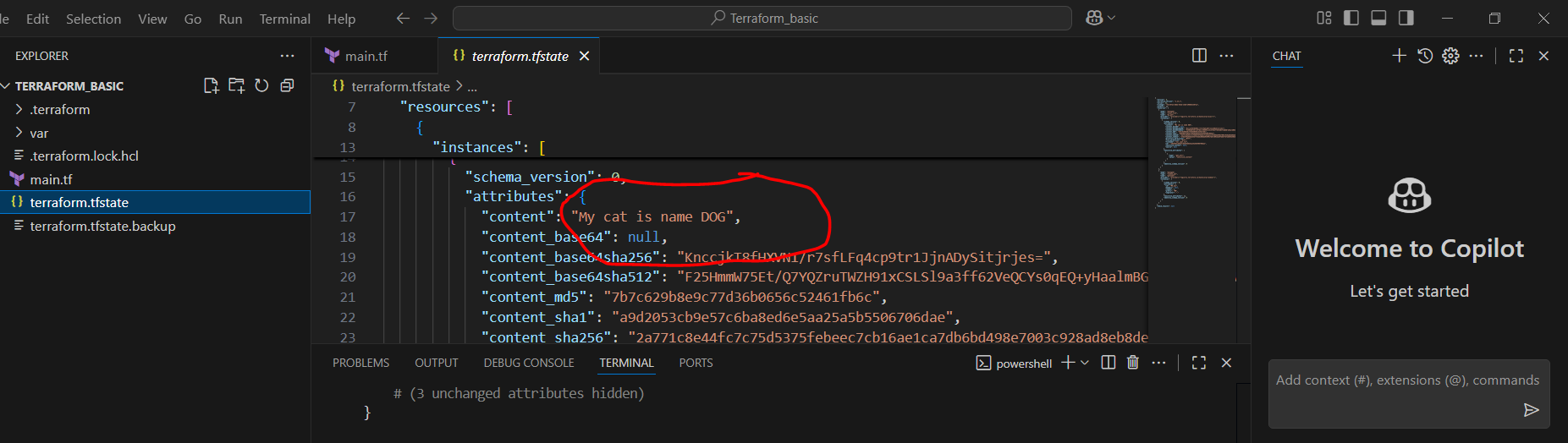


1. It will show instance cannot be destroyed because we gave **lifecycle ---- prevent\_destroye = true** and **terraform apply**

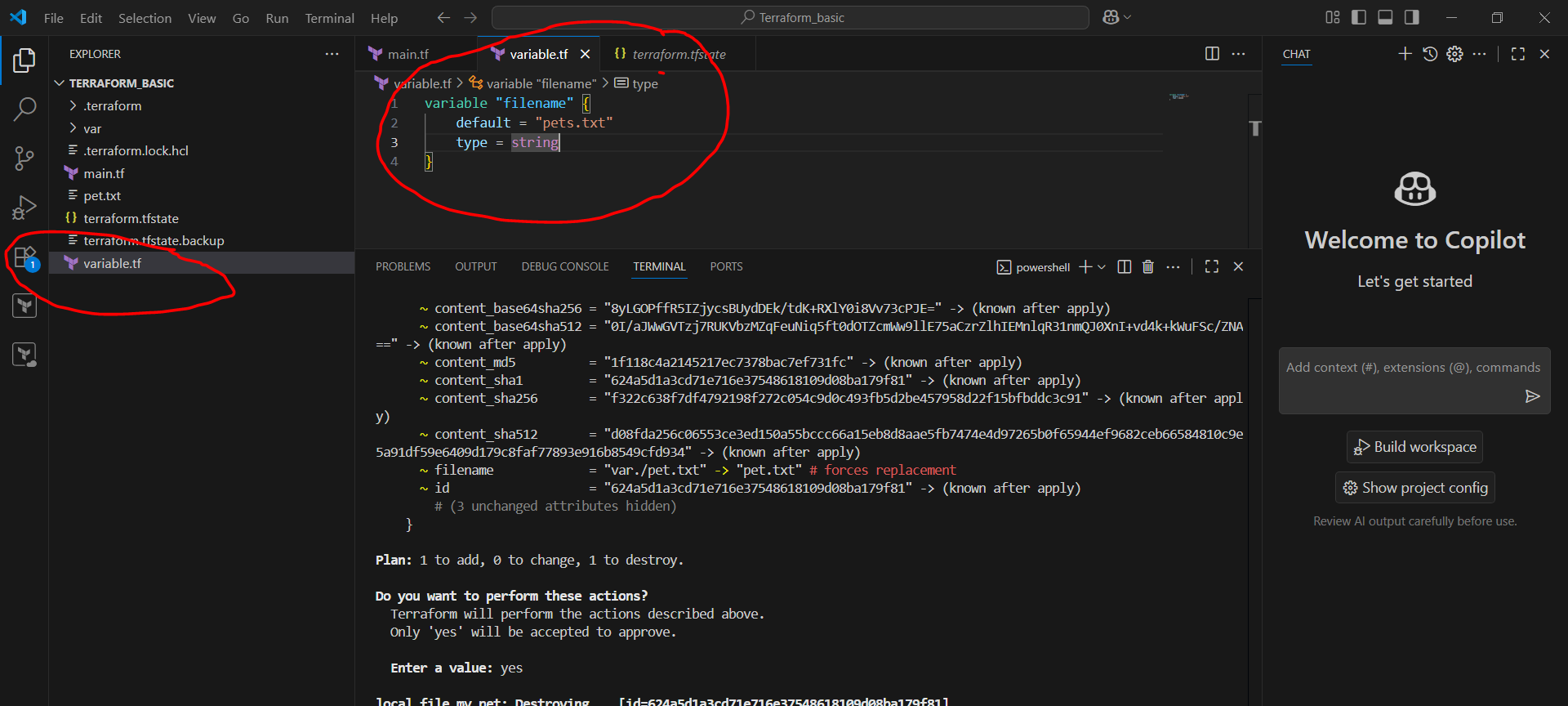


1. Now here we have updated the  **lifecycle --- ignore\_changes** of content, after changing the content it wont get any change of content, but in my **terraform .tfstate** file it was DOG and here I changed as APPLE so not get changed

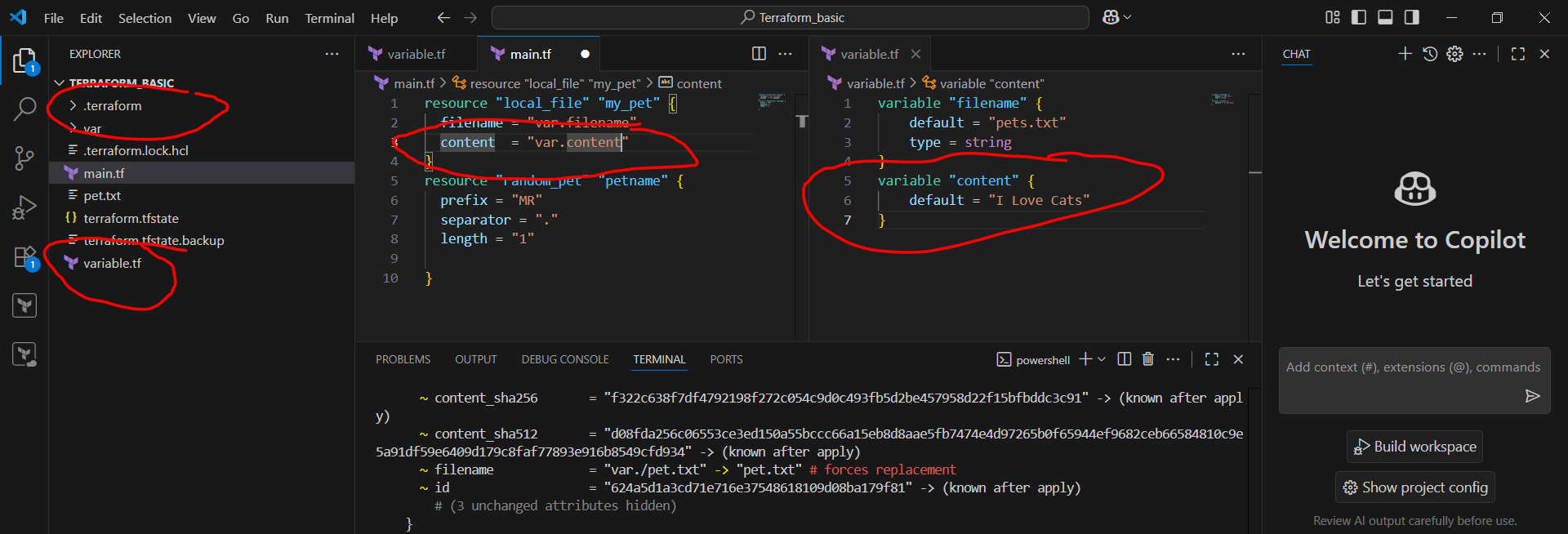


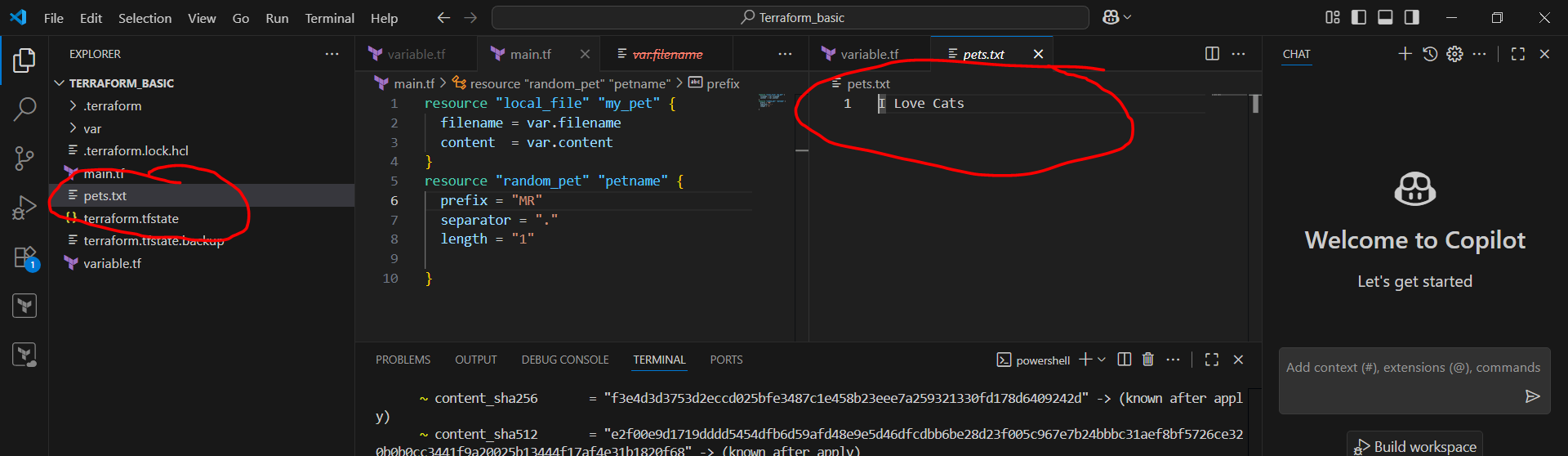


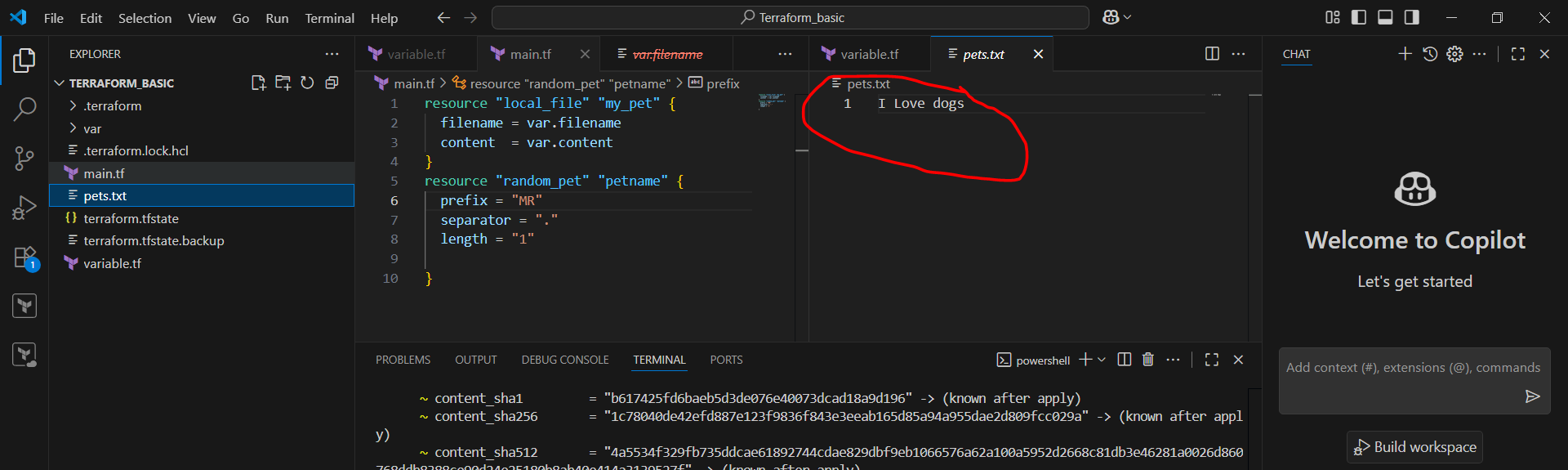
1. Now we have created on variable file and updated some content in that



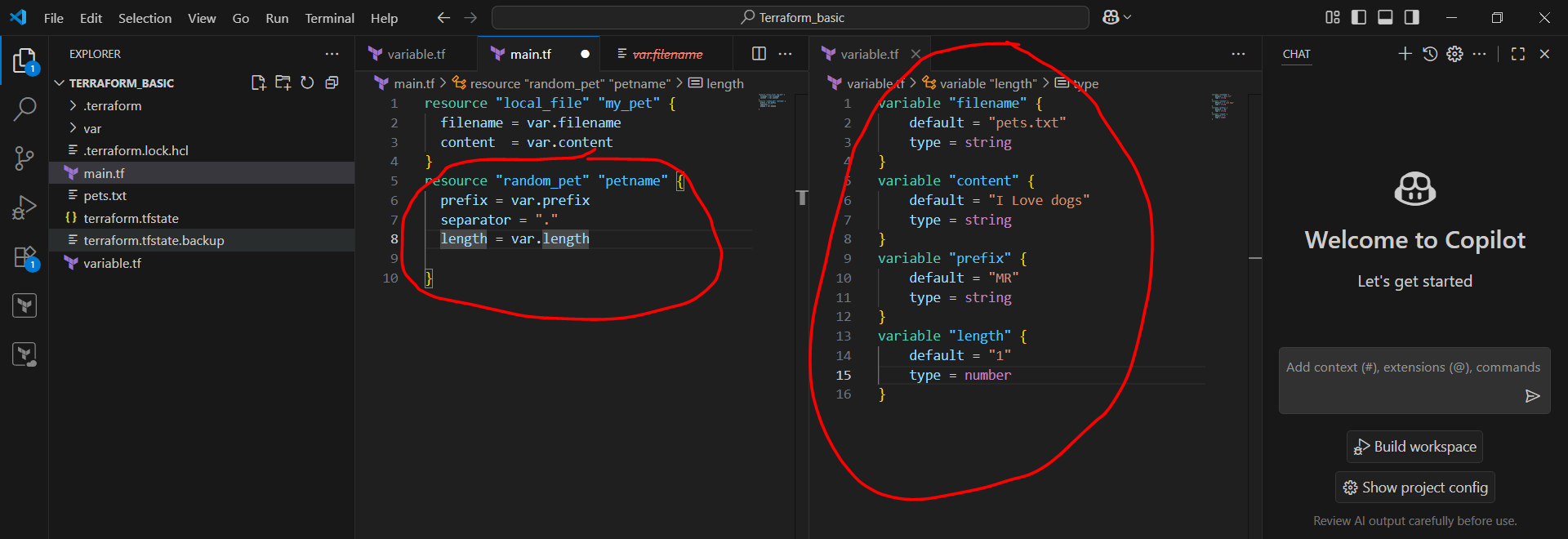
Here we have created two variable files in same directory and updated the content in both files **content = “var.content”** and **default = “I Love Cat”**



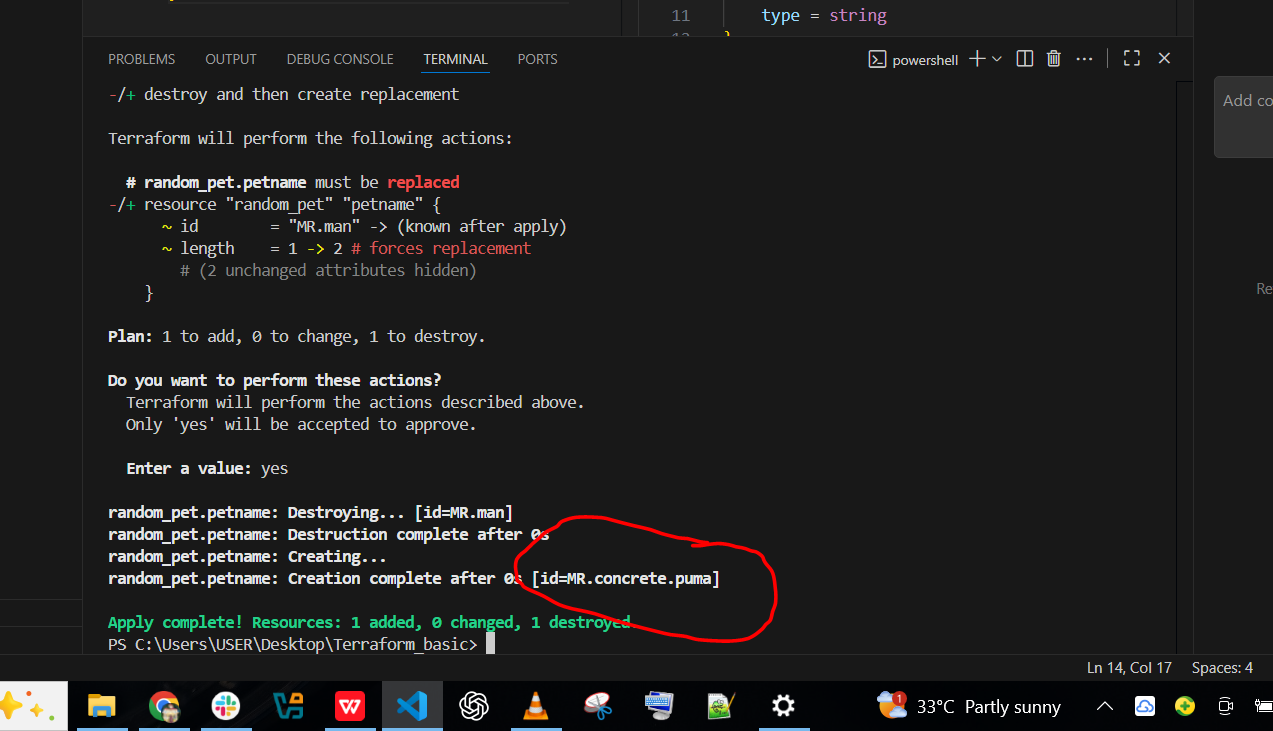
1. So we got the file name as pet.txt in which we have changed in variable file 
2. So instead of making chnges in main file file we can changes directly in variable file and it will get changed

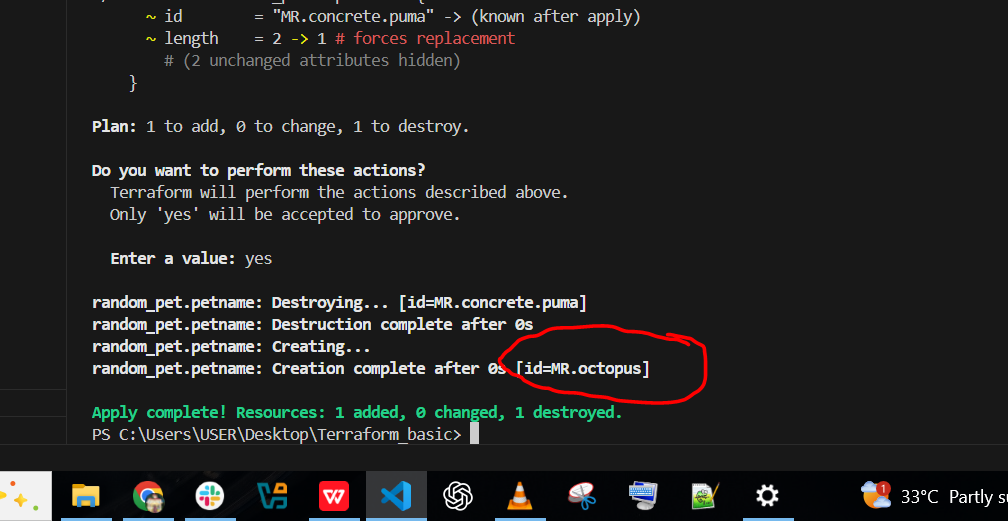


1. So here we have using the different type of variable in varibale file and changed in main.tf file also

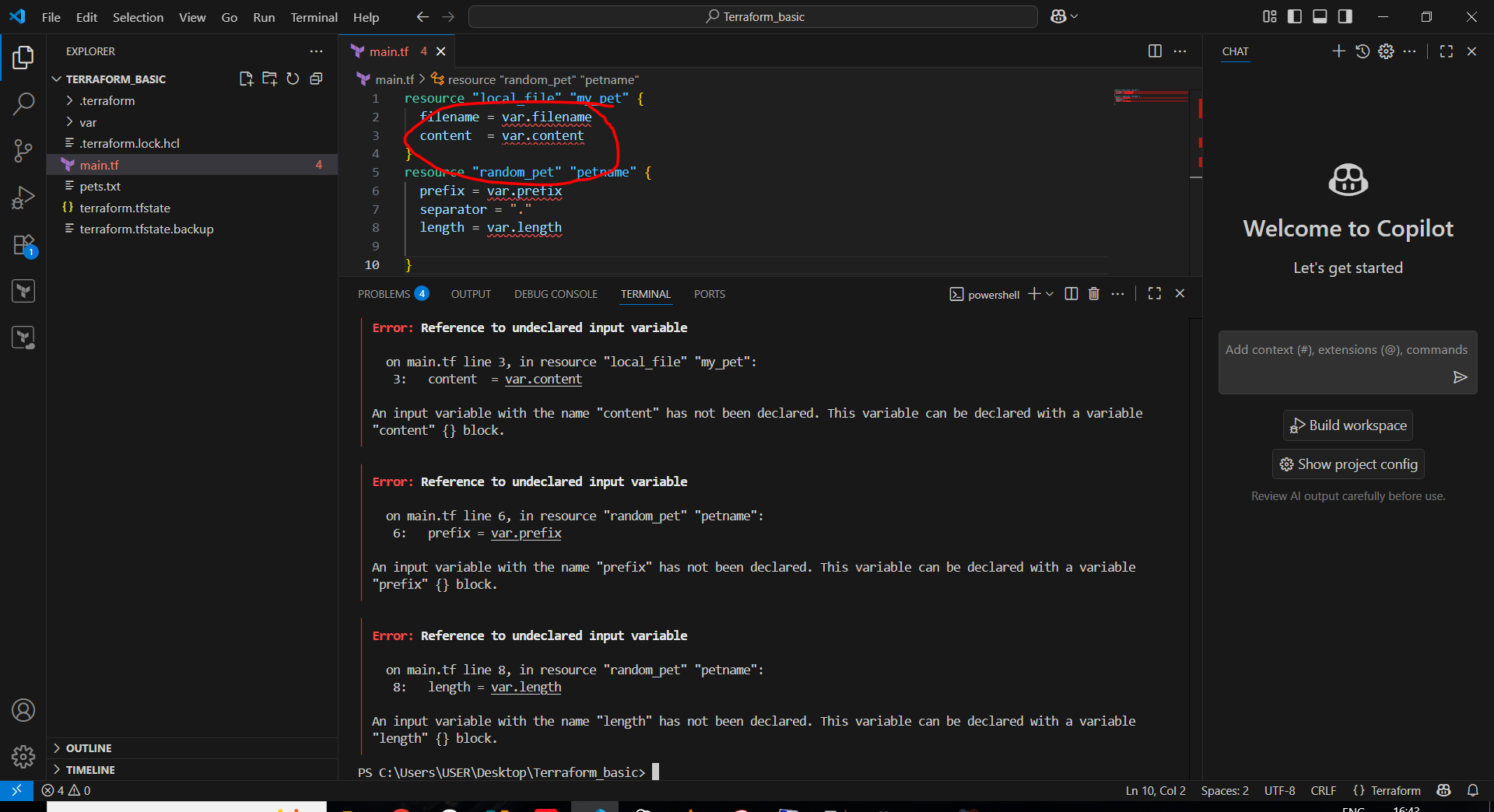


1. Here in variable file I have gave **variable length is 2** its show 2 name and then again 1 its shows one name in below picture

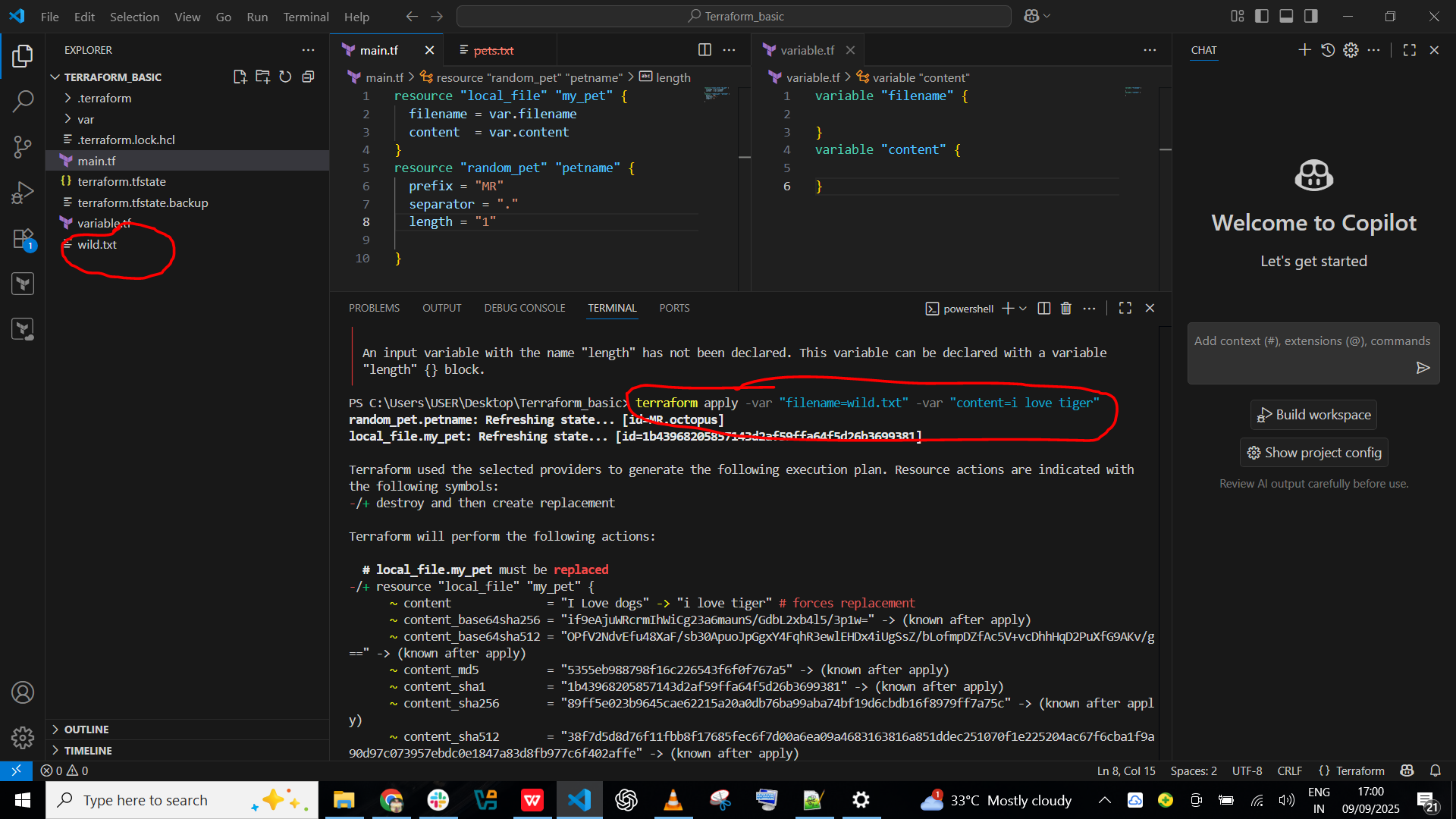




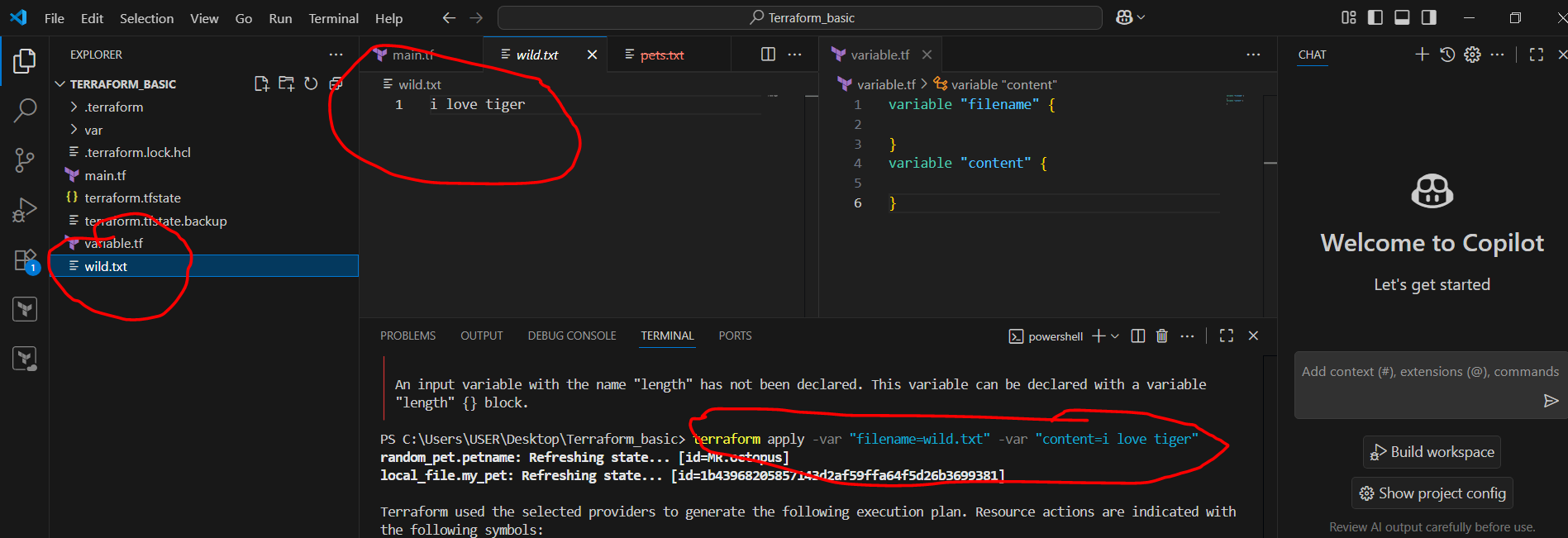
1. Here I have deleted the variable file in this, but the var.filename is there in main.tf file after the execute the **terraform apply** it will show this **Error: Reference to undeclared input variable,** because its is deleted



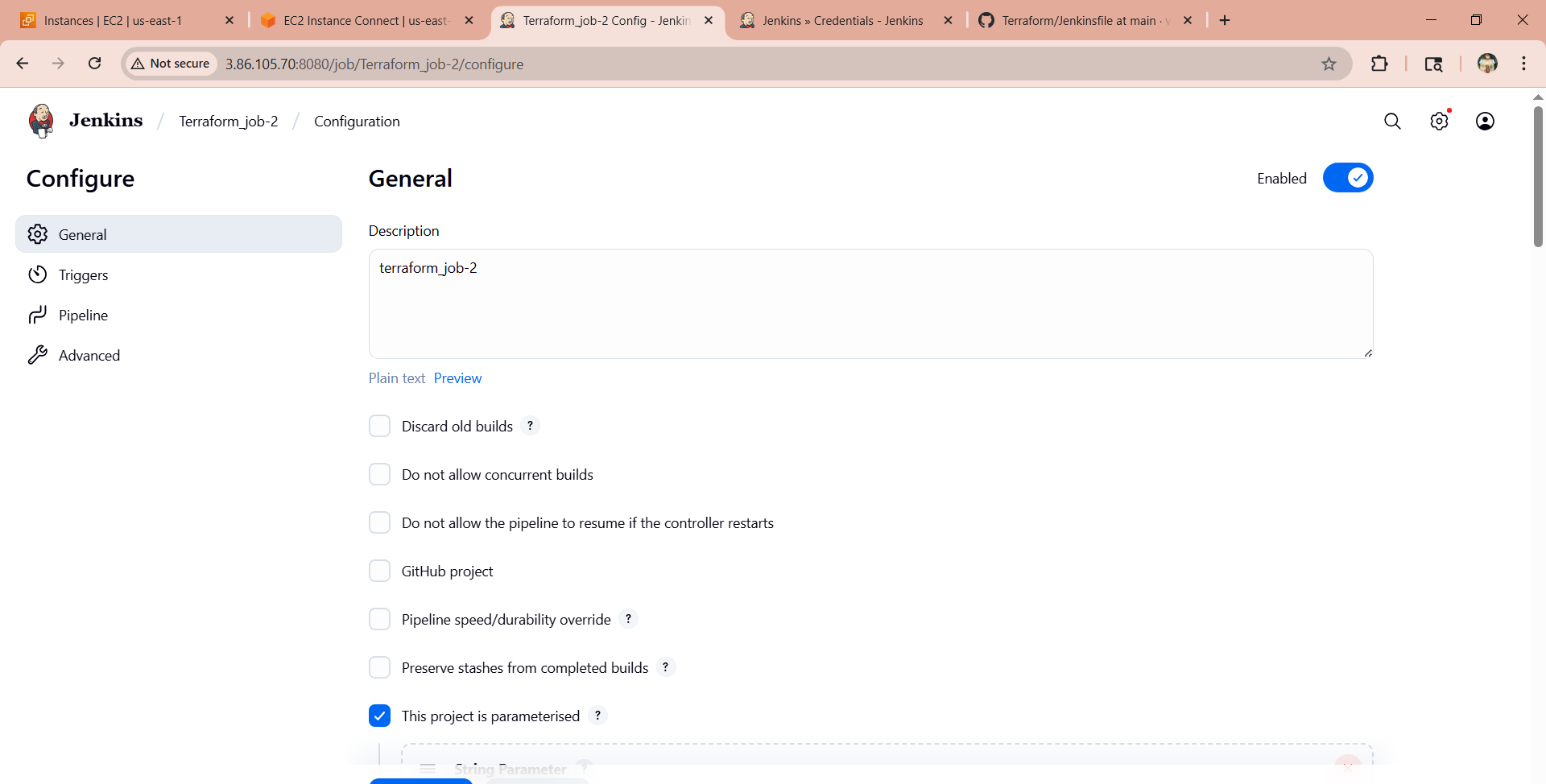
1. Here we have using the command **terraform apply -var "filename=wild.txt" -var "content=i love tiger"** and the wild.txt file will get created

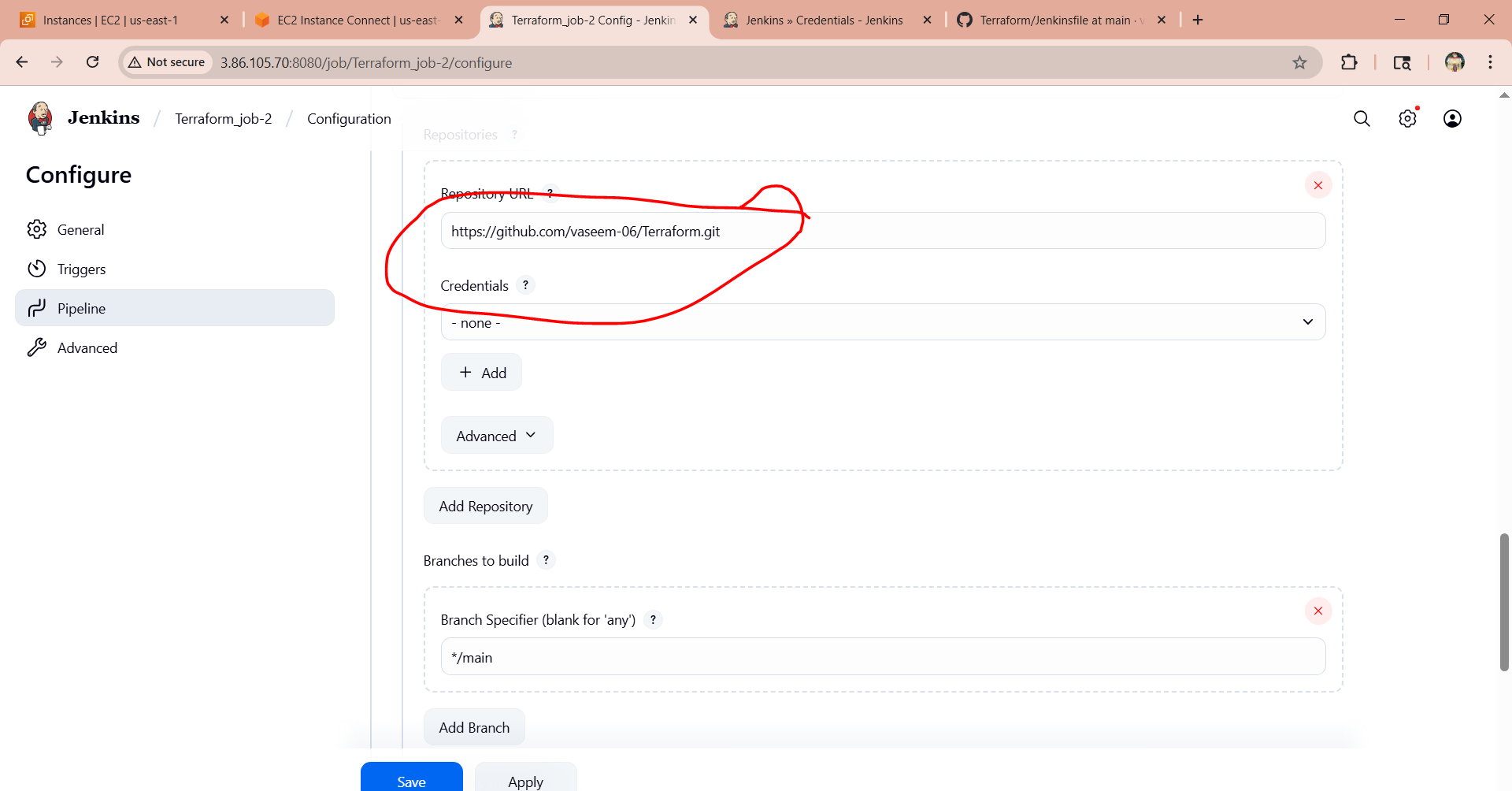


1. After going that file in content will get the **I love tiger**



**3) Integrate terraform in Jenkins using Terraform plugin.**

1. Need to create job as **Terraform\_job-2** and clone the github URL 



1. Now in github added the variable.tf file and in jenkins file updated the script as below

**pipeline {**

**agent any**

**parameters {**

**string(name: 'FILENAME', defaultValue: 'animals.txt', description: 'Name of the file to create')**

**string(name: 'CONTENT', defaultValue: 'some animals are human friendly', description: 'Content of the file')**

**}**

**environment {**

**SLACK\_CHANNEL = '#jenkins-integration' // replace with your channel**

**SLACK\_CREDENTIALS = 'slack-notification' // Jenkins credential ID for Slack token**

**}**

**stages {**

**stage('Checkout SCM') {**

**steps {**

**git branch: 'main', url: 'https://github.com/vaseem-06/Terraform.git'**

**}**

**}**

**stage('Terraform Init') {**

**steps {**

**sh 'terraform init'**

**}**

**}**

**stage('Terraform Plan') {**

**steps {**

**sh """**

**terraform plan -out=tfplan \**

**-var="filename=${params.FILENAME}" \**

**-var="content=${params.CONTENT}"**

**"""**

**}**

**}**

**stage('Terraform Apply') {**

**steps {**

**input message: ":warning: Do you want to apply Terraform changes?"**

**sh 'terraform apply -auto-approve tfplan'**

**}**

**}**

**}**

**post {**

**success {**

**echo ":white\_check\_mark: Terraform executed successfully. File: ${params.FILENAME}"**

**slackSend (**

**channel: "${env.SLACK\_CHANNEL}",**

**color: '#36A64F',**

**message: ":white\_check\_mark: Terraform executed successfully!\n\*File:\* ${params.FILENAME}\n\*Content:\* ${params.CONTENT} (By Imran Khan)"**

**)**

**}**

**failure {**

**echo ":x: Terraform pipeline failed!"**

**slackSend (**

**channel: "${env.SLACK\_CHANNEL}",**

**color: '#FF0000',**

**message: ":x: Terraform pipeline failed!"**

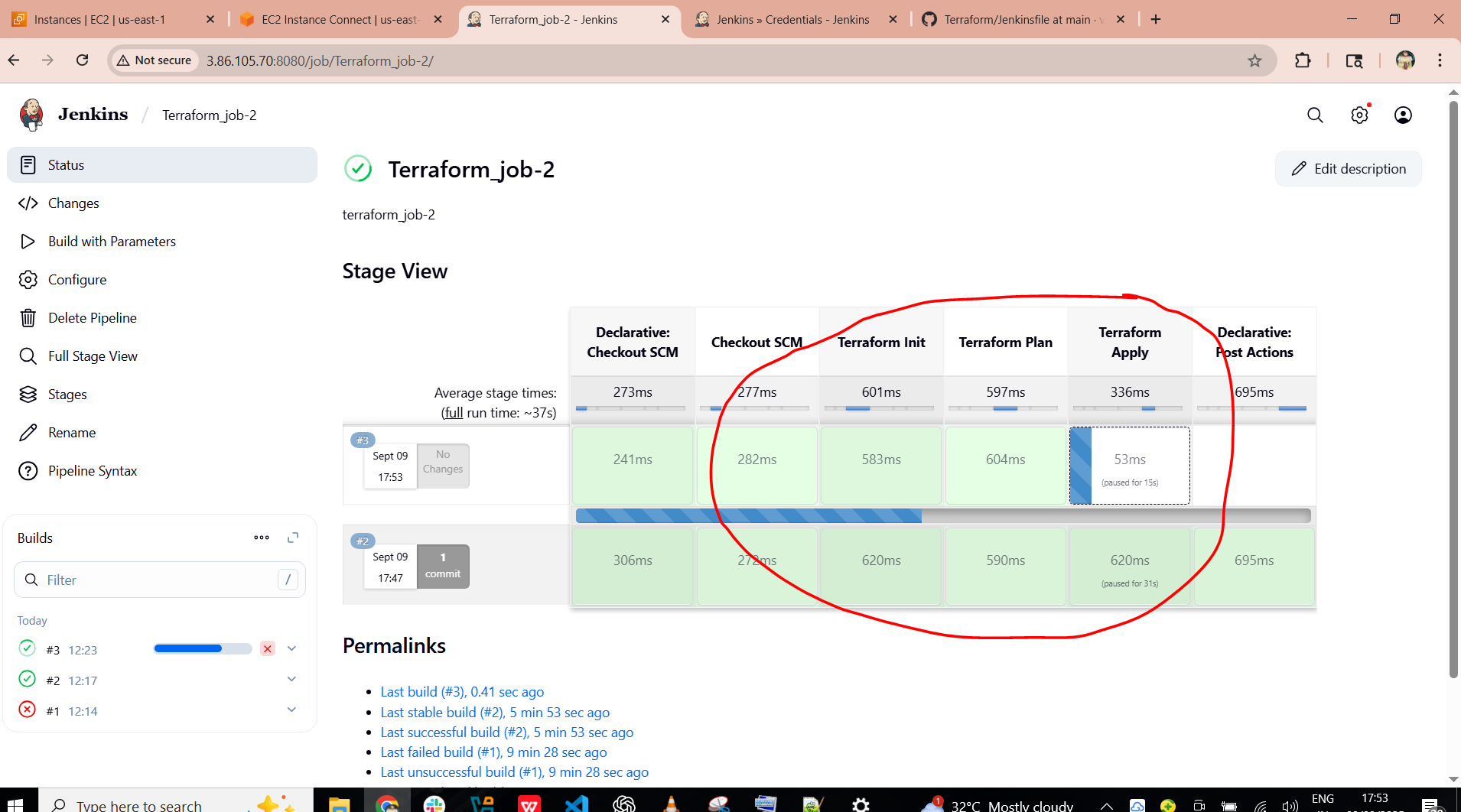
**)**

**}**

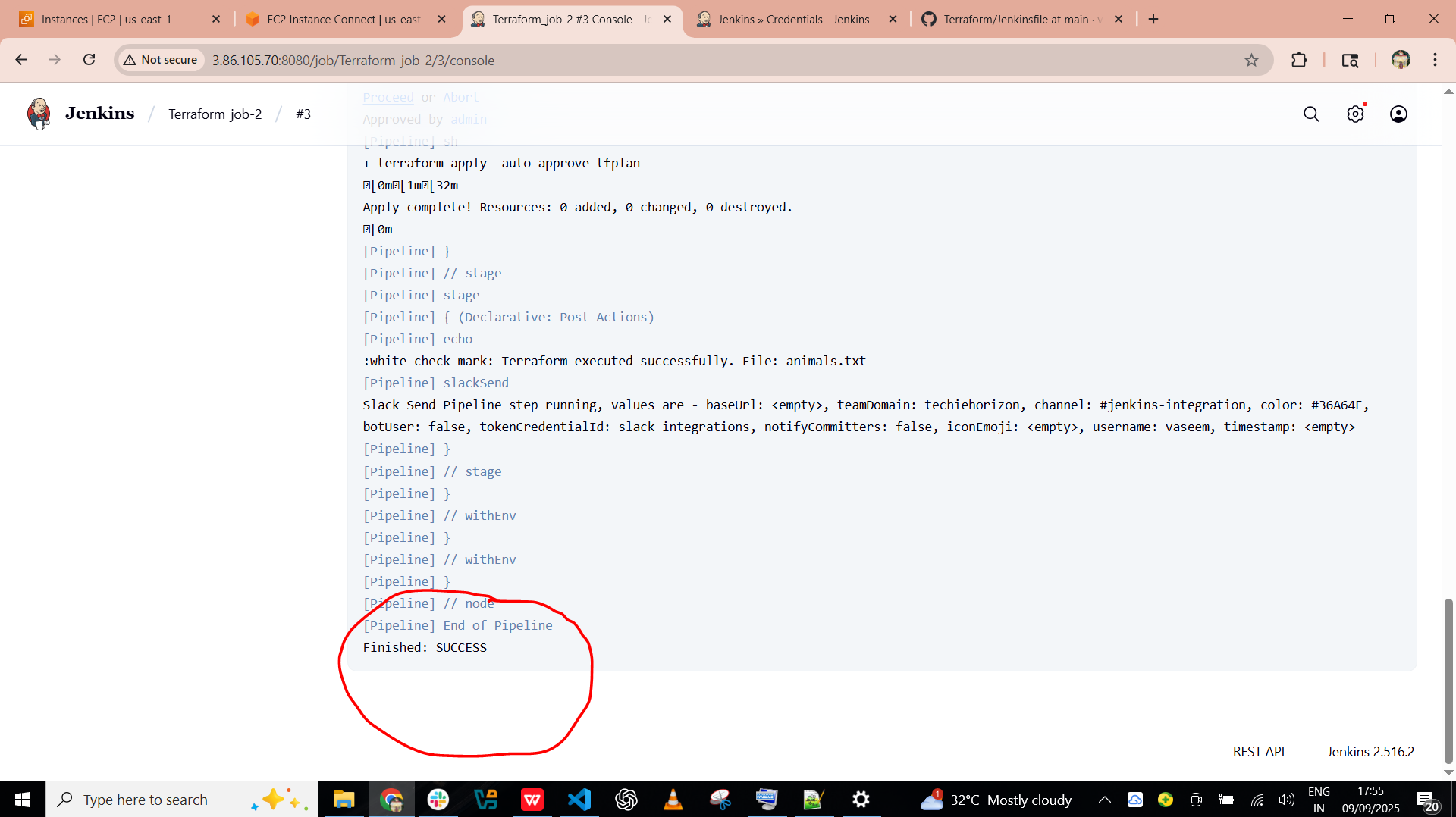
**}**

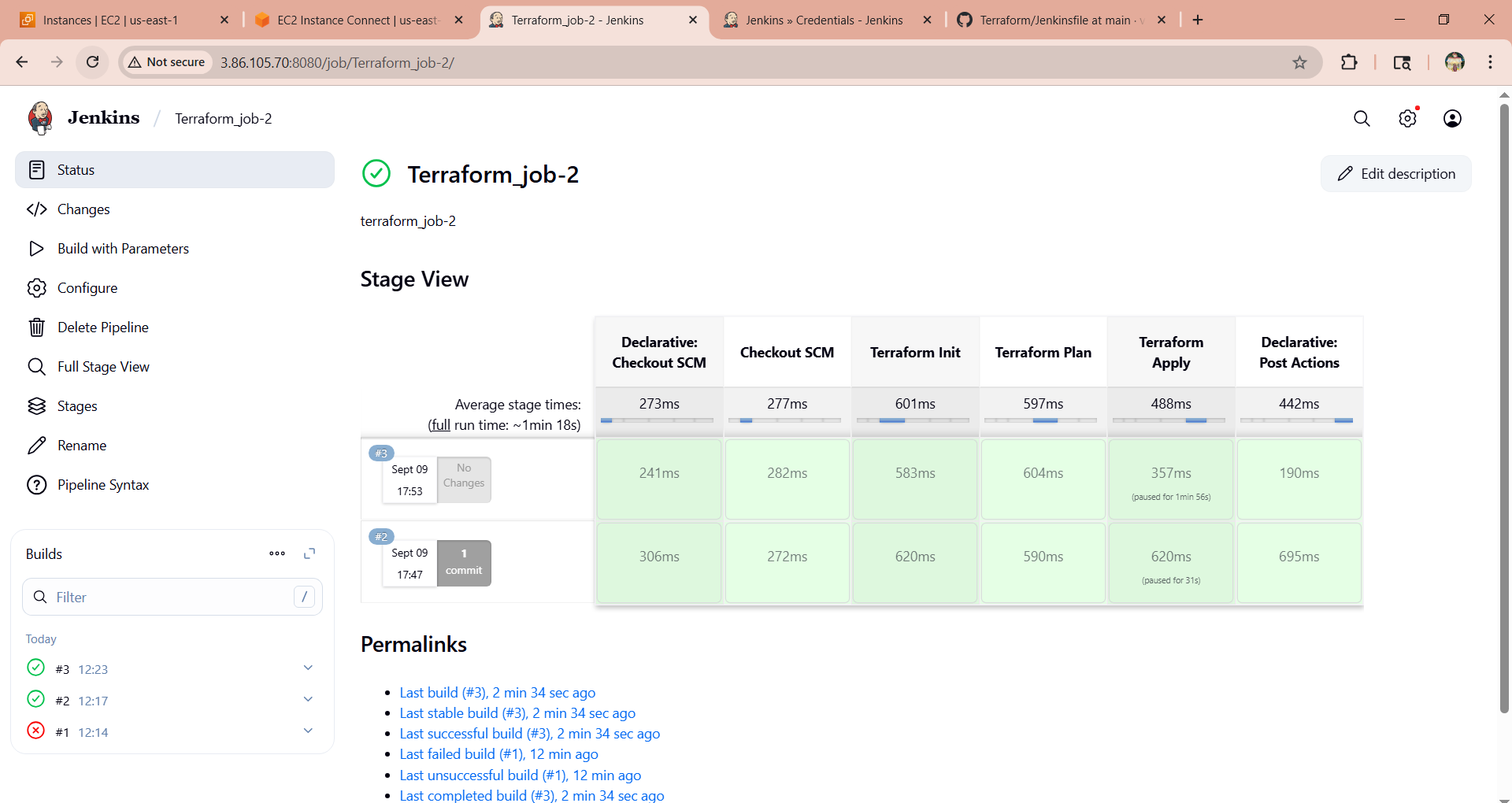
**}**

1. After updating that script in github need to run the job in jenkins and while running the job it will get stop here

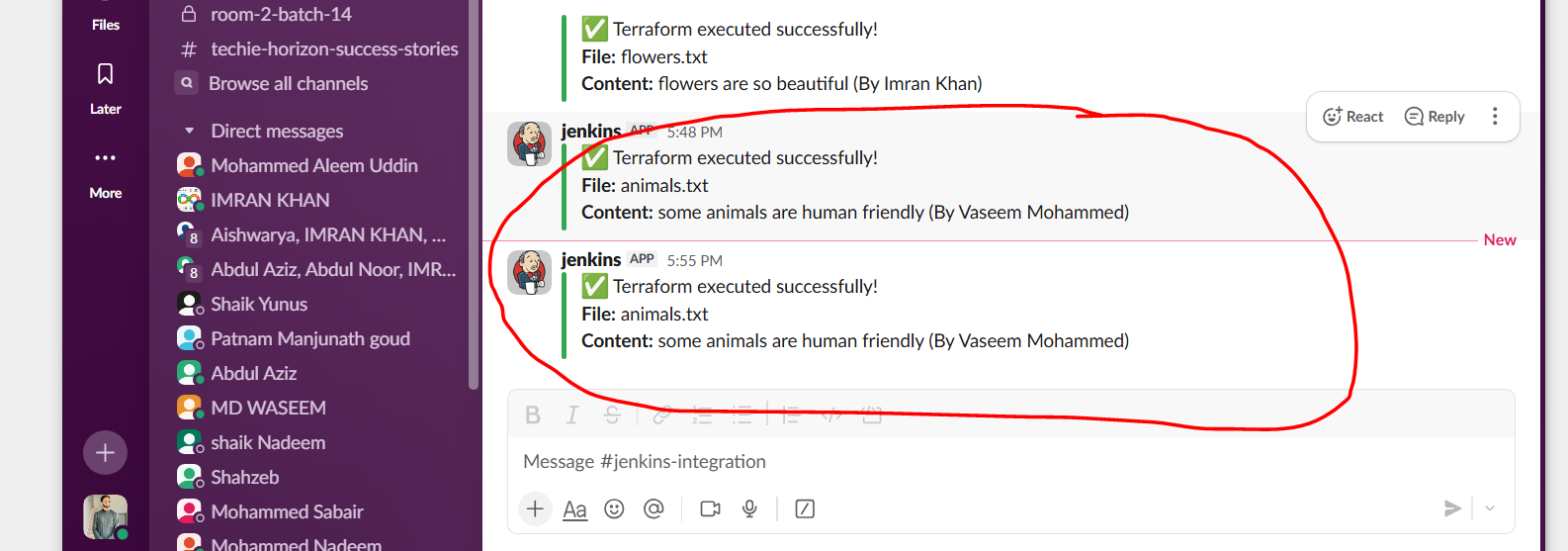


1. So we need to go in console output and click on **Proceed** than it will run the job after that only as shown in picture





1. Than will get notification our slack channel as well



1. Go to jenkins and configure --> cd /var/lib/jenkins/workspace --> ls --> cd Terraform\_job-2 --> ls

