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

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

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

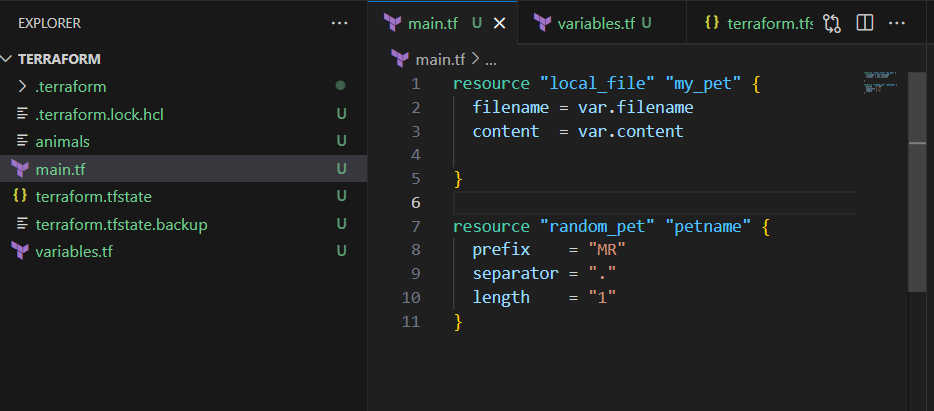
**1) Watch the terraform-02 video.**

**Completed.**

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

**Providers:**

1. **Official provided by terraform**
2. **Partner provided by third party vendors**
3. **Community individual who can create**

****

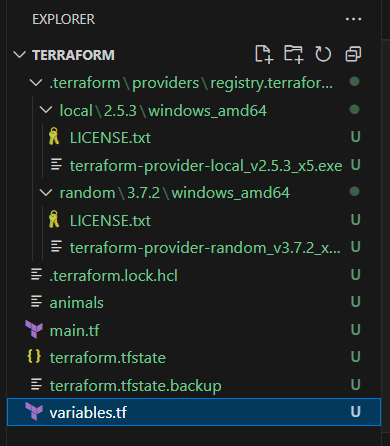
**Configuration Directory:**

**Main.tf -->main configuration file containing resource definition**

**variables.tf --> contains varibles declaration**

**output.tf --> contains outputs from resources**

**provider.tf --> contains the provider definition**

****

**Terraform mutable vs immutable infrastructure:**

**Terraform as a IAC tool uses immutable infrastructure strategy.**

**Immutable means deleting the older infra and creating a newer one with a new update.**

**Mutable means using the existing infra and updating the system with newer versions.**

**Lifecycle rules:**

**We can configure lifecycle rules to our resource file.**

**Terraform will destroy the file before creating a new file by default.**

**We can create a file before destruction by using the lifecycle rules.**

**resource "local\_file" "pet" {**

**filename = "/root/pets.txt"**

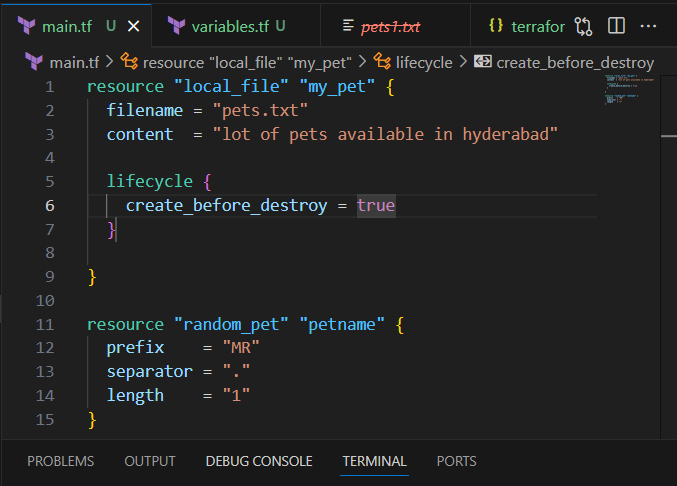
**content = "My cat is MR.CAT"**

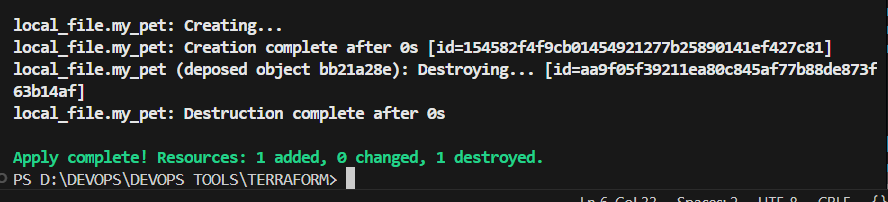
**lifecylce {**

**create\_before\_destory = true**

**}**

**}**

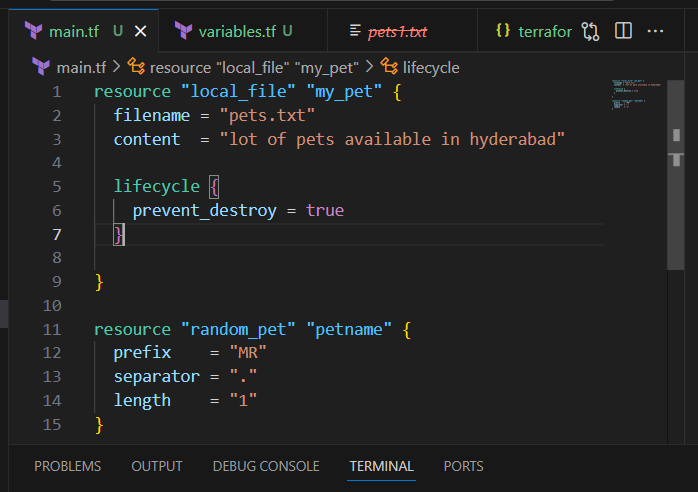
****

****

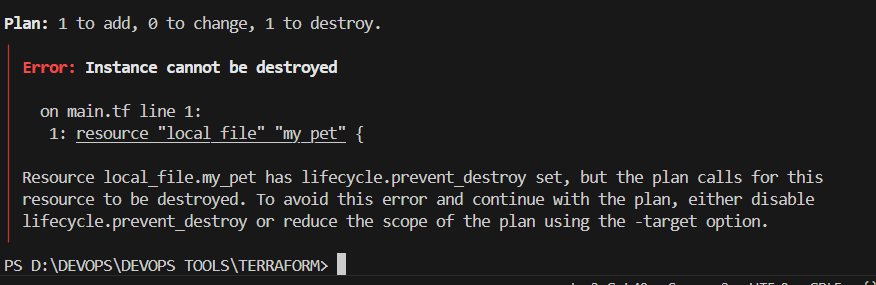
**lifecycle {**

**prevent\_destroy = true**

**}**

****

**If going to change content and doing terraform apply it will stops.**

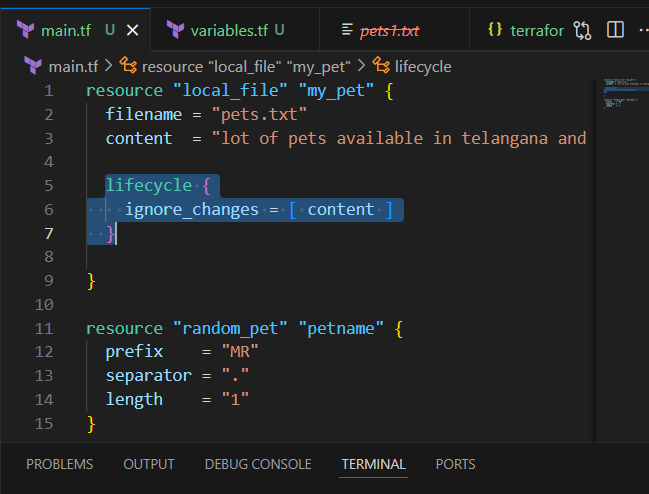
****

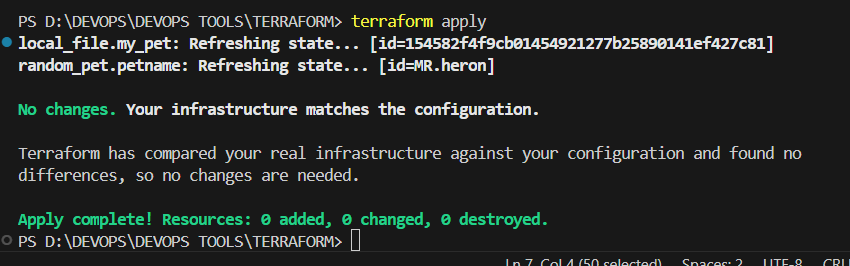
**lifecycle {**

**ignore\_changes = [ content ]**

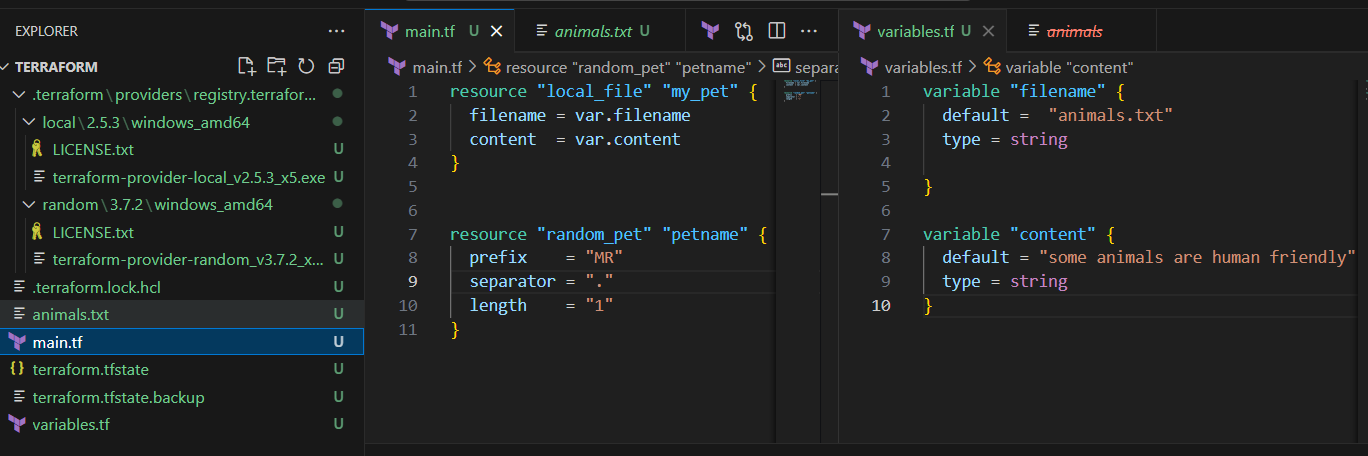
**}**

**If do changes then it will ignore and remain same.**

****

****

**Variables**

****

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

**filename = var.filename**

**content = var.content**

**}**

**resource "random\_pet" "petname" {**

**prefix = "MR"**

**separator = "."**

**length = "1"**

**}**

**variable "filename" {**

**default = "animals.txt"**

**type = string**

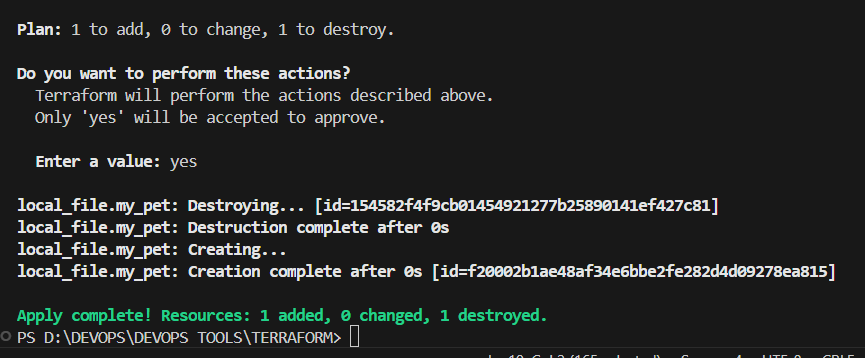
**}**

**variable "content" {**

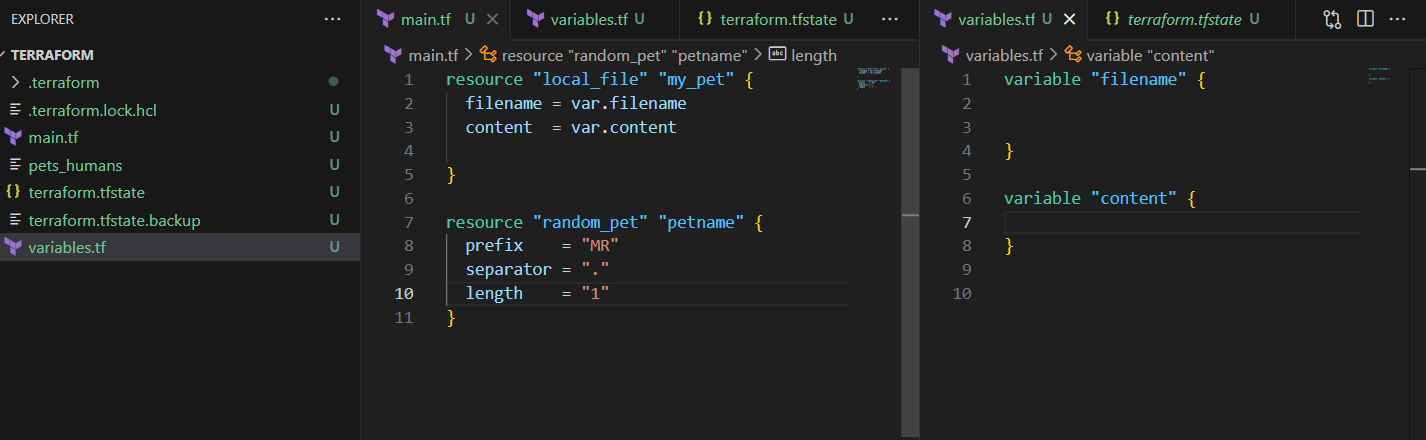
**default = "some animals are human friendly"**

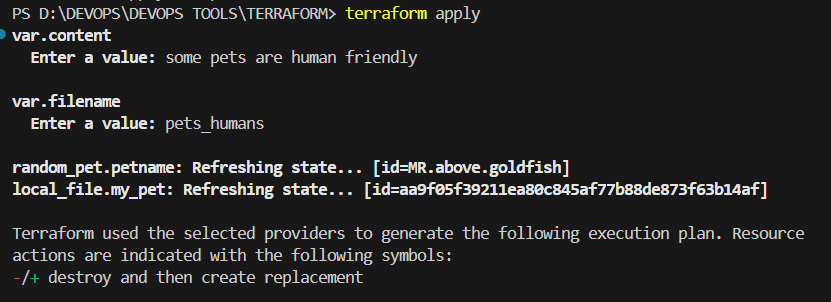
**type = string**

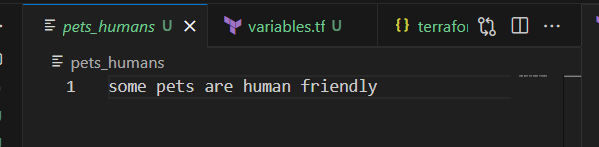
**}**

****

**Taking user input**

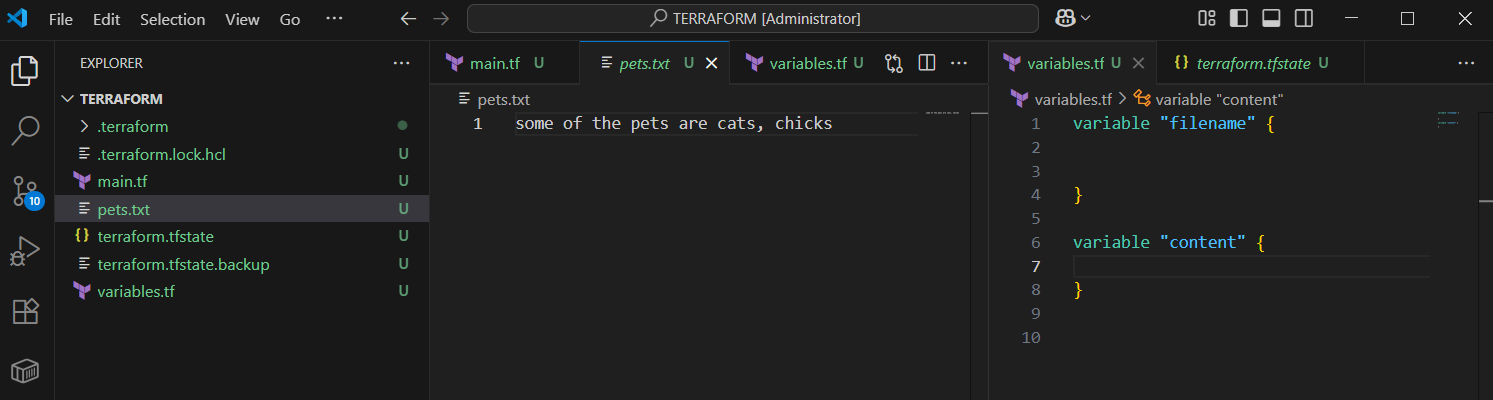
****

****

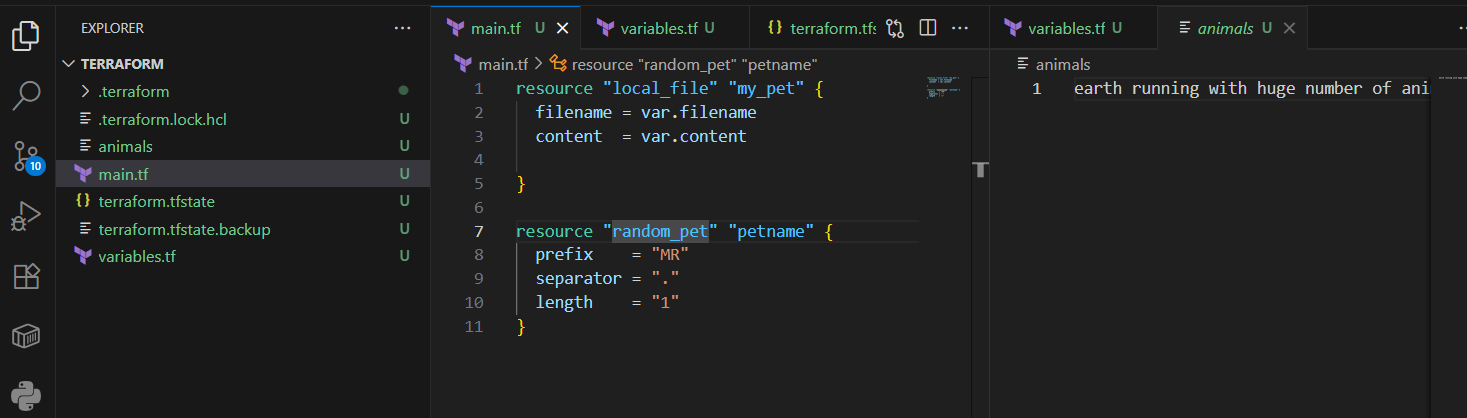
****

**Passing values using command**

**terraform apply -var "filename=pets.txt" -var "content=some of the pets are cats, chicks"**

****

**terraform apply -var "filename=animals" -var "content=earth running with huge number of animals"**

****

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

1. **Jenkins ec2 launched , git, java, jenkins installed.**
2. **In jenkins master terraform plugin installed.**
3. **Installed terraform**

**# Download latest version (Linux 64-bit)**

**wget** [**https://releases.hashicorp.com/terraform/1.13.1/terraform\_1.13.1\_linux\_amd64.zip**](https://releases.hashicorp.com/terraform/1.13.1/terraform_1.13.1_linux_amd64.zip)

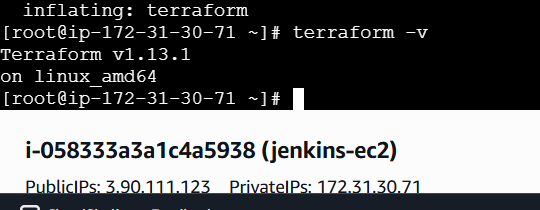
**# Unzip and move**

**unzip terraform\_1.13.1\_linux\_amd64.zip**

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

**# Verify**

**terraform -v**

****

**4. Using git repo 1 job created in jenkins master.**

[**https://github.com/imrankhanmohammad257/terraform.git**](https://github.com/imrankhanmohammad257/terraform.git)

**5. Declarative script taken in Jenkinsfile of the github repo.**

# **📘 Jenkins + Terraform + Slack Integration Documentation**

## **🔹 1. Prerequisites**

* **Jenkins Server installed and running.**
* **Terraform CLI installed on Jenkins (or configured via Jenkins tools).**
* **GitHub Repository with Terraform files (main.tf, variables.tf, Jenkinsfile).**
* **Slack Workspace with an incoming webhook or Slack App.**

## **🔹 2. Configure Jenkins**

### **(a) Install Required Plugins**

1. **Go to Manage Jenkins → Plugins → Available Plugins.**
2. **Install:**
   * **Terraform Plugin**
   * **Pipeline Plugin (if not already installed)**
   * **Git Plugin**
   * **Slack Notification Plugin, stage view plugin**

### **(b) Add GitHub Repo**

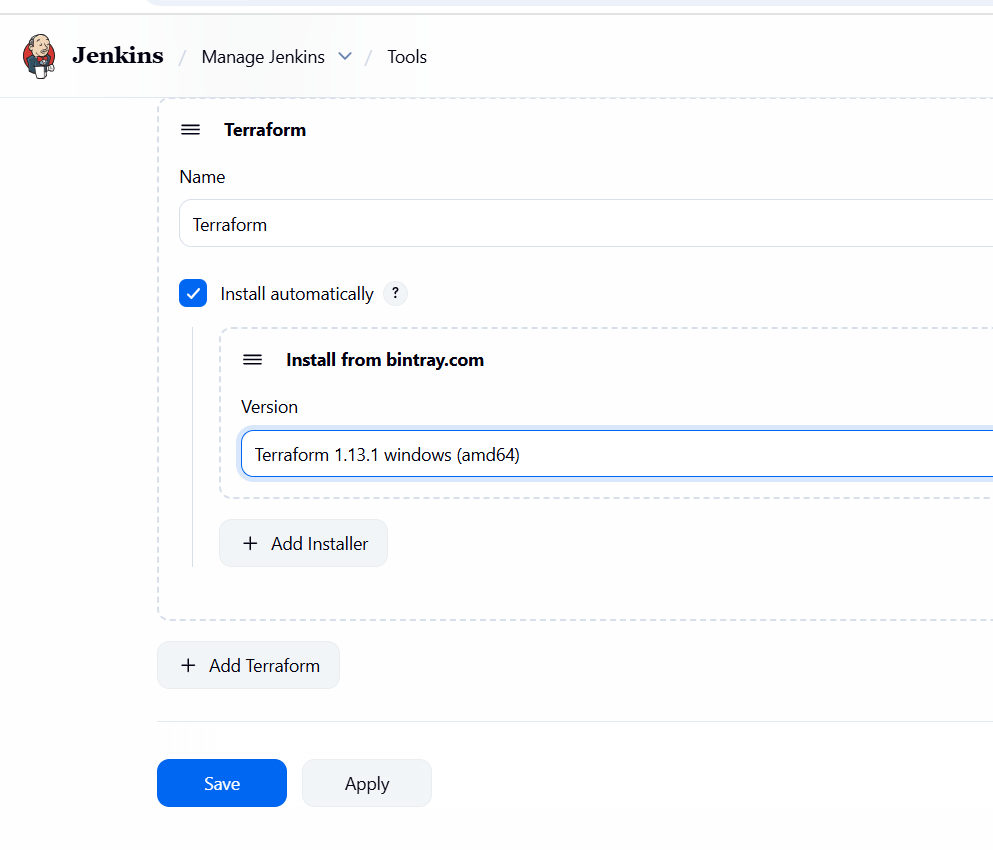
1. **Create a Pipeline Job in Jenkins → “New Item → Pipeline”.**
2. **Configure:**
   * **Select Pipeline script from SCM.**
   * **SCM: Git → Enter repo URL:**

[**https://github.com/imrankhanmohammad257/terraform.git**](https://github.com/imrankhanmohammad257/terraform.git)

**Branch: main.**

### **(c) Configure Terraform in Jenkins**

1. **Go to Manage Jenkins → Global Tool Configuration.**
2. **Add Terraform installation:**
   * **Name: Terraform**
   * **Version: 1.13.1**

****

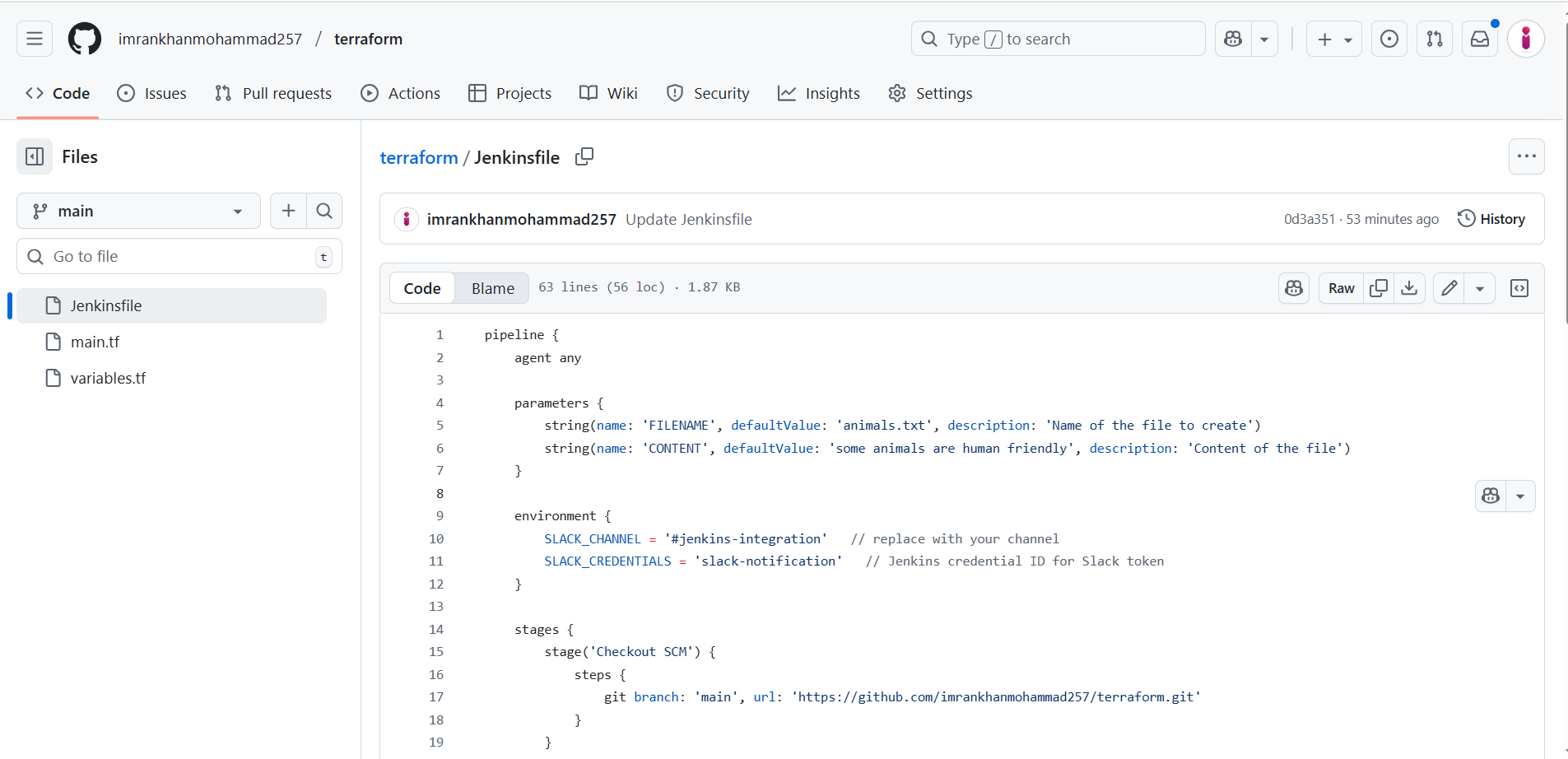
### **(d) Configure Slack in Jenkins**

1. **In Slack:**
   * **Go to Slack API → Create App → Bots → Incoming Webhooks.**
   * **Add Jenkins to your channel.**
   * **Copy the Webhook URL or Token.**
2. **In Jenkins:**
   * **Go to Manage Jenkins → Configure System → Slack.**
   * **Add:**
     + **Workspace: your Slack workspace**
     + **Channel: #jenkins-alerts (or your channel)**
     + **Integration Token: Save it in Jenkins Credentials (Secret Text).**
   * **Note the Credential ID (e.g., slack-token).**

## **🔹 3. GitHub Repo Structure**

**Your repo should look like this:**

****

****

**main.tf**

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

**filename = var.filename**

**content = var.content**

**}**

**resource "random\_pet" "petname" {**

**prefix = "MR"**

**separator = "."**

**length = 1**

**}**

**variables.tf**

**variable "filename" {**

**description = "The name of the file to create"**

**type = string**

**}**

**variable "content" {**

**description = "The content to put inside the file"**

**type = string**

**}**

**🔹 4. Jenkinsfile (Pipeline Script)**

**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/imrankhanmohammad257/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: "⚠️ Do you want to apply Terraform changes?"**

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

**}**

**}**

**}**

**post {**

**success {**

**echo "✅ Terraform executed successfully. File: ${params.FILENAME}"**

**slackSend (**

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

**color: '#36a64f',**

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

**)**

**}**

**failure {**

**echo "❌ Terraform pipeline failed!"**

**slackSend (**

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

**color: '#FF0000',**

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

**)**

**}**

**}**

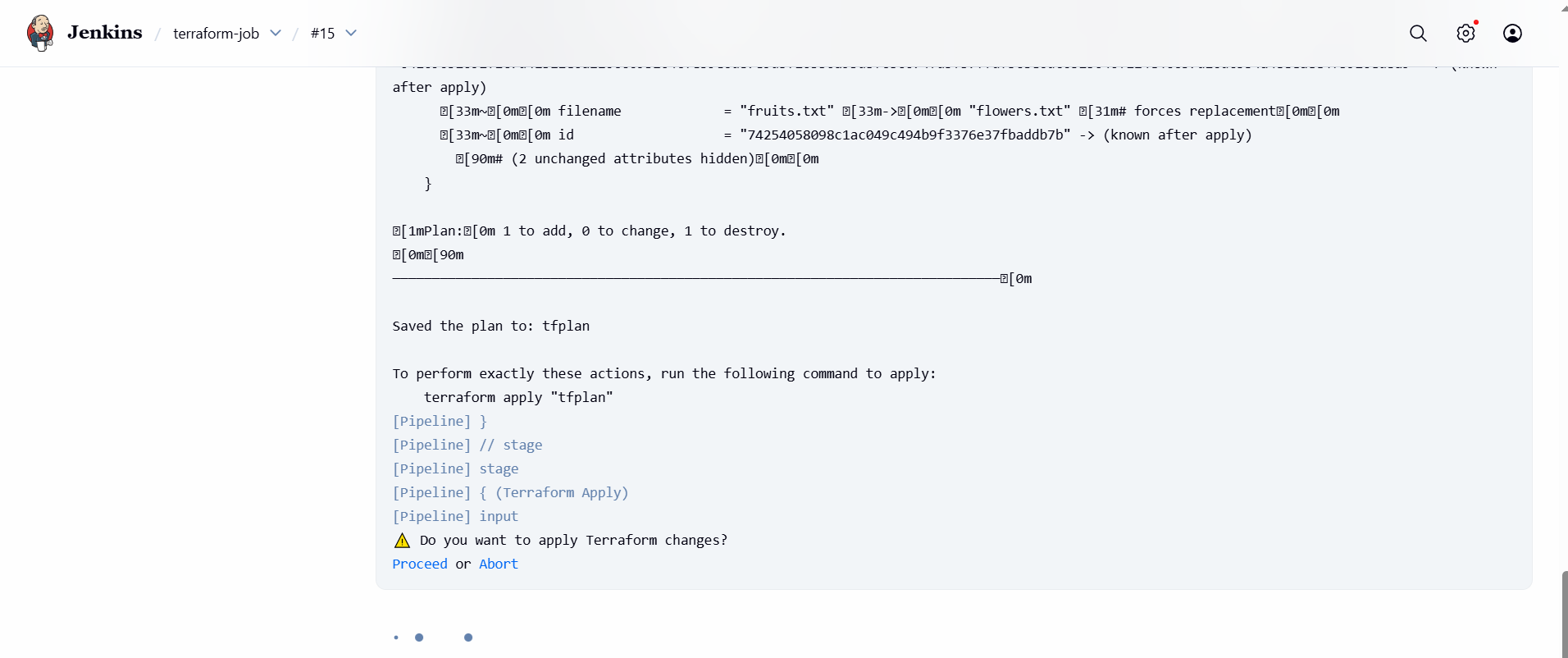
**}**

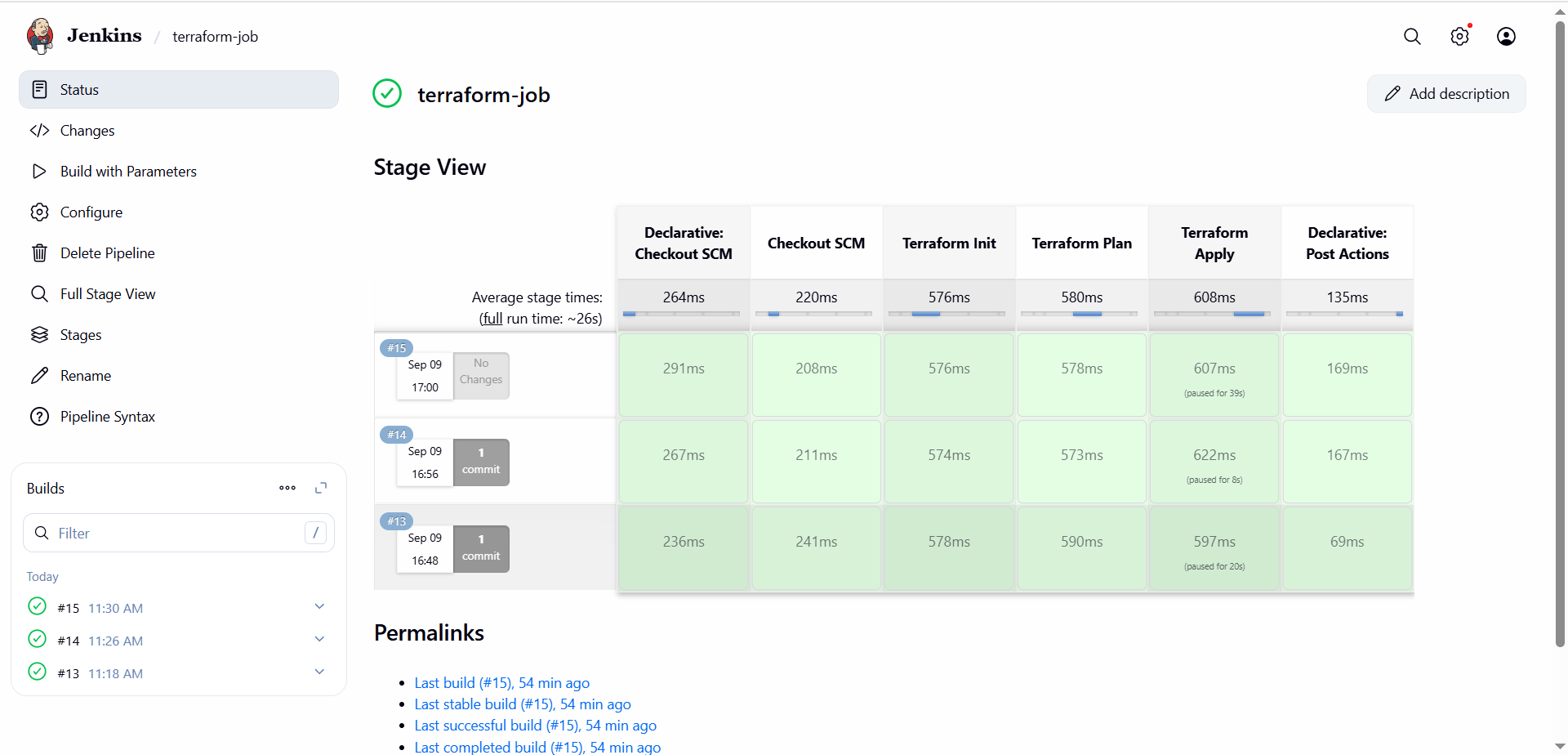
## **🔹 5. Run the Pipeline**

1. **In Jenkins, run the pipeline job.**
2. **Provide parameters (filename, content) if asked.**
3. **Jenkins will:**
   * **Checkout Git repo**
   * **Run terraform init**
   * **Run terraform plan with variables**
   * **Wait for manual approval**
   * **Run terraform apply**
   * **Send success/failure message to Slack.**

## 

**While running build, jenkins stop the job at apply stage need to open console output and click on proceed or abort.**

****



## 

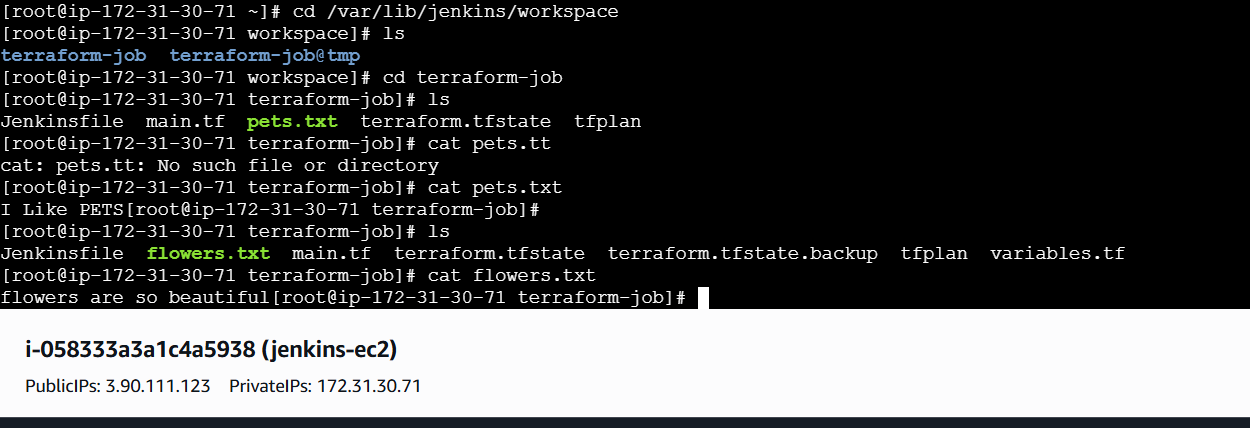
## 

## 

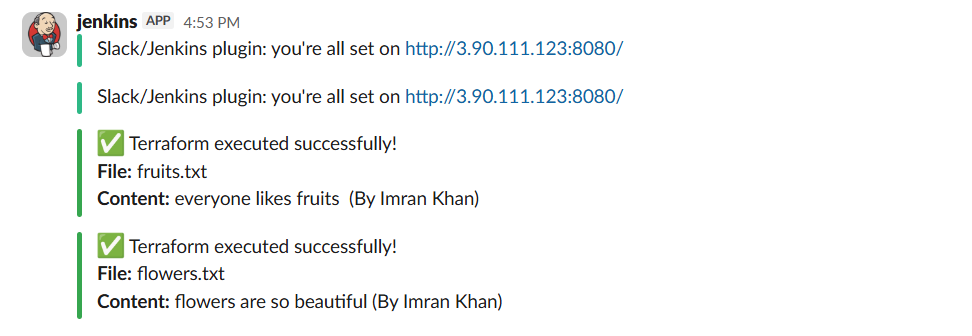
## **🔹 6. Verify the Output**

* **The file (flowers.txt) will be created inside the Jenkins workspace:**

**/var/lib/jenkins/workspace/terraform-job/animals.txt**

****

**Slack channel will receive:**

****