**Terraform Task-4**

1. **Watch Terraform-04 video.**

**Completed**

1. **Execute the script shown in the video.**

**1. Terraform Versions**

**version = "2.3.0" --> download the exact version**

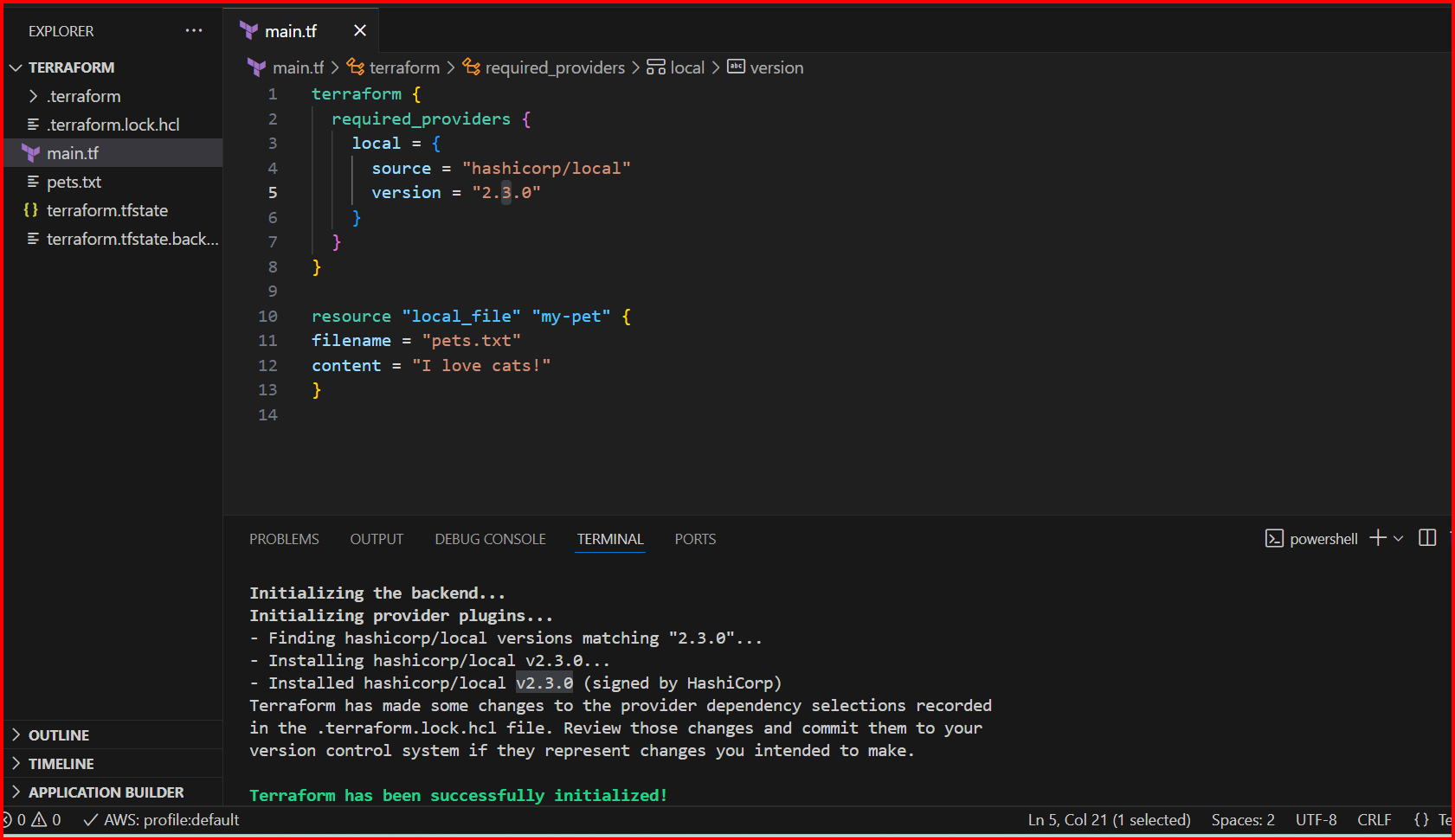
**version = "!=2.3.0" --> will not use the mentioned version**

**version = "< 2.3.0" --> lesses than the mention version**

**version = "> 2.3.0" --> greater than the given version**

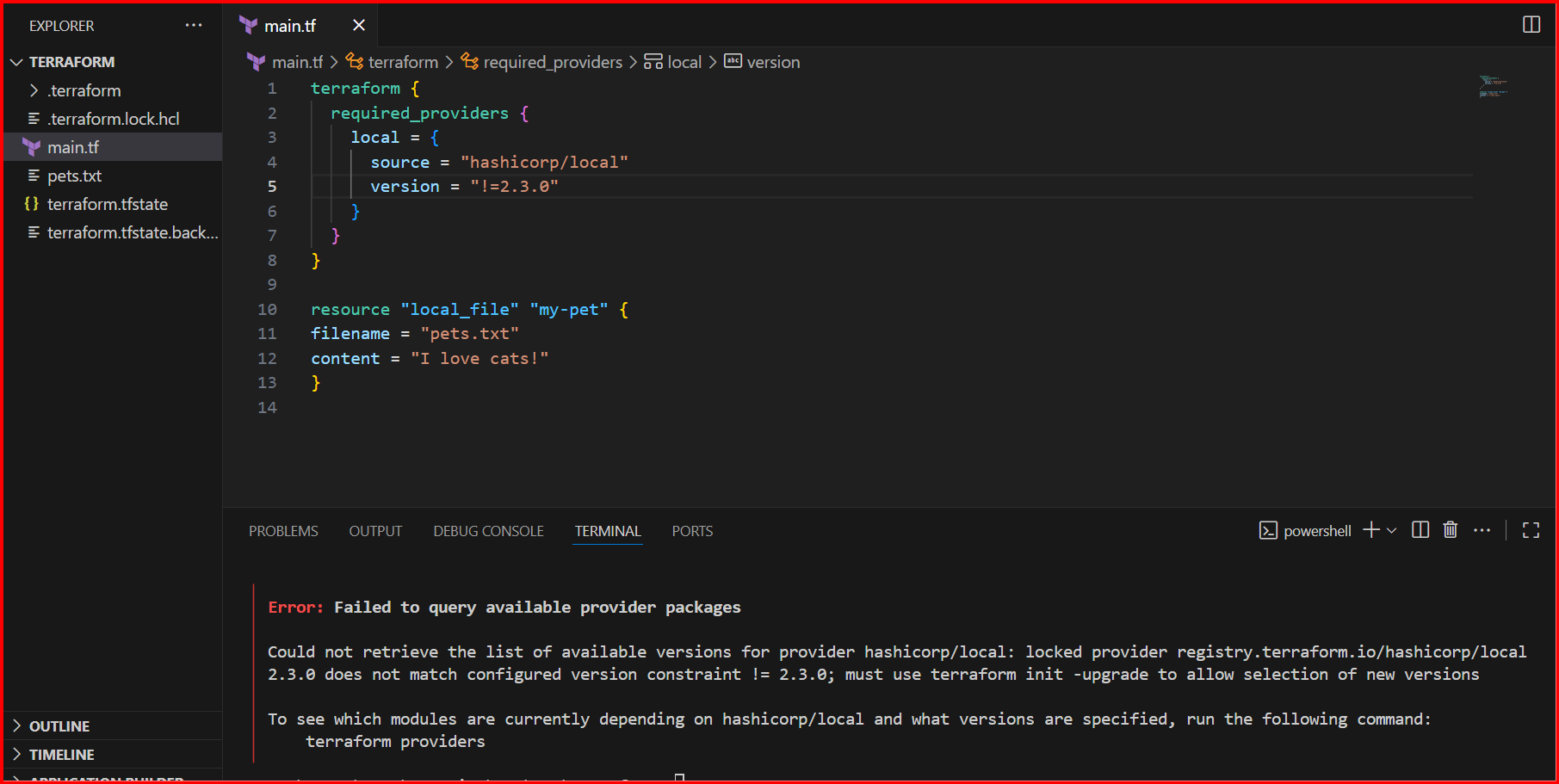
**version = "~> 2.3.0" --> specific version or higher version.**

**version = "2.3.0"**

****

**version = "!=2.3.0"**

**error**

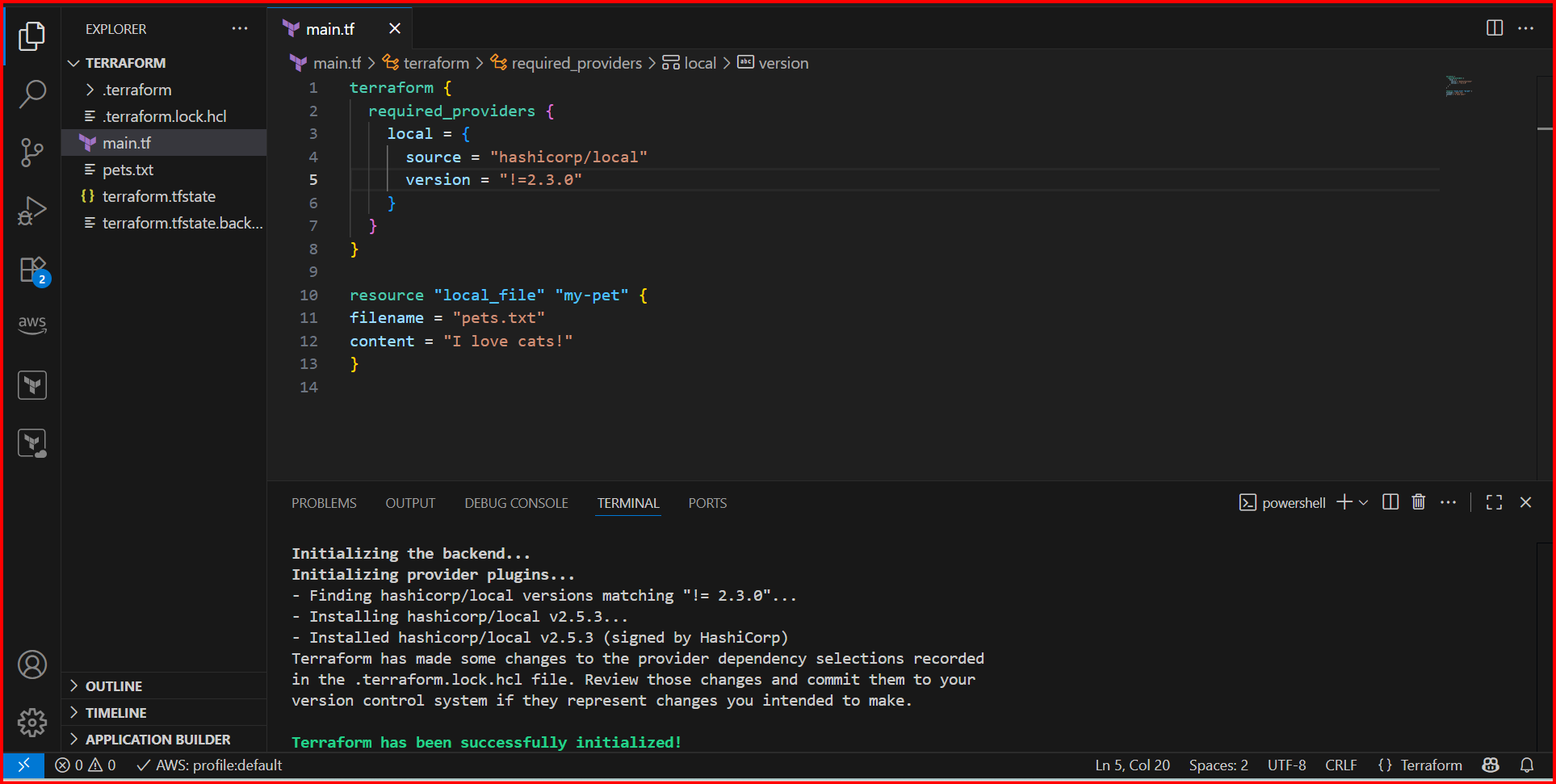
****

**For this you should do terraform init -upgrade**

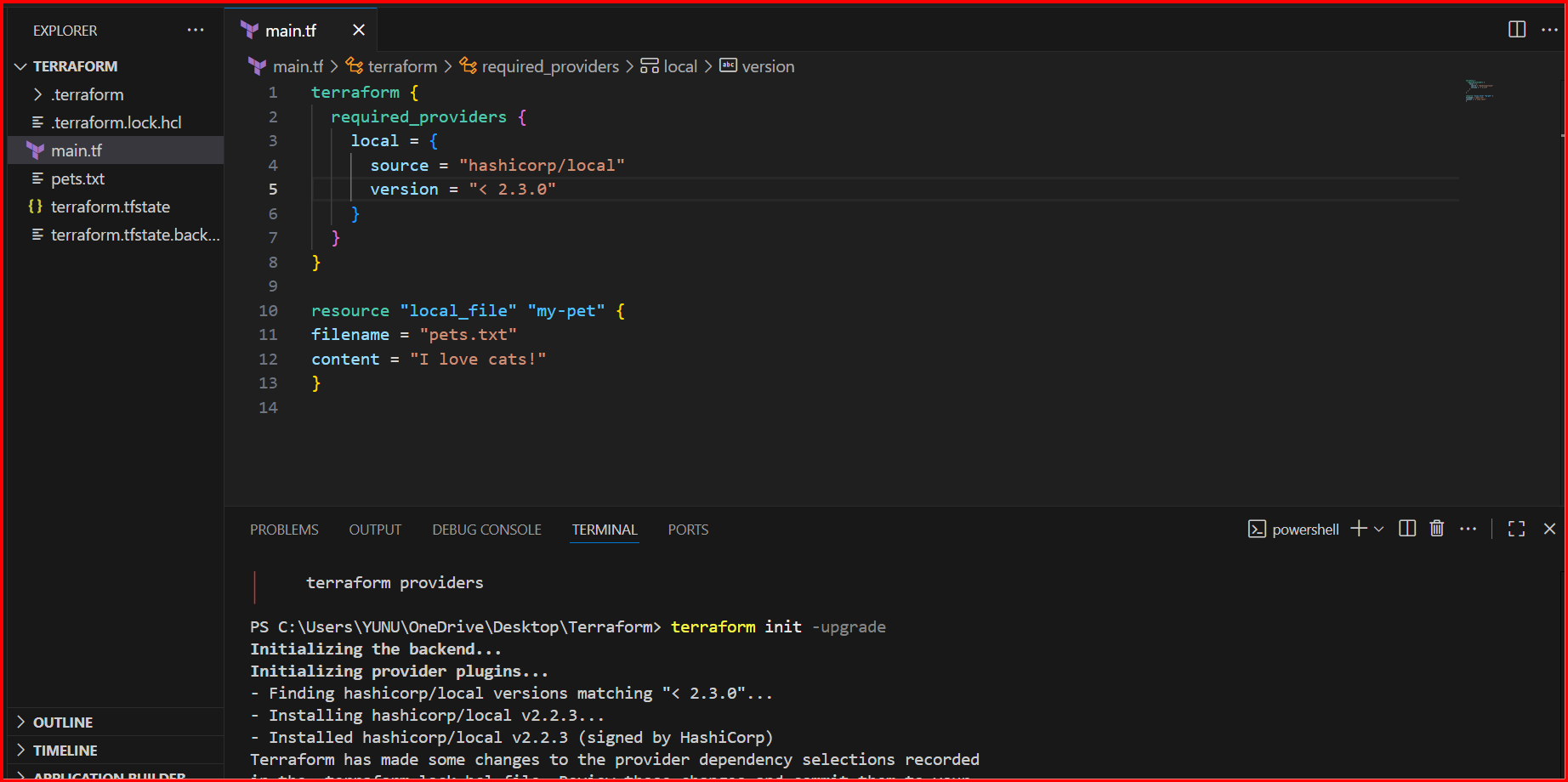
**You have a provider hashicorp/local in your project.**

**Your config or lock file (.terraform.lock.hcl) expects version 2.3.0, but Terraform sees version 2.5.3 available in your system/registry.**

**Terraform refuses to continue because of the mismatch.**

****

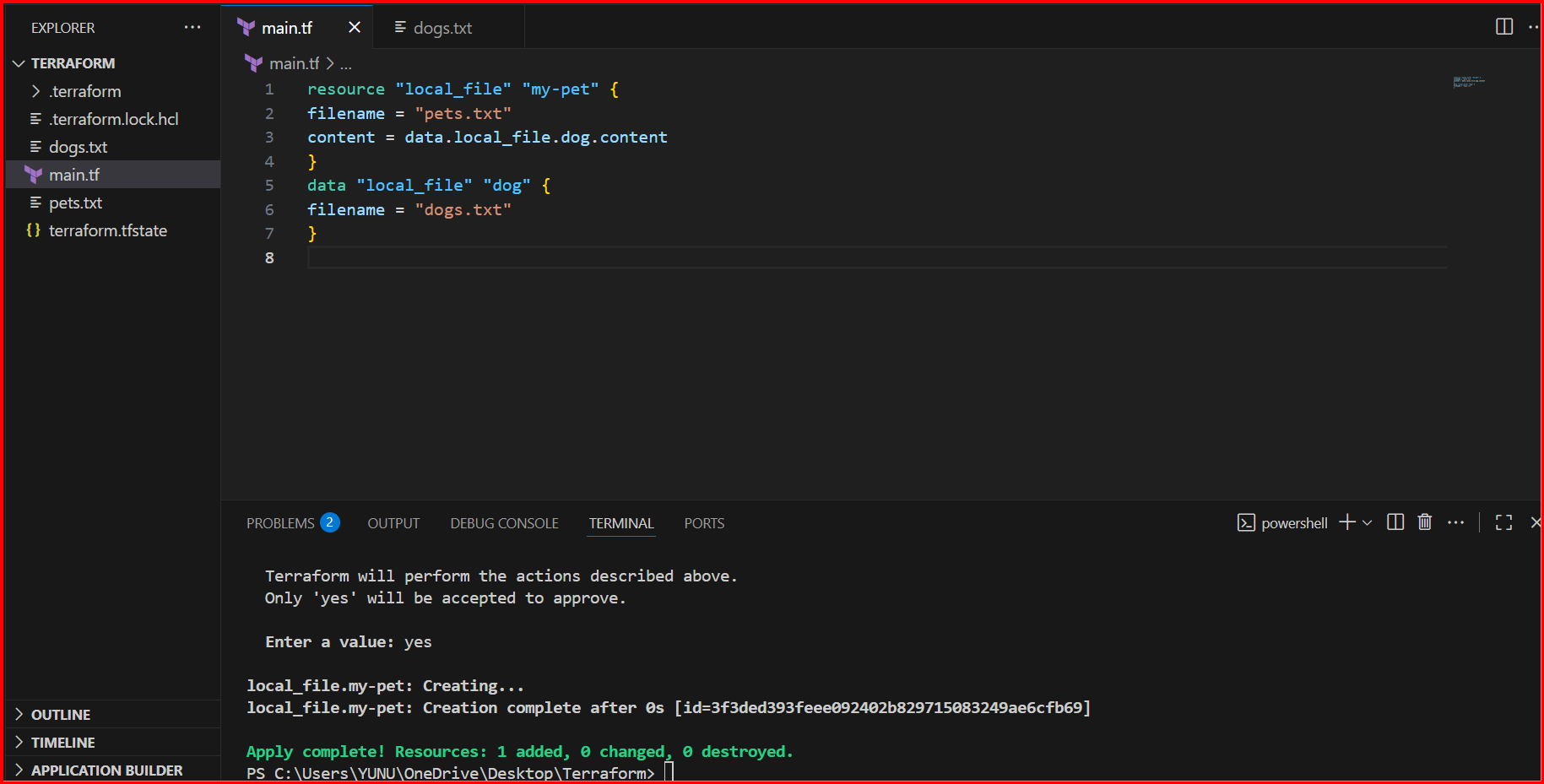
**version = "< 2.3.0" --> lesses than the mention version**

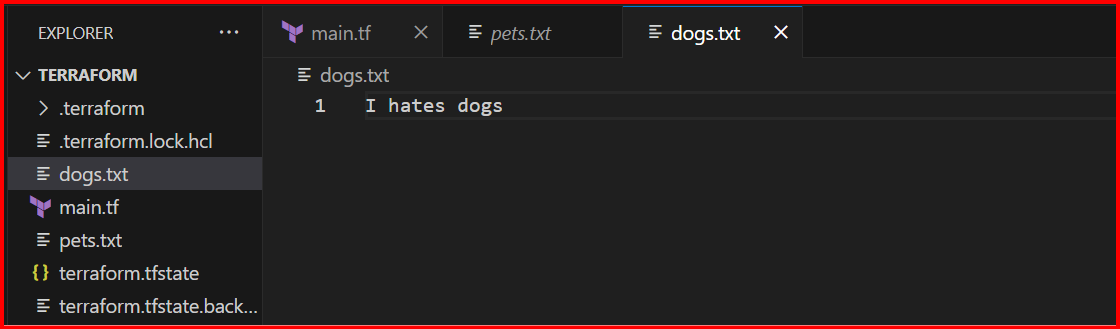
****

**Data sources:**

**Data sources are used to read the content of the infrastructure**

**Create txt file**

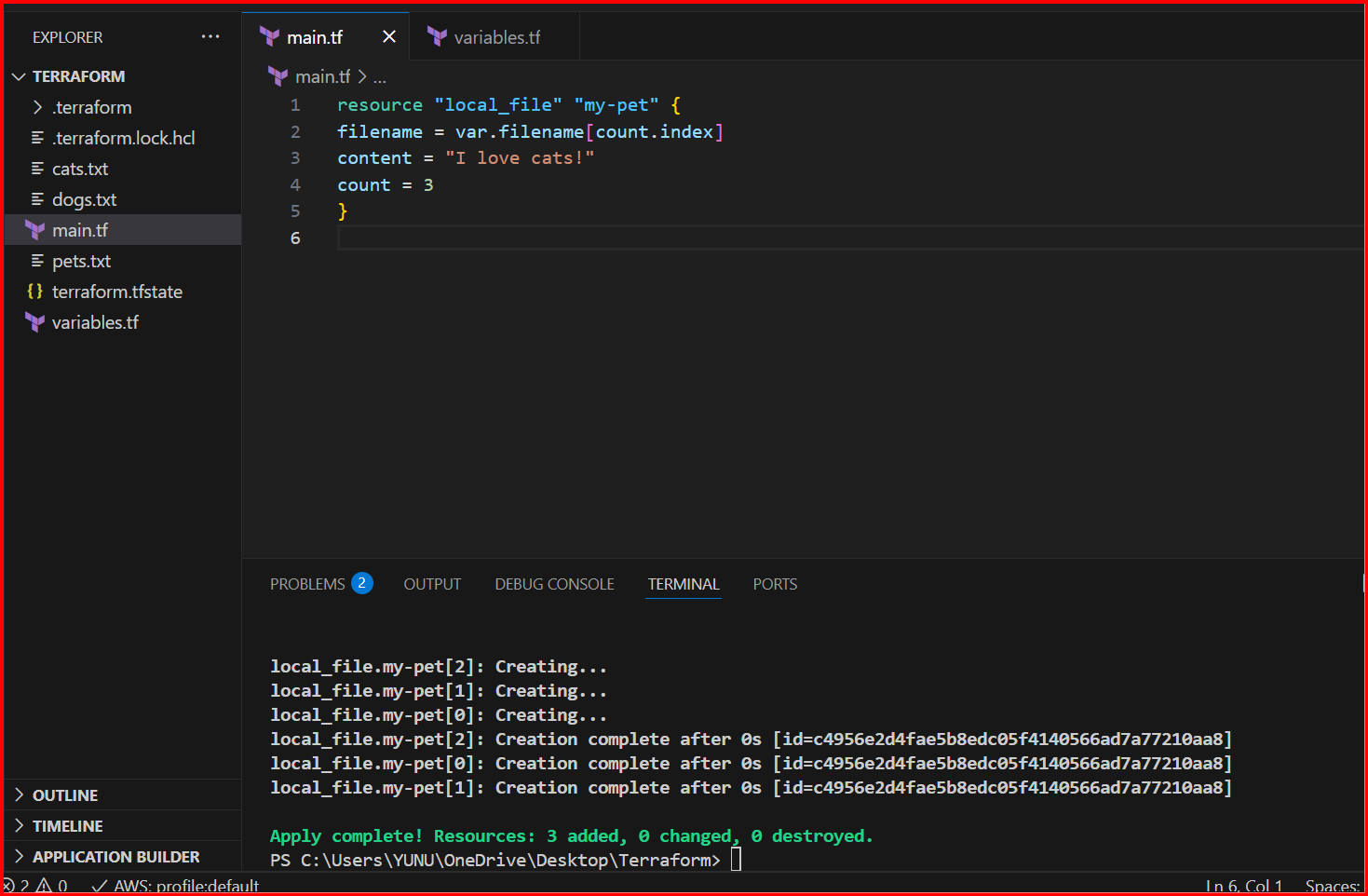
****

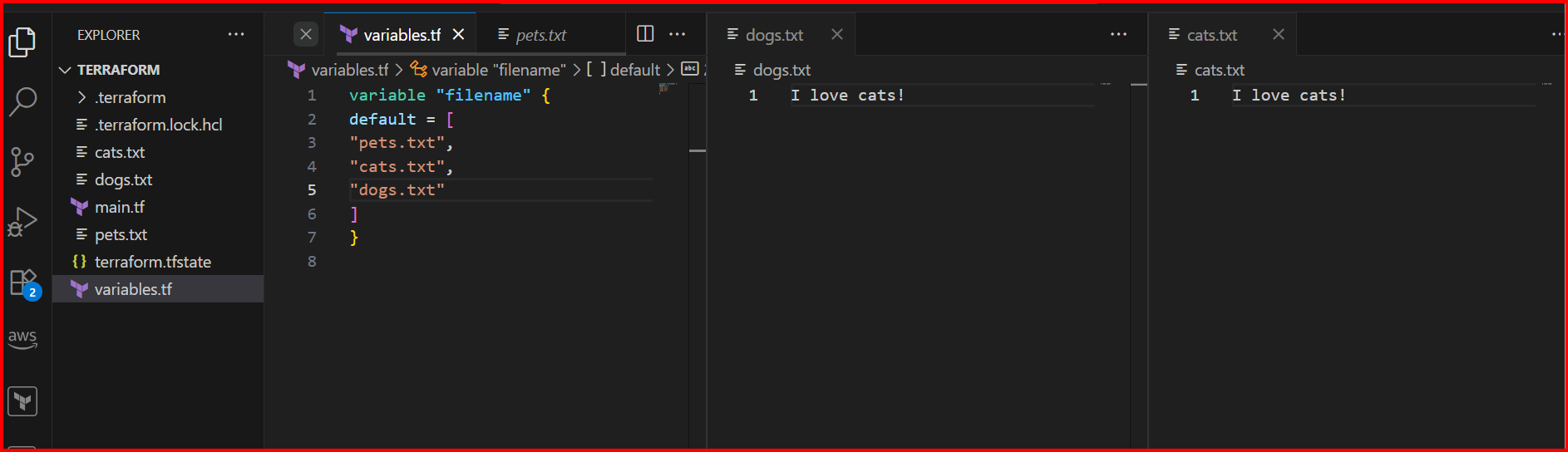
****

**Meta-Arguments:**

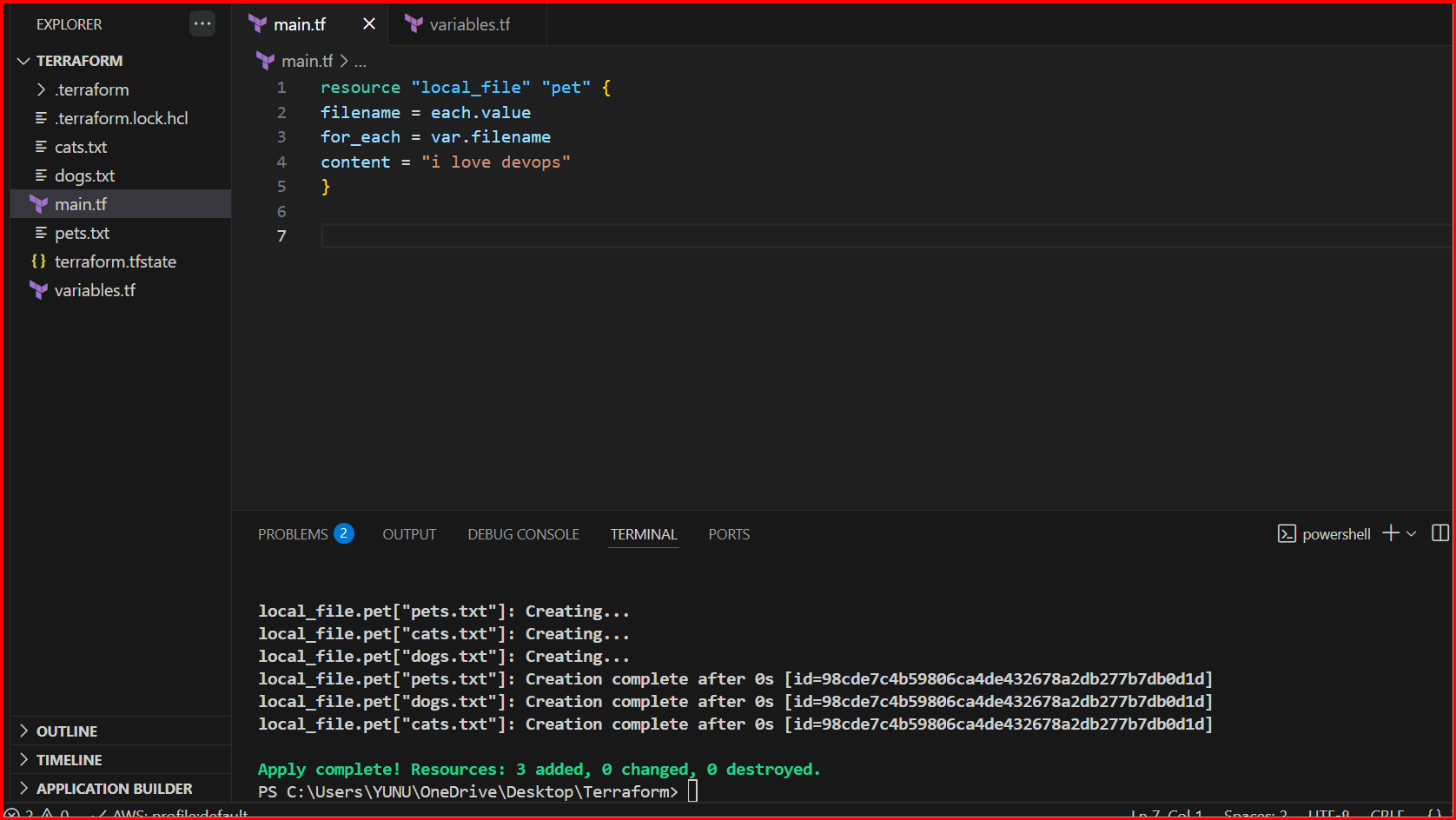
**Meta arguments are used if we want to create multiple resources.**

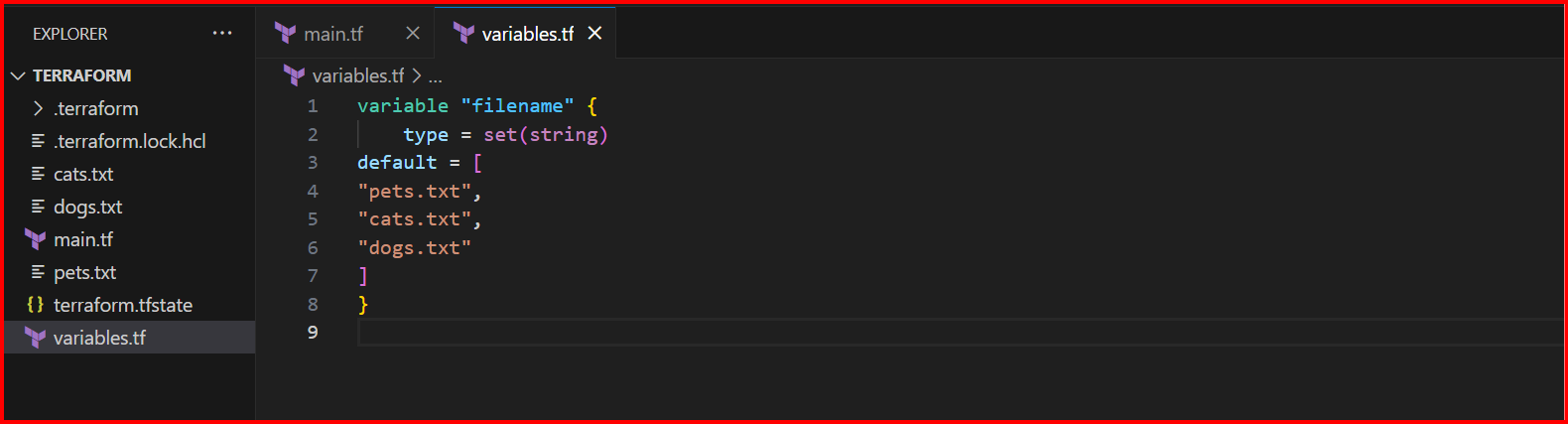
**Meta arguments can be used within any resource block to change the behaviour of the resources.**

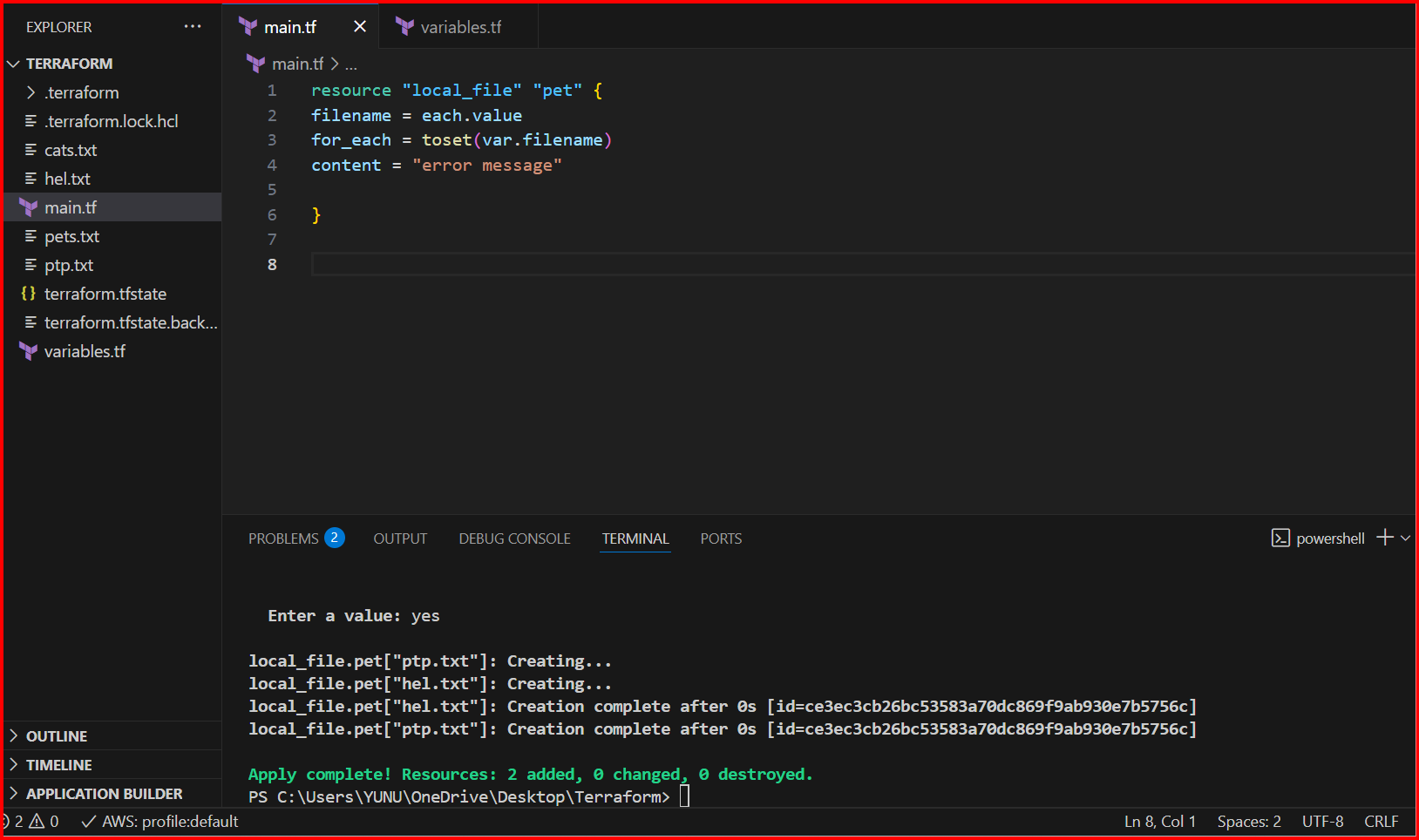
****

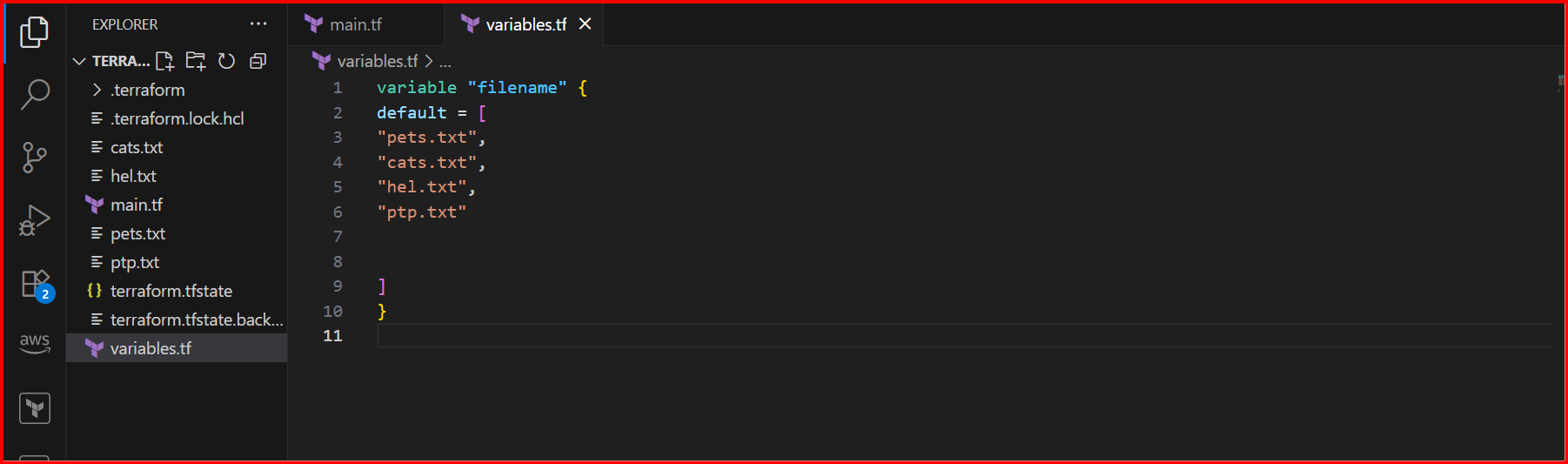
****

**For\_each:**

****

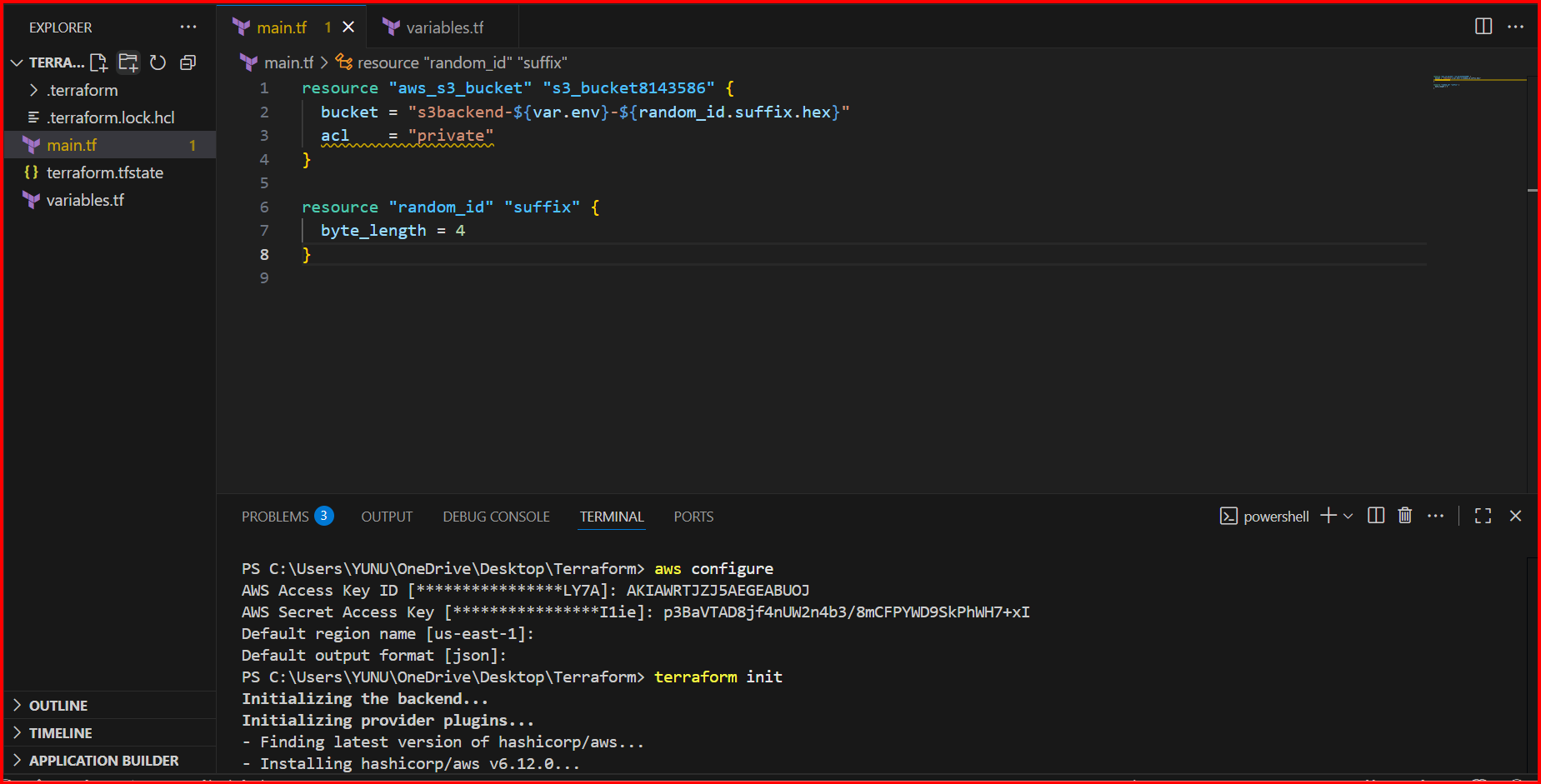
****

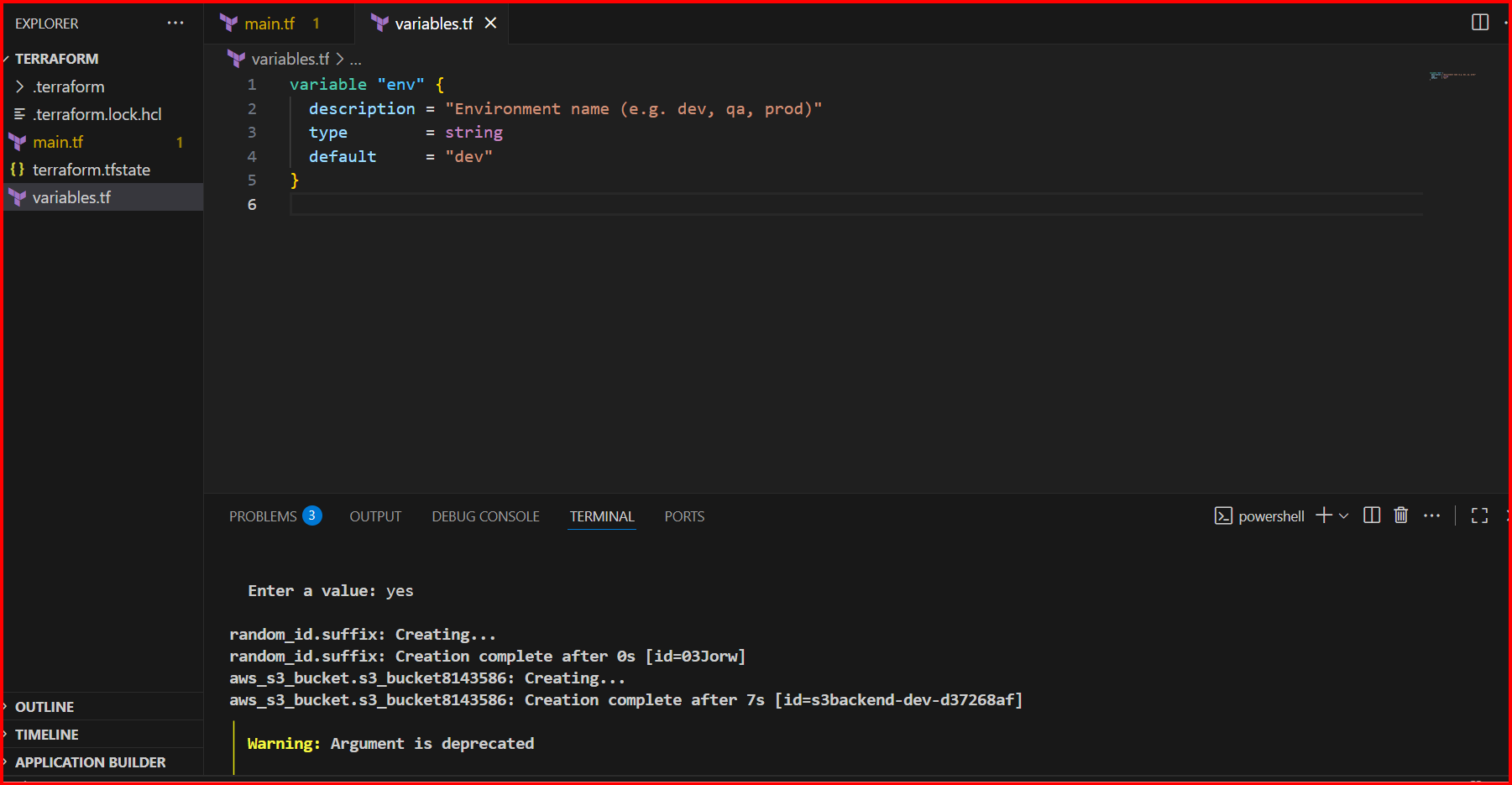
****

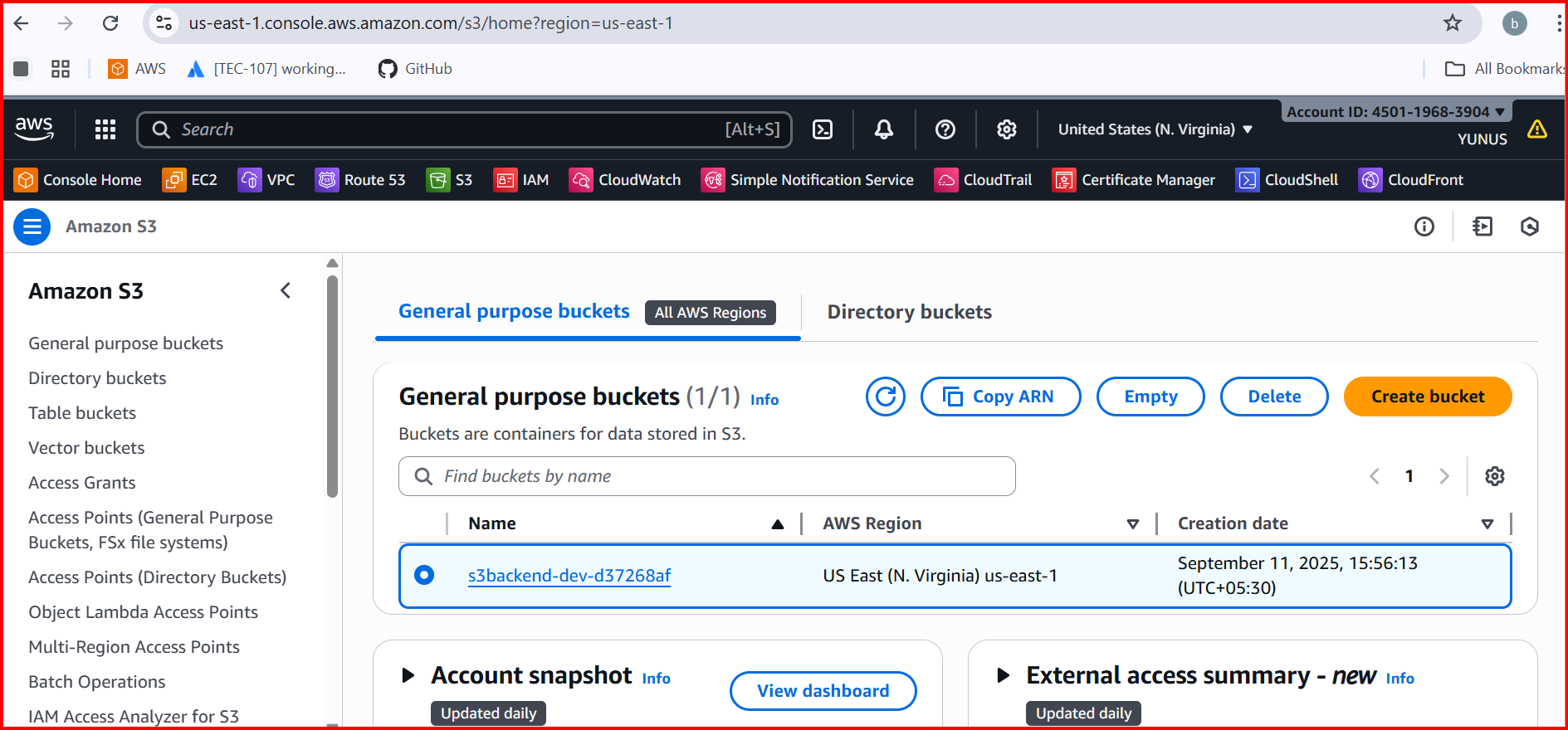
****

**AWS:**

**Created s3 bucket from terraform**

****

****

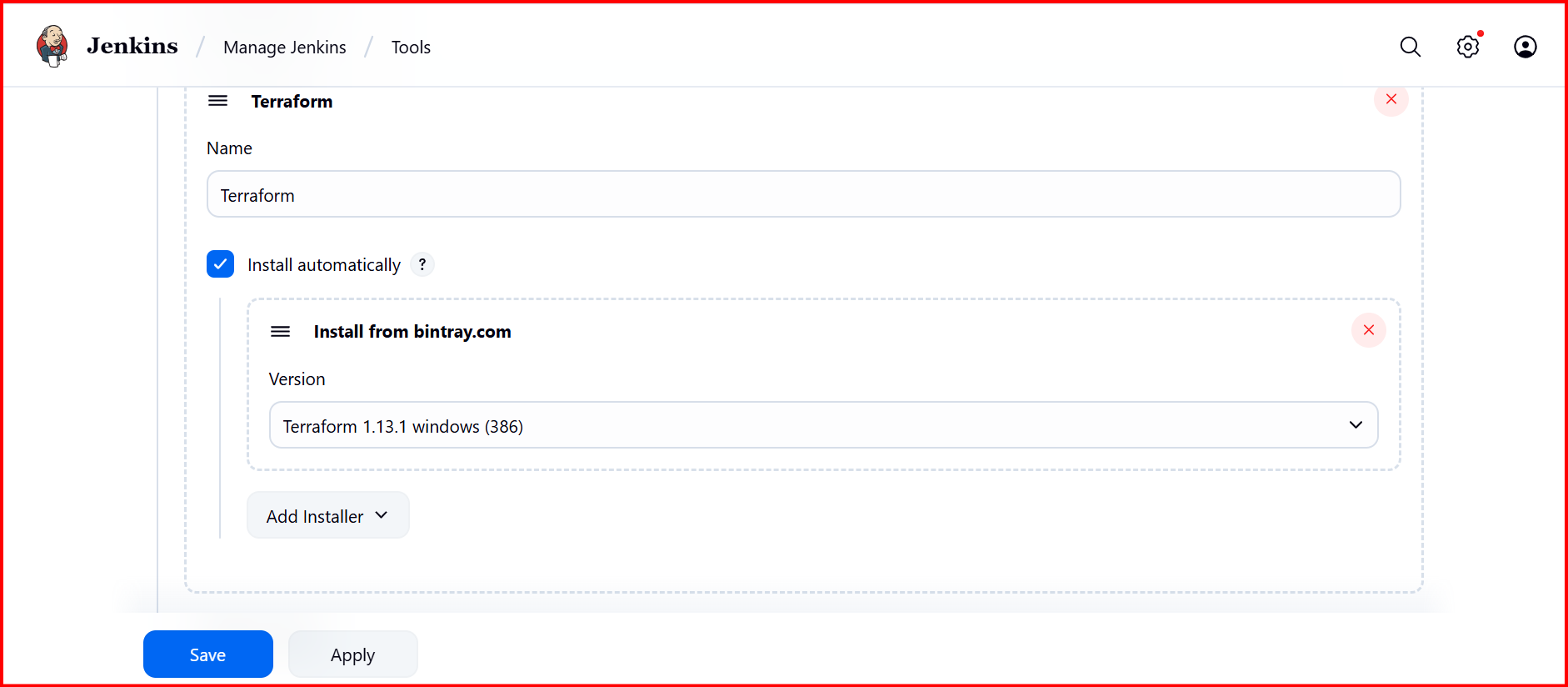
****

**3.Integrate Terraform in Jenkins using the Terraform plugin.**

**Installed Jenkins**

**Install Terraform plugin Jenkins**

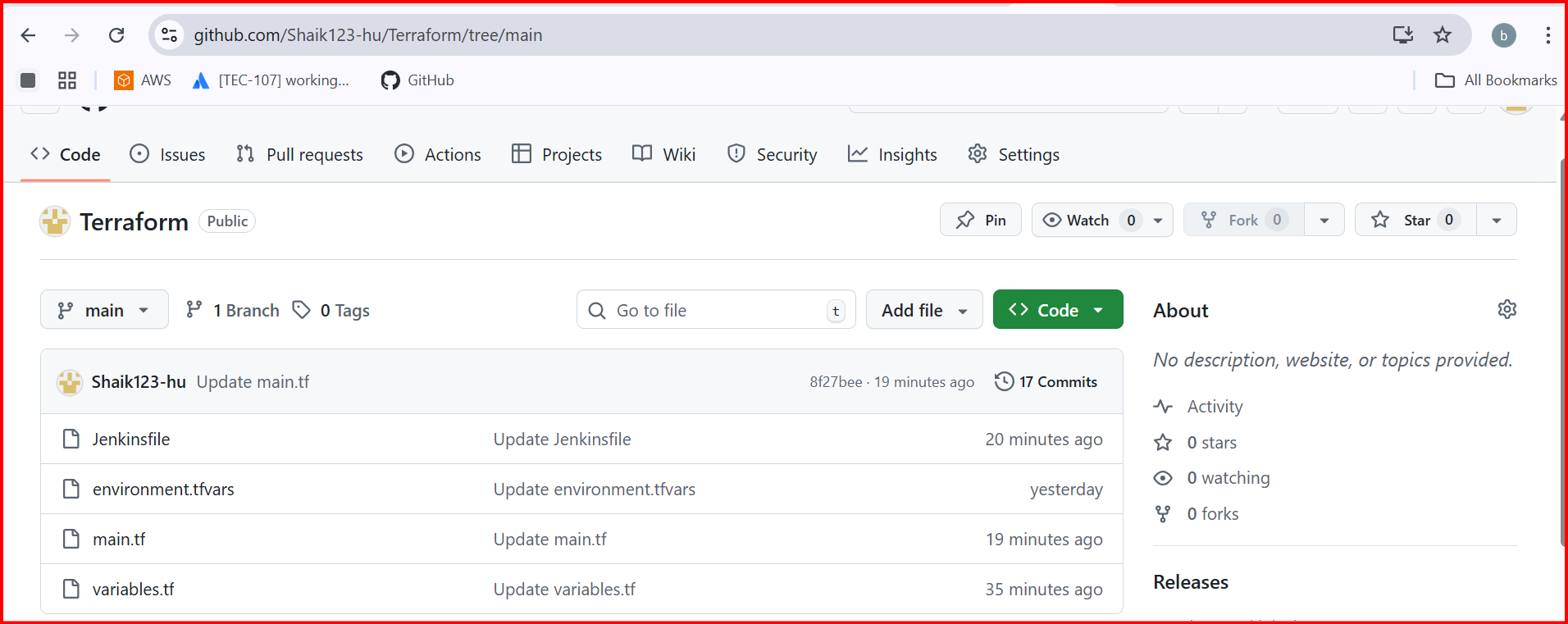
**Configure the terraform in Jenkins**

****

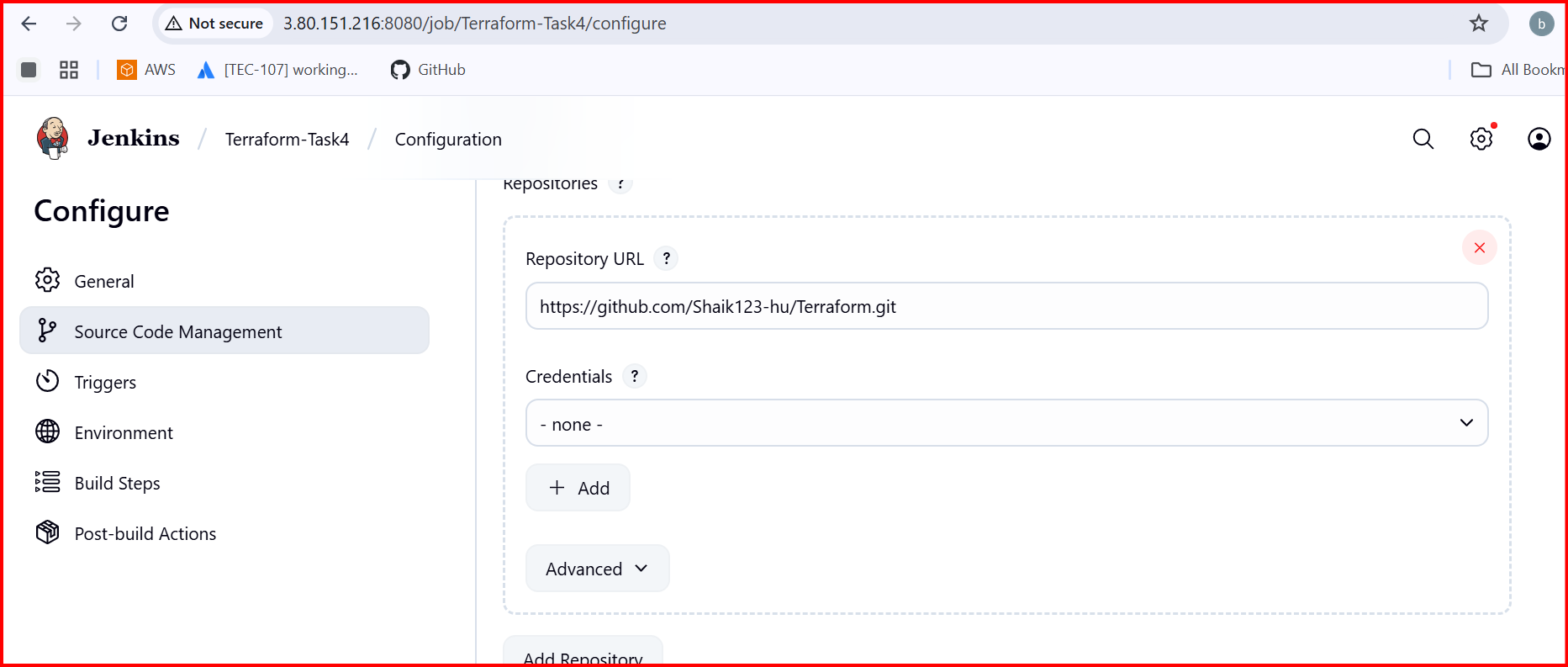
**GIT REPO:**

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

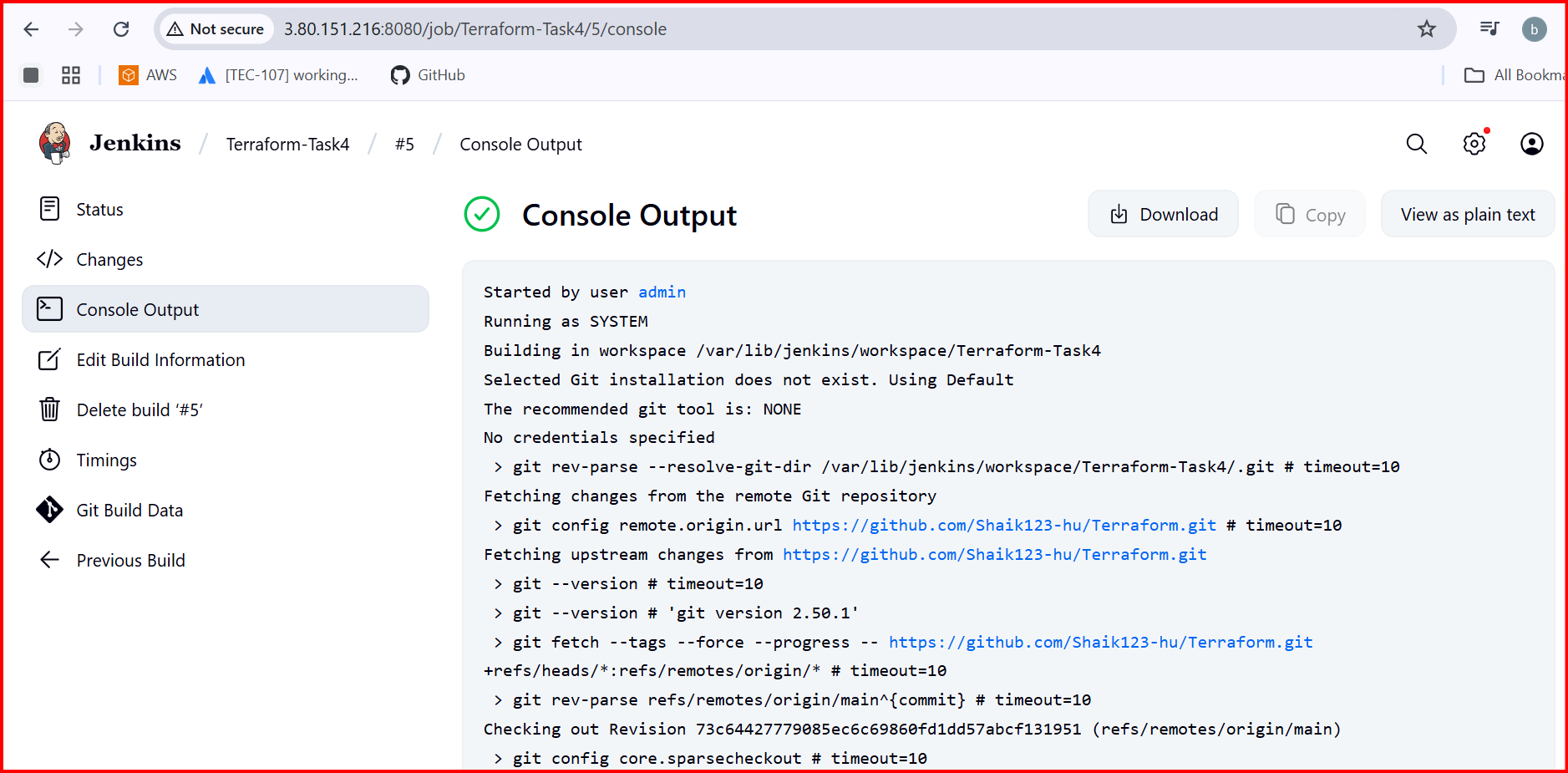
**update the main.tf variables.tf and Jenkins file**

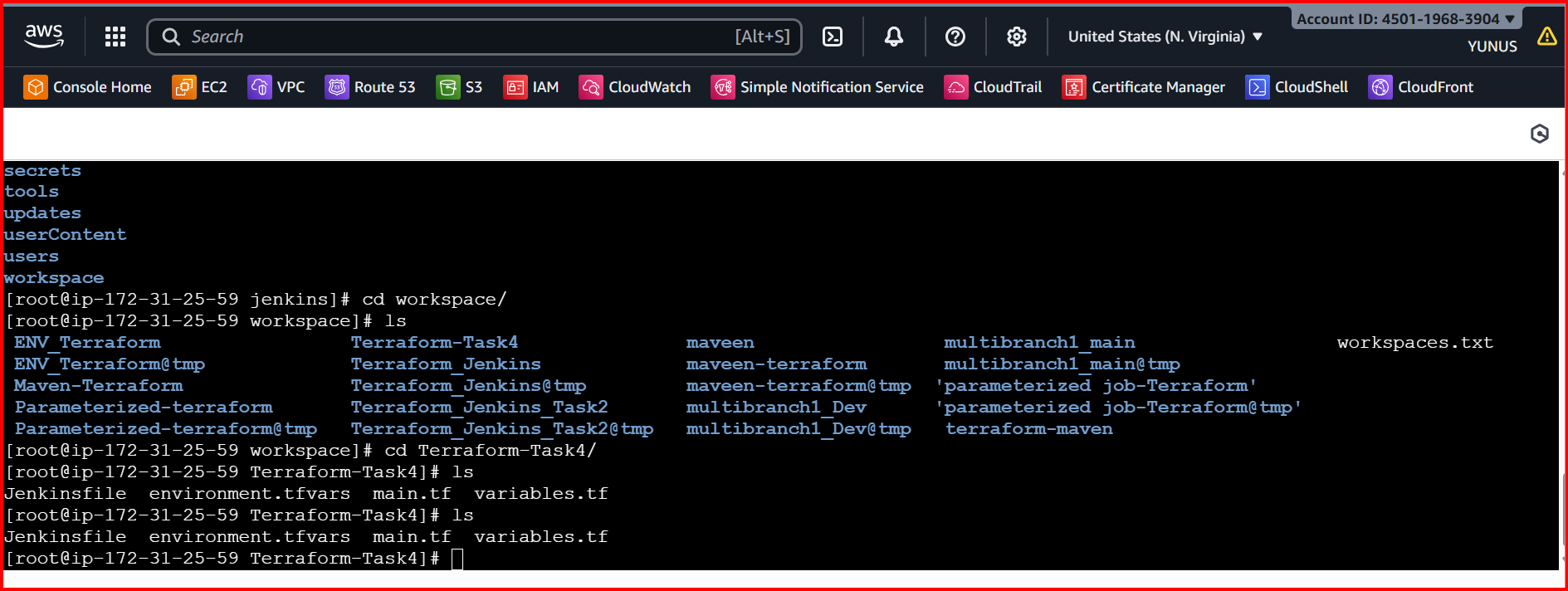
****

**In Jenkins create a pipeline and save**

****

**Once saved then buildnow**

****

****

**4.Create a CI/CD pipeline for a Nodejs Application:**[**https://github.com/betawins/Trading-UI.git**](https://github.com/betawins/Trading-UI.git)

**Install Nodejs**

**sudo yum update -y --> sudo yum install -y nodejs npm --> node -v --> npm -v**

**Fork above url**

**Install Nodejs Plugin in Jenkins**

**And configure the Nodejs**

****

**Go to git repo**

**Check the Jenkins file**

**pipeline {**

**agent any**

**tools {**

**nodejs "NodeJS14" // :point\_left: if need Use the new Node.js 20 installation**

**}**

**stages {**

**stage('Checkout') {**

**steps {**

**git url: 'https://github.com/Shaik123-hu/Trading-UI.git', branch: 'master'**

**}**

**}**

**stage('Install') {**

**steps {**

**sh '''**

**npm cache clean --force**

**rm -rf node\_modules package-lock.json**

**npm install --omit=optional**

**'''**

**}**

**}**

**stage('Test') {**

**steps {**

**sh 'npm test || echo ":warning: No tests found"'**

**}**

**}**

**stage('Build') {**

**steps {**

**withEnv(["CI=false"]) {**

**sh 'npm run build'**

**}**

**}**

**}**

**}**

**post {**

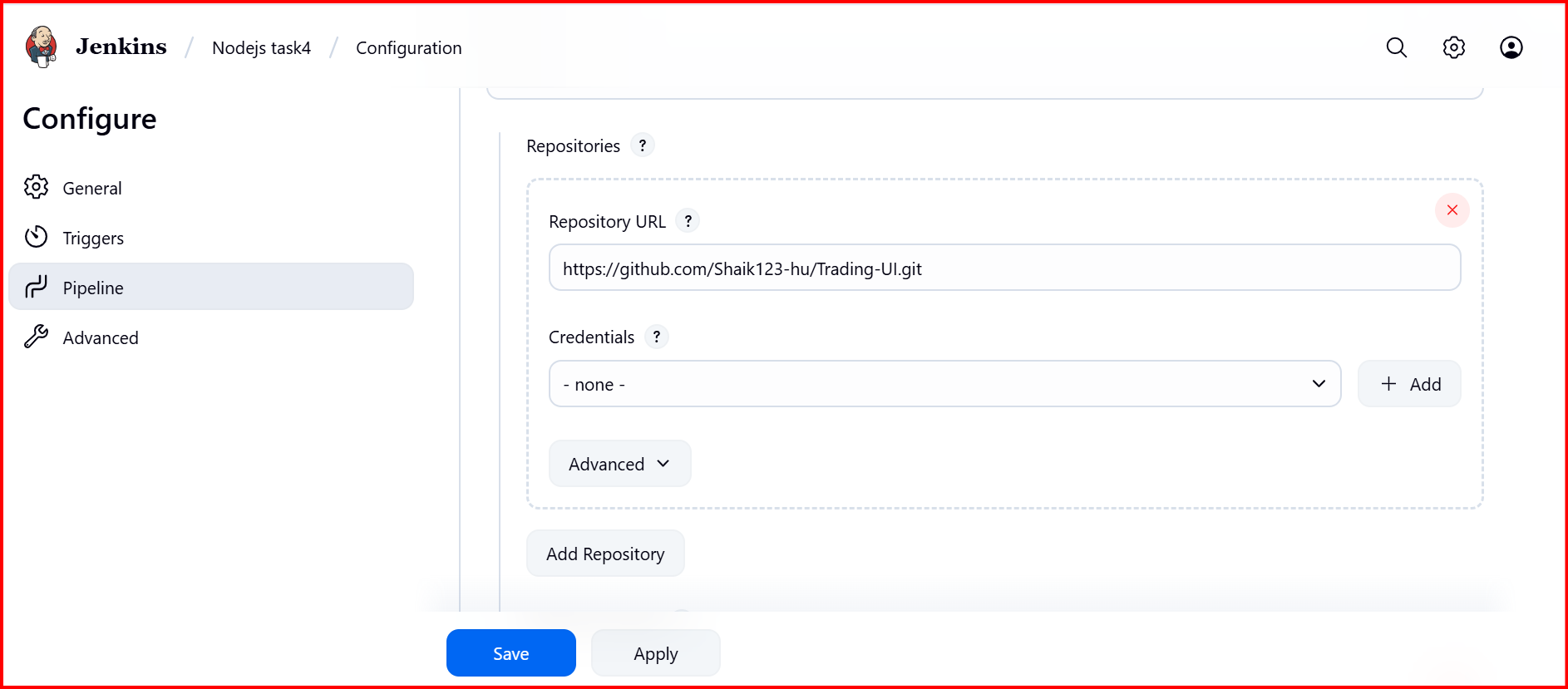
**success { echo ':white\_check\_mark: Node.js pipeline finished successfully.' }**

**failure { echo ':x: Node.js pipeline failed — check console output.' }**

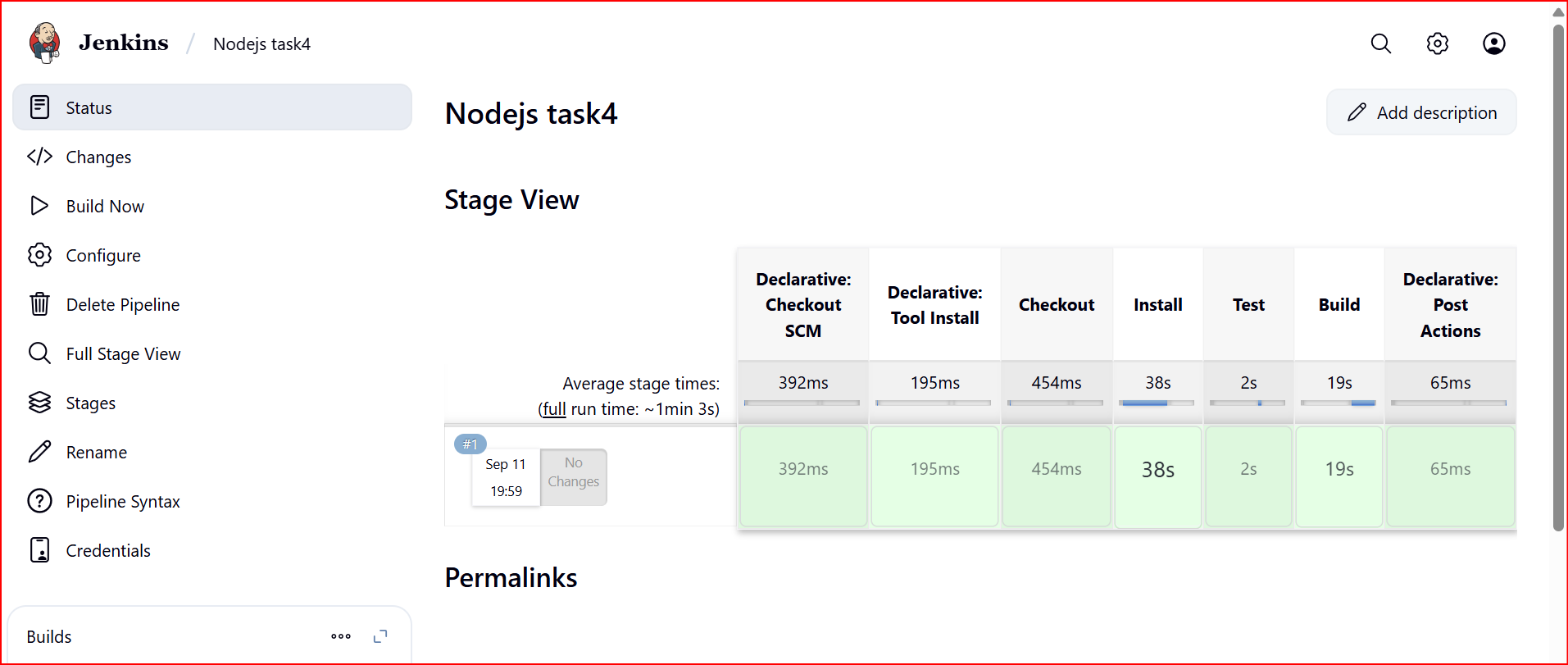
**}**

**}**

**Create job in jenkins**

****

**Save and build the job**

****

**5.Explain 10 Maven commands**

| **Command** | **Description** | **Example Usage** |
| --- | --- | --- |
| **mvn -version** | **Shows Maven and Java version installed.** | **mvn -version** |
| **mvn clean** | **Deletes target/ directory (removes old builds).** | **mvn clean** |
| **mvn compile** | **Compiles source code in src/main/java → target/classes.** | **mvn compile** |
| **mvn test** | **Runs unit tests in src/test/java.** | **mvn test** |
| **mvn package** | **Packages compiled code into JAR/WAR (defined in pom.xml).** | **mvn package** |
| **mvn install** | **Installs the package into local Maven repository (~/.m2/repository).** | **mvn install** |
| **mvn deploy** | **Deploys the package to a remote repository (e.g., Nexus, Artifactory).** | **mvn deploy** |
| **mvn site** | **Generates project documentation (reports, dependencies).** | **mvn site** |
| **mvn dependency:tree** | **Displays dependency hierarchy of the proj** |  |