**1) Install Terraform on your PC**

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

**3) Note down below points,**

**Terraform Init**

**Terraform Plan**

**Terraform Apply**

**Terraform Provider**

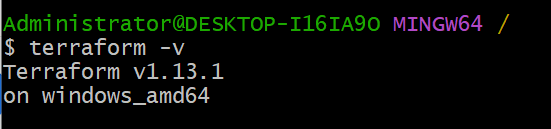
**4) Integrate a sample Terraform template in Jenkins.**

**1) Install Terraform on your PC**

## **Windows — Manual download**

1. **Go to https://www.terraform.io/downloads and download the Windows 64‑bit ZIP for the latest stable Terraform.**
2. **Unzip the downloaded file. You get terraform.exe.**
3. **Move terraform.exe to a folder in your PATH, e.g. C:\terraform and add that folder to your system PATH (System Properties → Advanced → Environment Variables → Path → Edit).**
4. **Open a new PowerShell window and run:**

**terraform -v**

****

## **Linux (Ubuntu/Debian) — A simple approach**

**# Download and unzip (example for Terraform 1.x; change to latest link if needed)**

**wget https://releases.hashicorp.com/terraform/<VERSION>/terraform\_<VERSION>\_linux\_amd64.zip**

**unzip terraform\_<VERSION>\_linux\_amd64.zip**

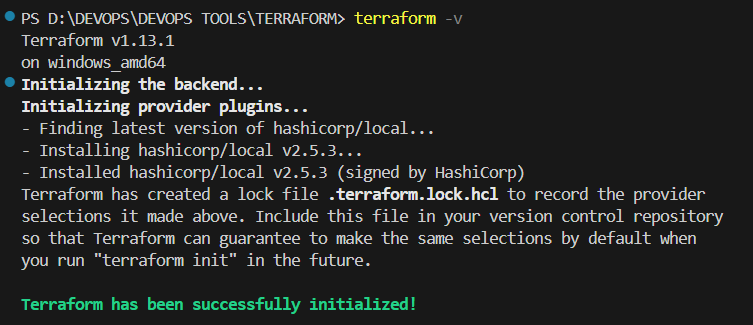
**sudo mv terraform /usr/local/bin/**

**terraform -v**

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

**Initializing terraform in VS.**

**In terminal terraform init**

****

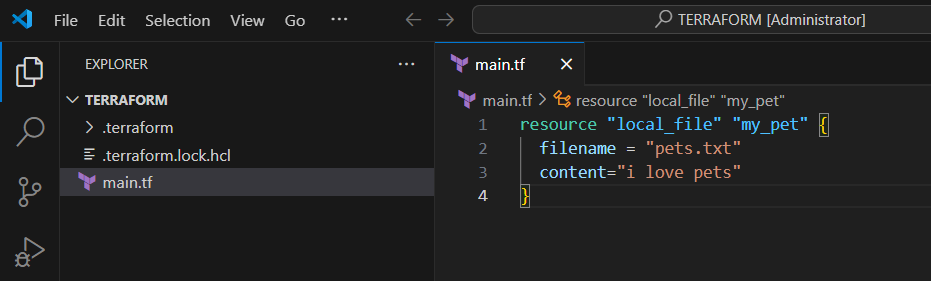
**Sample template.**

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

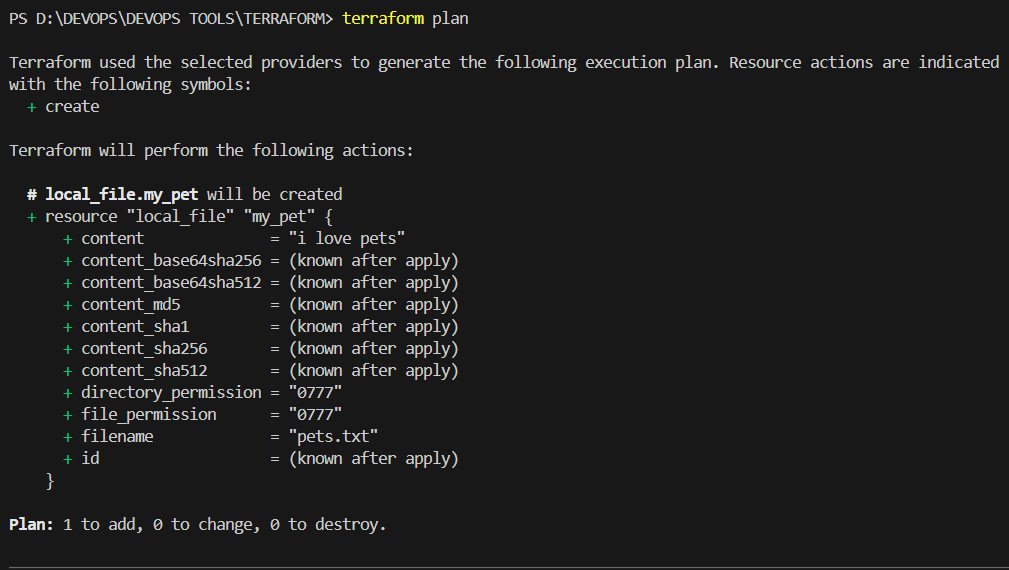
**filename = "pets.txt"**

**content="i love pets"**

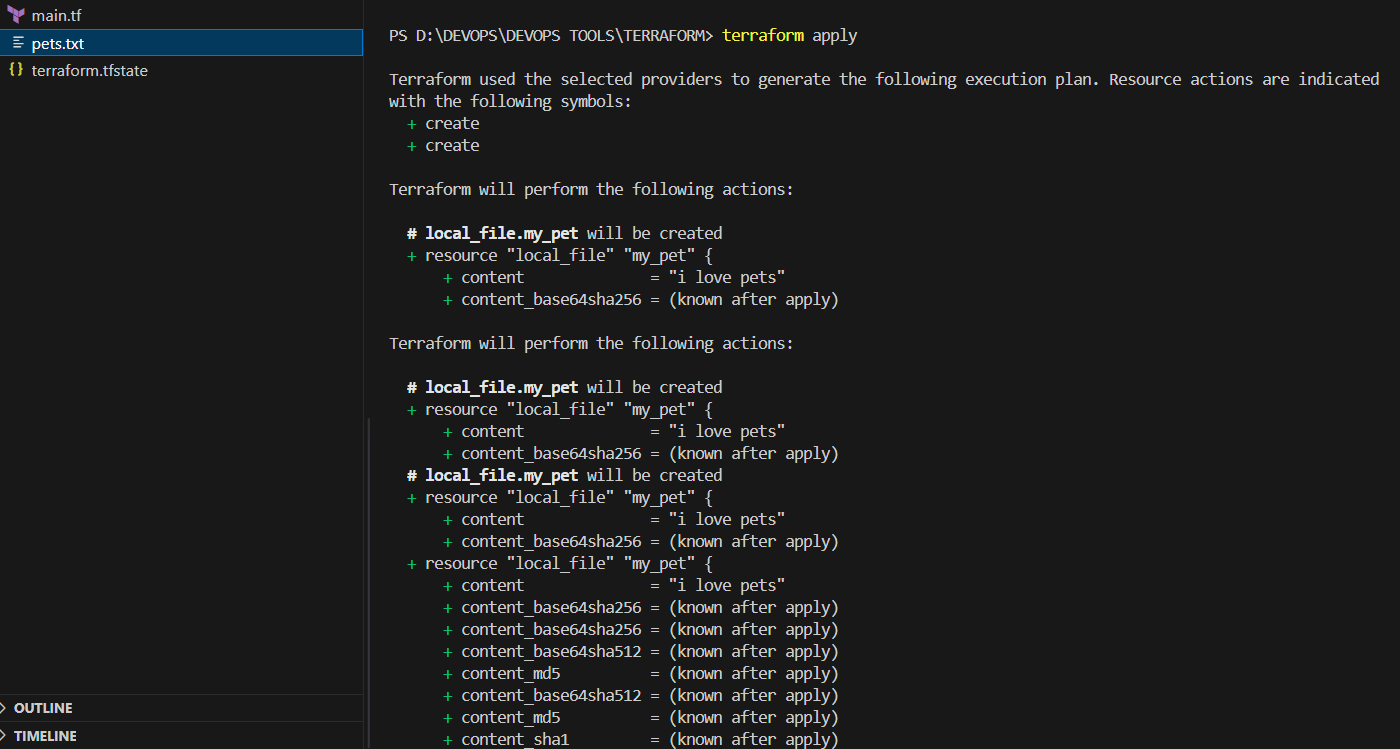
**}**

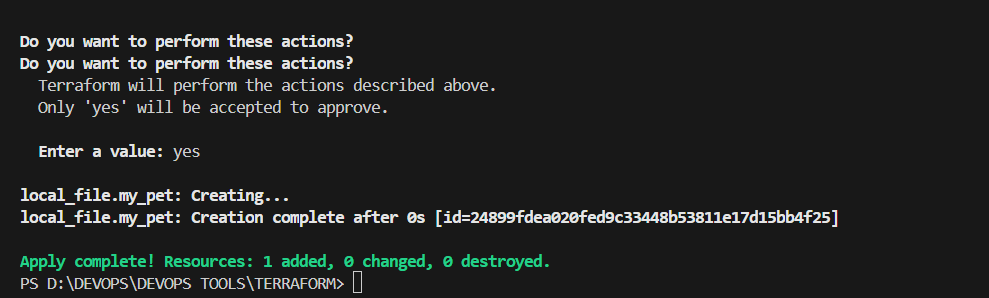
****

**Terraform PLAN**

****

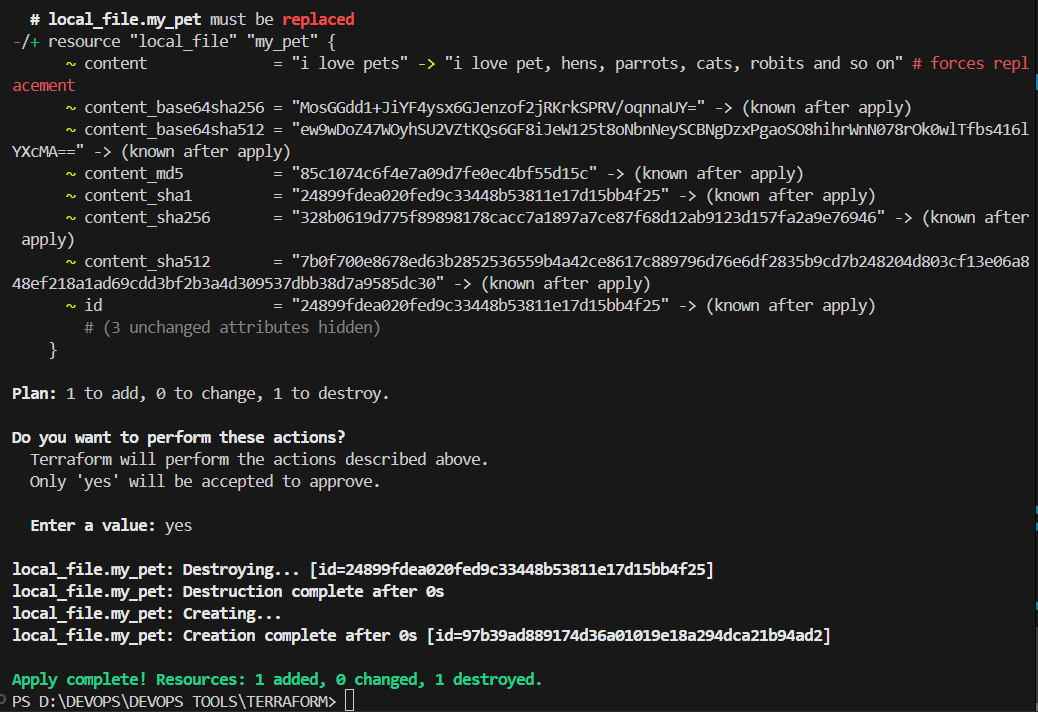
**Terraform Apply**

****

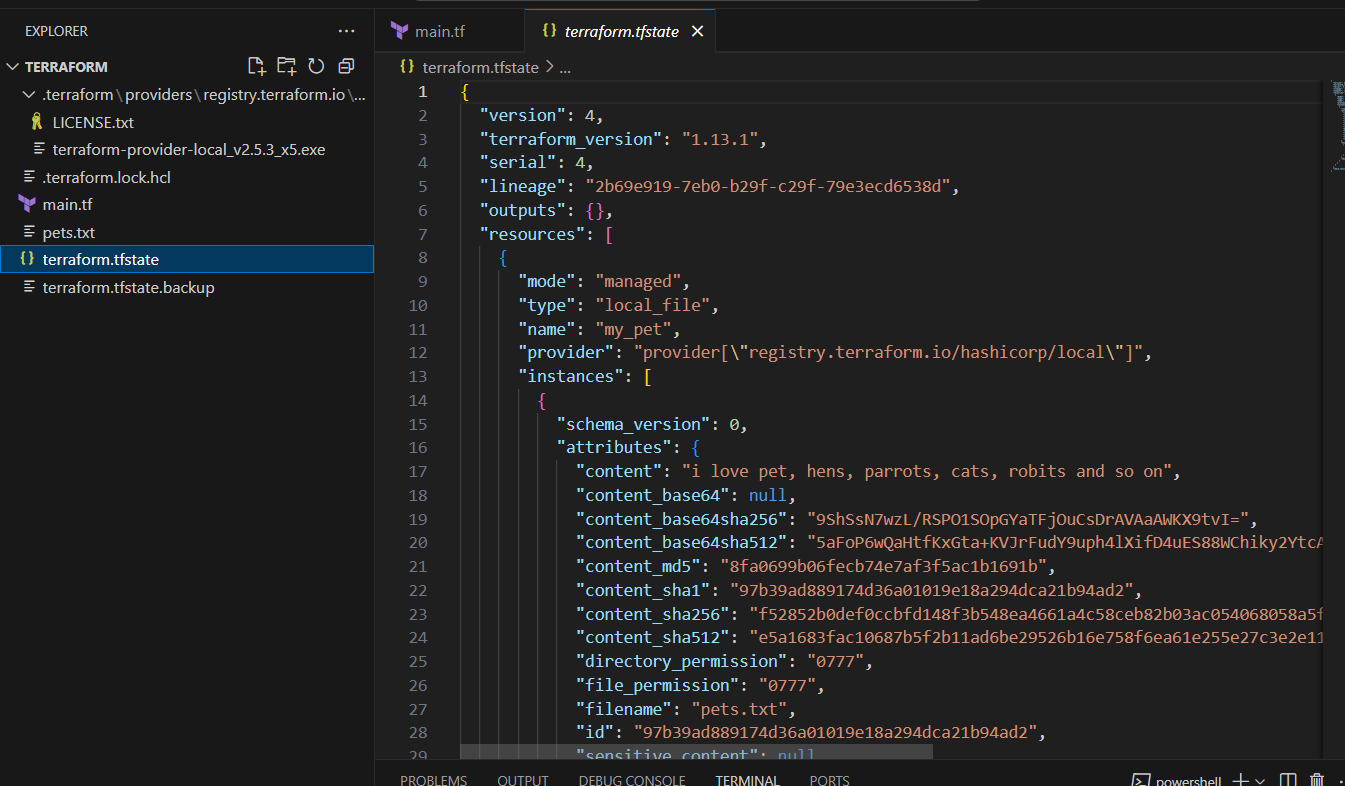
****

**Terraform plan → apply**

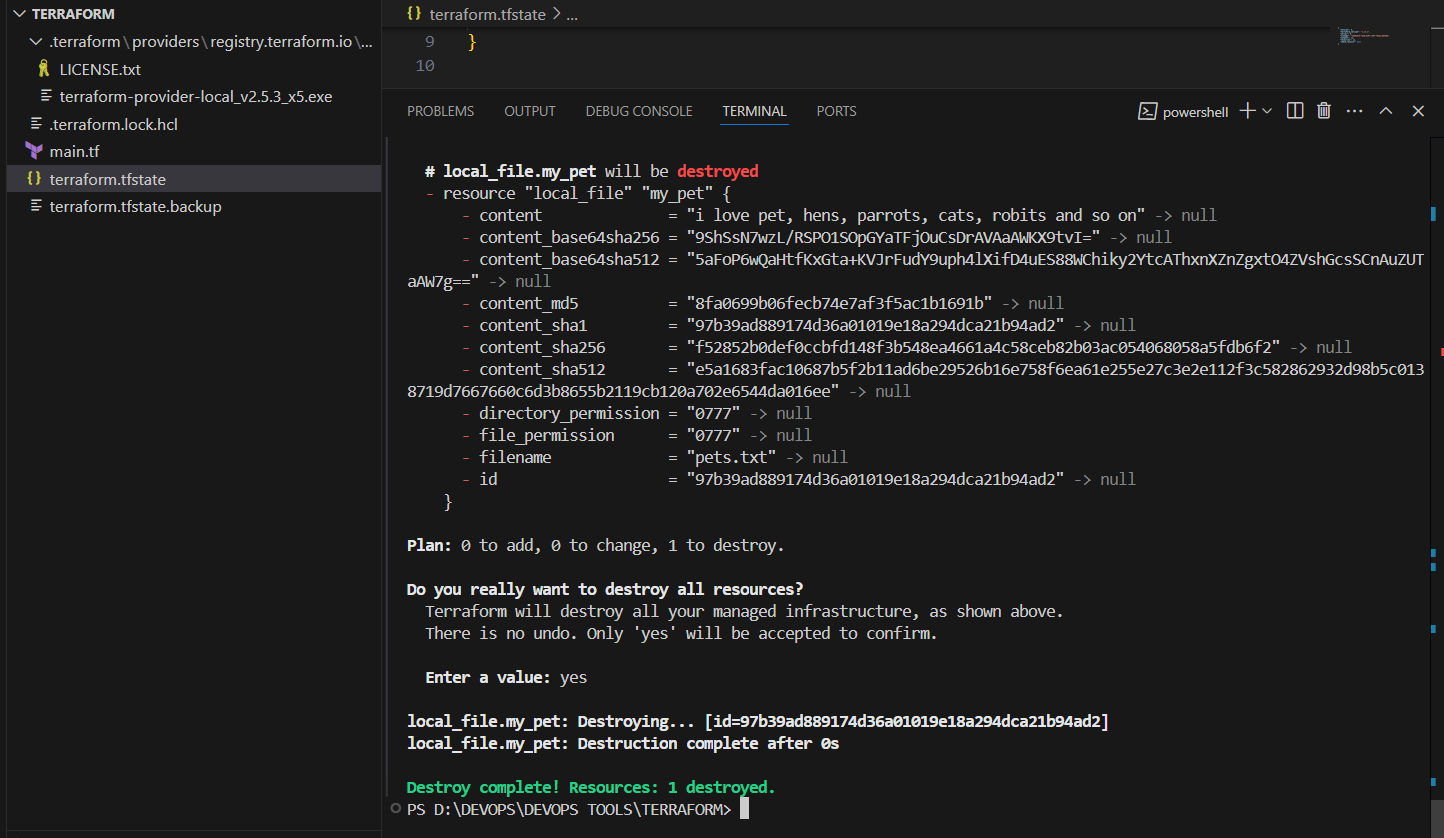
**Now modified the** [**main.tf**](http://main.tf)

****

**Terraform.tfstate containing all information.**

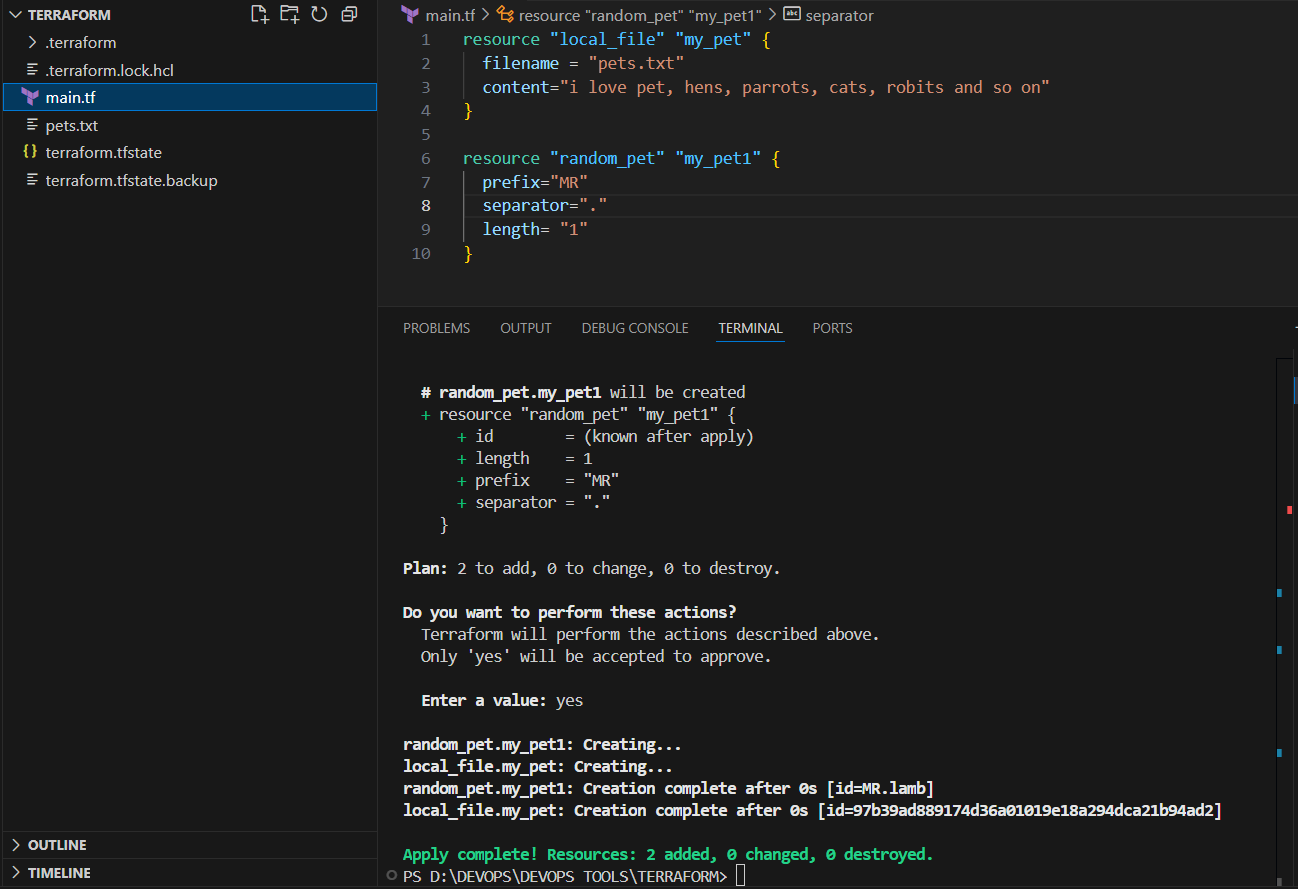
****

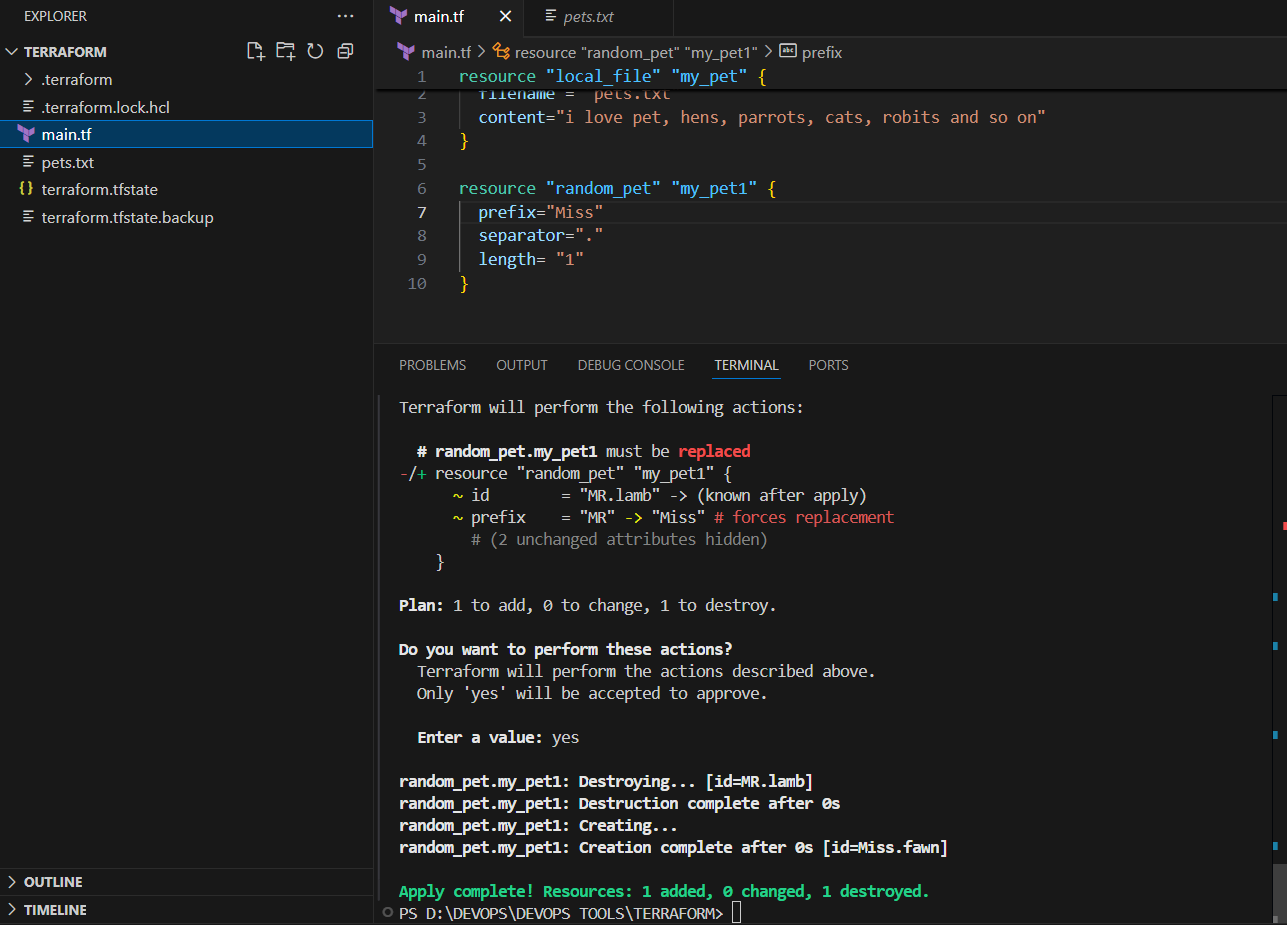
**Terraform Destroy**

****

**Created one more resource**

**Terraform init → plan → apply**

****

****

**Terraform fmt**

**Will format or organize your code.**

****

**3) Note down below points,**

**Terraform Init**

**Terraform Plan**

**Terraform Apply**

**Terraform Provider**

## **Terraform Init**

**What it does: Prepares the working directory for use with Terraform. Downloads provider plugins, configures backends, and sets up the local cache.**

**terraform init**

## **Terraform Plan**

**What it does: Builds an execution plan showing what resources Terraform will create, change or delete.**

**terraform plan -out=tfplan**

## **Terraform Apply**

**What it does: Applies the changes described by a plan to the provider (creates/updates/deletes cloud resources).**

**# Apply a saved plan**

**terraform apply tfplan**

**# or directly**

**terraform apply**

**# non-interactive**

**terraform apply -auto-approve**

## **Terraform Provider**

**What it is: A plugin that Terraform uses to manage a specific platform (e.g., hashicorp/aws, hashicorp/azurerm, hashicorp/google). Providers translate Terraform resources into API calls.**

**Key points to note in docs:**

* **Provider block example and required provider declaration.**
* **How credentials are provided (environment variables, shared credentials file, service principal, etc.).**
* **Provider version pinning using required\_providers.**

**Example (AWS):**

**terraform {**

**required\_providers {**

**aws = {**

**source = "hashicorp/aws"**

**version = "~> 5.0"**

**}**

**}**

**}**

**provider "aws" {**

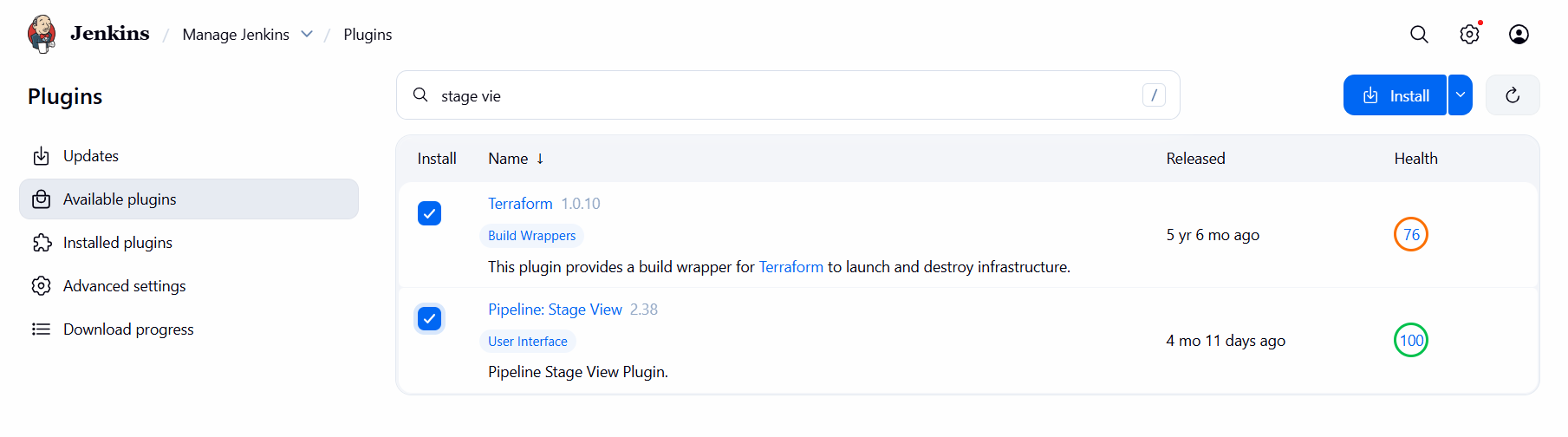
**region = "ap-south-1"**

**}**

**Auth methods (examples):**

* **AWS: AWS\_ACCESS\_KEY\_ID / AWS\_SECRET\_ACCESS\_KEY env vars, or ~/.aws/credentials profile.**
* **Azure: az login for local, or service principal credentials for automation.**
* **GCP: service account JSON key file referenced by GOOGLE\_APPLICATION\_CREDENTIALS.**

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

1. **Launched jenkins-ec2**
2. **Installed plugins terraform**
3. ****
4. **In Jenkins-ec2 installed terraform**

**# Remove old binary**

**sudo rm -f /usr/local/bin/terraform**

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

**wget https://releases.hashicorp.com/terraform/1.13.1/terraform\_1.13.1\_linux\_amd64.zip**

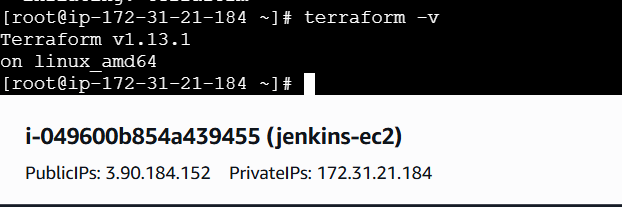
**# Unzip and move**

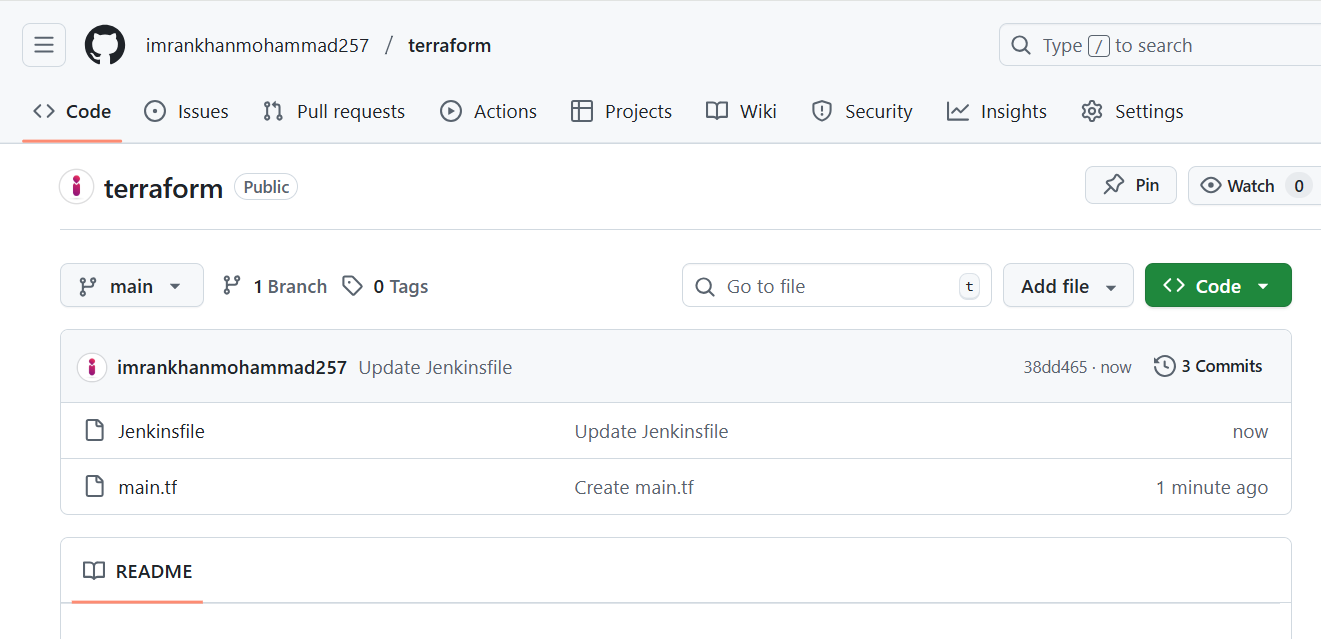
**unzip terraform\_1.13.1\_linux\_amd64.zip**

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

**# Verify**

**terraform -v**

****

1. **Now created a repo in github and added or created Jenkinsfile ,** [**main.tf**](http://main.tf)
2. ****
3. **Jenkinsfile having below declarative script**

**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/imrankhanmohammad257/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'**

**}**

**}**

**}**

**}**

1. [**main.tf**](http://main.tf) **have following code**

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

**filename = "pets.txt"**

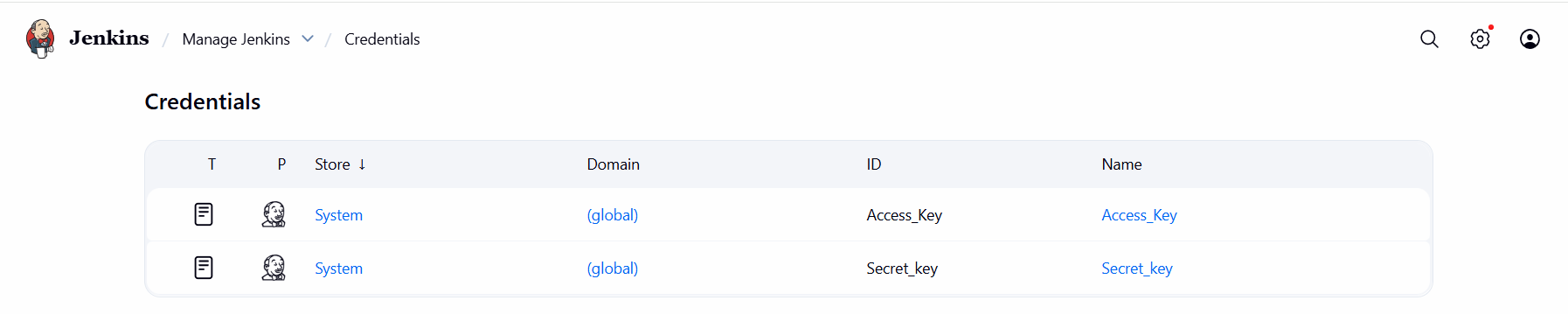
**content = "I Like PETS"**

**}**

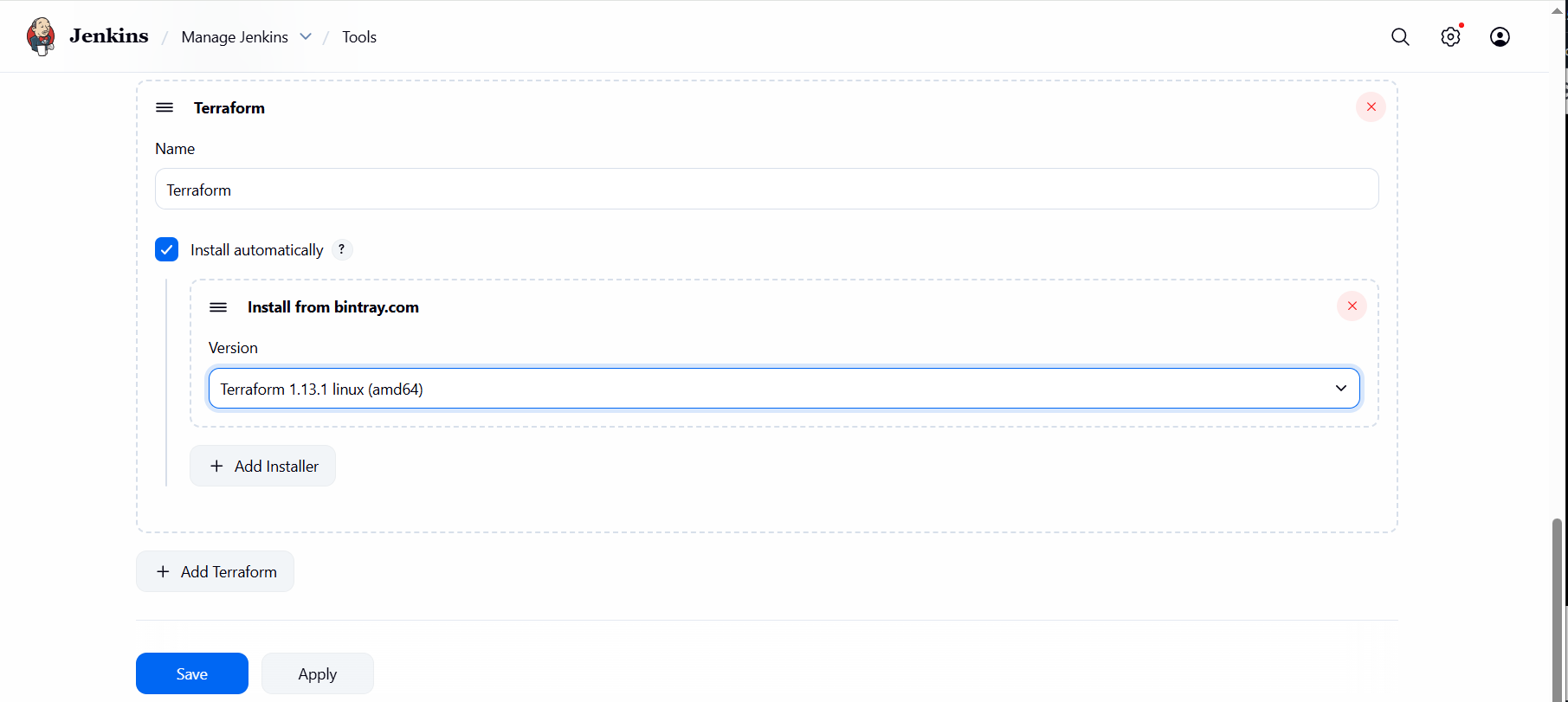
1. **Now create and add credentials of AWS in Jenkins Home page.**

**Create access keys, Access\_Key, Secret\_key**

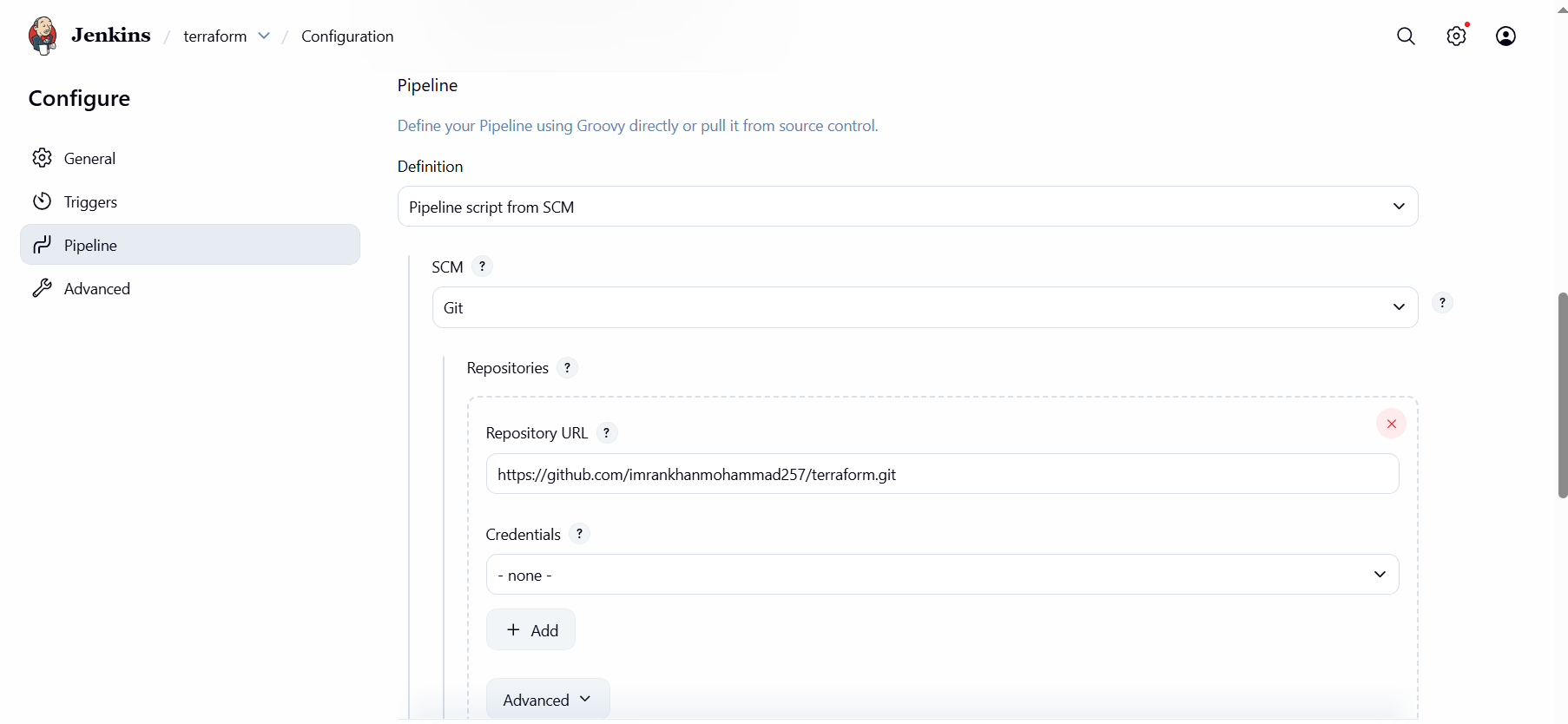
**$ cat ~/.aws/credentials (check credentials)**

****

1. **In Jenkins tools added terrform**

****

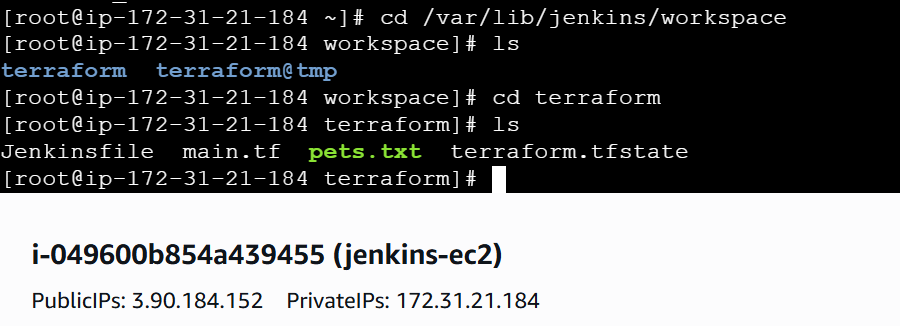
1. **In jenkins create a terraform job.**

****

1. **Build job**

****

**J. check in jenkins ec2 workspace.**

****