1. Watch terraform-03 video.

done

2) Execute the script shown in video.

**Environment variables**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Variable definition file**

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Precedency order**

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Resource attribute reference:**

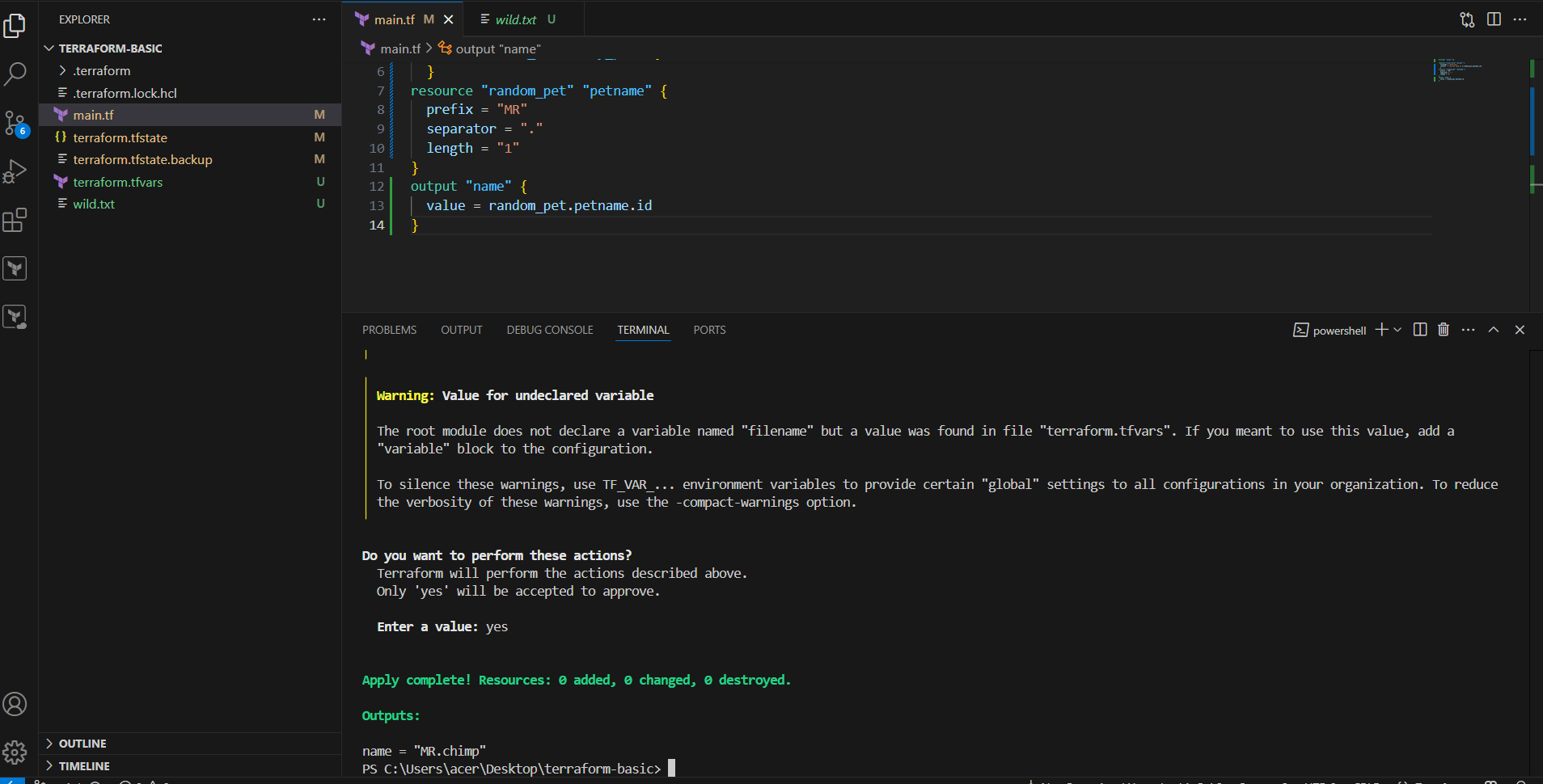
A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Output:**



3)Intergrate terrafrom in jenkins using Terraform plugin.

Already done this task in terraform-02 tasks

4) Create one jenkins job using MAVEN PROJECT for the below code with two stages.

stage 1: Git clone stage

2: Maven Compilation Code: <https://github.com/betawins/java-Working-app.git>

**Create a New Maven Project**

1. From the Jenkins **Dashboard**, click on **“New Item”**
2. Enter a name for the job ( JavaWorkingApp)
3. Select **“Maven Project”**
4. Click **“OK”**

A screenshot of a computer

AI-generated content may be incorrect.

**Select Git**

* In **Repository URL**, paste:

https://github.com/betawins/java-Working-app.git

A screenshot of a chat

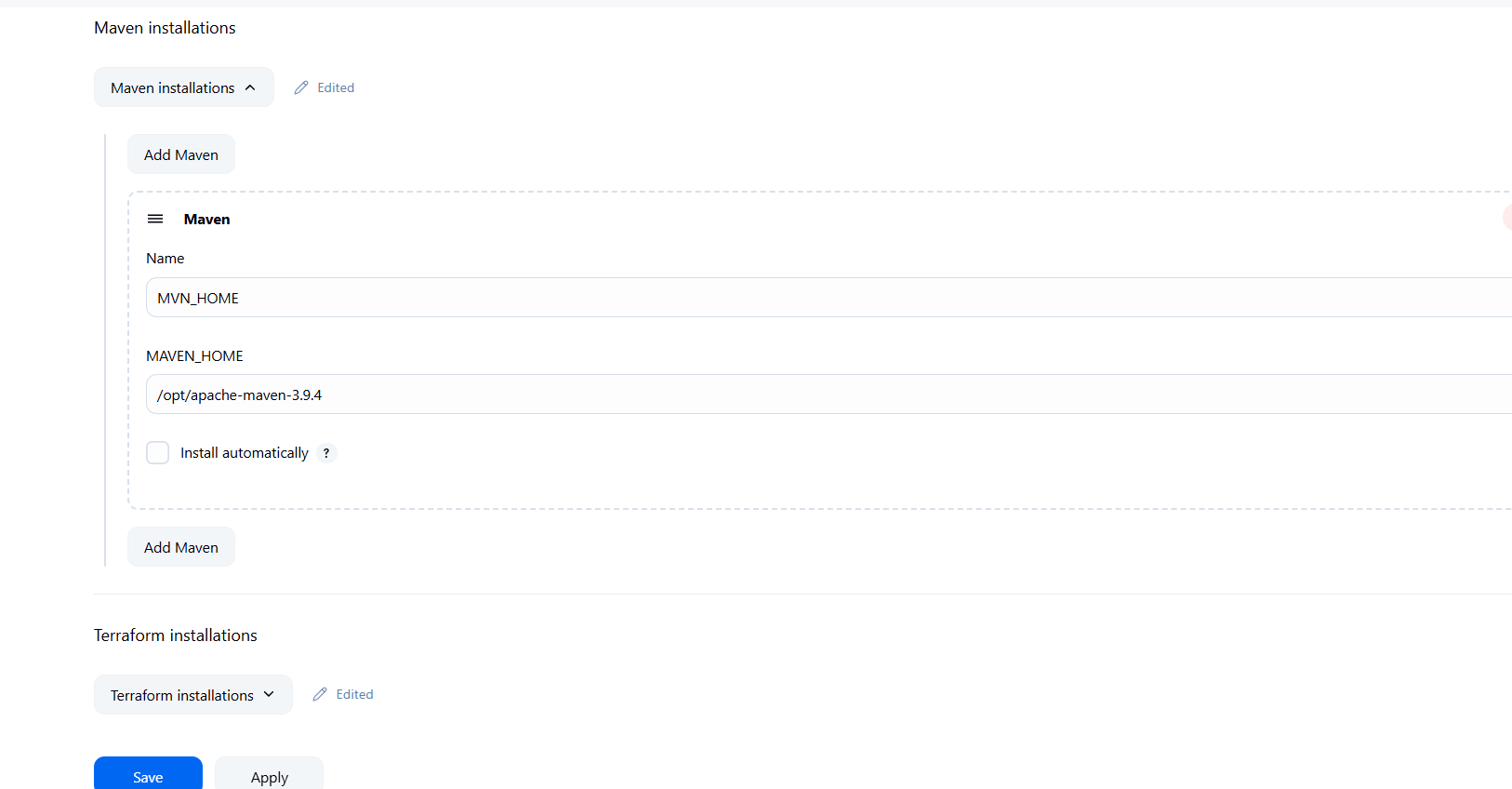
AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

Go to **Manage Jenkins > Global Tool Configuration**

Under **Maven**, click **Add Maven**



click **“Build Now”**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

5) Use the below code and create a parameterized job in jenkins

stage 1: Git clone stage

2: Maven Compilation Code: <https://github.com/betawins/java-Working-app.git>

Go to your **Jenkins Dashboard**

Click **“New Item”**

Enter the name: JavaParameterizedBuild

Select **“Freestyle project”**

Click **OK**

A screenshot of a computer

AI-generated content may be incorrect.

Scroll to **“This project is parameterized”**

Check the box ✅

Click **“Add Parameter”** → choose:

* **String Parameter**
  + Name: BRANCH\_NAME
  + Default value: main
  + Description: Git branch to build
* **String Parameter**
  + Name: MAVEN\_GOALS
  + Default value: clean install
  + Description: Maven build command

A screenshot of a computer

AI-generated content may be incorrect.

**String Parameter**

* Name: MAVEN\_GOALS
* Default value: clean install
* Description: Maven build command

A screenshot of a computer

AI-generated content may be incorrect.

Under **Source Code Management** → choose **Git**

In **Repository URL**: https://github.com/betawins/java-Working-app.git

A screenshot of a chat

AI-generated content may be incorrect.

Scroll to **Build**

Click **Add Build Step** → choose **Invoke top-level Maven targets**

In **Goals** field, use:

${MAVEN\_GOALS}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

6) What are the global varaiables in jenkins?

**Types of Global Variables**

1. **Built-in Variables**  
   Automatically provided by Jenkins:
   * BUILD\_NUMBER: Current build number
   * JOB\_NAME: Name of the Jenkins job
   * WORKSPACE: Path to job workspace
   * GIT\_BRANCH: Git branch being built
   * GIT\_COMMIT: Git commit hash
2. **User-defined Global Variables**  
   Defined in:  
   ➤ *Manage Jenkins → Configure System → Global Properties*  
   Example:
   * ENV = dev
   * REGION = us-east-1
3. **Pipeline-Specific Variables**  
   Used in Jenkins Pipelines:
   * env: Access to all environment variables
   * params: Access to build parameters
   * currentBuild: Info about the current build
4. **Parameter Variables**  
   Created using “This project is parameterized”:
   * params.BRANCH: Branch name from user input
   * params.VERSION: Version tag input