# *Terraform* 03&04

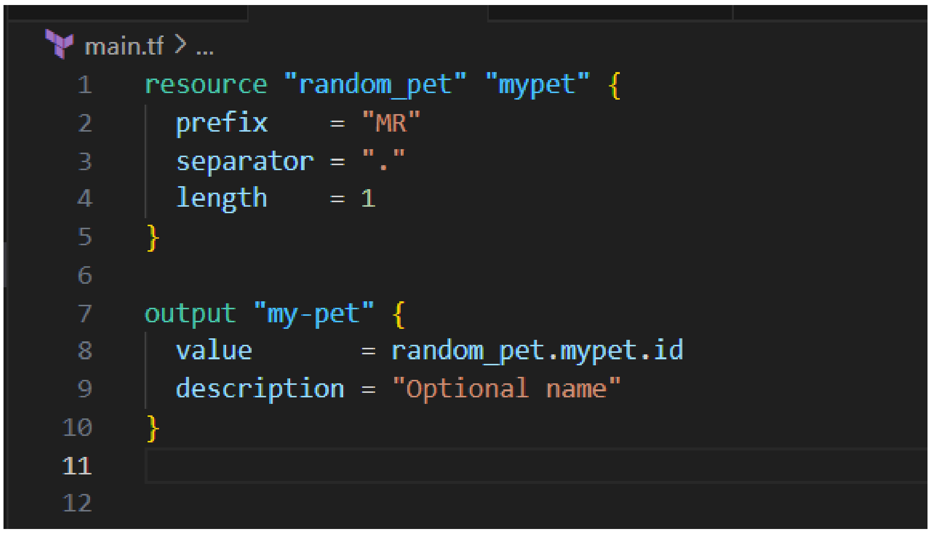
1) **Watch terraform-03 video.**

   **--completed**

2) **Execute the script shown in video.**

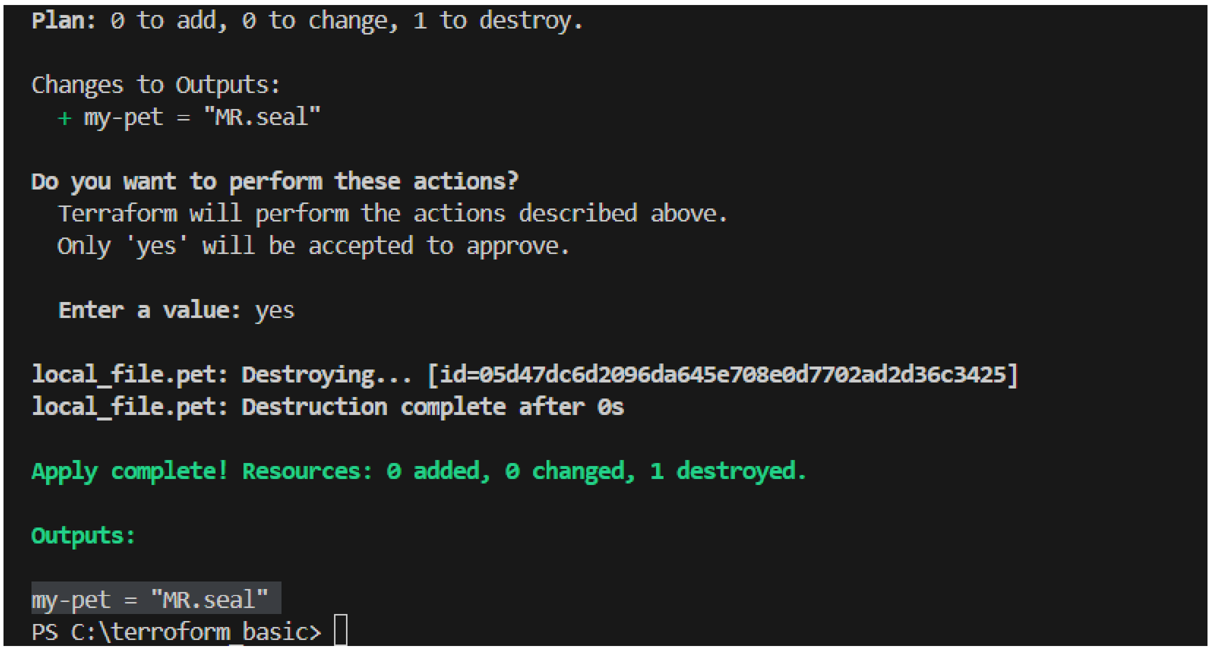
   **Output variables:**

   **--template**



```
main.tf > ...
resource "random_pet" "mypet" {
  prefix    = "MR"
  separator = "."
  length    = 1
}

output "my-pet" {
  value       = random_pet.mypet.id
  description = "Optional name"
}


```

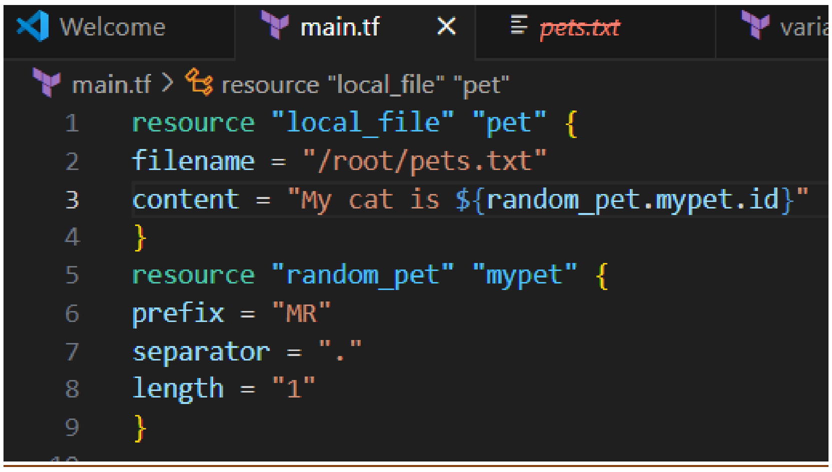
**--execution**



```
Plan: 0 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + my-pet = "MR.seal"

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.pet: Destroying... [id=05d47dc6d2096da645e708e0d7702ad2d36c3425]
local_file.pet: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 1 destroyed.

Outputs:

my-pet = "MR.seal"
PS C:\terroform_basic>
```

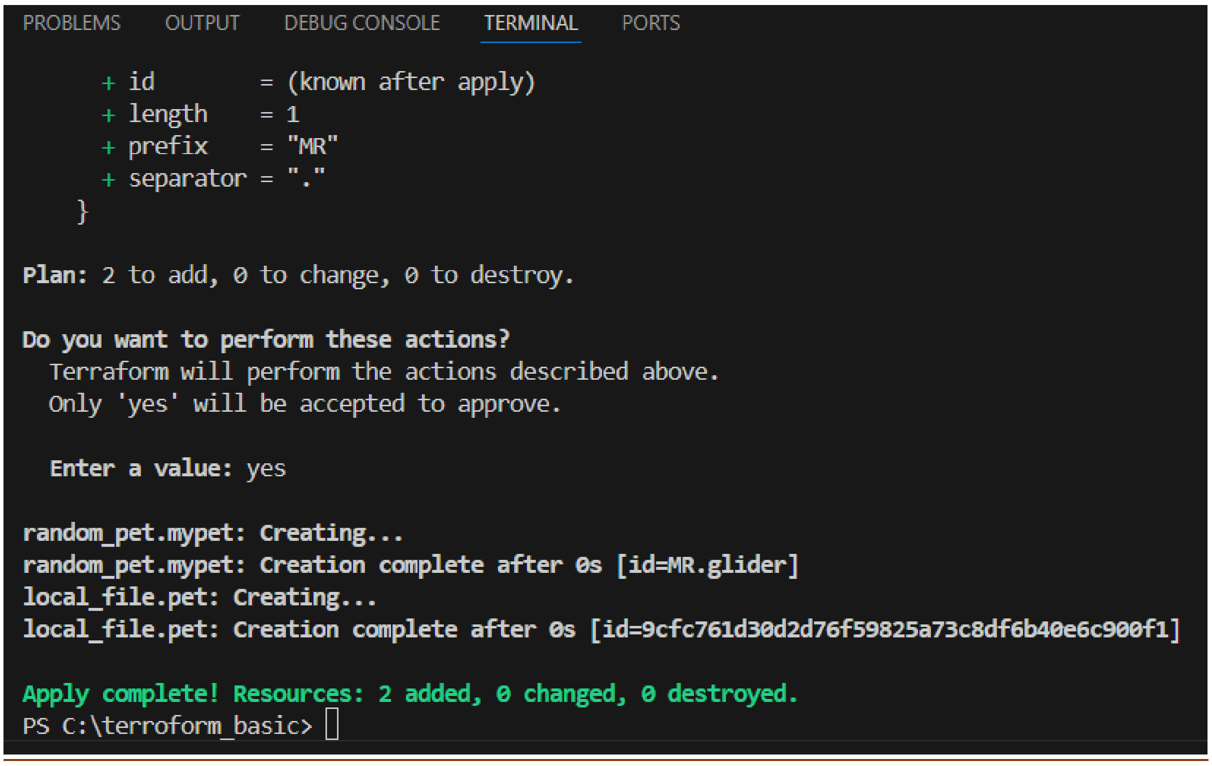
**Resource Attribute reference:**

**--template**



```
resource "local_file" "pet" {
filename = "/root/pets.txt"
content = "My cat is ${random_pet.mypet.id}"
}
resource "random_pet" "mypet" {
prefix = "MR"
separator = "."
length = "1"
}
```

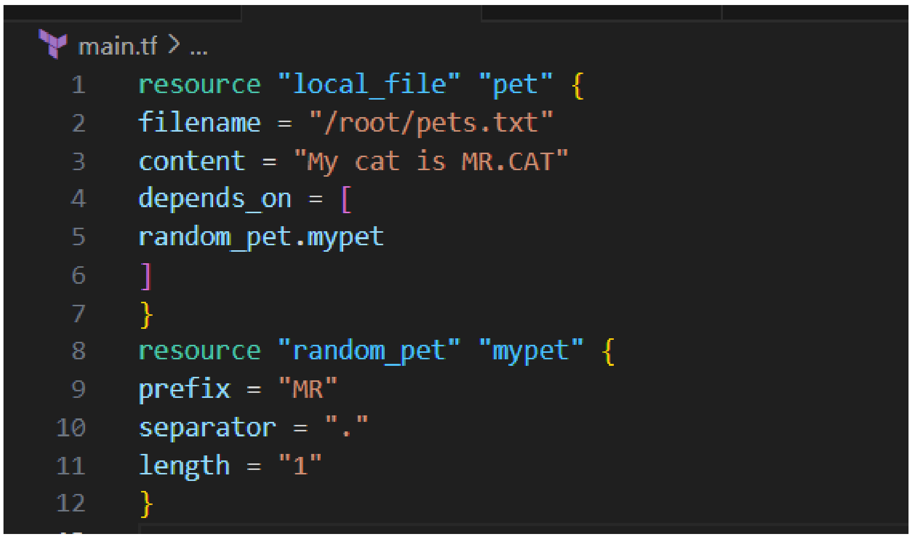
**--execution**



```
      + id        = (known after apply)
      + length    = 1
      + prefix    = "MR"
      + separator = "."
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

random_pet.mypet: Creating...
random_pet.mypet: Creation complete after 0s [id=MR.glider]
local_file.pet: Creating...
local_file.pet: Creation complete after 0s [id=9cfc761d30d2d76f59825a73c8df6b40e6c900f1]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\terroform_basic>
```

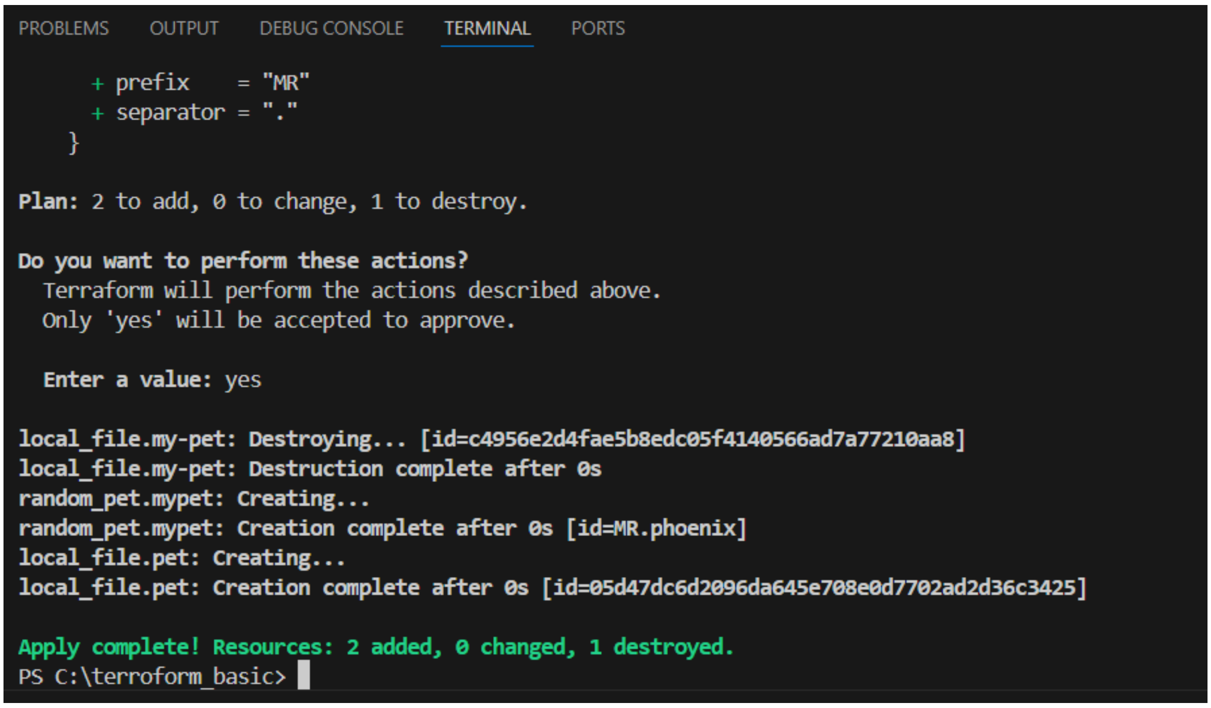
## Resource Dependencies:

**--template**



```
main.tf > ...
resource "local_file" "pet" {
filename = "/root/pets.txt"
content = "My cat is MR.CAT"
depends_on = [
random_pet.mypet
]
}
resource "random_pet" "mypet" {
prefix = "MR"
separator = "."
length = "1"
}
```

**--execution**



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
      + prefix    = "MR"
      + separator = "."
    }

Plan: 2 to add, 0 to change, 1 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.my-pet: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet: Destruction complete after 0s
random_pet.mypet: Creating...
random_pet.mypet: Creation complete after 0s [id=MR.phoenix]
local_file.pet: Creating...
local_file.pet: Creation complete after 0s [id=05d47dc6d2096da645e708e0d7702ad2d36c3425]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.
PS C:\terroform_basic>
```

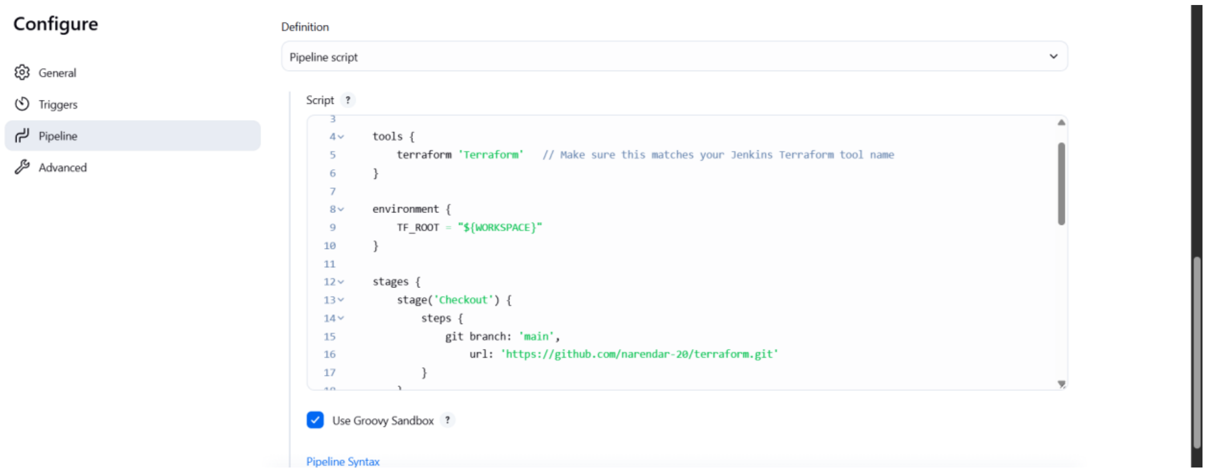
## 3) Intergrate terrafrom in jenkins using Terraform plugin.

**--install Terraform plugin in Jenkins server**

**--add that plugin in tools**

**--create job**

**-- add pipeline to it**



Definition

Pipeline script

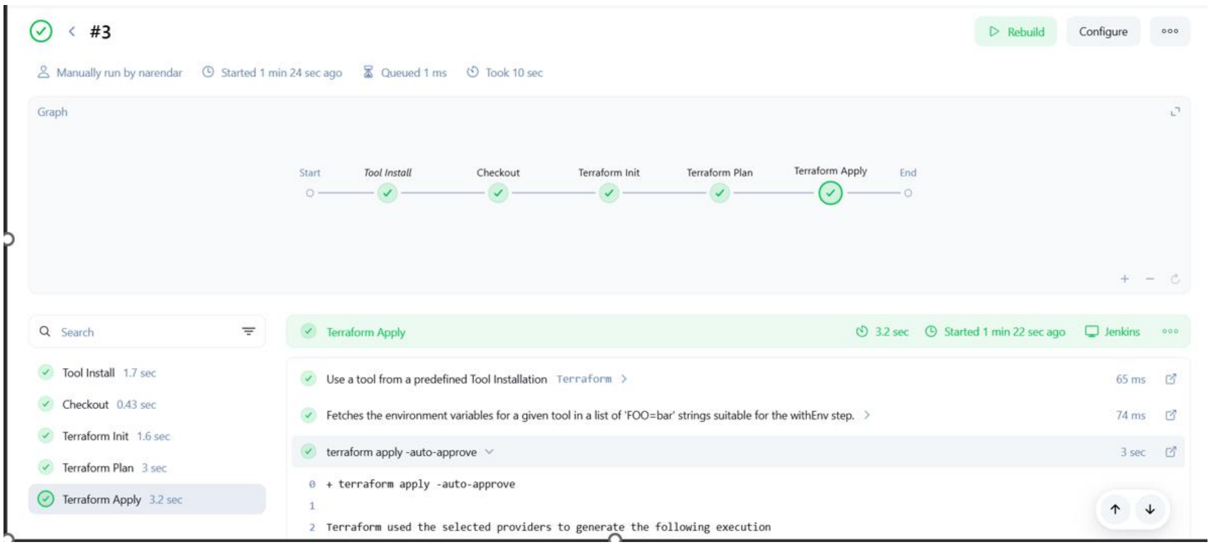
Script

```
    tools {
        terraform 'Terraform'   // Make sure this matches your Jenkins Terraform tool name
    }

    environment {
        TF_ROOT = "${WORKSPACE}"
    }

    stages {
        stage('Checkout') {
            steps {
                git branch: 'main',
                    url: 'https://github.com/narendar-20/terraform.git'
            }
```

[✓] Use Groovy Sandbox

Pipeline Syntax

**--execute the job**



Tool Install 1.7 sec
Checkout 0.43 sec
Terraform Init 1.6 sec
Terraform Plan 3 sec
Terraform Apply 3.2 sec

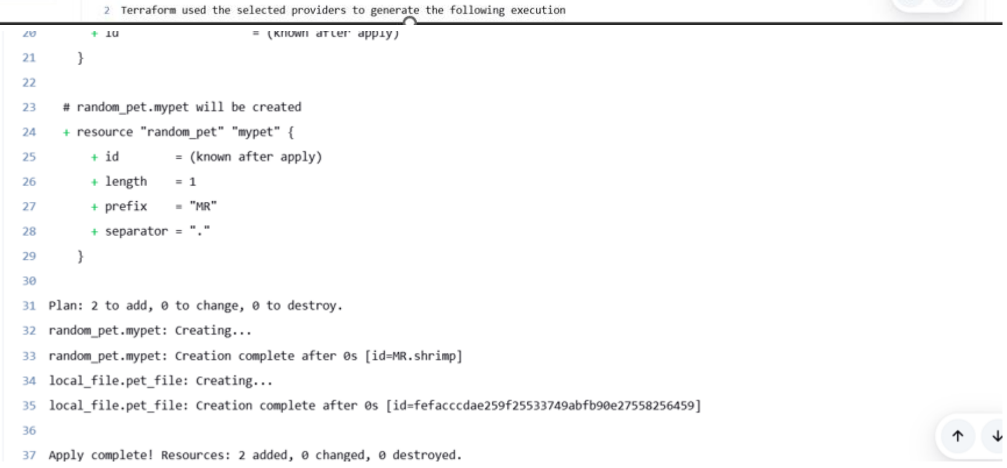
Terraform Apply 3.2 sec Started 1 min 22 sec ago Jenkins

Use a tool from a predefined Tool Installation Terraform 65 ms

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. 74 ms

terraform apply -auto-approve 3 sec

```
+ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution
```



```
      + id                    = (known after apply)
    }

  # random_pet.mypet will be created
  + resource "random_pet" "mypet" {
      + id        = (known after apply)
      + length    = 1
      + prefix    = "MR"
      + separator = "."
    }

Plan: 2 to add, 0 to change, 0 to destroy.
random_pet.mypet: Creating...
random_pet.mypet: Creation complete after 0s [id=MR.shrimp]
local_file.pet_file: Creating...
local_file.pet_file: Creation complete after 0s [id=fefacccdae259f25533749abfb90e27558256459]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

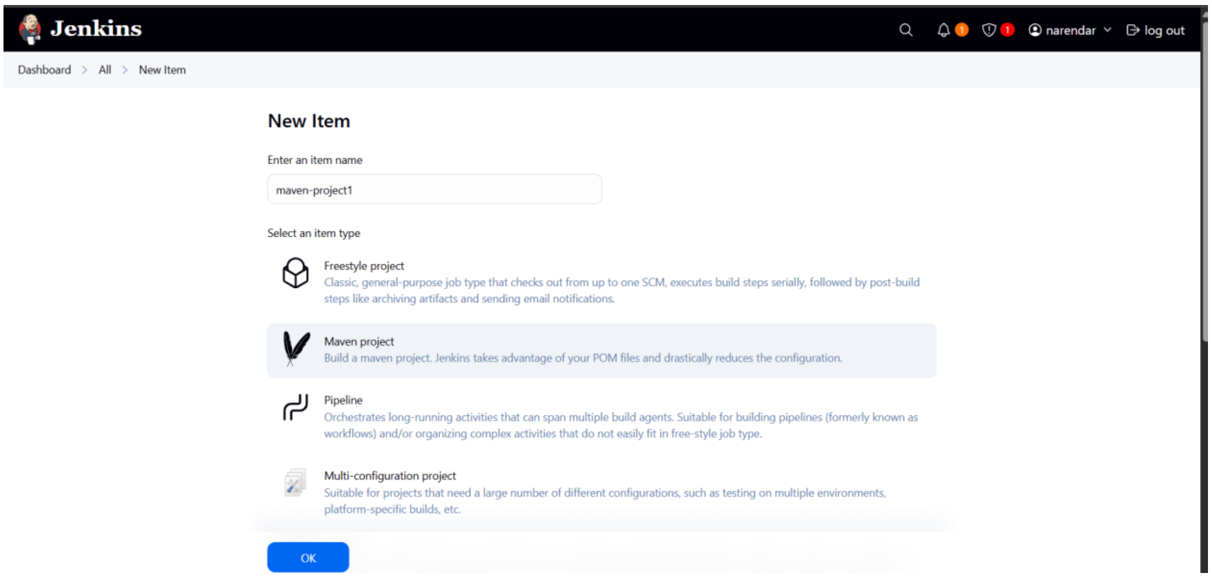
**4) Create one jenkins job using MAVEN PROJECT for the below code with two stages.**

**stage 1: Git clone**

**stage 2: Maven Compilation**

**Code: [https://github.com/betawins/java-Working-app.git](https://github.com/betawins/java-Working-app.git)**

**--create a Maven project job**



Jenkins — narendar — log out

Dashboard > All > New Item

New Item

Enter an item name

maven-project1

Select an item type

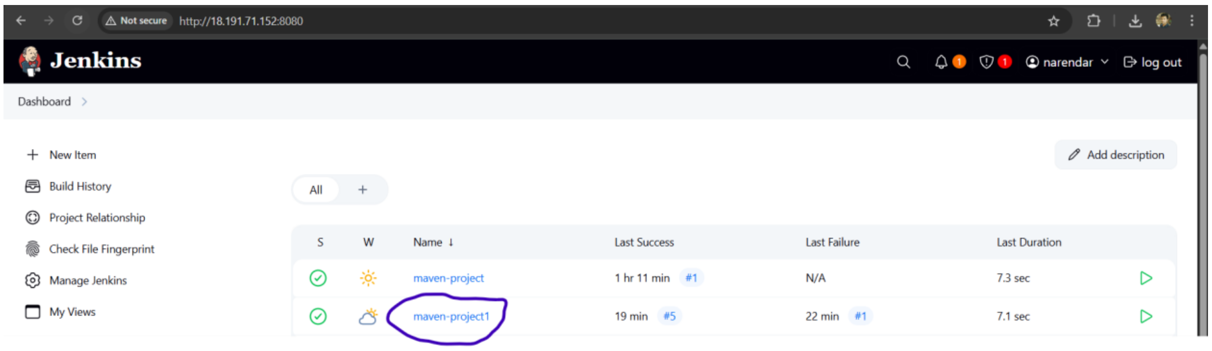
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK



Not secure http://18.191.71.152:8080

Jenkins — narendar — log out

Dashboard >

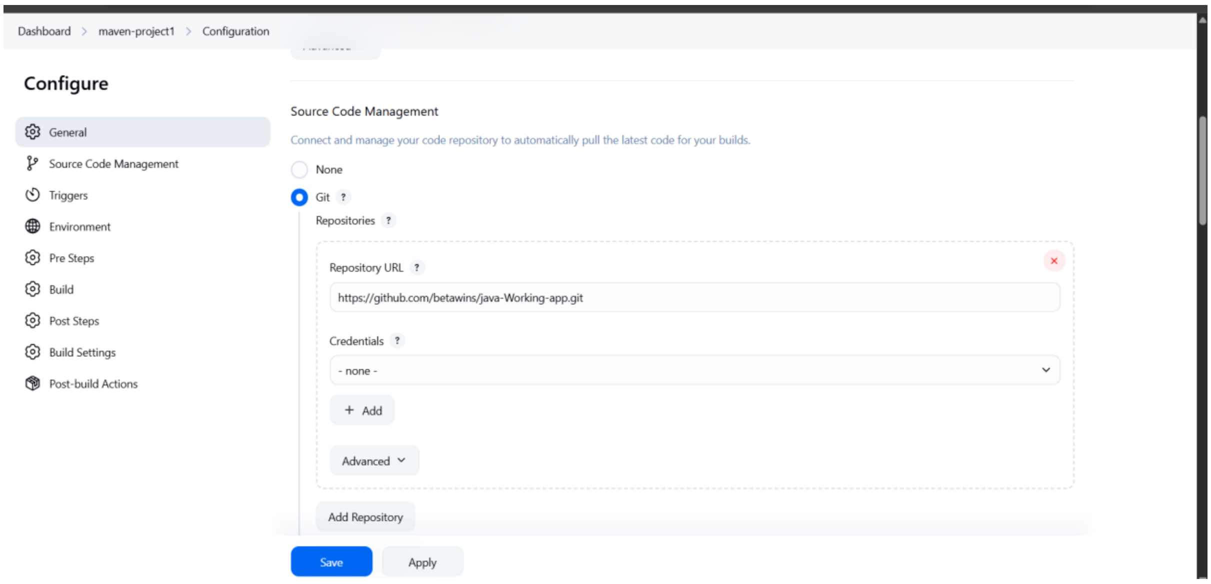
New Item
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views

Add description

All +

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| ✓ | ☀ | maven-project | 1 hr 11 min #1 | N/A | 7.3 sec |
| ✓ | ⛅ | maven-project1 | 19 min #5 | 22 min #1 | 7.1 sec |

## --add git url



Dashboard > maven-project1 > Configuration

Configure

Source Code Management

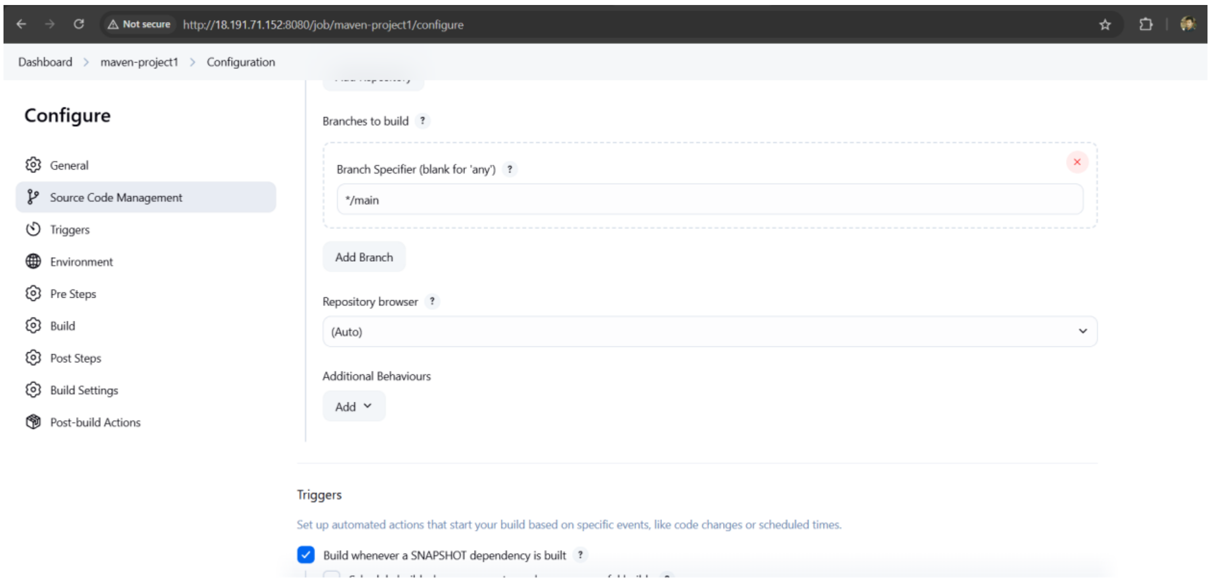
Connect and manage your code repository to automatically pull the latest code for your builds.

None [✓ Git]

Repositories

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

Credentials: - none -



http://18.191.71.152:8080/job/maven-project1/configure

Dashboard > maven-project1 > Configuration

Configure

Branches to build

Branch Specifier (blank for 'any'): */main

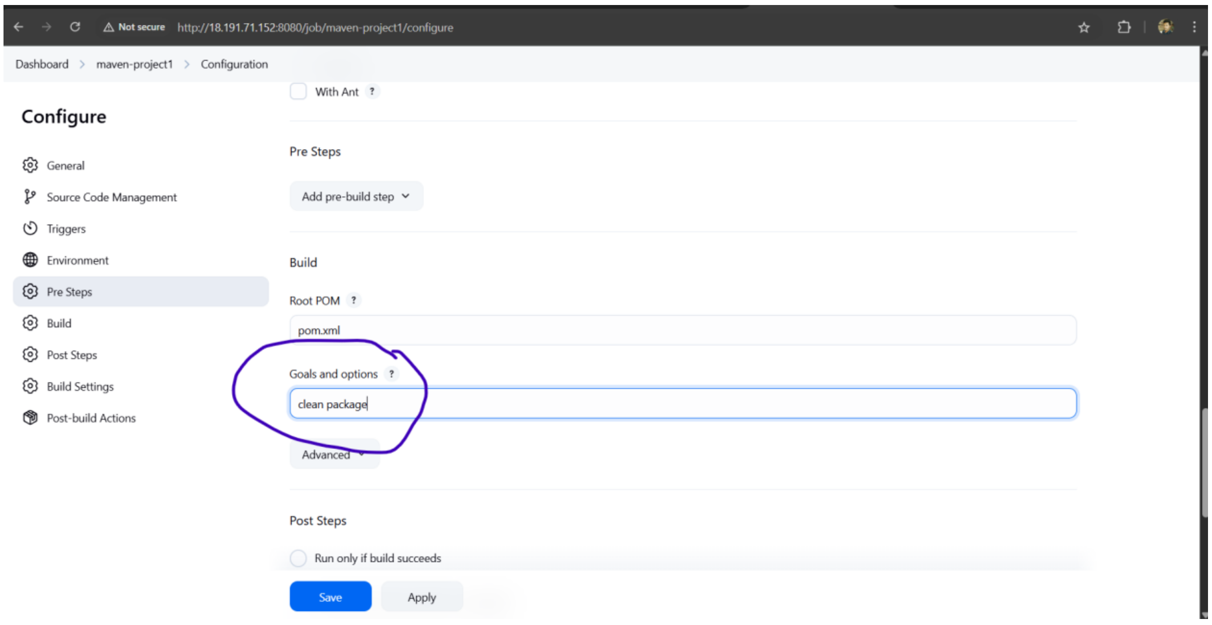
Repository browser: (Auto)

Additional Behaviours

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

[✓] Build whenever a SNAPSHOT dependency is built


http://18.191.71.152:8080/job/maven-project1/configure

Dashboard > maven-project1 > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

With Ant

Pre Steps

Add pre-build step

Build

Root POM: pom.xml

Goals and options: clean package

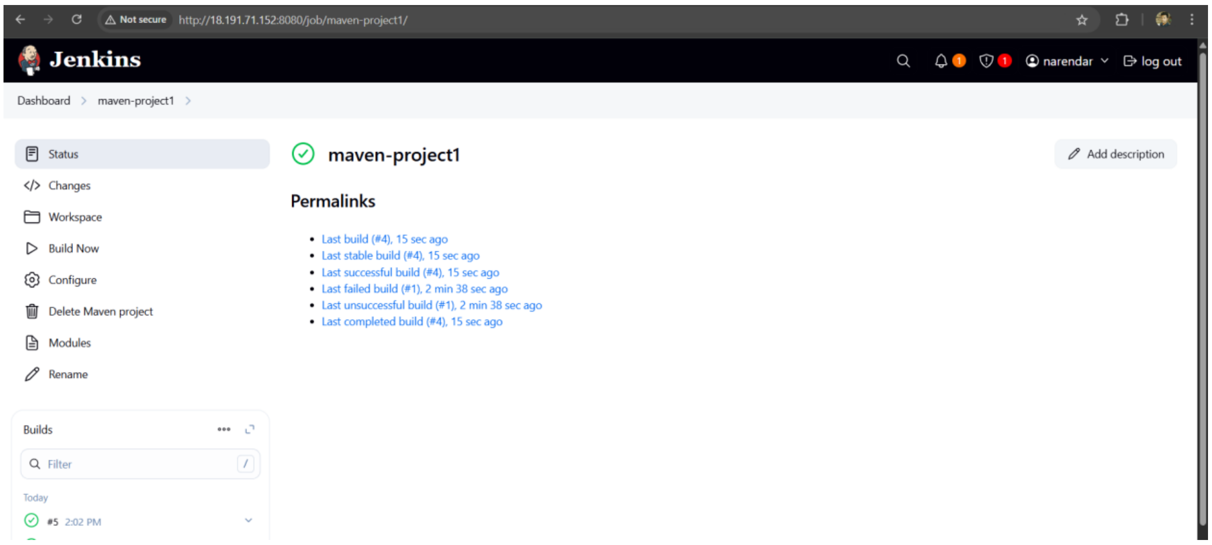
Advanced

Post Steps

Run only if build succeeds

Save | Apply

## --execution


http://18.191.71.152:8080/job/maven-project1/

Jenkins — narendar — log out

Dashboard > maven-project1 >

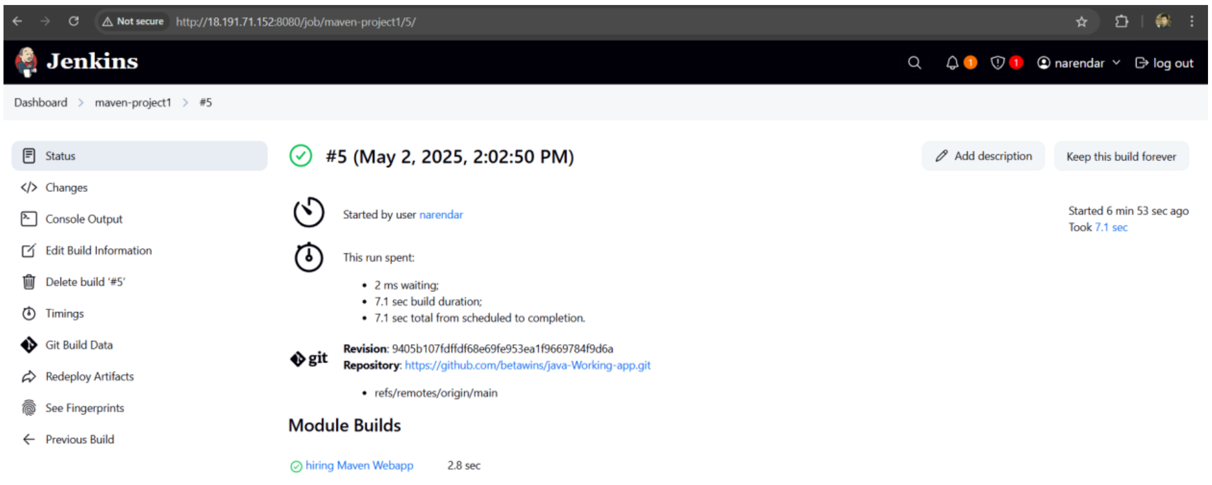
- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Maven project
- Modules
- Rename

Builds — Filter — Today — #5 2:02 PM

maven-project1 — Add description

Permalinks

- Last build (#4), 15 sec ago
- Last stable build (#4), 15 sec ago
- Last successful build (#4), 15 sec ago
- Last failed build (#1), 2 min 38 sec ago
- Last unsuccessful build (#1), 2 min 38 sec ago
- Last completed build (#4), 15 sec ago


http://18.191.71.152:8080/job/maven-project1/5/

Jenkins — narendar — log out

Dashboard > maven-project1 > #5

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#5'
- Timings
- Git Build Data
- Redeploy Artifacts
- See Fingerprints
- Previous Build

#5 (May 2, 2025, 2:02:50 PM) — Add description — Keep this build forever

Started by user narendar

Started 6 min 53 sec ago
Took 7.1 sec

This run spent:

- 2 ms waiting;
- 7.1 sec build duration;
- 7.1 sec total from scheduled to completion.

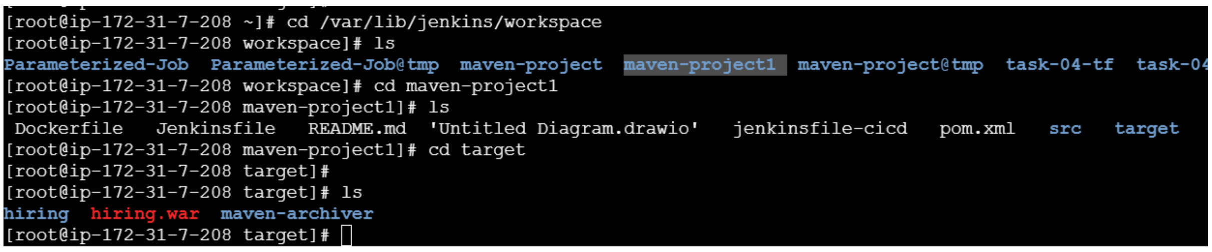
git **Revision**: 9405b107fdffdf68e69fe953ea1f9669784f9d6a
**Repository**: https://github.com/betawins/java-Working-app.git

- refs/remotes/origin/main

Module Builds

hiring Maven Webapp 2.8 sec

**--check in Jenkins server .war file got created**



```
[root@ip-172-31-7-208 ~]# cd /var/lib/jenkins/workspace
[root@ip-172-31-7-208 workspace]# ls
Parameterized-Job  Parameterized-Job@tmp  maven-project  maven-project1  maven-project@tmp  task-04-tf  task-04
[root@ip-172-31-7-208 workspace]# cd maven-project1
[root@ip-172-31-7-208 maven-project1]# ls
 Dockerfile   Jenkinsfile   README.md  'Untitled Diagram.drawio'   jenkinsfile-cicd   pom.xml   src   target
[root@ip-172-31-7-208 maven-project1]# cd target
[root@ip-172-31-7-208 target]#
[root@ip-172-31-7-208 target]# ls
hiring  hiring.war  maven-archiver
[root@ip-172-31-7-208 target]#
```

**i-06add05e8e88f0610 (jenkins)**

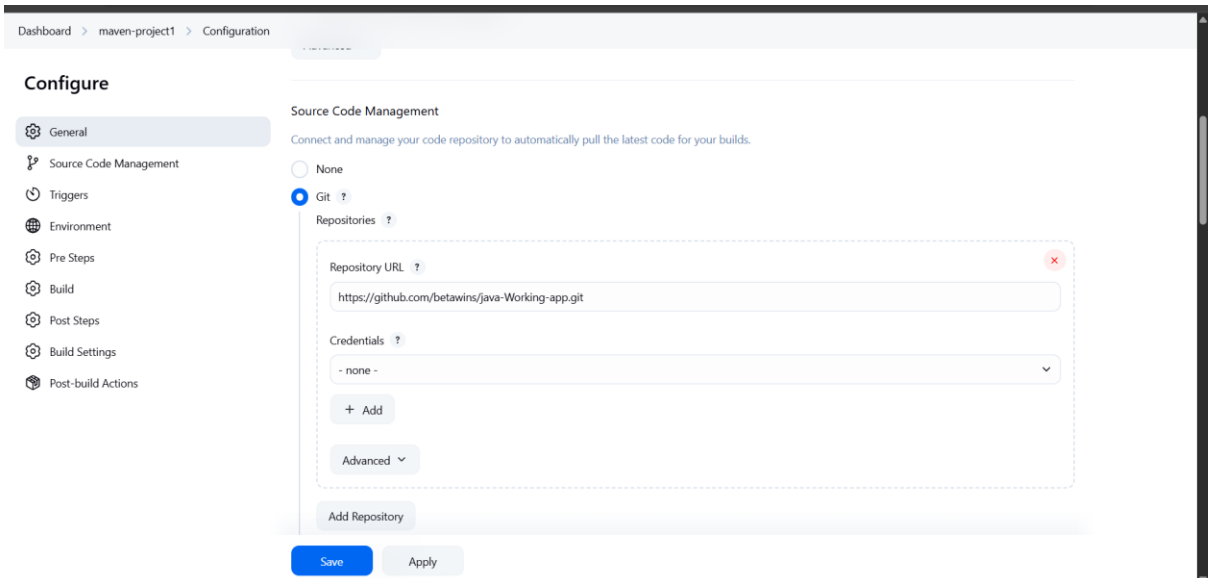
## 5) Use the below code and create a parameterized job in jenkins

**stage 1: Git clone**

**stage 2: Maven Compilation**

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

**--install maven integration plugin**



Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

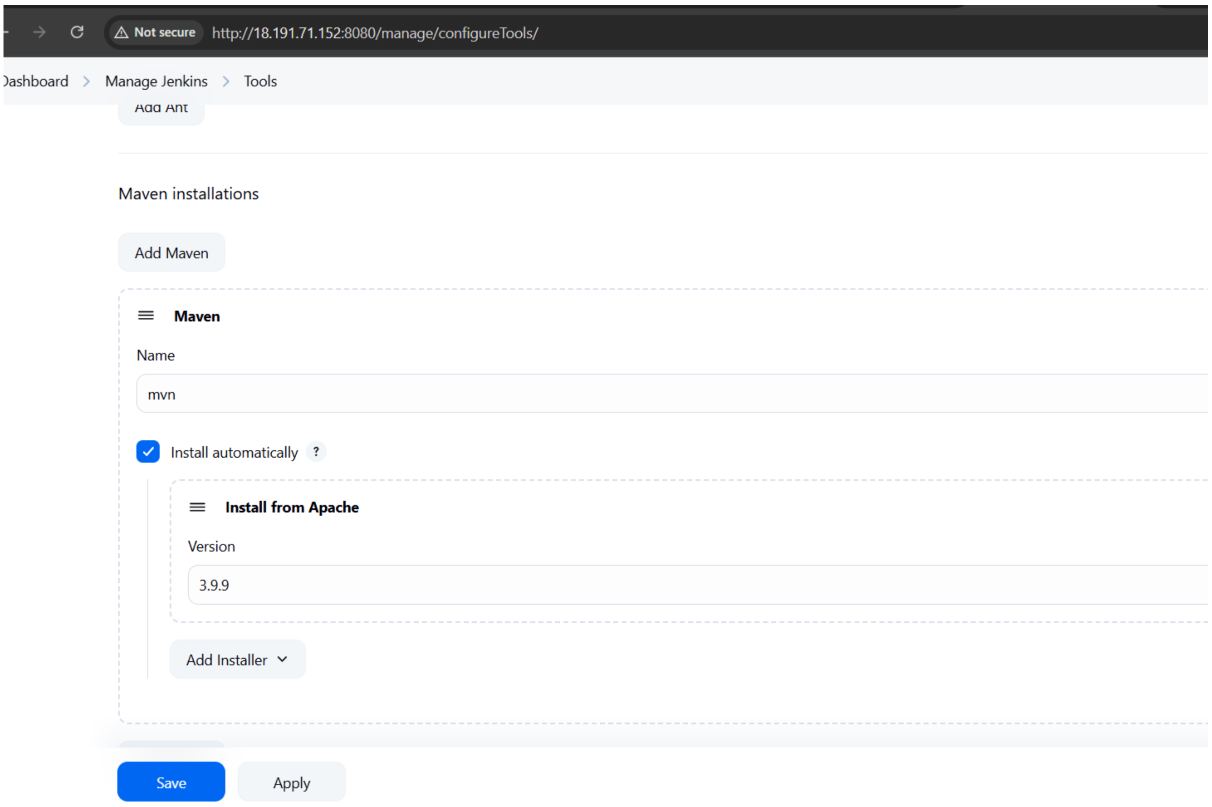
Git [✓]

Repositories

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

Credentials: - none -

## --add plugin in tools



http://18.191.71.152:8080/manage/configureTools/

Dashboard > Manage Jenkins > Tools

Maven installations

Maven

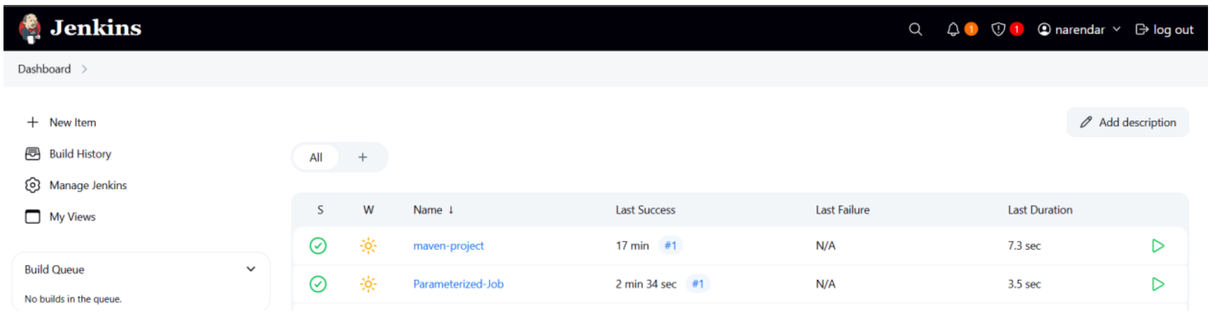
Name: mvn

Install automatically ✓

Install from Apache

Version: 3.9.9

## --create a job

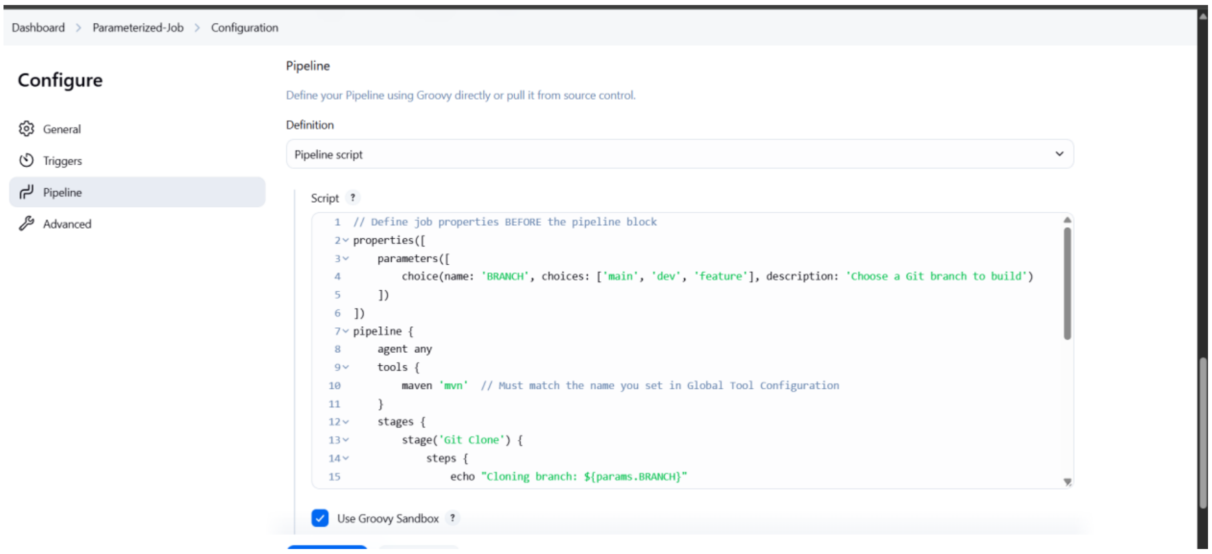


| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| ✓ | ☀ | maven-project | 17 min #1 | N/A | 7.3 sec |
| ✓ | ☀ | Parameterized-Job | 2 min 34 sec #1 | N/A | 3.5 sec |

Build Queue

No builds in the queue.

## --pipeline


Dashboard > Parameterized-Job > Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

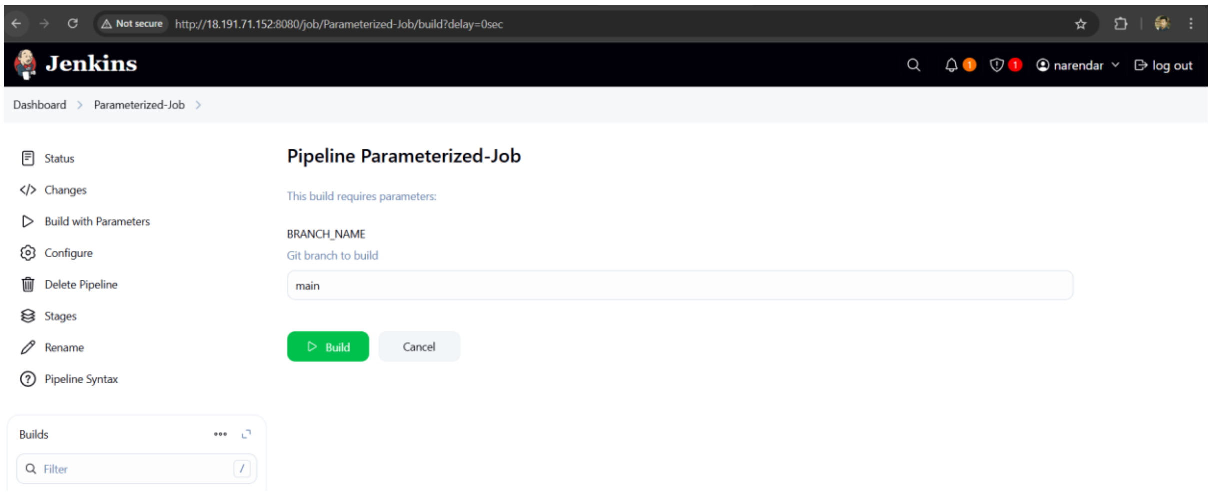
Script

```
// Define job properties BEFORE the pipeline block
properties([
    parameters([
        choice(name: 'BRANCH', choices: ['main', 'dev', 'feature'], description: 'Choose a Git branch to build')
    ])
])
pipeline {
    agent any
    tools {
        maven 'mvn'  // Must match the name you set in Global Tool Configuration
    }
    stages {
        stage('Git Clone') {
            steps {
                echo "Cloning branch: ${params.BRANCH}"
```

Use Groovy Sandbox

## --execution

## --build with parameters


http://18.191.71.152:8080/job/Parameterized-Job/build?delay=0sec

Jenkins

Dashboard > Parameterized-Job >

- Status
- Changes
- Build with Parameters
- Configure
- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax

Builds

Pipeline Parameterized-Job

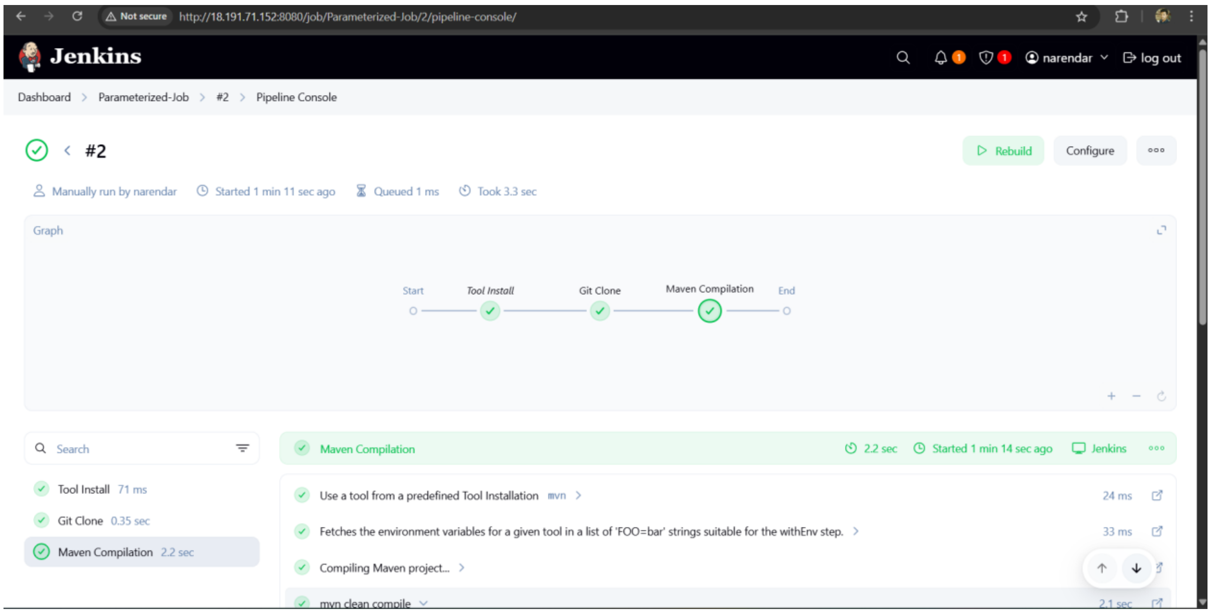
This build requires parameters:

BRANCH_NAME

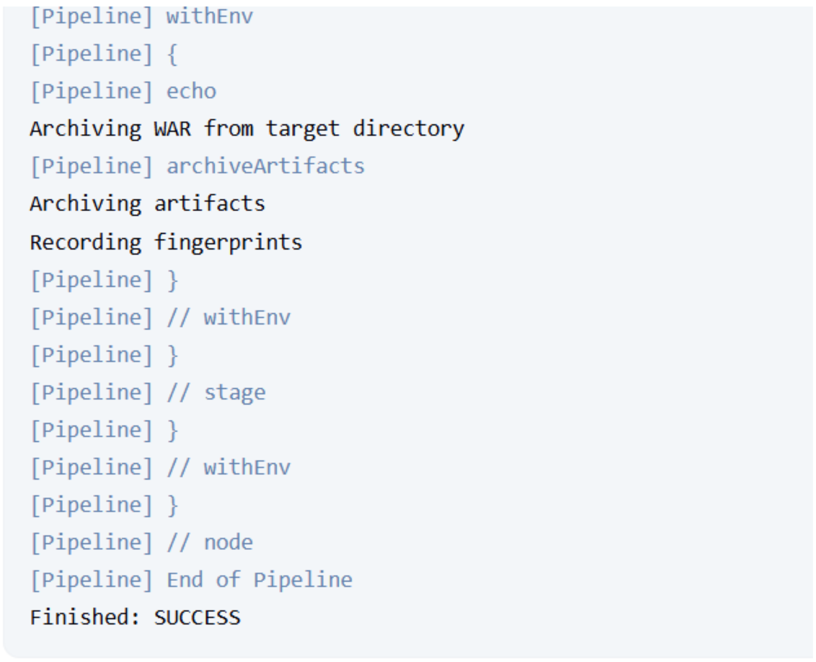
Git branch to build

main

Build Cancel

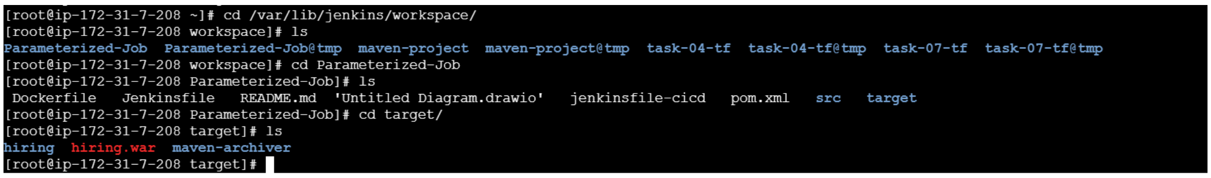


```
Use a tool from a predefined Tool Installation mvn                                24 ms
Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.   33 ms
Compiling Maven project...
mvn clean compile                                                                 2.1 sec
```



```
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
Archiving WAR from target directory
[Pipeline] archiveArtifacts
Archiving artifacts
Recording fingerprints
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## --check in Jenkins server .war file has created or not



```
[root@ip-172-31-7-208 ~]# cd /var/lib/jenkins/workspace/
[root@ip-172-31-7-208 workspace]# ls
Parameterized-Job  Parameterized-Job@tmp  maven-project  maven-project@tmp  task-04-tf  task-04-tf@tmp  task-07-tf  task-07-tf@tmp
[root@ip-172-31-7-208 workspace]# cd Parameterized-Job
[root@ip-172-31-7-208 Parameterized-Job]# ls
 Dockerfile   Jenkinsfile   README.md  'Untitled Diagram.drawio'   jenkinsfile-cicd   pom.xml   src   target
[root@ip-172-31-7-208 Parameterized-Job]# cd target/
[root@ip-172-31-7-208 target]# ls
hiring  hiring.war  maven-archiver
[root@ip-172-31-7-208 target]#
```

## 6) What are the global varaiables in jenkins?

### 1. env

**Definition:** Access environment variables.
**Example:**

```
echo "Path is: ${env.PATH}"
```

### 2. params

**Definition:** Access build parameters.
**Example:**

```
echo "Username: ${params.USERNAME}"
```

### 3. currentBuild

**Definition:** Metadata about the current build.
**Example:**

```
echo "Build number: ${currentBuild.number}"
```

### 4. scm

**Definition:** Refers to the source control (Git, SVN).
**Example:**

```
 checkout scm
```

### 5. node

**Definition:** Defines the Jenkins agent (node) to run the job.
**Example:**

```
node {

  echo "Running on a node"

}
```

### 6. tool

**Definition:** Uses a tool configured in Jenkins (e.g., Maven).
**Example:**

```
    def mvnHome = tool 'Maven 3.8.5'
```

### 7. pipeline

**Definition:** Refers to the pipeline script object (used in libraries).
**Example:**

```
pipeline {
   agent any
}
```

**8. Shared Library Variable**

**Definition:** Custom global functions from vars/ directory in a shared library.

```
helloWorld('Narendar')  // Defined in vars/helloWorld.groovy
```

**9. User-defined environment variable**

**Definition:** Custom env variable declared in pipeline.
**Example:**

```
environment {
   DEPLOY_ENV = 'dev'
}
```

**10. User-defined Groovy variable**

**Definition**: Custom variable in scripted pipeline.
**Example:**

```
   def appName = 'MyApp'
   echo "Deploying ${appName}"
```

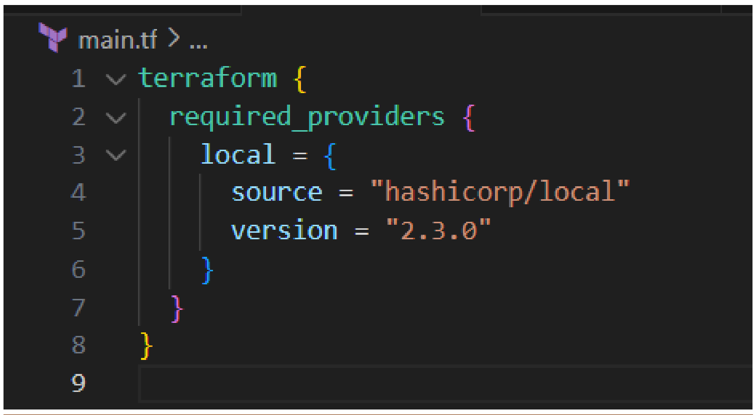
## 7) Watch terraform-04 video.

**--completed**

## 8) Execute the script shown in video.

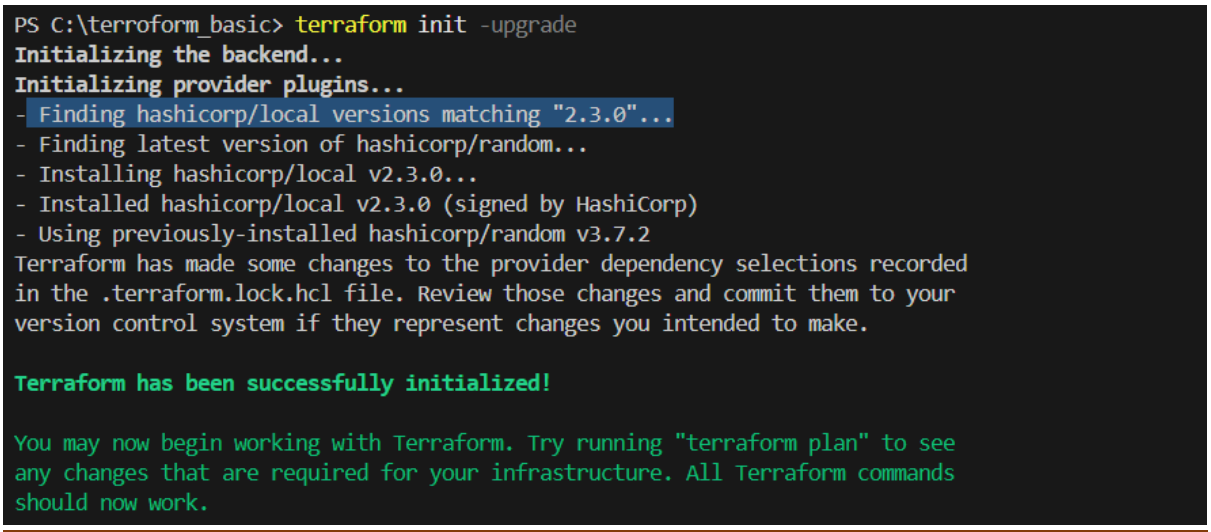
**Version Constraints:**

**--template**

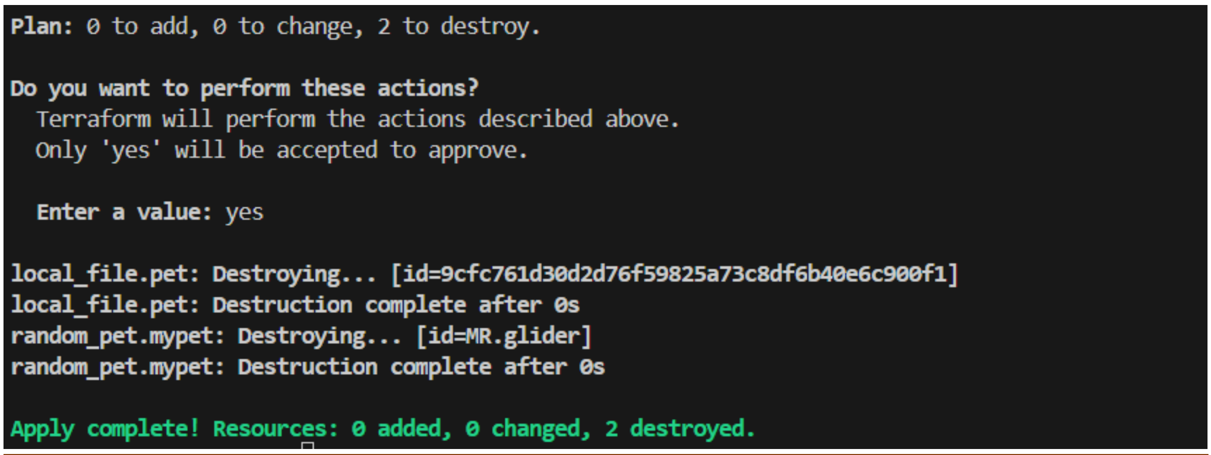


```
main.tf > ...
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "2.3.0"
    }
  }
}
```

**--execution**



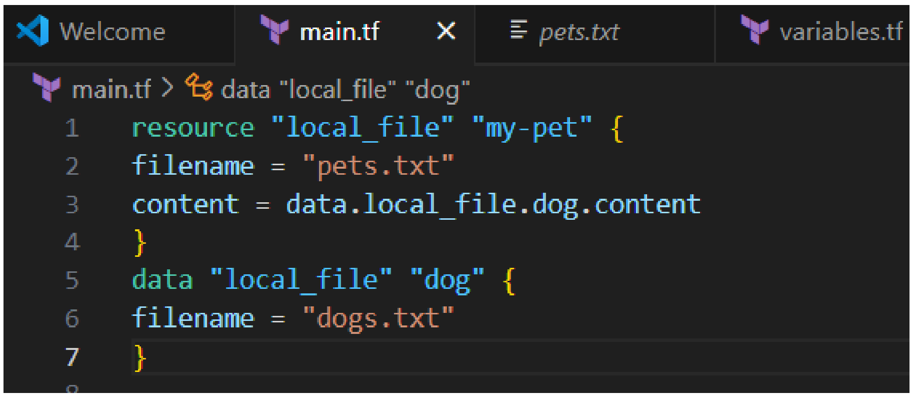
```
PS C:\terroform_basic> terraform init -upgrade
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/local versions matching "2.3.0"...
- Finding latest version of hashicorp/random...
- Installing hashicorp/local v2.3.0...
- Installed hashicorp/local v2.3.0 (signed by HashiCorp)
- Using previously-installed hashicorp/random v3.7.2
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```



```
Plan: 0 to add, 0 to change, 2 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.pet: Destroying... [id=9cfc761d30d2d76f59825a73c8df6b40e6c900f1]
local_file.pet: Destruction complete after 0s
random_pet.mypet: Destroying... [id=MR.glider]
random_pet.mypet: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 2 destroyed.
```

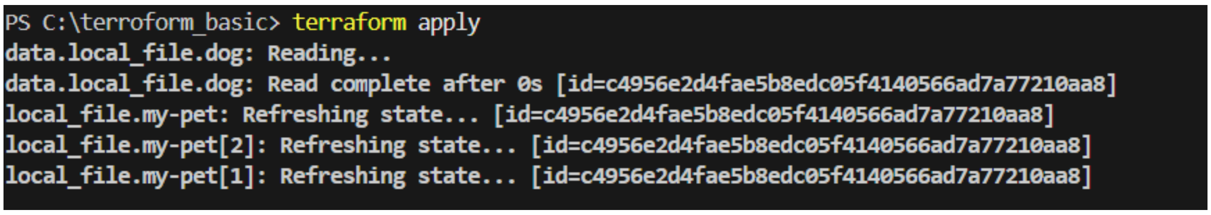
## Data Sources:

### --template

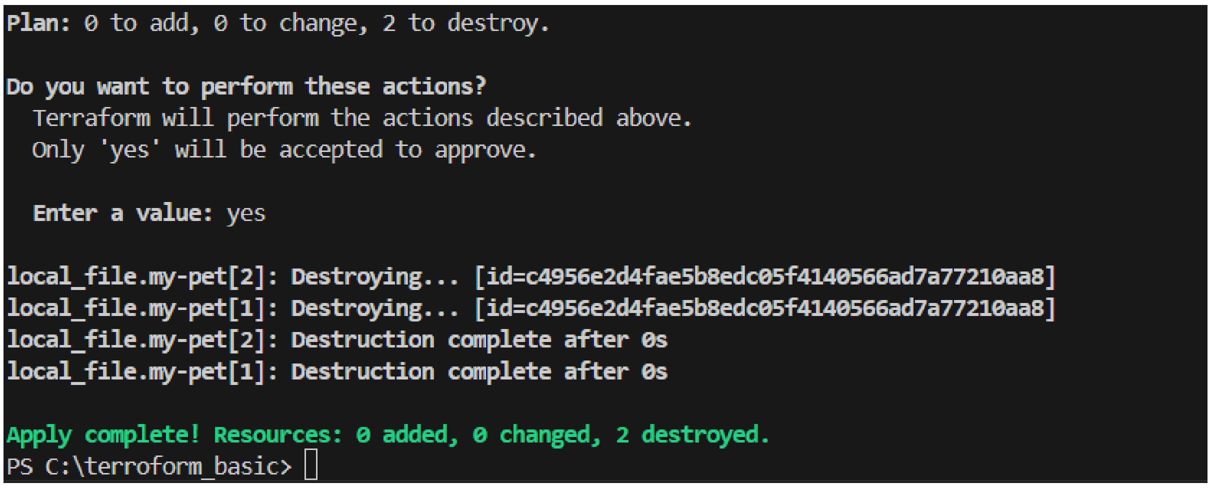


```
resource "local_file" "my-pet" {
filename = "pets.txt"
content = data.local_file.dog.content
}
data "local_file" "dog" {
filename = "dogs.txt"
}
```

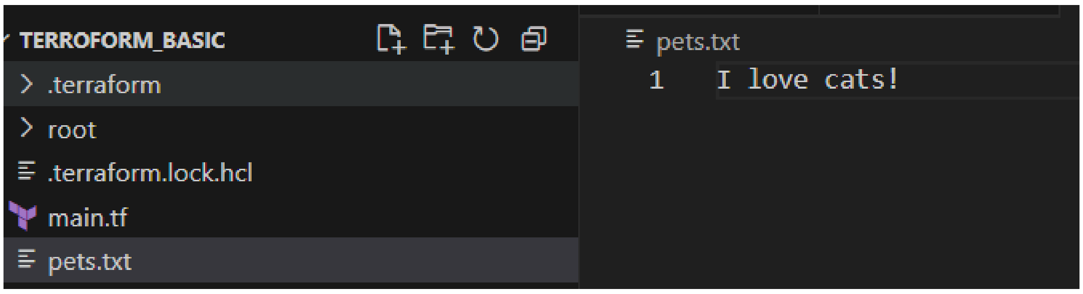
### --execution



```
PS C:\terroform_basic> terraform apply
data.local_file.dog: Reading...
data.local_file.dog: Read complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet: Refreshing state... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[2]: Refreshing state... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Refreshing state... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
```



```
Plan: 0 to add, 0 to change, 2 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.my-pet[2]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[2]: Destruction complete after 0s
local_file.my-pet[1]: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 2 destroyed.
PS C:\terroform_basic>
```



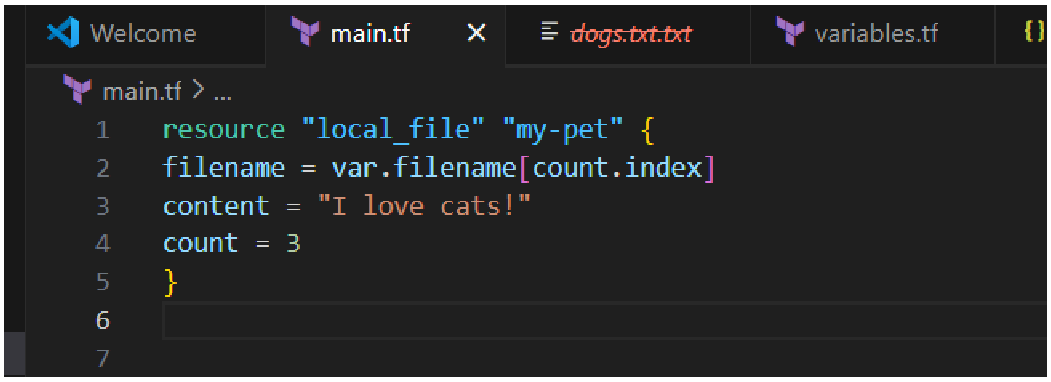
```
I love cats!
```

## Meta-Arguments:

### Example of count

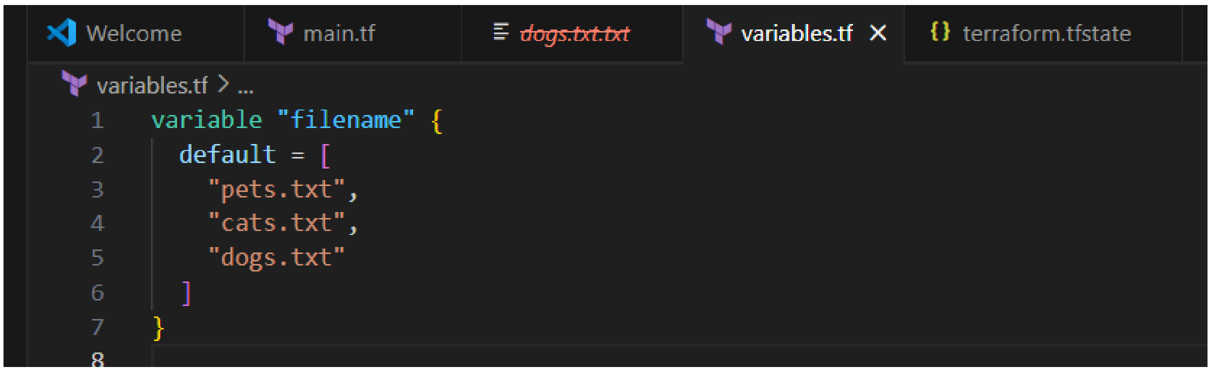
**--templates**

**Main.tf**



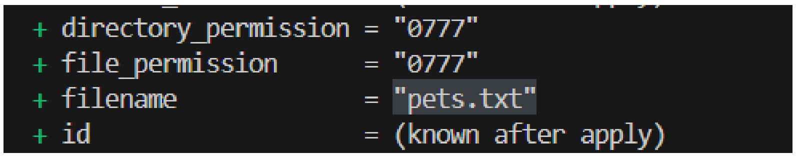
```
resource "local_file" "my-pet" {
filename = var.filename[count.index]
content = "I love cats!"
count = 3
}
```

**Variables.tf**

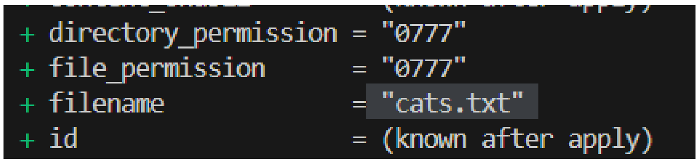


```
variable "filename" {
  default = [
    "pets.txt",
    "cats.txt",
    "dogs.txt"
  ]
}
```

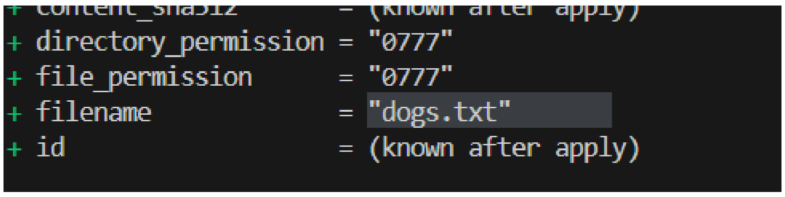
**--execution**



```
+ directory_permission = "0777"
+ file_permission      = "0777"
+ filename             = "pets.txt"
+ id                   = (known after apply)
```

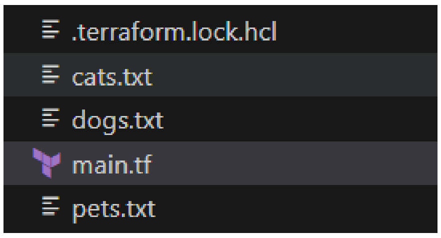


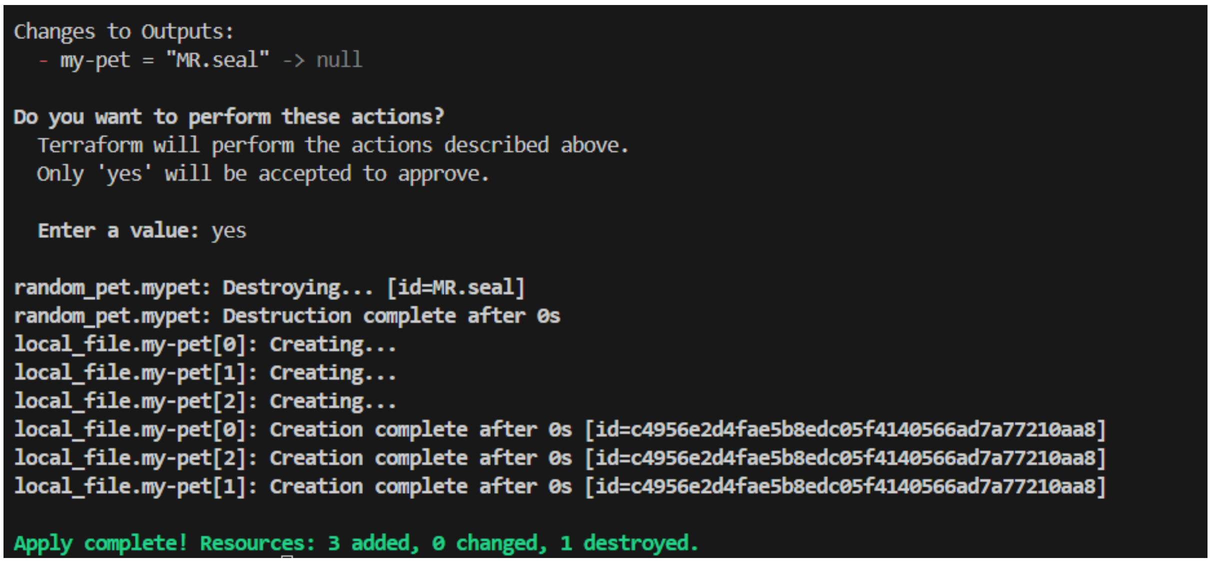
```
+ directory_permission = "0777"
+ file_permission      = "0777"
+ filename             = "cats.txt"
+ id                   = (known after apply)
```



```
+ directory_permission = "0777"
+ file_permission      = "0777"
+ filename             = "dogs.txt"
+ id                   = (known after apply)
```

**3 files are created**

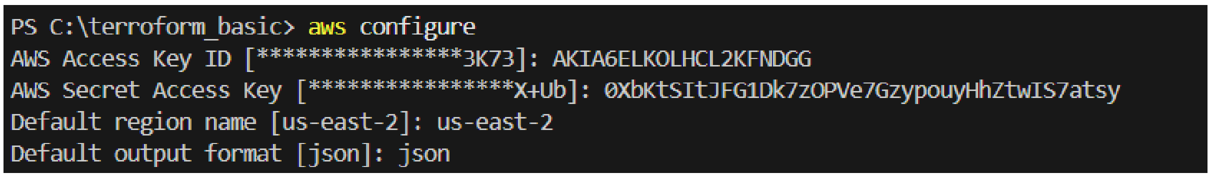




```
Changes to Outputs:
  - my-pet = "MR.seal" -> null

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

random_pet.mypet: Destroying... [id=MR.seal]
random_pet.mypet: Destruction complete after 0s
local_file.my-pet[0]: Creating...
local_file.my-pet[1]: Creating...
local_file.my-pet[2]: Creating...
local_file.my-pet[0]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[2]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]

Apply complete! Resources: 3 added, 0 changed, 1 destroyed.
```

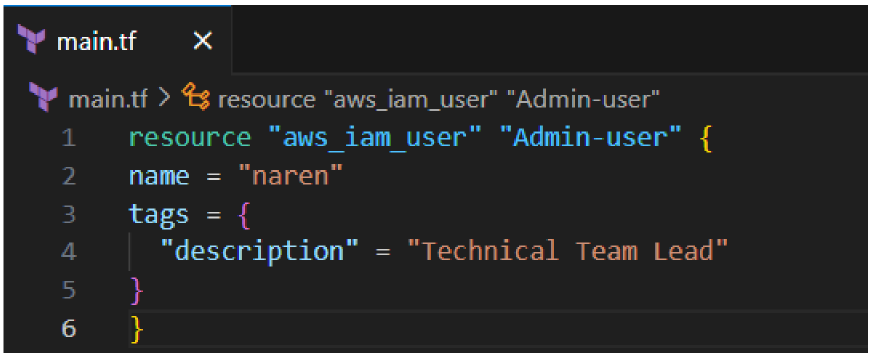
## Create an AWS IAM user:

### --do aws configure



```
PS C:\terroform_basic> aws configure
AWS Access Key ID [****************3K73]: AKIA6ELKOLHCL2KFNDGG
AWS Secret Access Key [****************X+Ub]: 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
Default region name [us-east-2]: us-east-2
Default output format [json]: json
```

### --template

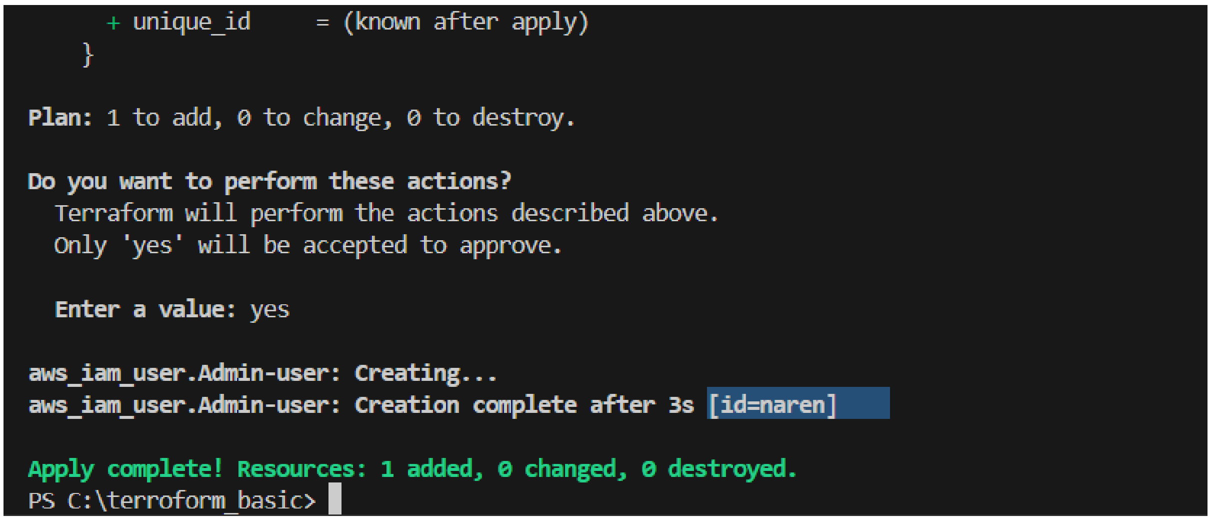


main.tf ×

main.tf > resource "aws_iam_user" "Admin-user"

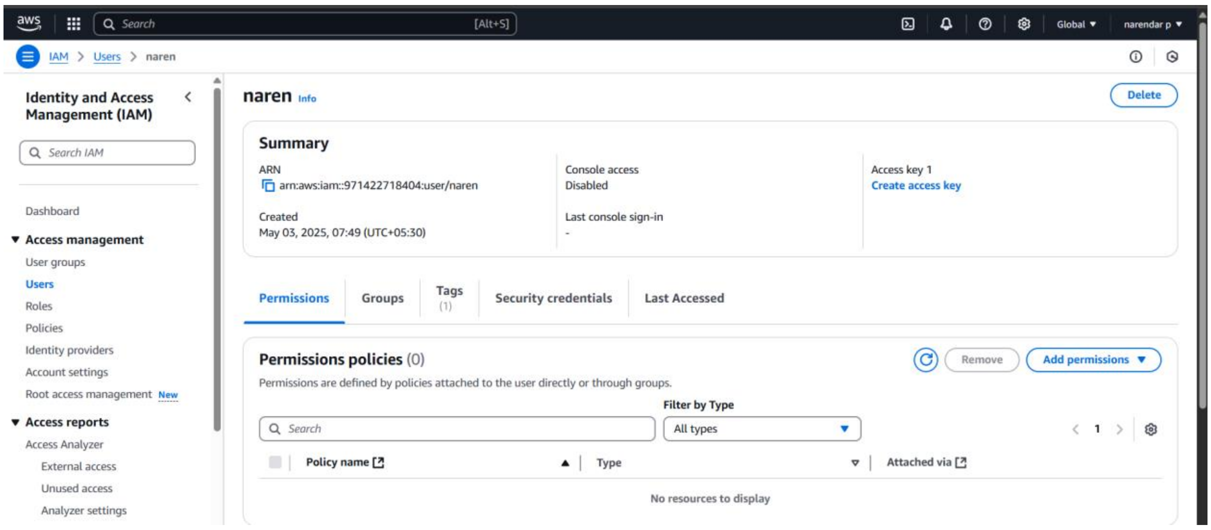
```
resource "aws_iam_user" "Admin-user" {
name = "naren"
tags = {
  "description" = "Technical Team Lead"
}
}
```

### --execution



```
      + unique_id     = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_iam_user.Admin-user: Creating...
aws_iam_user.Admin-user: Creation complete after 3s [id=naren]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\terroform_basic>
```

**--check in aws account IAM user has been created**



IAM > Users > naren

Identity and Access Management (IAM)

Dashboard

Access management
- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management New

Access reports
- Access Analyzer
  - External access
  - Unused access
  - Analyzer settings

**naren** Info

Summary

ARN
arn:aws:iam::971422718404:user/naren

Created
May 03, 2025, 07:49 (UTC+05:30)

Console access
Disabled

Last console sign-in
-

Access key 1
Create access key

Permissions | Groups | Tags (1) | Security credentials | Last Accessed

Permissions policies (0)

Permissions are defined by policies attached to the user directly or through groups.

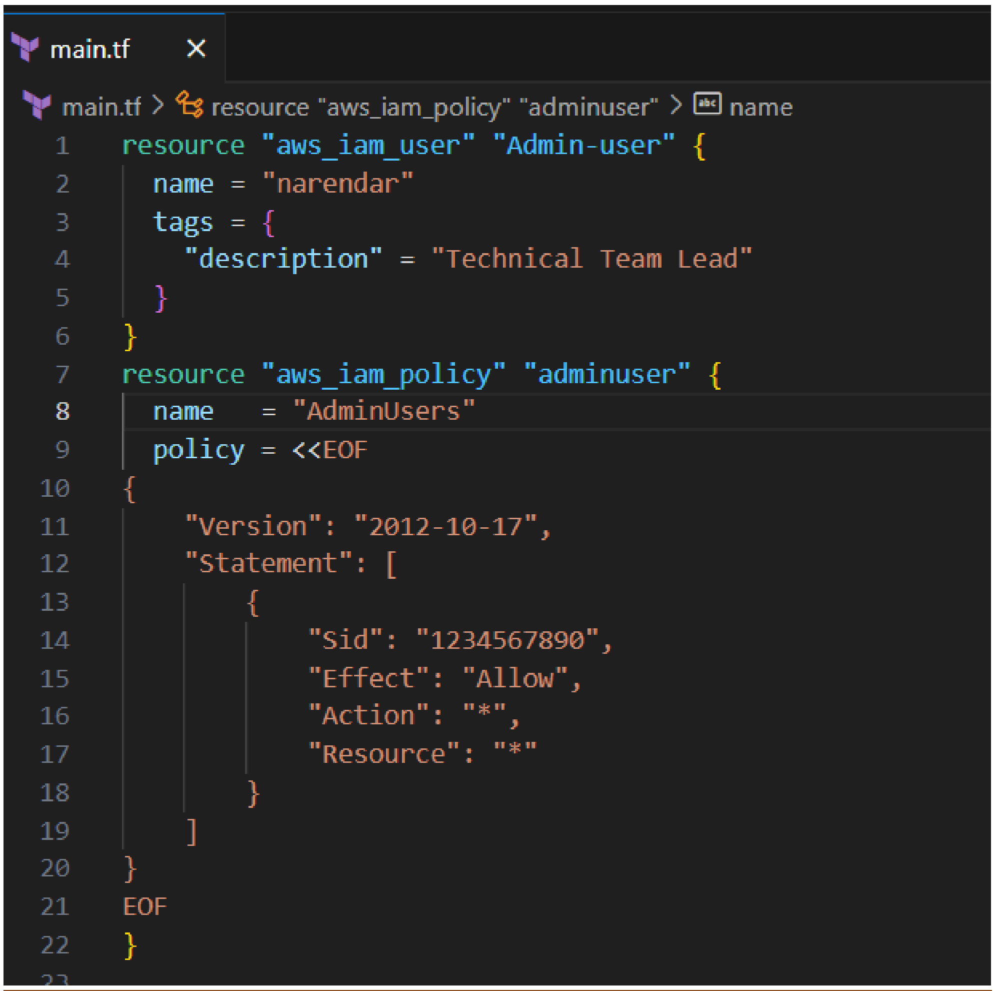
Filter by Type: All types

| Policy name | Type | Attached via |
|---|---|---|

No resources to display

## Create an AWS IAM user with policy attached to the user narendar:

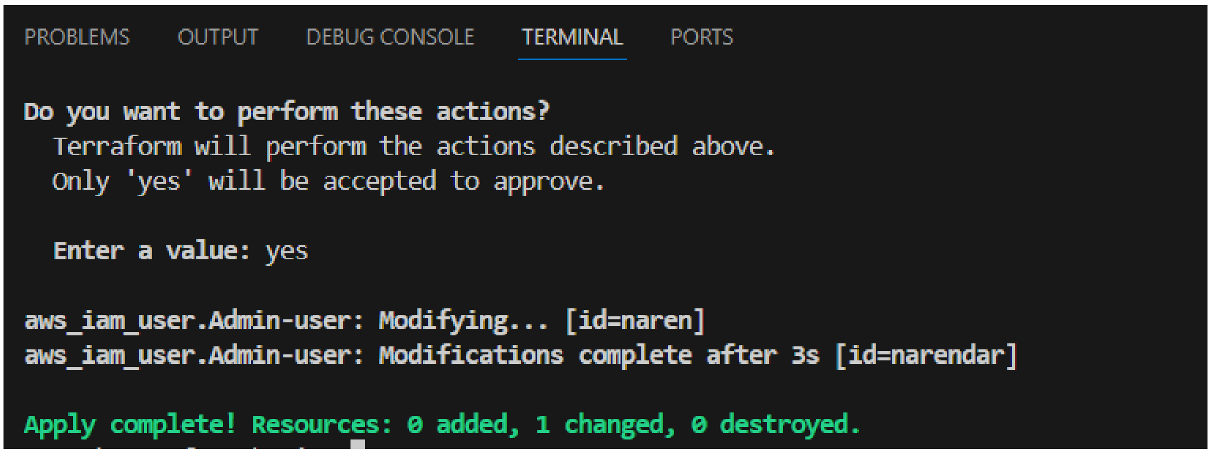
**--template**



main.tf > resource "aws_iam_policy" "adminuser" > name

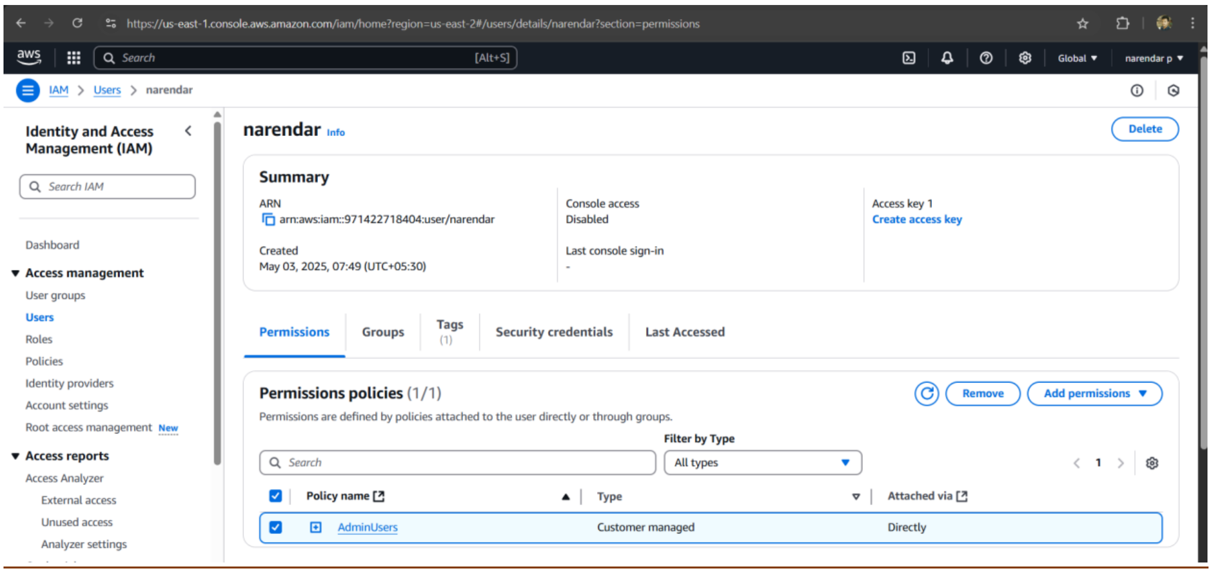
```
resource "aws_iam_user" "Admin-user" {
  name = "narendar"
  tags = {
    "description" = "Technical Team Lead"
  }
}
resource "aws_iam_policy" "adminuser" {
  name    = "AdminUsers"
  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1234567890",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
EOF
}
```

**--execution**



```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_iam_user.Admin-user: Modifying... [id=naren]
aws_iam_user.Admin-user: Modifications complete after 3s [id=narendar]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

**--check in aws account narendar user got created with attach policy**



https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-2#/users/details/narendar?section=permissions

narendar

Summary

ARN: arn:aws:iam::971422718404:user/narendar
Created: May 03, 2025, 07:49 (UTC+05:30)
Console access: Disabled
Last console sign-in: -
Access key 1: Create access key

Permissions policies (1/1)

Permissions are defined by policies attached to the user directly or through groups.

| Policy name | Type | Attached via |
|---|---|---|
| AdminUsers | Customer managed | Directly |

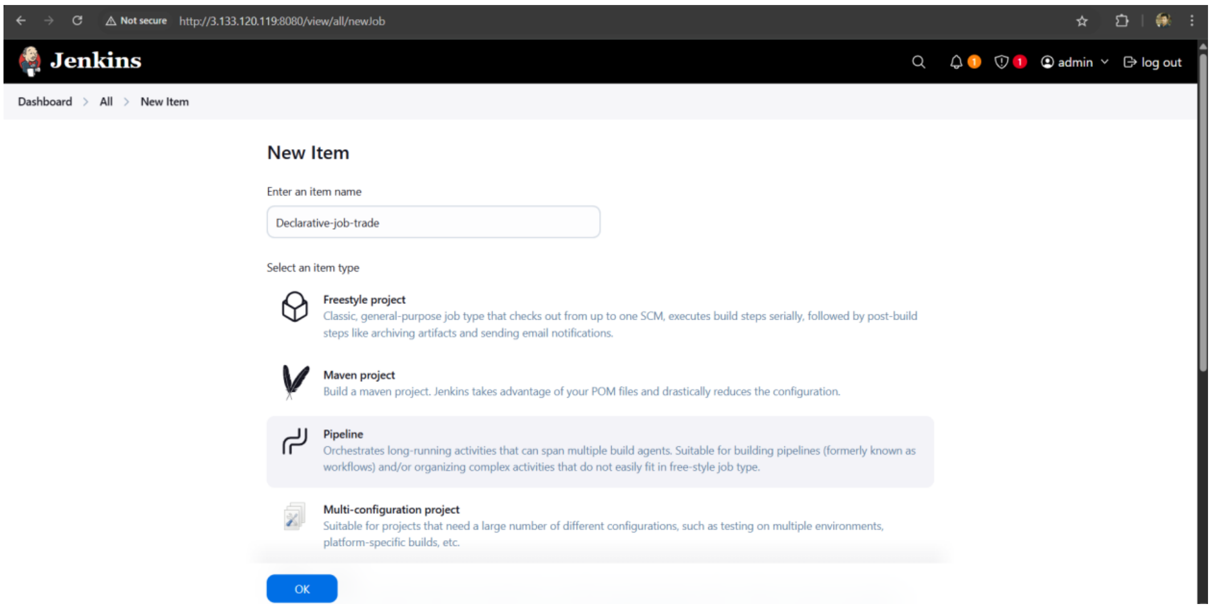
## 9) Integrate terrafrom in jenkins using Terraform plugin.

**--done in 3rd Task**

## 10) Create CICD pipeline for Nodejs Application.

[https://github.com/betawins/Trading-UI.git](https://github.com/betawins/Trading-UI.git)

--create job


Not secure http://3.133.120.119:8080/view/all/newJob
Jenkins
admin
log out
Dashboard > All > New Item

New Item

Enter an item name

Declarative-job-trade

Select an item type

**Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

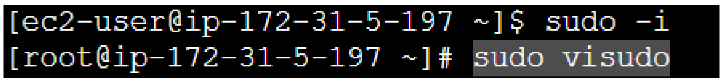
**Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

--before build the job do this in Jenkins server

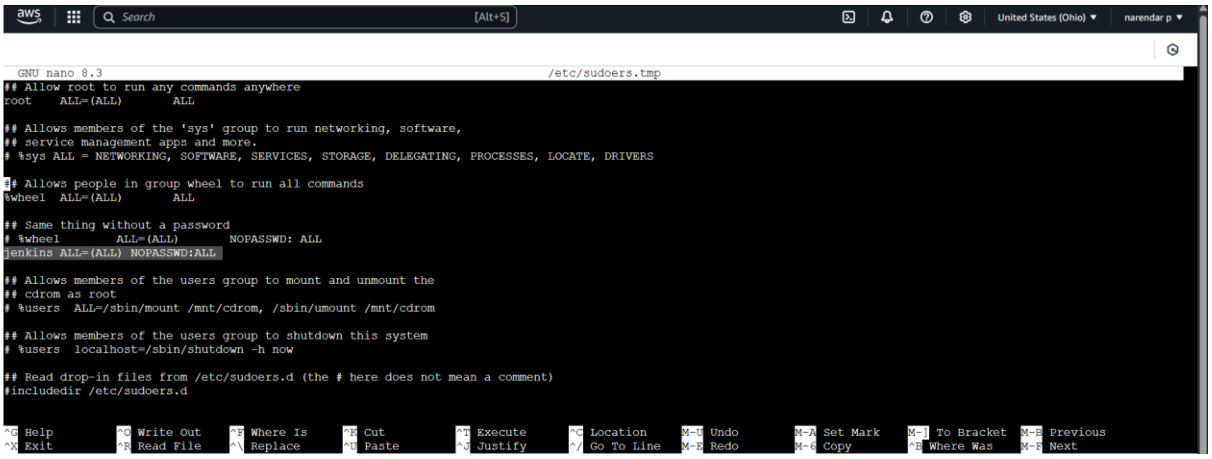
-do this visudo



```
[ec2-user@ip-172-31-5-197 ~]$ sudo -i
[root@ip-172-31-5-197 ~]# sudo visudo
```

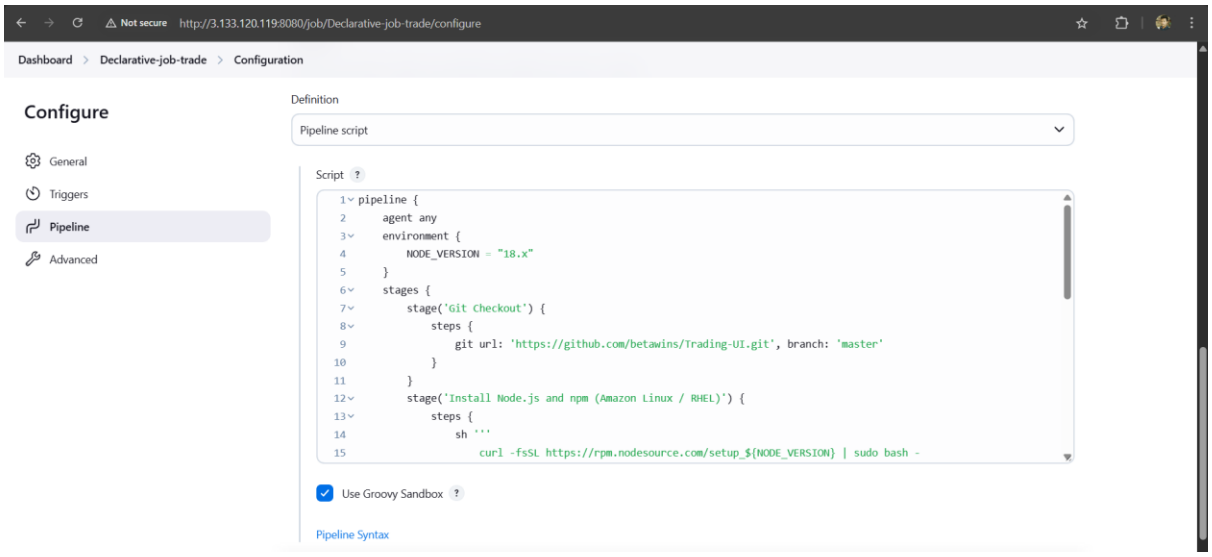
-then add this

jenkins ALL=(ALL) NOPASSWD:ALL



```
  GNU nano 8.3                                                                  /etc/sudoers.tmp
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
jenkins ALL=(ALL) NOPASSWD:ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-B Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was  M-F Next
```

## --pipeline

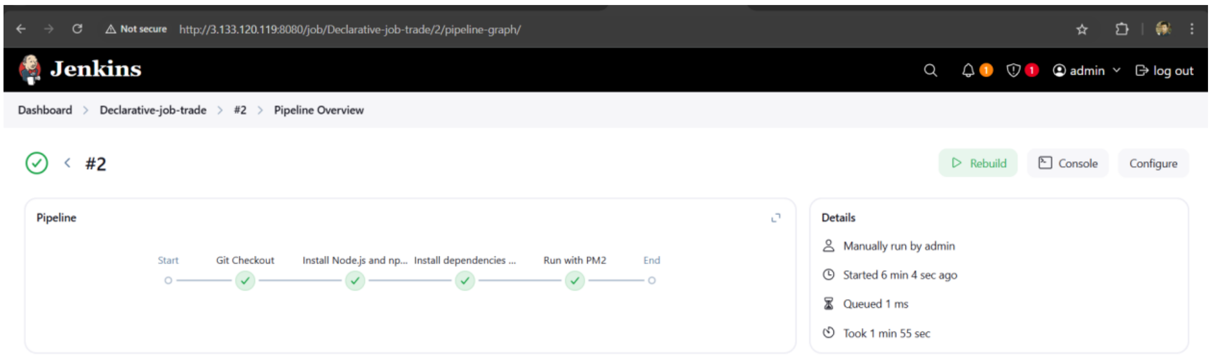


```
pipeline {
    agent any
    environment {
        NODE_VERSION = "18.x"
    }
    stages {
        stage('Git Checkout') {
            steps {
                git url: 'https://github.com/betawins/Trading-UI.git', branch: 'master'
            }
        }
        stage('Install Node.js and npm (Amazon Linux / RHEL)') {
            steps {
                sh '''
                    curl -fsSL https://rpm.nodesource.com/setup_${NODE_VERSION} | sudo bash -
```

```
pipeline {
  agent any
  environment {
    NODE_VERSION = "18.x"
  }
  stages {
    stage('Git Checkout') {
      steps {
        git url: 'https://github.com/betawins/Trading-UI.git', branch: 'master'
      }
    }
    stage('Install Node.js and npm (Amazon Linux / RHEL)') {
      steps {
        sh '''
          curl -fsSL https://rpm.nodesource.com/setup_${NODE_VERSION} | sudo bash -
          sudo yum install -y nodejs
        '''
      }
    }
```

```
        stage('Install dependencies & build') {
            steps {
                sh 'npm install'
            }
        }
        stage('Run with PM2') {
            steps {
                sh '''
                    sudo npm install -g pm2
                    pm2 start app.js || true
                '''
            }
        }
    }
    post {
        failure {
            echo ":x: Pipeline failed. Please check the logs above."
        }
    }
}
```

**-- execution**



Not secure http://3.133.120.119:8080/job/Declarative-job-trade/2/pipeline-graph/

Jenkins

admin log out

Dashboard > Declarative-job-trade > #2 > Pipeline Overview

#2

Rebuild Console Configure

Pipeline

Start — Git Checkout — Install Node.js and np... — Install dependencies ... — Run with PM2 — End

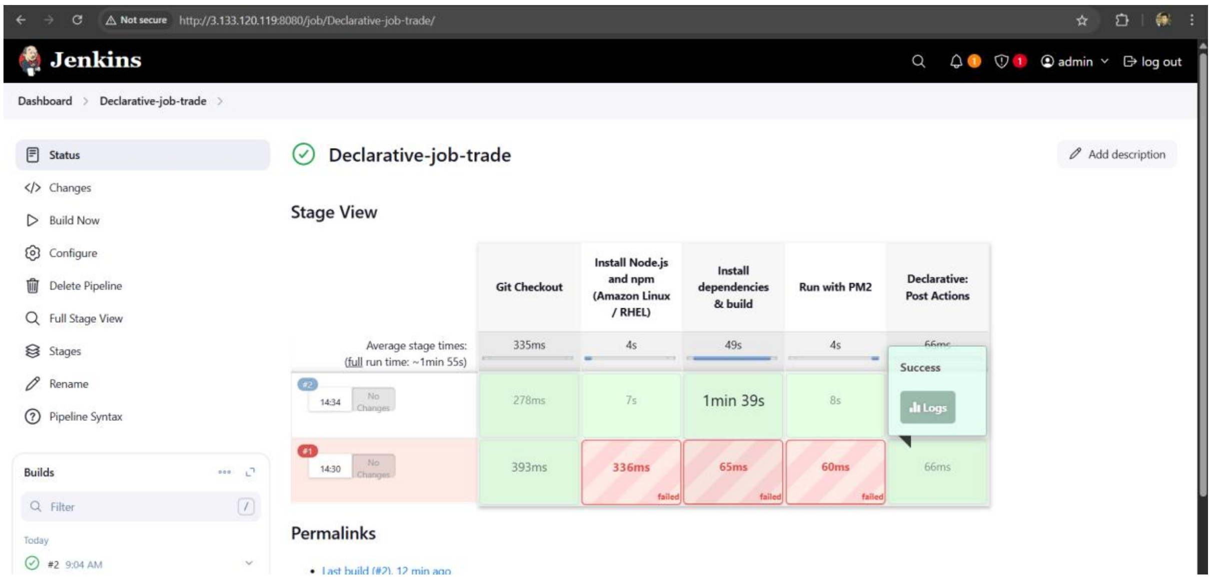
Details

Manually run by admin

Started 6 min 4 sec ago

Queued 1 ms

Took 1 min 55 sec



Declarative-job-trade

Stage View

| | Git Checkout | Install Node.js and npm (Amazon Linux / RHEL) | Install dependencies & build | Run with PM2 | Declarative: Post Actions |
|---|---|---|---|---|---|
| Average stage times: (full run time: ~1min 55s) | 335ms | 4s | 49s | 4s | 66ms |
| #2 14:34 No Changes | 278ms | 7s | 1min 39s | 8s | Success |
| #1 14:30 No Changes | 393ms | 336ms failed | 65ms failed | 60ms failed | 66ms |

Permalinks

- Last build (#2), 12 min ago

## 11) Explain 10 Maven commands.

### 1. mvn clean

- **Purpose**: Deletes the target directory (where Maven builds the project).
- **Use Case**: Clean up compiled files before rebuilding.

### 2. mvn compile

- **Purpose**: Compiles the source code of the project.
- **Use Case**: When you want to check if code compiles without packaging.

### 3. mvn test

- **Purpose**: Runs unit tests using a testing framework (like JUnit).
- **Use Case**: Ensures your application logic is working as expected.

### 4. mvn package

- **Purpose**: Compiles, tests, and packages the code into a .jar or .war.
- **Use Case**: Used when preparing a distributable artifact.

### 5. mvn install

- **Purpose**: Installs the built .jar/.war into the local Maven repository (~/.m2).
- **Use Case**: Makes the artifact available for other local projects.

### 6. mvn deploy

- **Purpose**: Uploads the artifact to a remote repository (like Nexus).
- **Use Case**: Used in CI/CD pipelines for deploying build outputs.

### 7. mvn validate

- **Purpose**: Validates if the project is correct and all necessary information is available.
- **Use Case**: Used early in the build process for project sanity checks.

### 8. mvn site

- **Purpose**: Generates a site or documentation for the project.
- **Use Case**: Used to produce project reports, javadocs, and metrics.

### 9. mvn dependency:tree

- **Purpose**: Displays the dependency tree of the project.
- **Use Case**: Debugging and analyzing transitive dependencies.

### 10. mvn versions:display-dependency-updates

- **Purpose**: Shows newer versions of project dependencies.
- **Use Case**: Helps in updating dependencies to the latest versions.