SOFTWARE ENGINEERING

UNIT – 3

CHAPTER – 6

BUILDING JENKINS PIPELINE

I. Definition of Jenkins Pipeline

Jenkins Pipeline (or simply Pipeline with a capital P) plugins supports implementing and integrating continuous delivery pipelines into Jenkins. Jenkins Pipeline is a suite of plugins for the Jenkins automation server that allows you to define and manage your software delivery pipeline as code. Instead of using the traditional graphical user interface (GUI) to create and configure jobs in Jenkins, you can use a Jenkinsfile to describe your build, test, and deployment process in a script format.

A Jenkinsfile is a text file that defines the steps of your pipeline, including build, test, and deployment phases. It can be written in either Declarative Pipeline Syntax or Scripted Pipeline Syntax. Declarative syntax is a more structured and opinionated way to define pipelines, while scripted syntax provides more flexibility and is based on a Groovy-based domain-specific language.

Approaches to Defining Pipeline Script

In Jenkins, there are two main approaches to defining pipeline scripts: Declarative Pipeline Syntax and Scripted Pipeline Syntax. Each approach has its own style and use cases.

1. Declarative Pipeline Syntax:

Declarative Pipeline Syntax provides a more structured and simplified way of defining pipelines. It is designed to be easy to read and write, making it accessible to users with less scripting experience. It uses a predefined structure with specified sections for defining the pipeline stages, steps, and other configurations.

Example of a Declarative Pipeline:

```
pipeline {
    agent any

    stages {

        stage('Build') {

                steps {

                    // Build steps go here

                    }

                }
        stage('Test') {
                steps {
                    // Test steps go here
                    }
                }
        stage('Deploy') {
                steps {
                    // Deployment steps go here
                    }
                }
        }
    }
```

<u>2. Scripted Pipeline Syntax:</u>

Scripted Pipeline Syntax, on the other hand, is more flexible and allows you to write pipelines using a Groovy-based scripting language. It is suitable for users who are comfortable with programming and provides greater control over the flow of the pipeline.

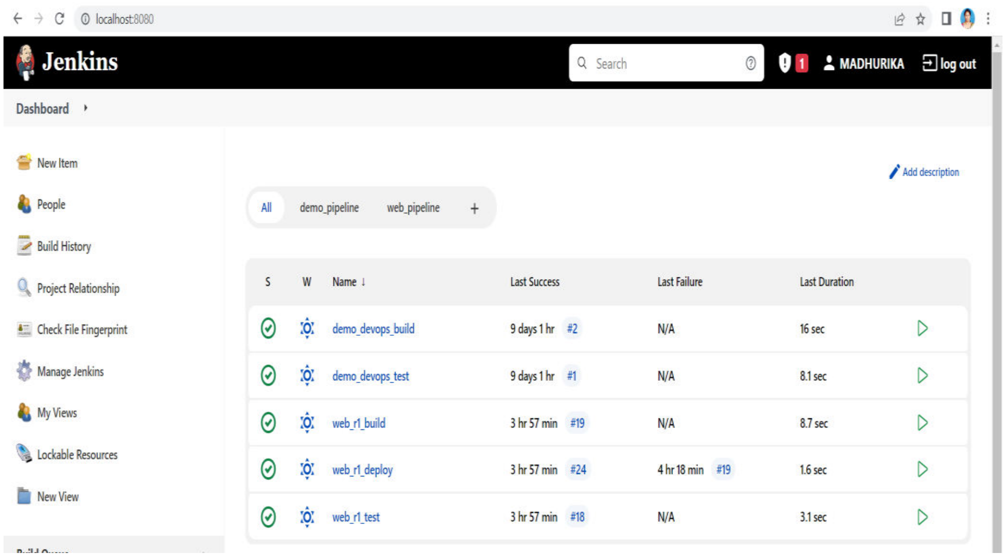
Example of a Scripted Pipeline:

```
node {

    stage('Build') {

            // Build steps go here

            }

    stage('Test') {

            // Test steps go here

            }
```

```
        stage('Deploy') {

                        // Deployment steps go here

                         }

        }
```

II. Creating a Simple Pipeline using user interface for Java project without Jenkins Script.

1. Open Jenkins in local host:8080 and create a new item



localhost:8080

Jenkins | Search | MADHURIKA | log out

Dashboard

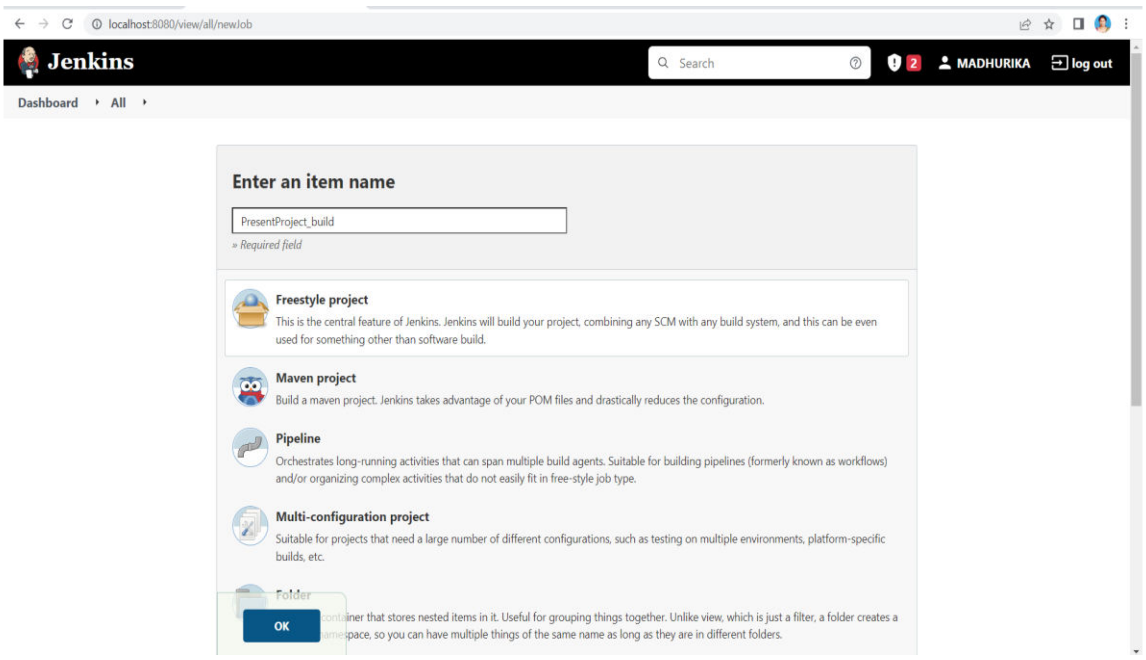
- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Add description

All | demo_pipeline | web_pipeline | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☼	demo_devops_build	9 days 1 hr #2	N/A	16 sec
✓	☼	demo_devops_test	9 days 1 hr #1	N/A	8.1 sec
✓	☼	web_r1_build	3 hr 57 min #19	N/A	8.7 sec
✓	☼	web_r1_deploy	3 hr 57 min #24	4 hr 18 min #19	1.6 sec
✓	☼	web_r1_test	3 hr 57 min #18	N/A	3.1 sec

2. Select a new Freestyle Project give name (eg. PresentProject_build) and then click ok


localhost:8080/view/all/newJob

Jenkins | Search | 2 | MADHURIKA | log out

Dashboard › All ›

Enter an item name

PresentProject_build

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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

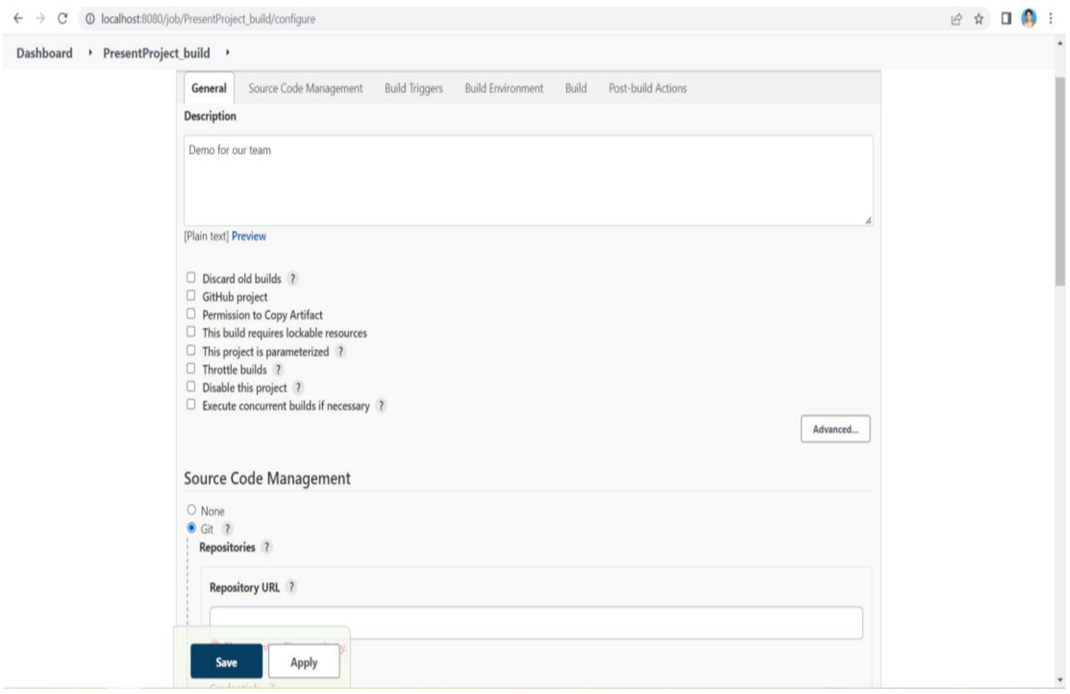
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

Folder
container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

3. In description type e.g., Build demo
4. Give the Git repository URL of the project to be built


localhost:8080/job/PresentProject_build/configure

Dashboard › PresentProject_build ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Description

Demo for our team

[Plain text] Preview

- ☐ Discard old builds
- ☐ GitHub project
- ☐ Permission to Copy Artifact
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

Advanced...

Source Code Management

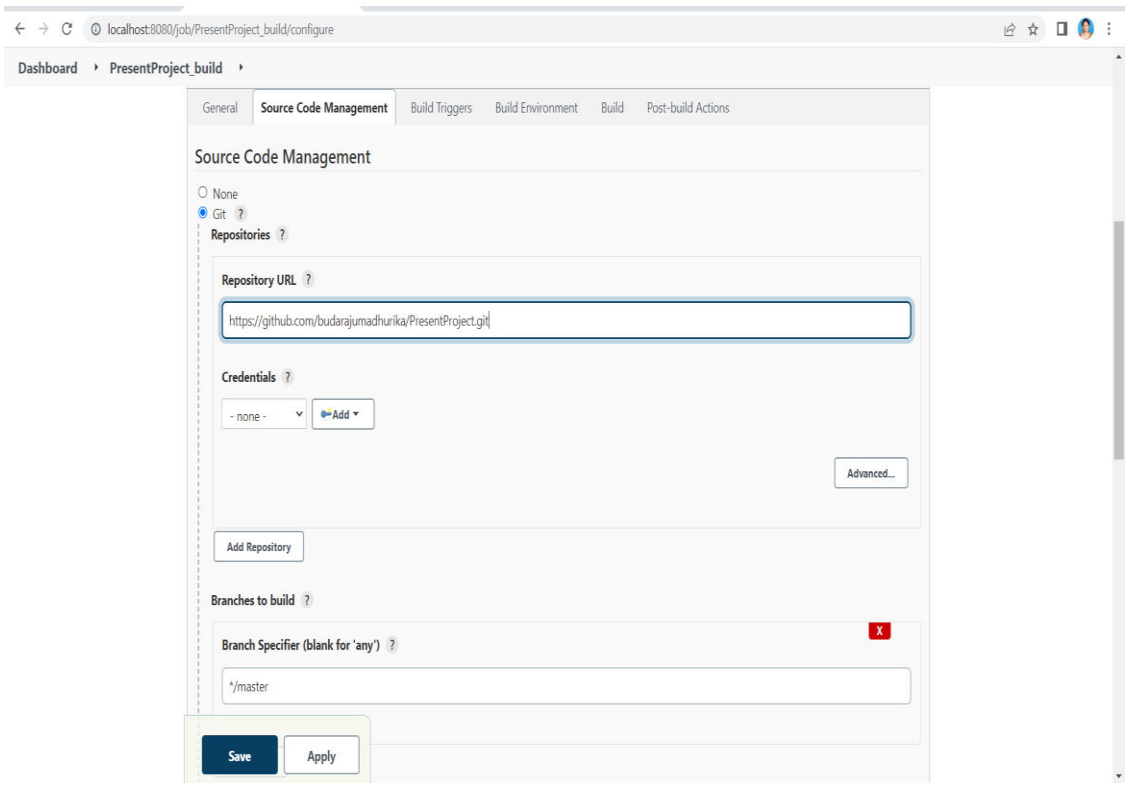
- ○ None
- ● Git

Repositories

Repository URL

Save | Apply

5. After adding the repository URL, Specify the Branch as either Master/Main as it is in the GitHub (eg. My project is in master branch so I have mentioned */master)


localhost:8080/job/PresentProject_build/configure

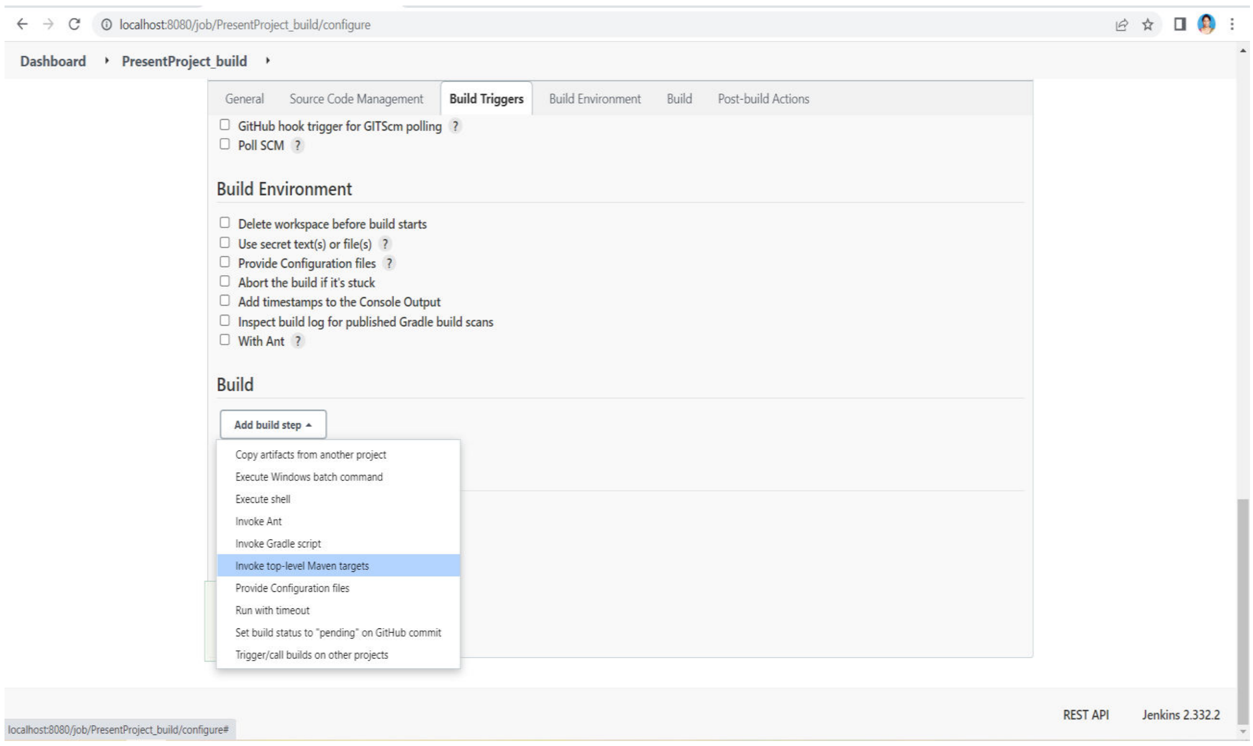
Dashboard › PresentProject_build ›

General | **Source Code Management** | Build Triggers | Build Environment | Build | Post-build Actions

Source Code Management

○ None
● Git ?
Repositories ?
Repository URL ?
https://github.com/budarajumadhurika/PresentProject.git
Credentials ?
- none - | Add
Advanced...
Add Repository
Branches to build ?
Branch Specifier (blank for 'any') ?
*/master
Save | Apply

6. Now in Build, select the "Invoke top-level Maven targets"


localhost:8080/job/PresentProject_build/configure

Dashboard › PresentProject_build ›

General | Source Code Management | **Build Triggers** | Build Environment | Build | Post-build Actions

☐ GitHub hook trigger for GITScm polling ?
☐ Poll SCM ?

Build Environment

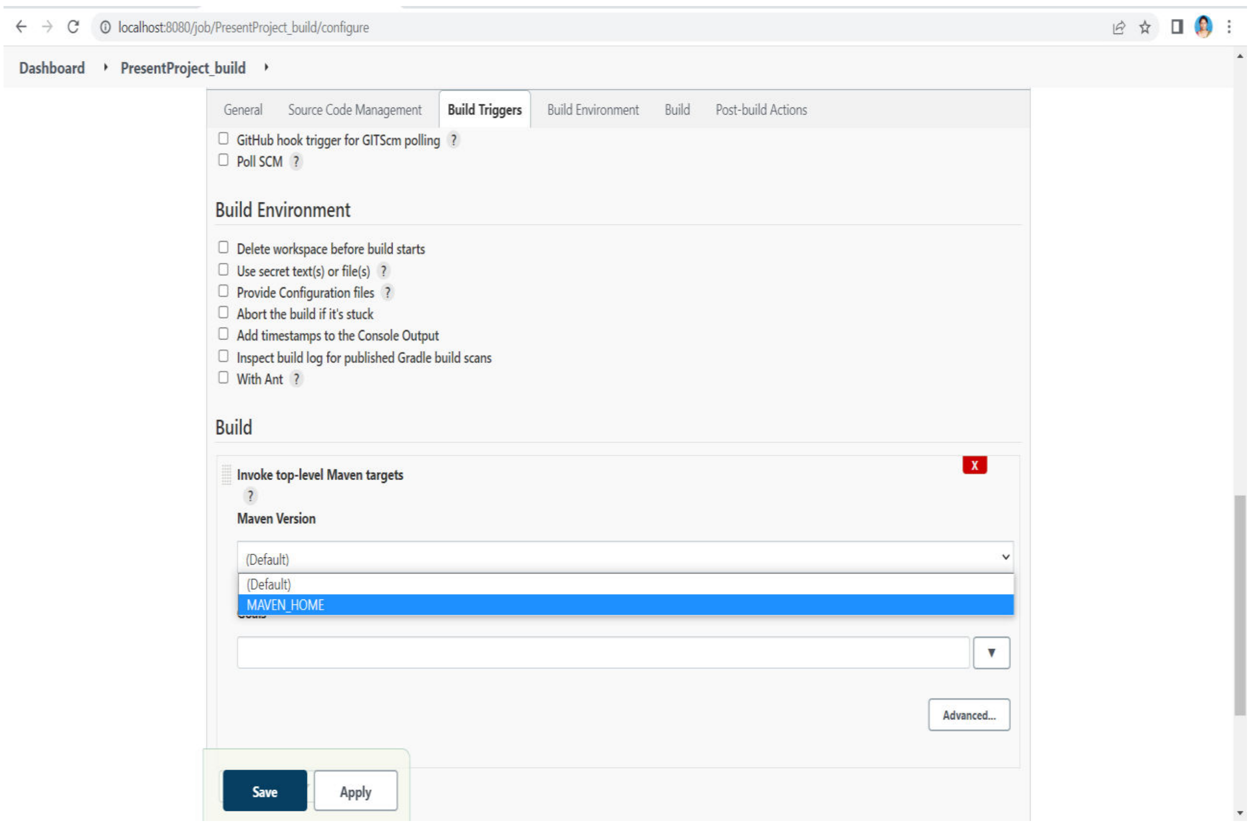
☐ Delete workspace before build starts
☐ Use secret text(s) or file(s) ?
☐ Provide Configuration files ?
☐ Abort the build if it's stuck
☐ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant ?

Build

Add build step ▴
- Copy artifacts from another project
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Provide Configuration files
- Run with timeout
- Set build status to "pending" on GitHub commit
- Trigger/call builds on other projects

localhost:8080/job/PresentProject_build/configure#
REST API Jenkins 2.332.2

7. Select the Maven path which is already set in the global credentials in Manage Jenkins (eg MAVEN_HOME)


localhost:8080/job/PresentProject_build/configure
Dashboard › PresentProject_build ›
General Source Code Management Build Triggers Build Environment Build Post-build Actions

GitHub hook trigger for GITScm polling
Poll SCM

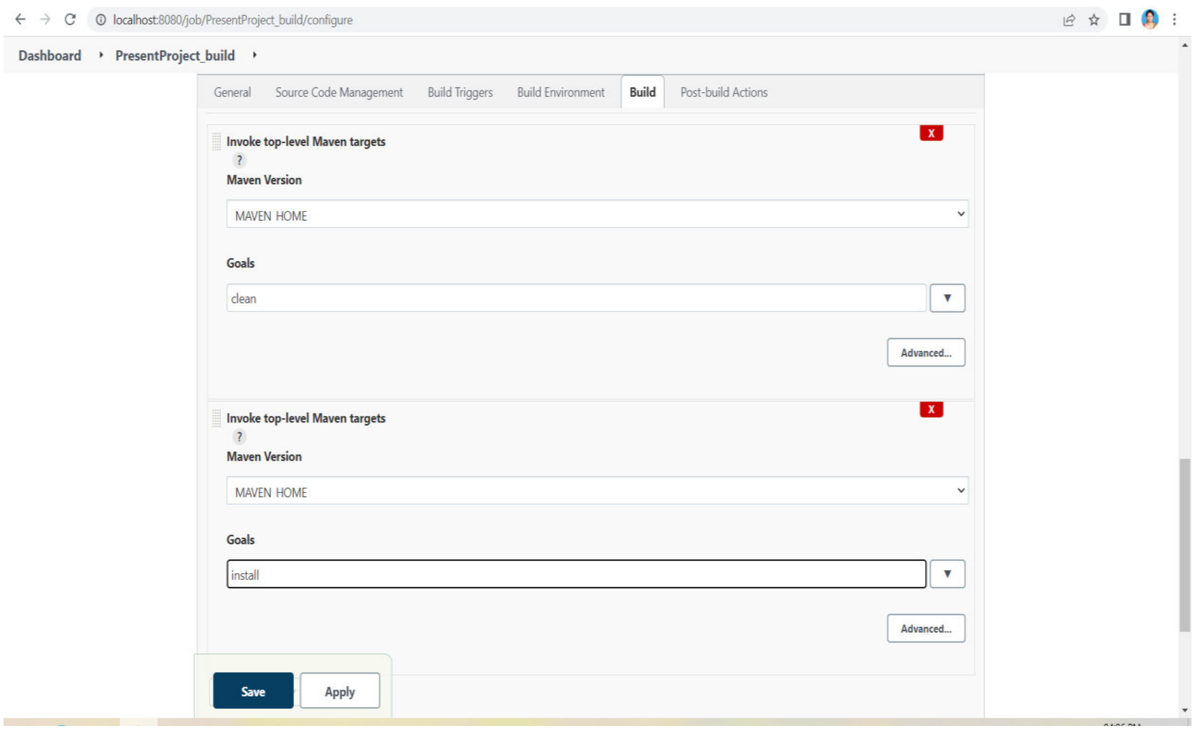
Build Environment

Delete workspace before build starts
Use secret text(s) or file(s)
Provide Configuration files
Abort the build if it's stuck
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
With Ant

Build

Invoke top-level Maven targets
Maven Version
(Default)
(Default)
MAVEN_HOME

8. Set Goals field to clean as done in eclipse


localhost:8080/job/PresentProject_build/configure
Dashboard › PresentProject_build ›
General Source Code Management Build Triggers Build Environment Build Post-build Actions

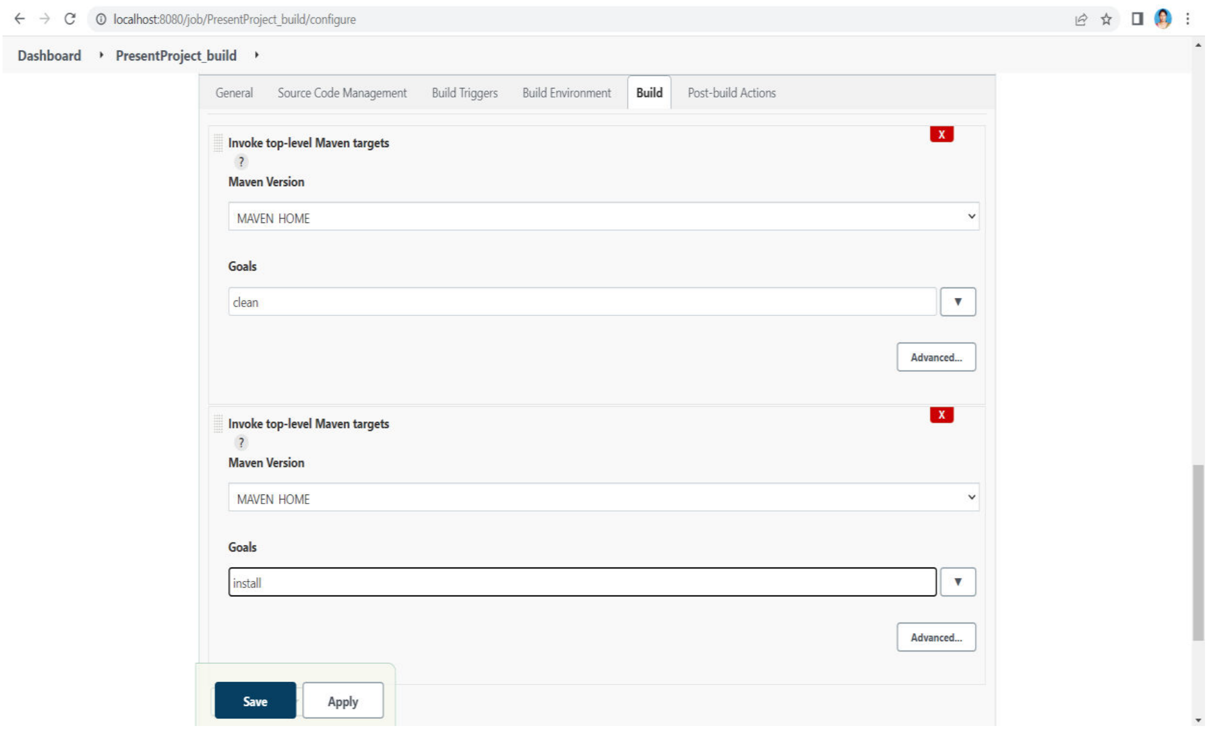
Invoke top-level Maven targets
Maven Version
MAVEN HOME
Goals
clean

Invoke top-level Maven targets
Maven Version
MAVEN HOME
Goals
install

Again, in Build, select the “Invoke top-level Maven targets”

9. Select the Maven path which is already set in the global credentials in Manage Jenkins (eg MAVEN_HOME)

10. Set Goals field to **Install** as done in eclipse


localhost:8080/job/PresentProject_build/configure

Dashboard › PresentProject_build ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Invoke top-level Maven targets

Maven Version: MAVEN HOME

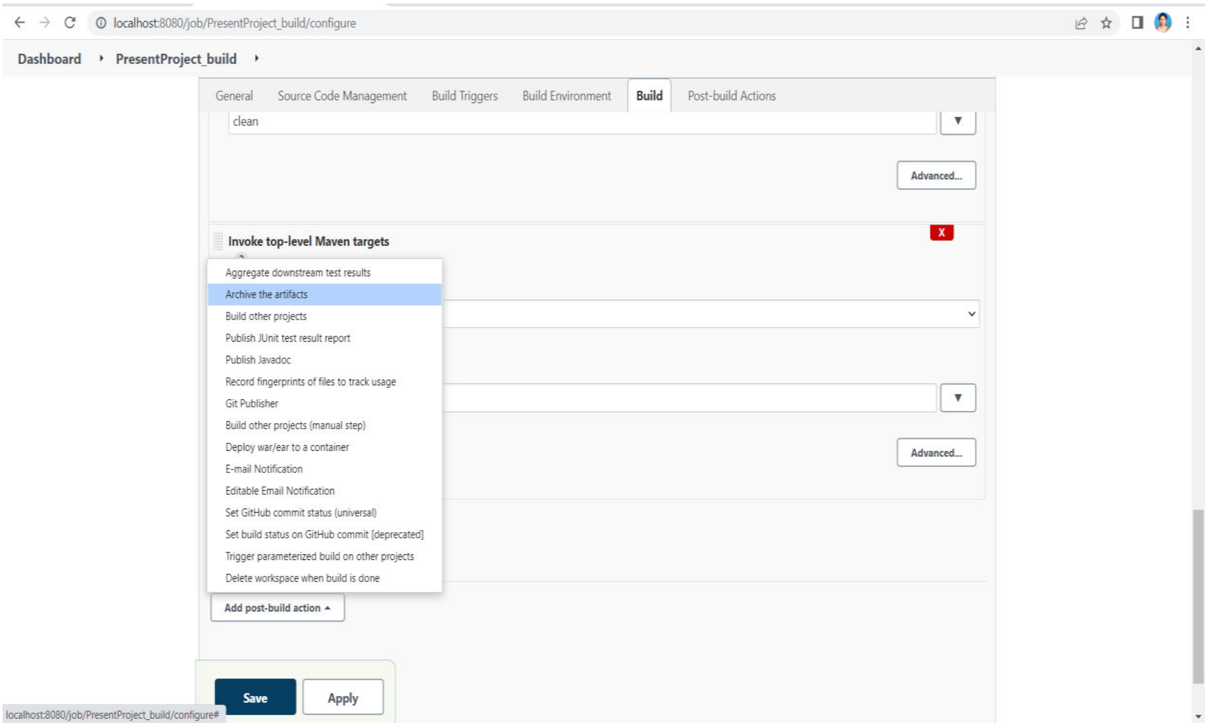
Goals: clean

Invoke top-level Maven targets

Maven Version: MAVEN HOME

Goals: install

11. Now in post build actions-> select "**Archive the artifacts**", to send the output of build project to the testing team


localhost:8080/job/PresentProject_build/configure

Dashboard › PresentProject_build ›

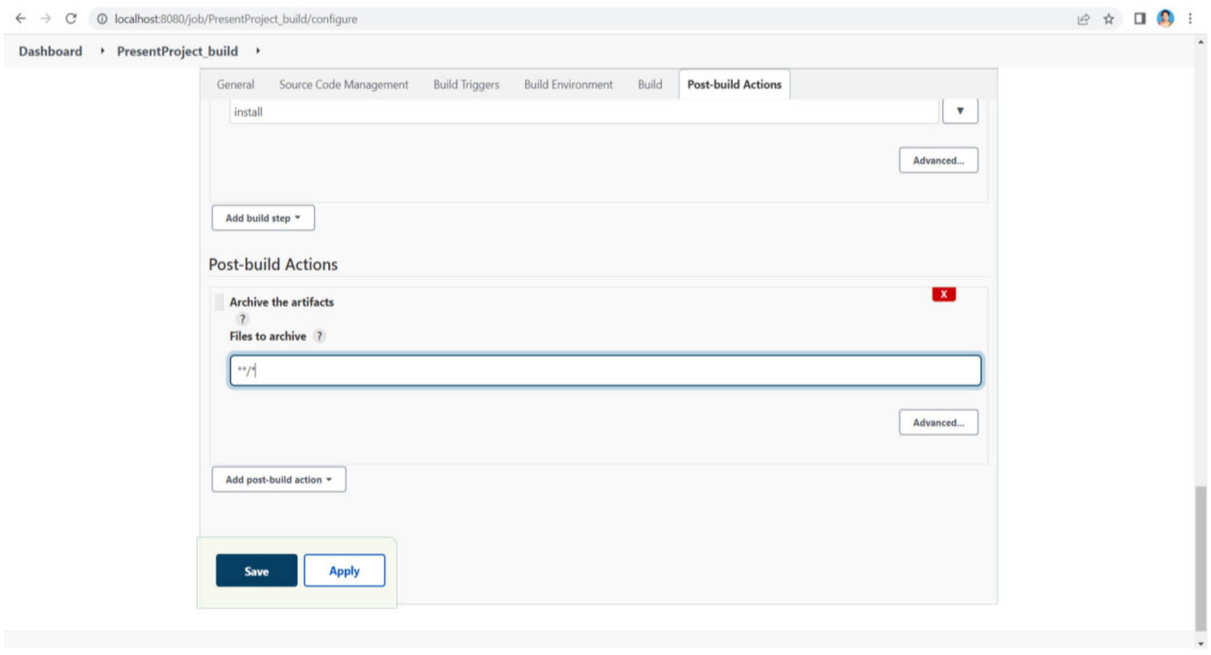
General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

clean

Invoke top-level Maven targets

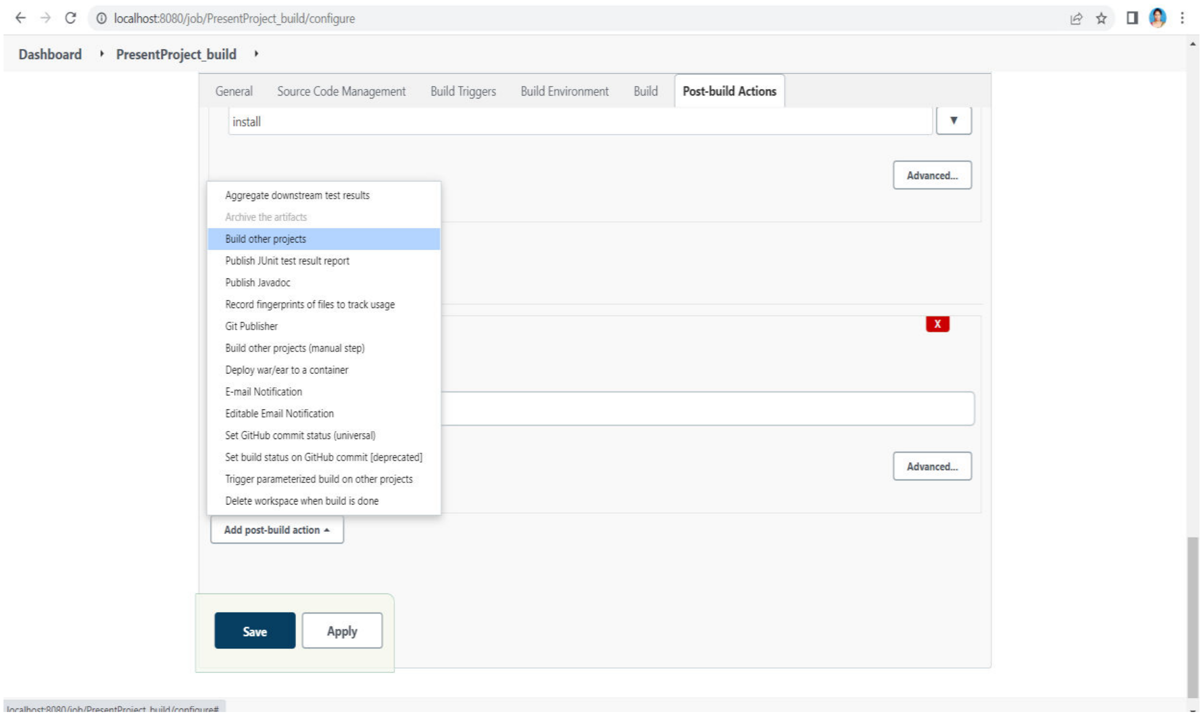
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Git Publisher
Build other projects (manual step)
Deploy war/ear to a container
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Trigger parameterized build on other projects
Delete workspace when build is done
Add post-build action
Advanced...
Save
Apply
localhost:8080/job/PresentProject_build/configure#

12. If we want to archive all the artifacts type **/* in Files to Archive

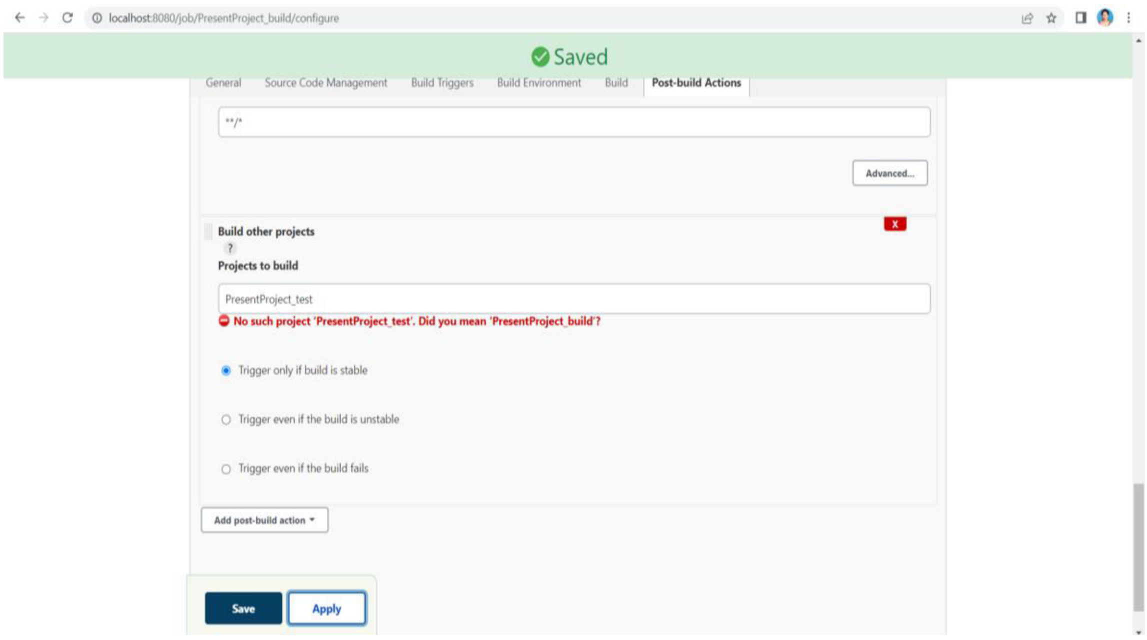


13. Now the next step is to build other projects, where we will create a test project which will be triggered by the build project.

For this **in <u>Add Post build Action</u> select "Build other projects"**


localhost:8080/job/PresentProject_build/configure
Dashboard › PresentProject_build ›
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
install
Advanced...
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Git Publisher
Build other projects (manual step)
Deploy war/ear to a container
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Trigger parameterized build on other projects
Delete workspace when build is done
Advanced...
Add post-build action
Save
Apply

14. In Projects to build enter the next project name as "**PresentProject_test**"
15. Next press on Apply and Save


localhost:8080/job/PresentProject_build/configure

Saved

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

**/*

Advanced...

Build other projects

Projects to build

PresentProject_test

No such project 'PresentProject_test'. Did you mean 'PresentProject_build'?

(●) Trigger only if build is stable

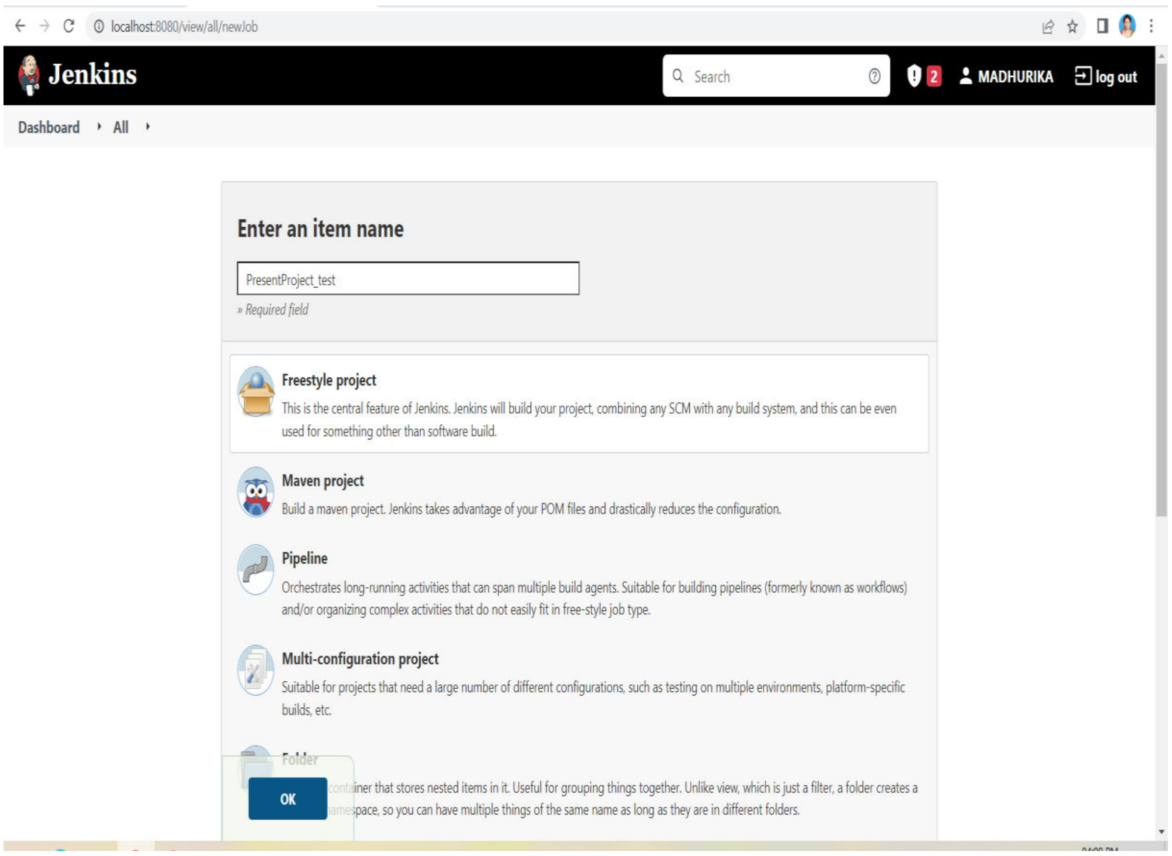
( ) Trigger even if the build is unstable

( ) Trigger even if the build fails

Add post-build action

Save | Apply

16. Go to dashboard -> New item-> Freestyle Project, and then give next project name as **PresentProject_test**, then press on OK


localhost:8080/view/all/newJob

Jenkins | Search | 2 | MADHURIKA | log out

Dashboard > All >

Enter an item name

PresentProject_test

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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

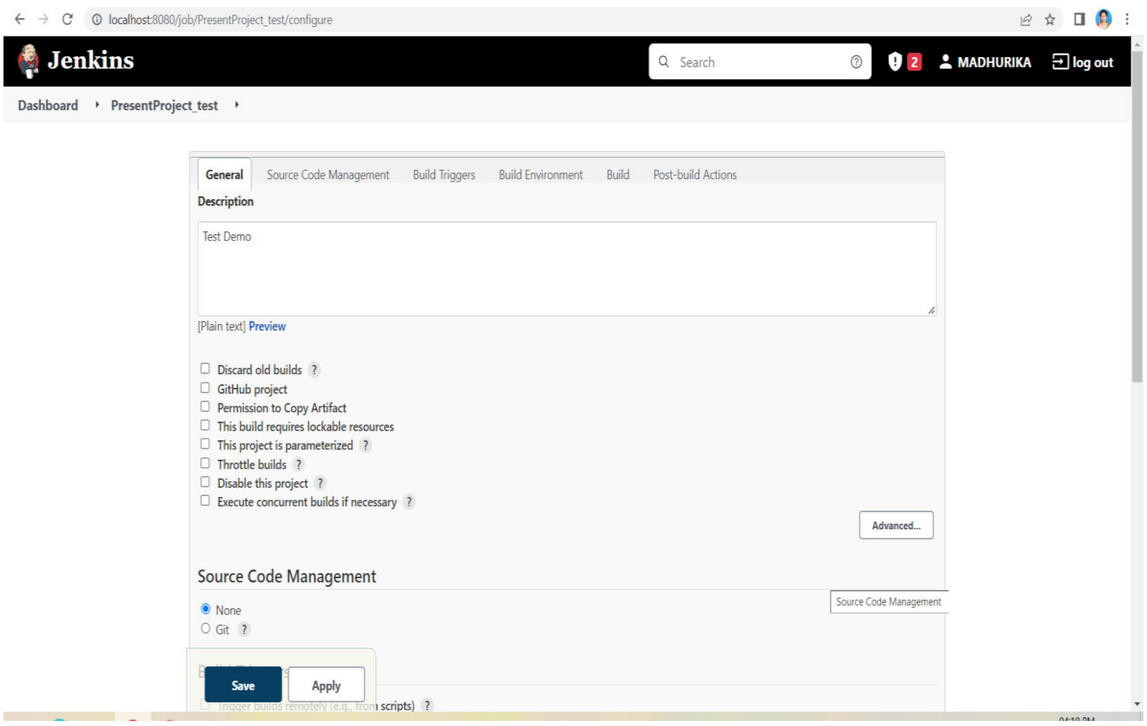
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

Folder
container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

17. In description type eg. Test demo



Description

Test Demo

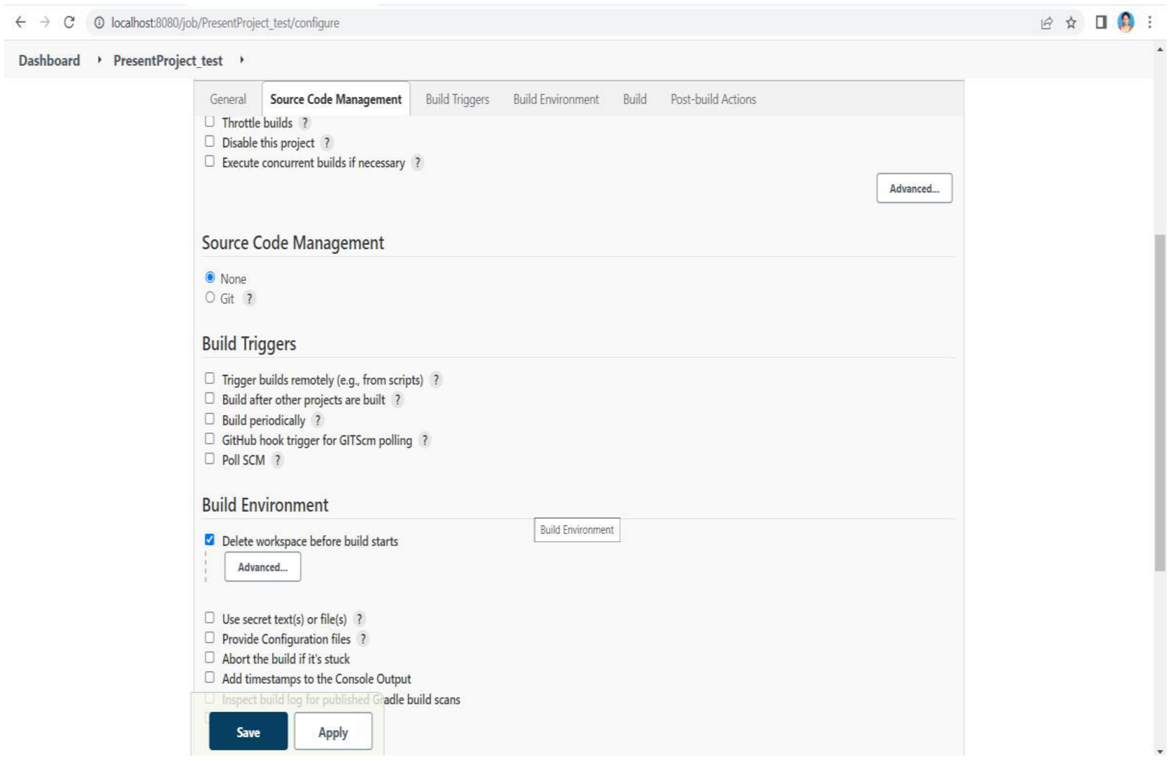
[Plain text] Preview

- ☐ Discard old builds
- ☐ GitHub project
- ☐ Permission to Copy Artifact
- ☐ This build requires lockable resources
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

Source Code Management

- ● None
- ○ Git

18. In Build environment, check the box with name "Delete the workspace before build starts". This is to discard old builds



- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

Source Code Management

- ● None
- ○ Git

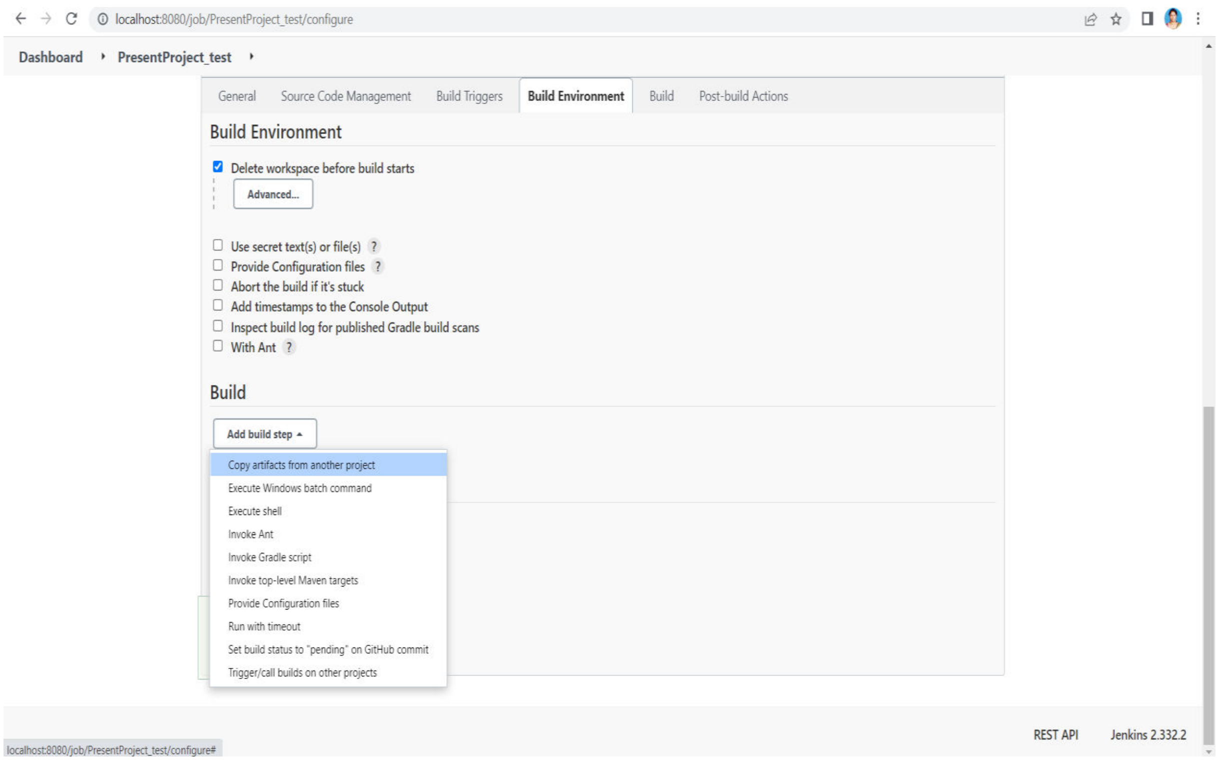
Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

Build Environment

- ☑ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Provide Configuration files
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans

19. In Build select "**copy the artifacts from another project**" to forward the artifacts of the previous project to the current test project i.e., PresentProject_test



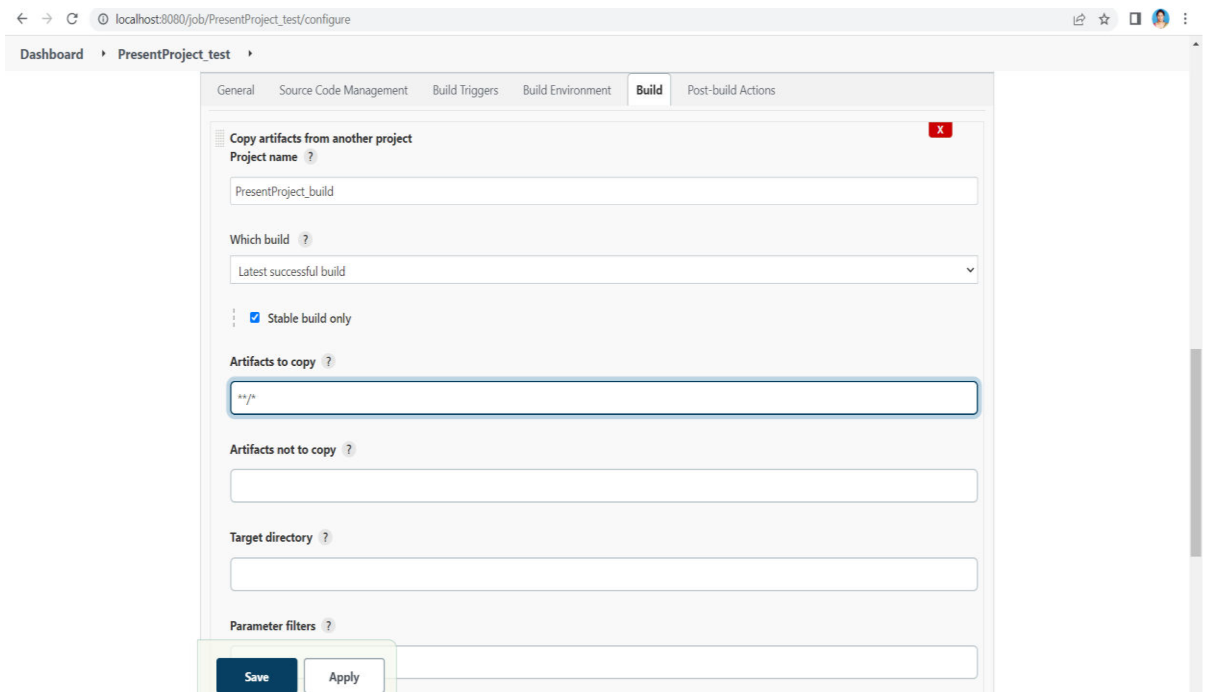
Build Environment

- [✓] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Provide Configuration files
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant

Build

Add build step:
- Copy artifacts from another project
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Provide Configuration files
- Run with timeout
- Set build status to "pending" on GitHub commit
- Trigger/call builds on other projects

20. Give the name of the project from which we want to copy the artifacts (eg. PresentProject_build) and check the box ->stable build only->to copy all the artifacts type **/*



Copy artifacts from another project

Project name: PresentProject_build

Which build: Latest successful build

[✓] Stable build only

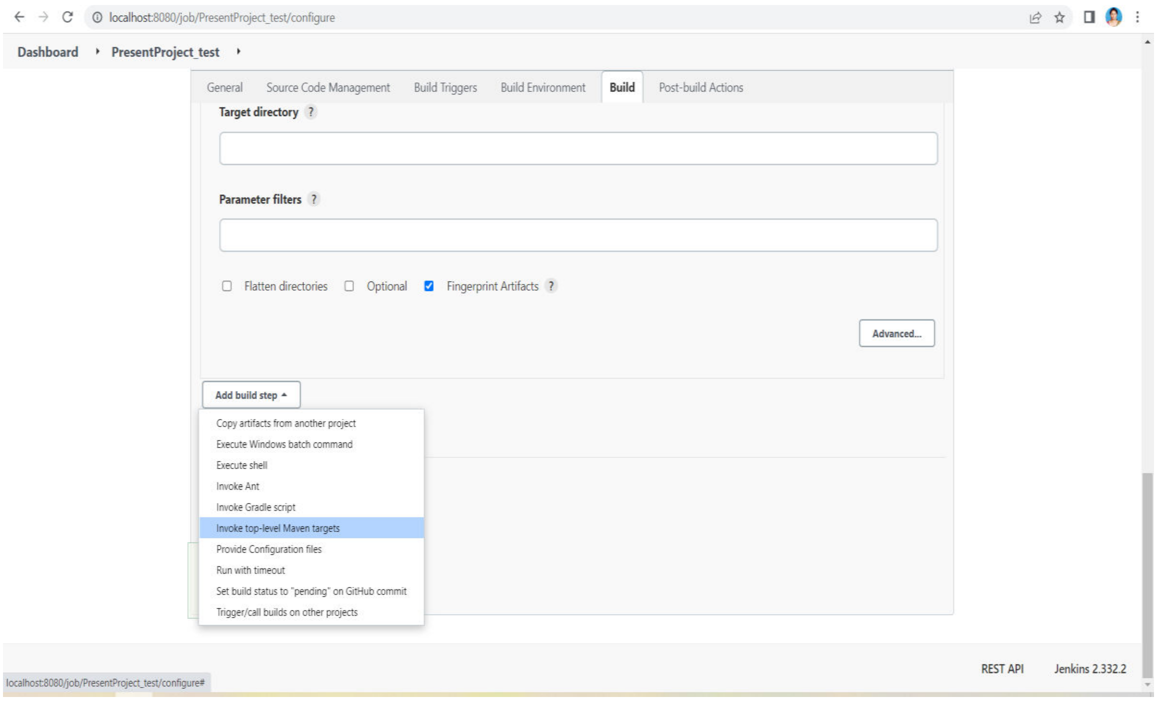
Artifacts to copy: **/*

Artifacts not to copy:

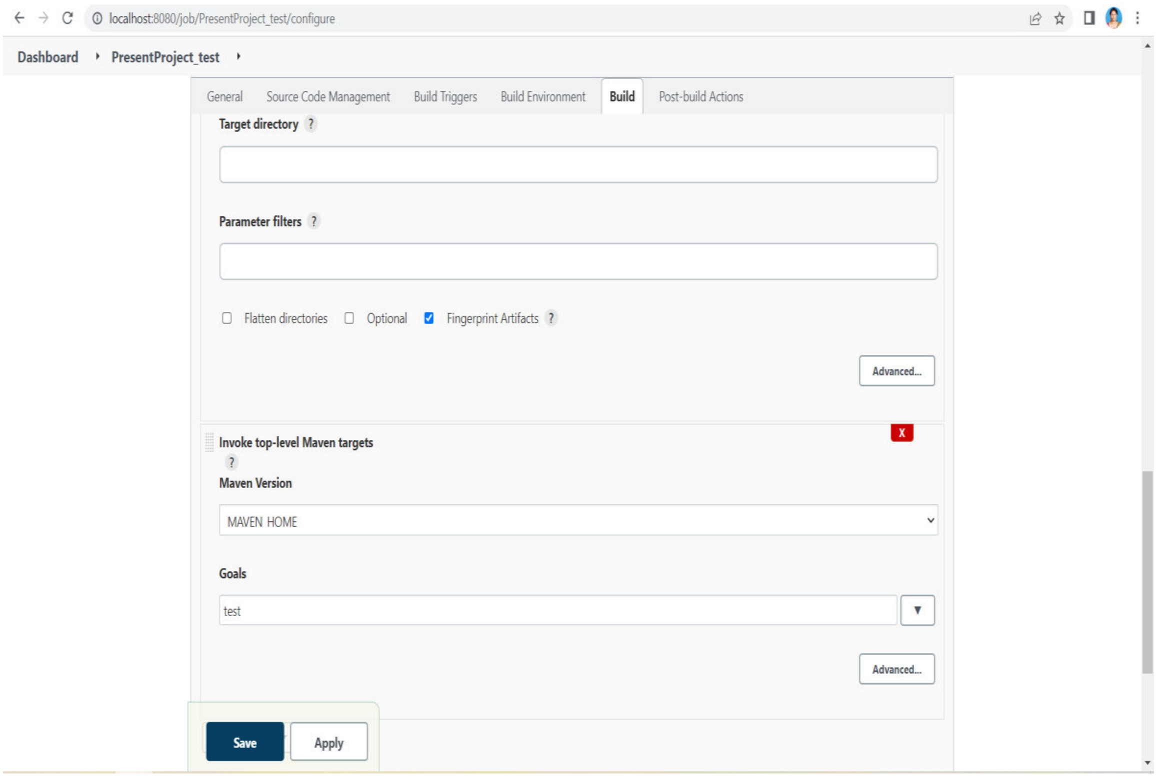
Target directory:

Parameter filters:

21. Now select Invoke top-level Maven targets in build



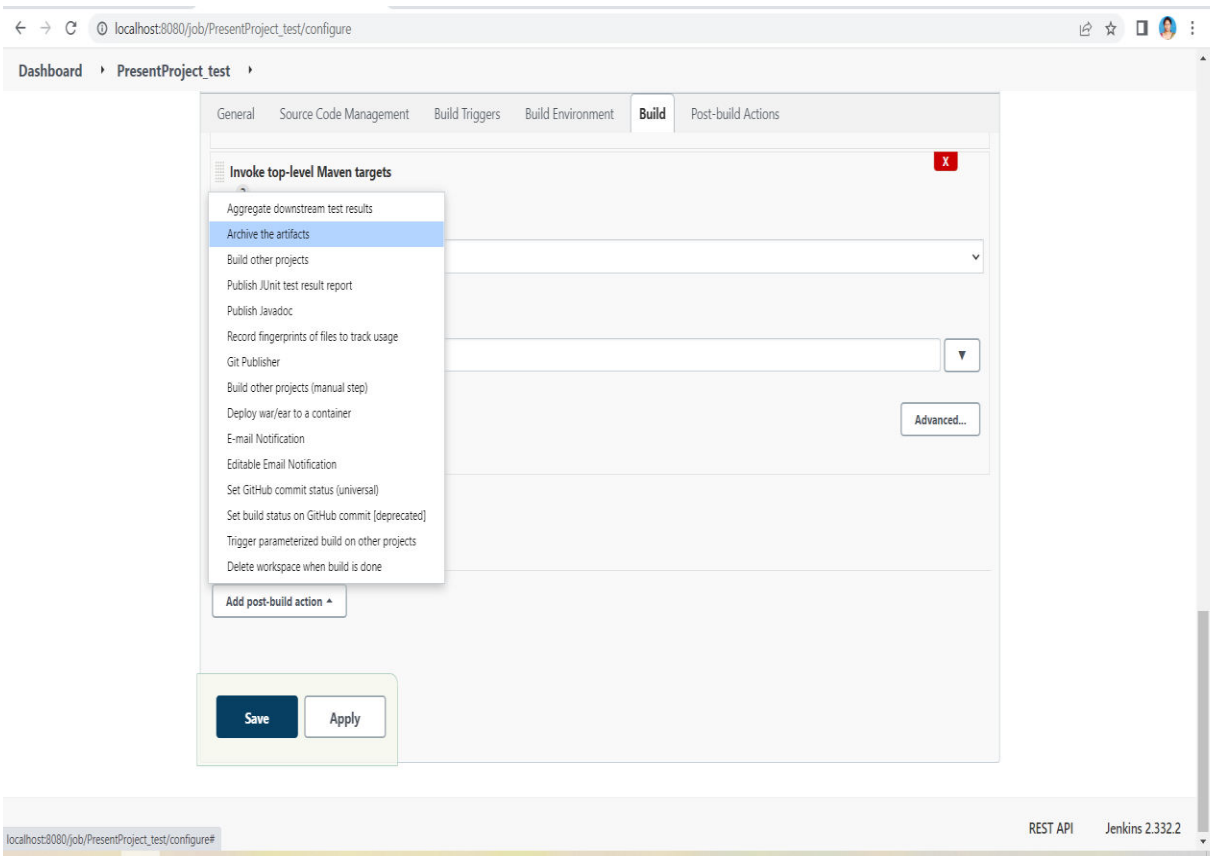
22. This time give the goal as test after selecting the Maven version



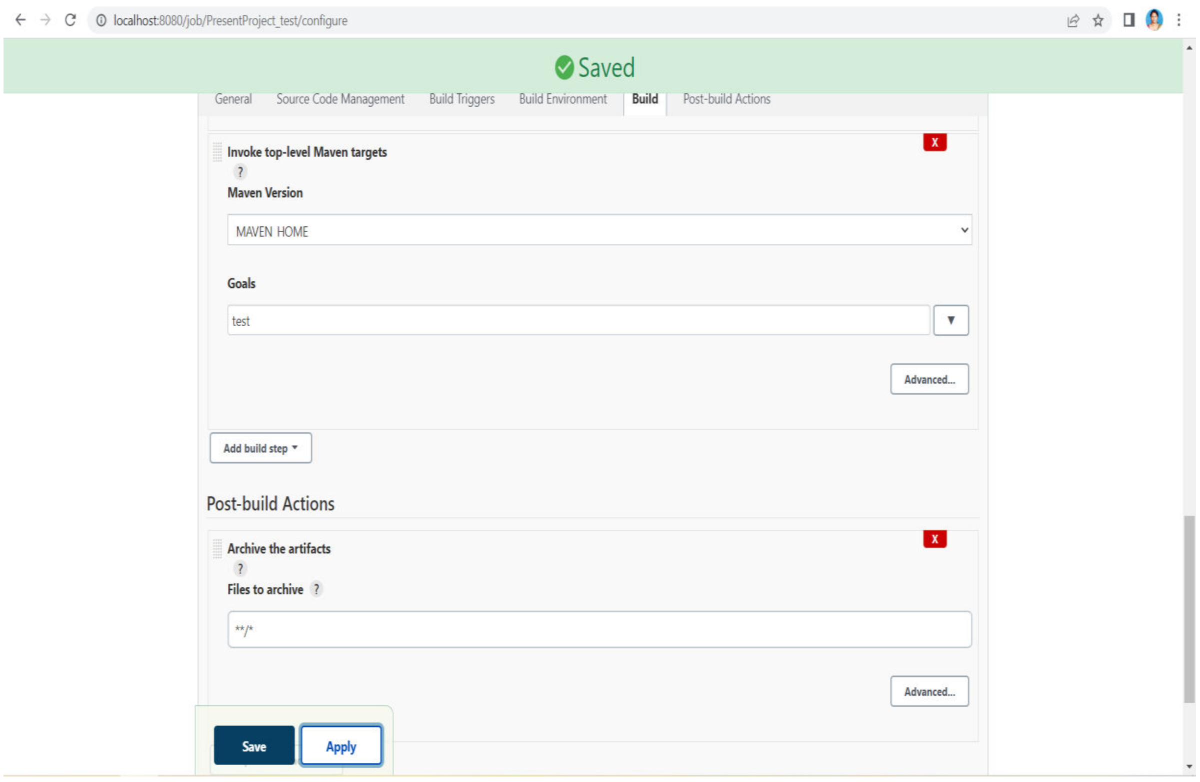
Maven Version: MAVEN HOME

Goals: test

23. In post-build actions->select Archive the artifacts


localhost:8080/job/PresentProject_test/configure
Dashboard ▸ PresentProject_test ▸
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Invoke top-level Maven targets
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Git Publisher
Build other projects (manual step)
Deploy war/ear to a container
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Trigger parameterized build on other projects
Delete workspace when build is done
Advanced...
Add post-build action
Save
Apply
localhost:8080/job/PresentProject_test/configure#
REST API
Jenkins 2.332.2

24. To save all the artifacts->type **/* and Apply->Save

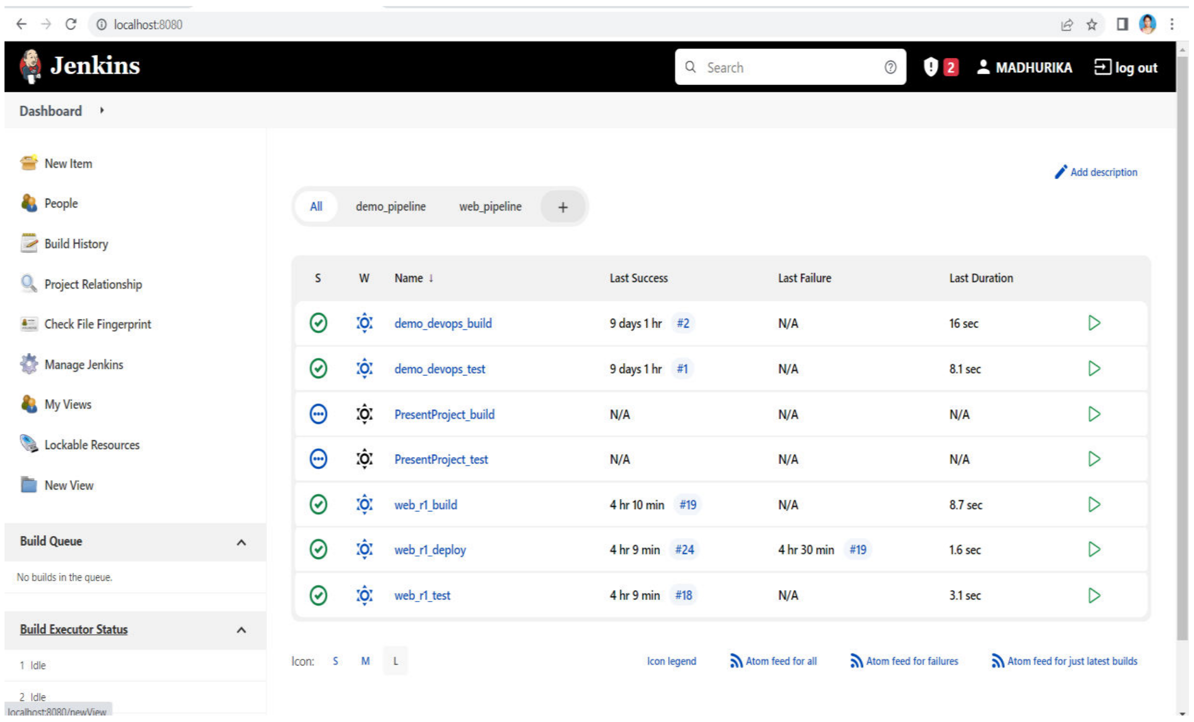


Maven Version: MAVEN HOME

Goals: test

Files to archive: **/*

25. Create a pipeline by clicking on + symbol in the dashboard ->a pipeline is a collection of events or jobs which are interlinked with one another in a sequence

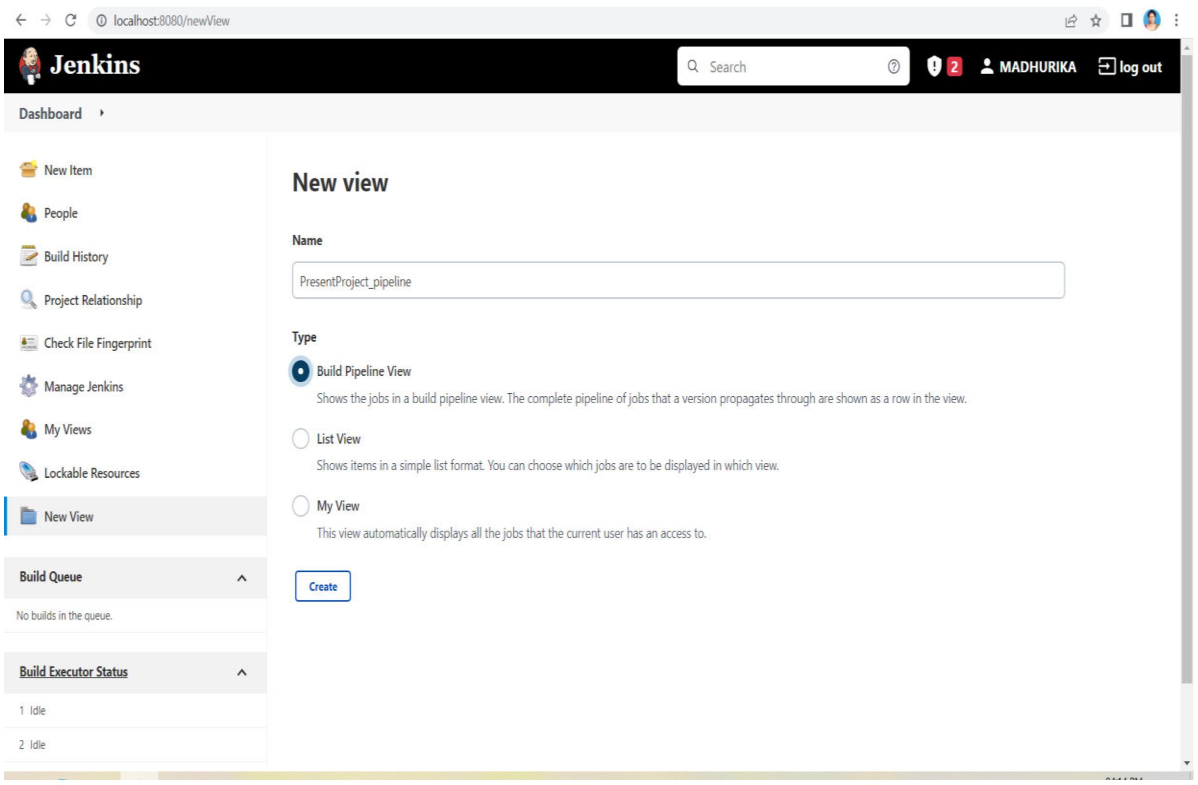


No builds in the queue.

1 Idle
2 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☼	demo_devops_build	9 days 1 hr #2	N/A	16 sec
✓	☼	demo_devops_test	9 days 1 hr #1	N/A	8.1 sec
⊖	☼	PresentProject_build	N/A	N/A	N/A
⊖	☼	PresentProject_test	N/A	N/A	N/A
✓	☼	web_r1_build	4 hr 10 min #19	N/A	8.7 sec
✓	☼	web_r1_deploy	4 hr 9 min #24	4 hr 30 min #19	1.6 sec
✓	☼	web_r1_test	4 hr 9 min #18	N/A	3.1 sec

26. Give a name to the pipeline->select Build Pipeline View->create



New view

Name: PresentProject_pipeline

Type

[✓ Build Pipeline View]
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

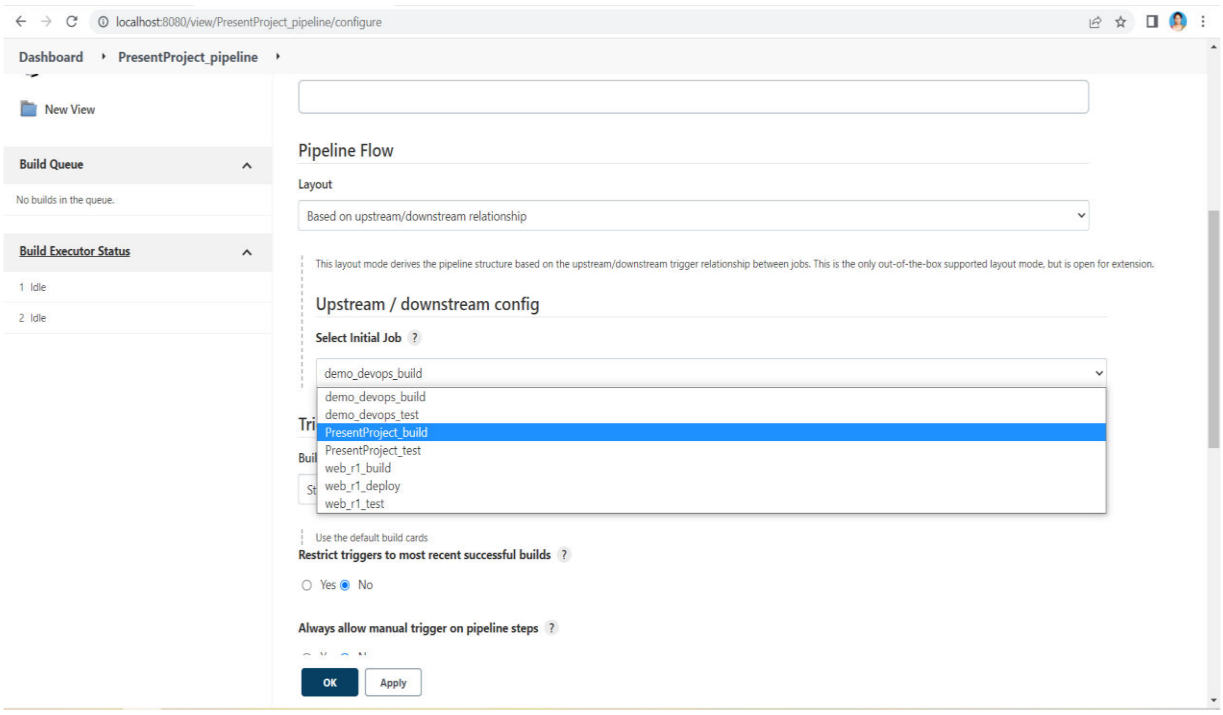
List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

My View
This view automatically displays all the jobs that the current user has an access to.

No builds in the queue.

1 Idle
2 Idle

27. Select the first project to trigger the execution->build project



localhost:8080/view/PresentProject_pipeline/configure

Dashboard ▸ PresentProject_pipeline ▸

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Pipeline Flow

Layout

Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job ?

demo_devops_build

demo_devops_build
demo_devops_test
PresentProject_build
PresentProject_test
web_r1_build
web_r1_deploy
web_r1_test

Use the default build cards

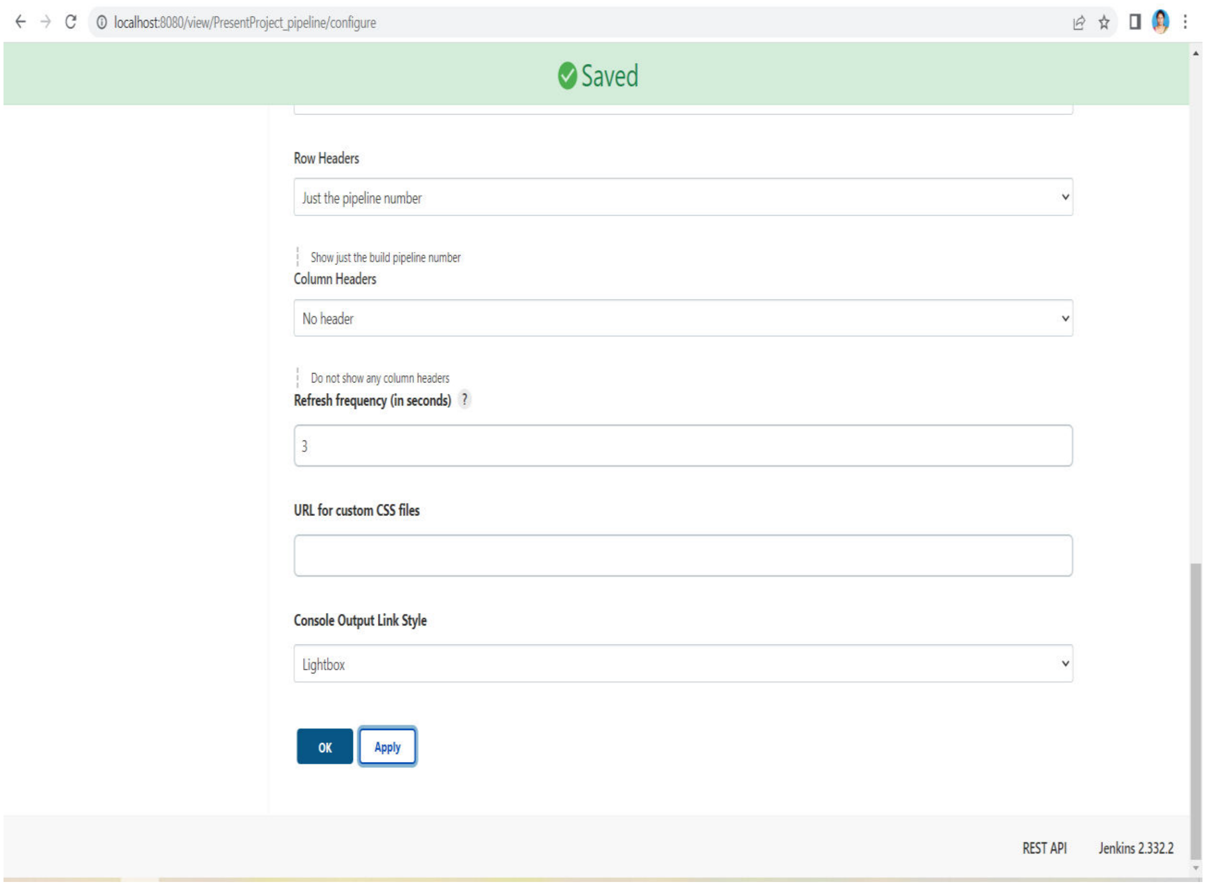
Restrict triggers to most recent successful builds ?

Yes [✓ No]

Always allow manual trigger on pipeline steps ?

OK Apply

28. Press on Apply->Ok



localhost:8080/view/PresentProject_pipeline/configure

Saved

Row Headers

Just the pipeline number

Show just the build pipeline number

Column Headers

No header

Do not show any column headers

Refresh frequency (in seconds) ?

3

URL for custom CSS files

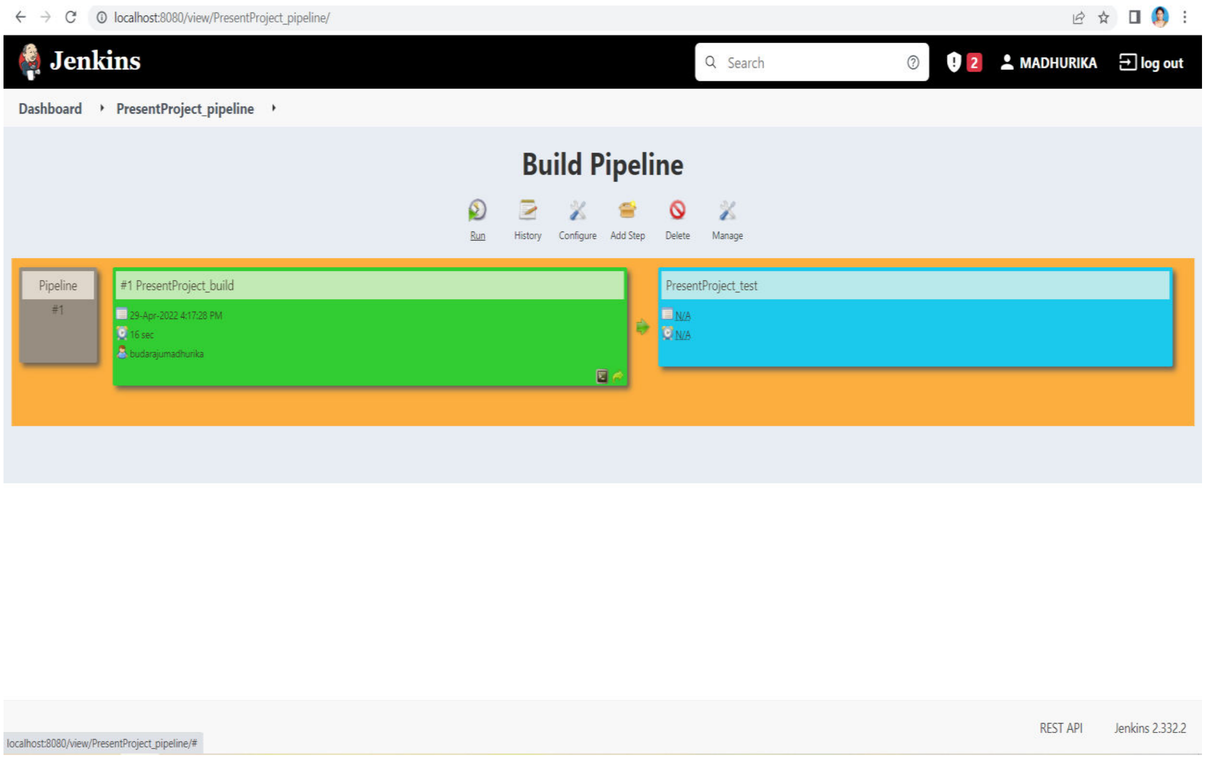
Console Output Link Style

Lightbox

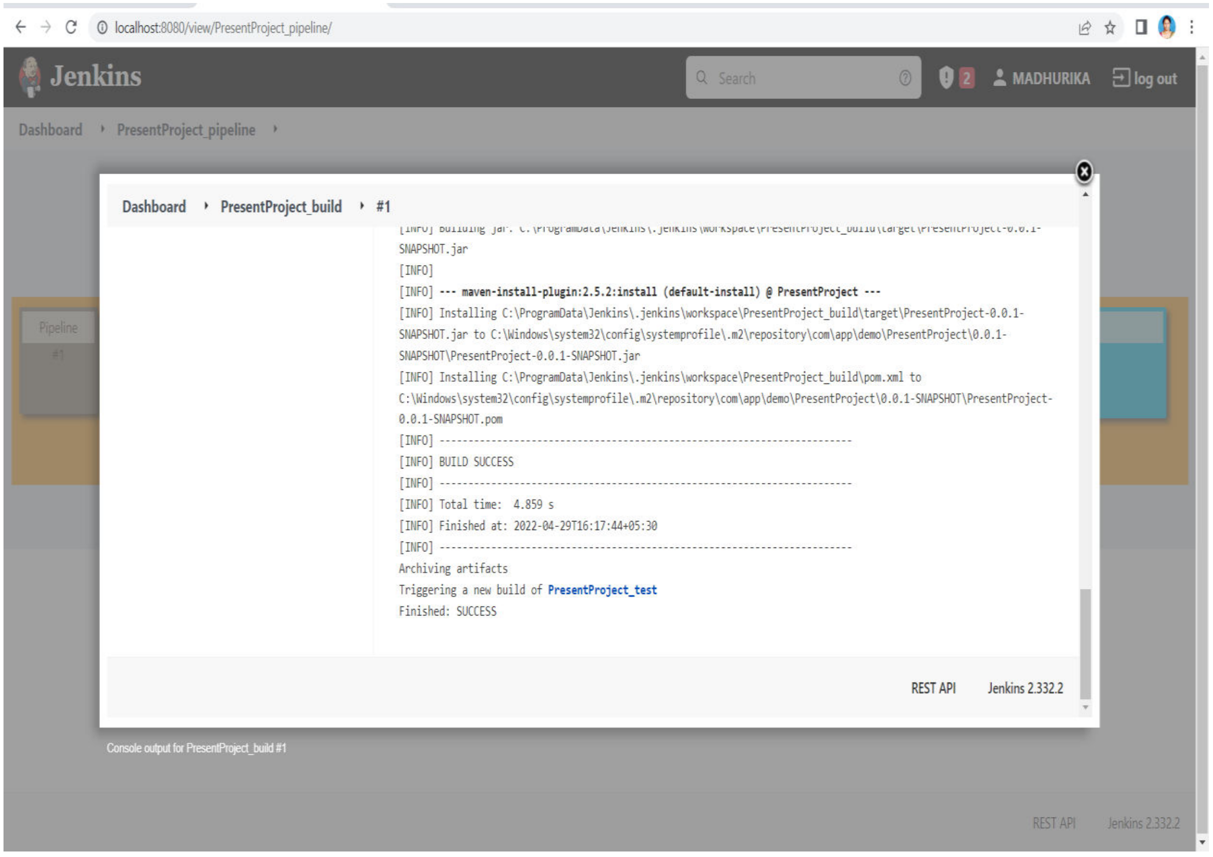
OK Apply

REST API Jenkins 2.332.2

29. Click on Run -> click on the small black box to open the console to check if the build is success

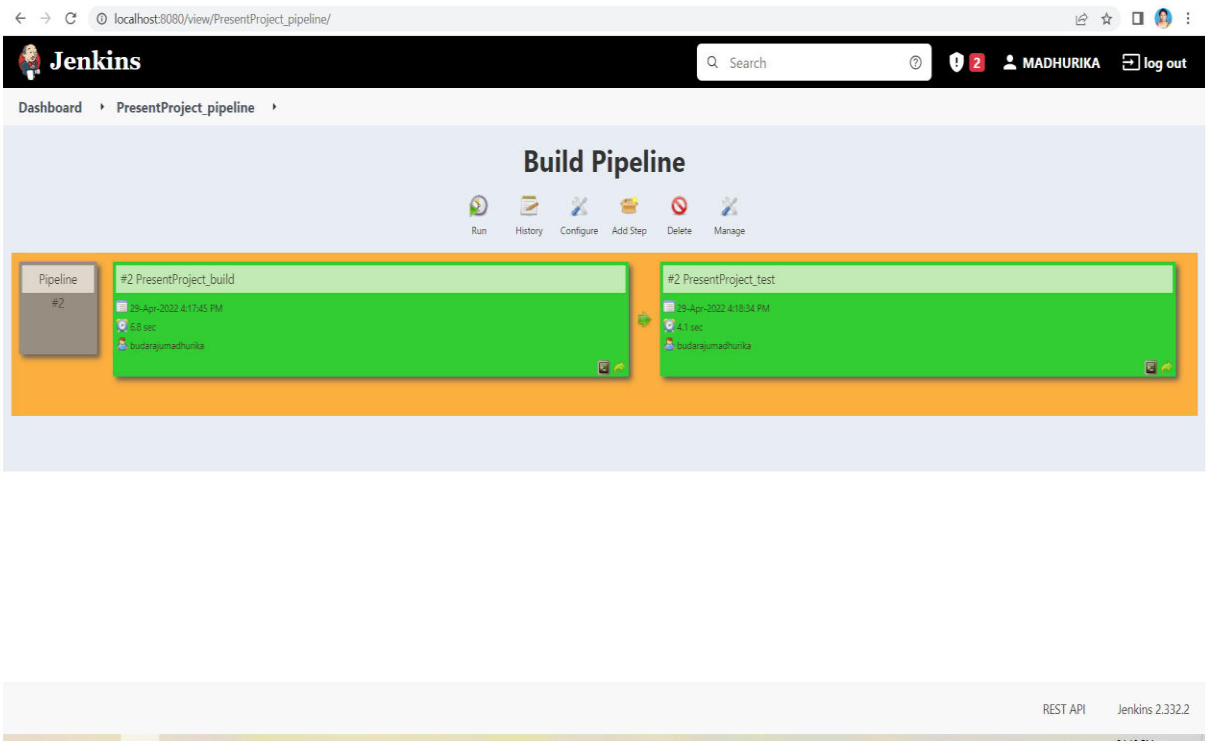


30. We can see that the build is success and the test project is also automatically triggered

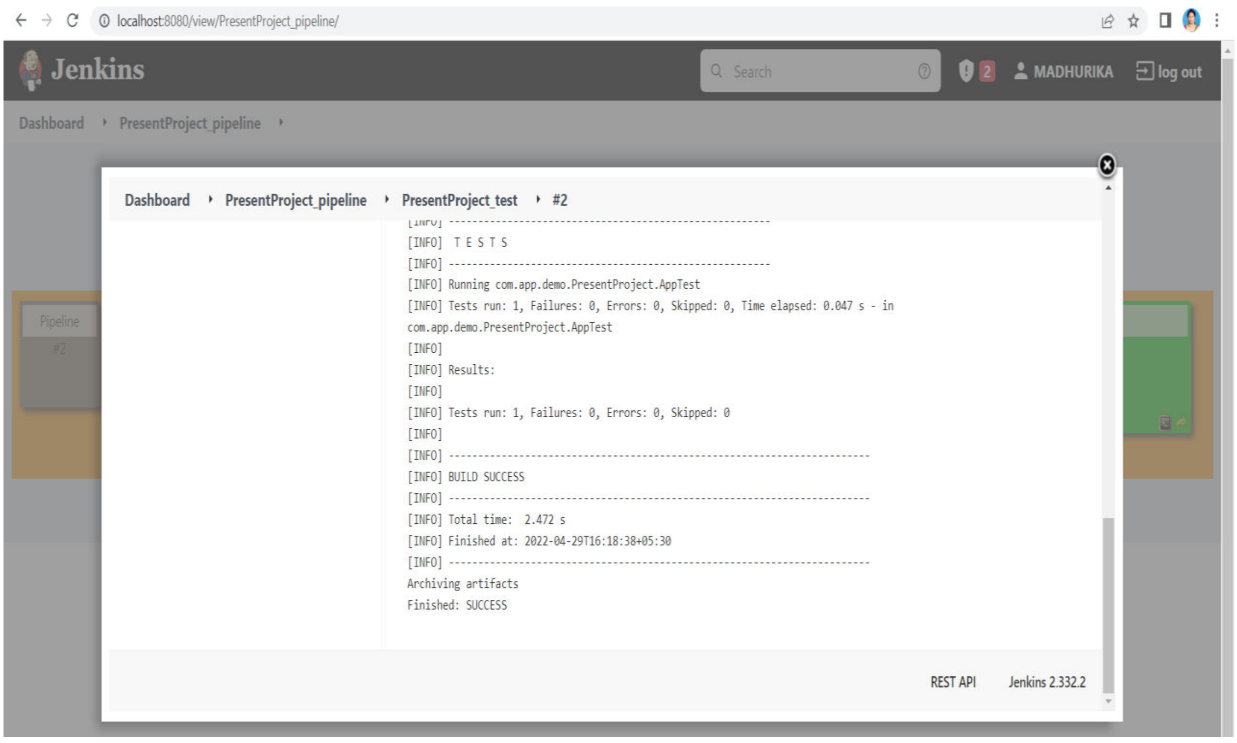


```
SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ PresentProject ---
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\PresentProject_build\target\PresentProject-0.0.1-
SNAPSHOT.jar to C:\Windows\system32\config\systemprofile\.m2\repository\com\app\demo\PresentProject\0.0.1-
SNAPSHOT\PresentProject-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\PresentProject_build\pom.xml to
C:\Windows\system32\config\systemprofile\.m2\repository\com\app\demo\PresentProject\0.0.1-SNAPSHOT\PresentProject-
0.0.1-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.859 s
[INFO] Finished at: 2022-04-29T16:17:44+05:30
[INFO] ------------------------------------------------------------------------
Archiving artifacts
Triggering a new build of PresentProject_test
Finished: SUCCESS
```

31. The pipeline is successful if it is in green color as shown ->check the console of the test project



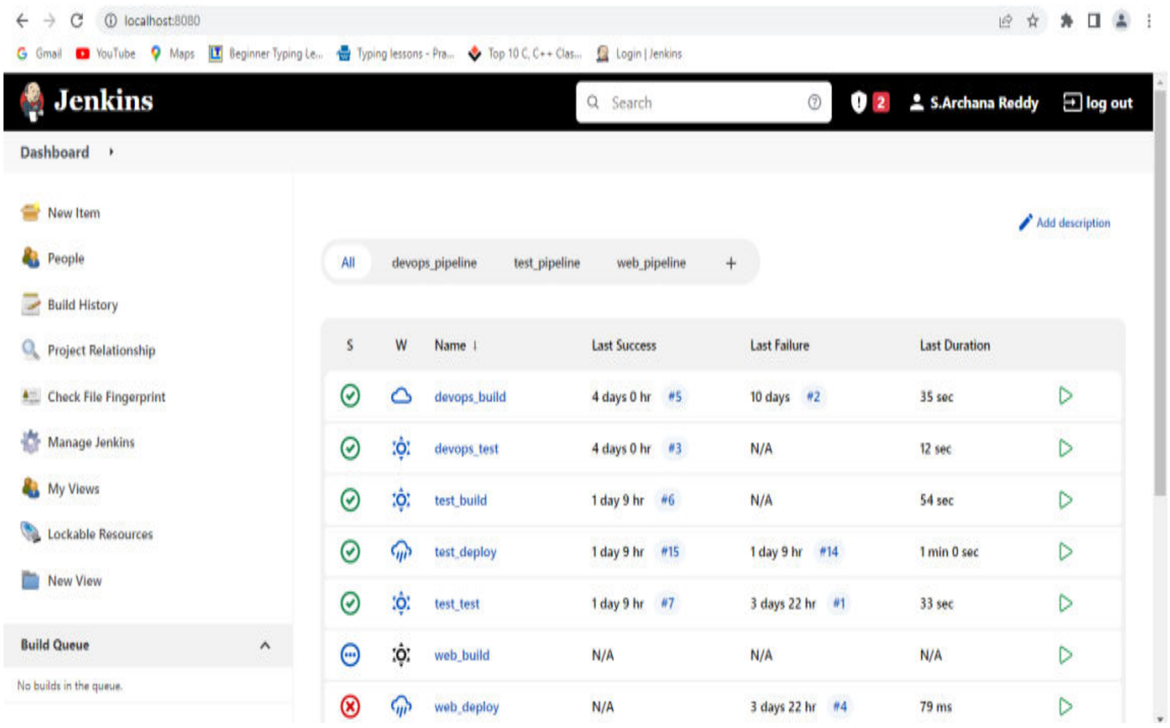
32. The test project is successful and all the artifacts are archived successfully


Dashboard › PresentProject_pipeline › PresentProject_test › #2

```
[INFO] T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.app.demo.PresentProject.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 s - in
com.app.demo.PresentProject.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.472 s
[INFO] Finished at: 2022-04-29T16:18:38+05:30
[INFO] ------------------------------------------------------------------------
Archiving artifacts
Finished: SUCCESS
```

III. Creating a Simple Pipeline using user interface for Web Project

1. Open Jenkins in local host:8080 and **create the new item**



localhost:8080

Jenkins | Search | S.Archana Reddy | log out

Dashboard

New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Lockable Resources
New View

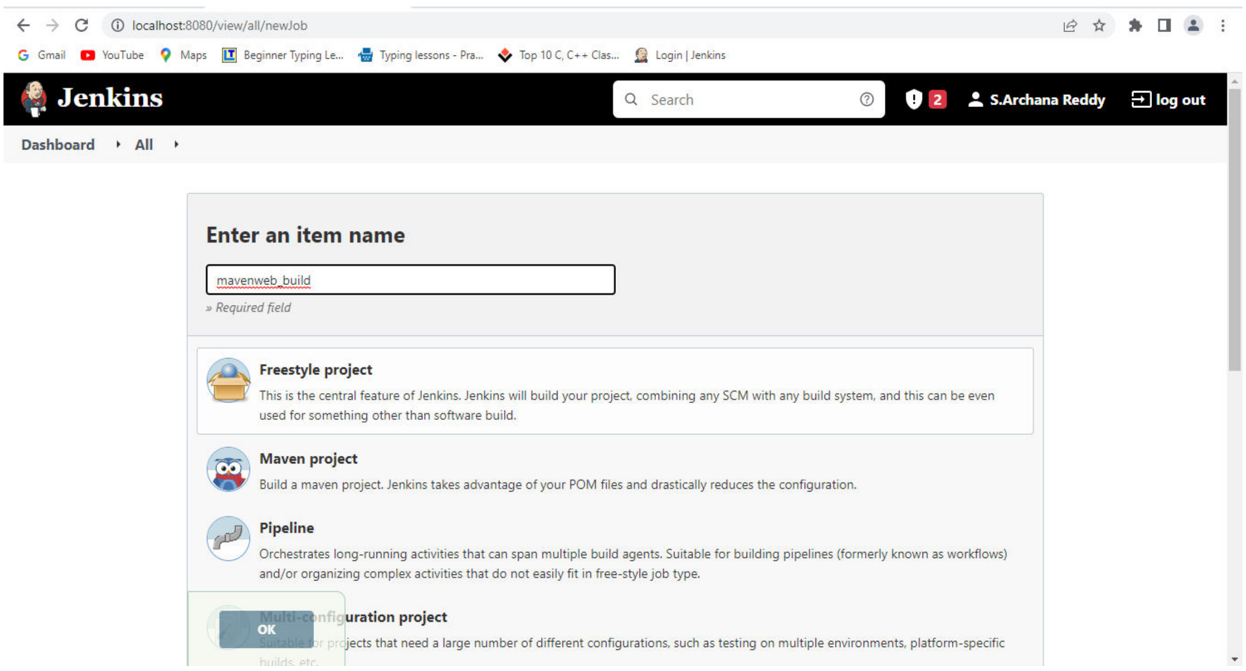
Build Queue
No builds in the queue.

Add description

All | devops_pipeline | test_pipeline | web_pipeline | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	devops_build	4 days 0 hr #5	10 days #2	35 sec
✓	☀	devops_test	4 days 0 hr #3	N/A	12 sec
✓	☀	test_build	1 day 9 hr #6	N/A	54 sec
✓	🌧	test_deploy	1 day 9 hr #15	1 day 9 hr #14	1 min 0 sec
✓	☀	test_test	1 day 9 hr #7	3 days 22 hr #1	33 sec
⋯	☀	web_build	N/A	N/A	N/A
✗	🌧	web_deploy	N/A	3 days 22 hr #4	79 ms

2. Select a new Freestyle Project give name (eg. mavenweb_build) and then click ok
3. In description type eg. Build demo



localhost:8080/view/all/newJob

Jenkins | Search | S.Archana Reddy | log out

Dashboard › All ›

Enter an item name

mavenweb_build

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

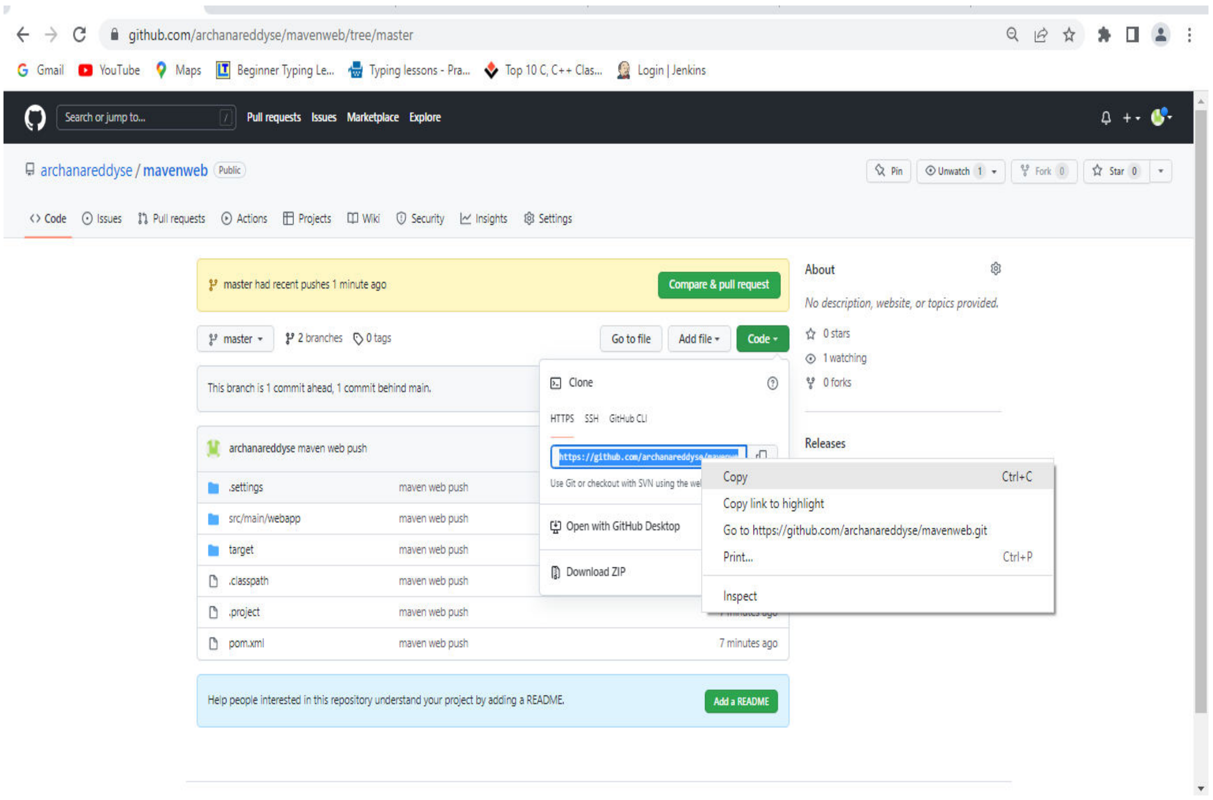
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

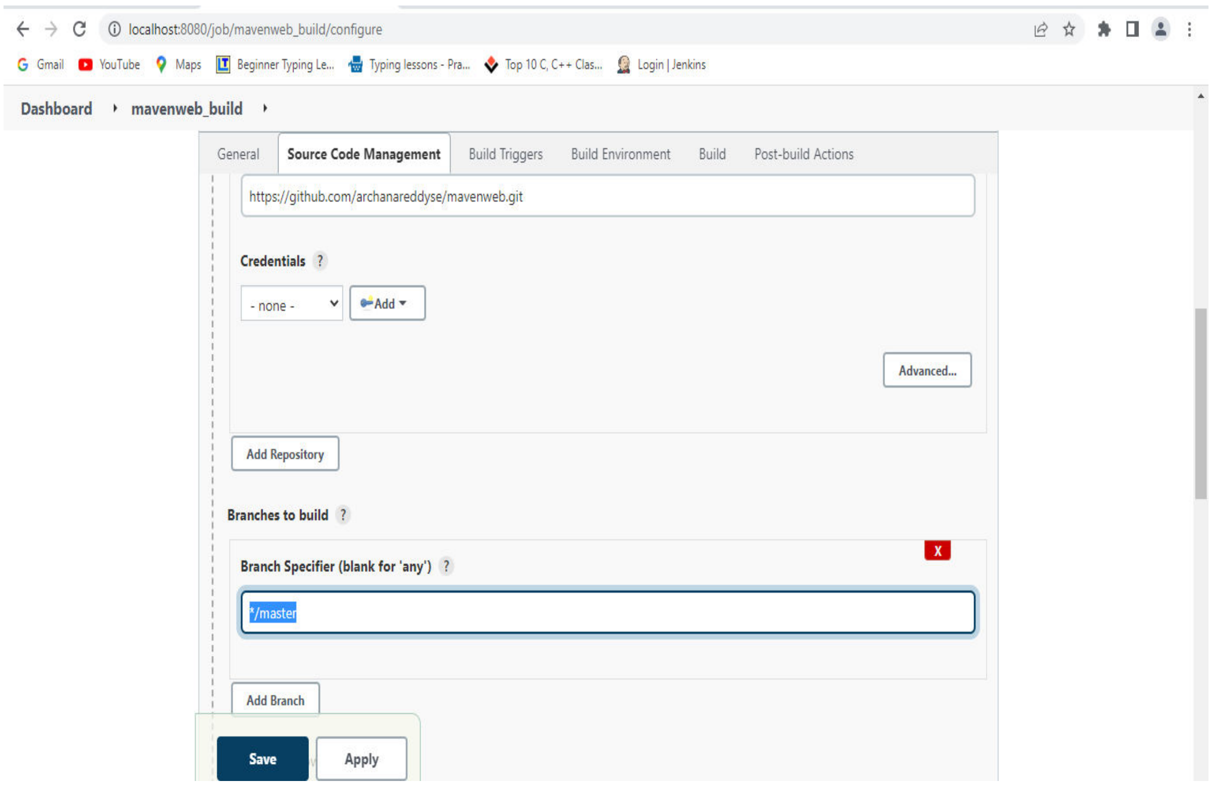
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

4. **Copy the git hub http code**


github.com/archanareddyse/mavenweb/tree/master
archanareddyse / mavenweb Public
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
master had recent pushes 1 minute ago
Compare & pull request
master
2 branches
0 tags
Go to file
Add file
Code
Clone
HTTPS SSH GitHub CLI
https://github.com/archanareddyse/maven
Copy Ctrl+C
Copy link to highlight
Go to https://github.com/archanareddyse/mavenweb.git
Print... Ctrl+P
Inspect
Open with GitHub Desktop
Download ZIP
.settings
src/main/webapp
target
.classpath
.project
pom.xml
Add a README

5. Paste the code in GIT URL, check for */master or */main.

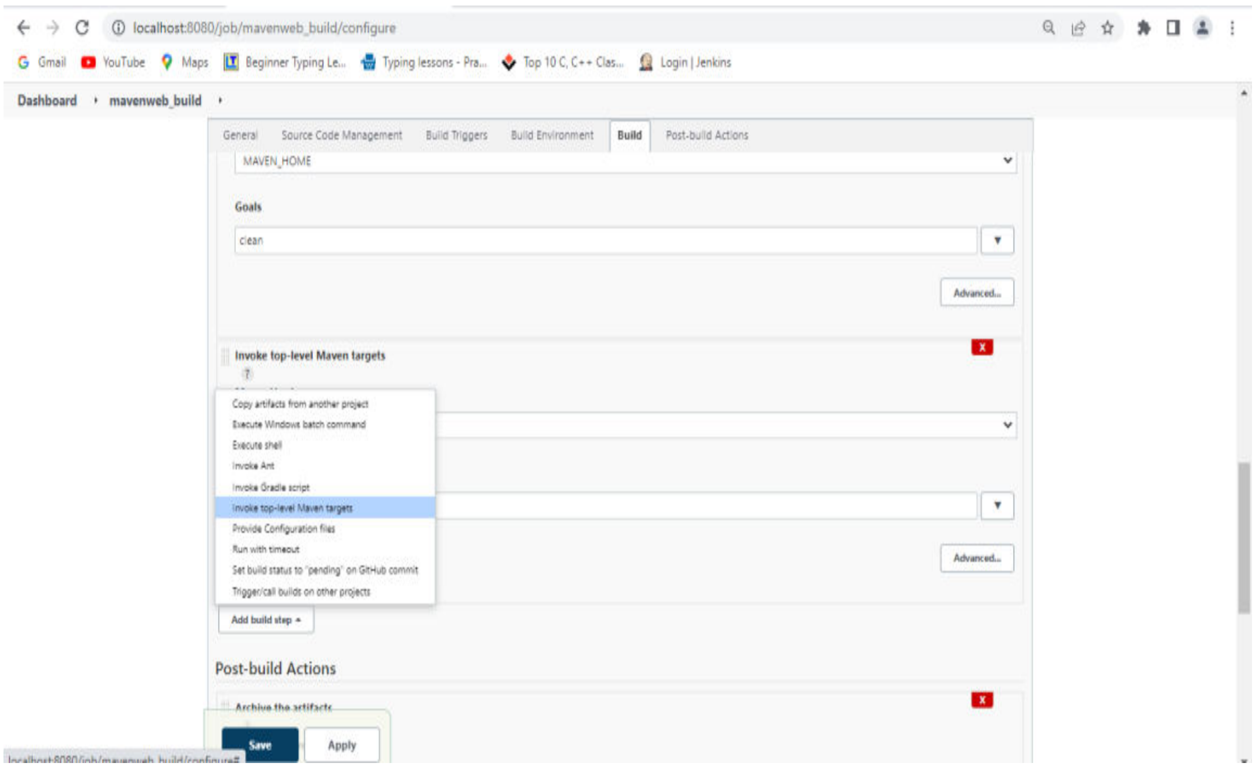


Repository URL: https://github.com/archanareddyse/mavenweb.git

Credentials: - none -

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

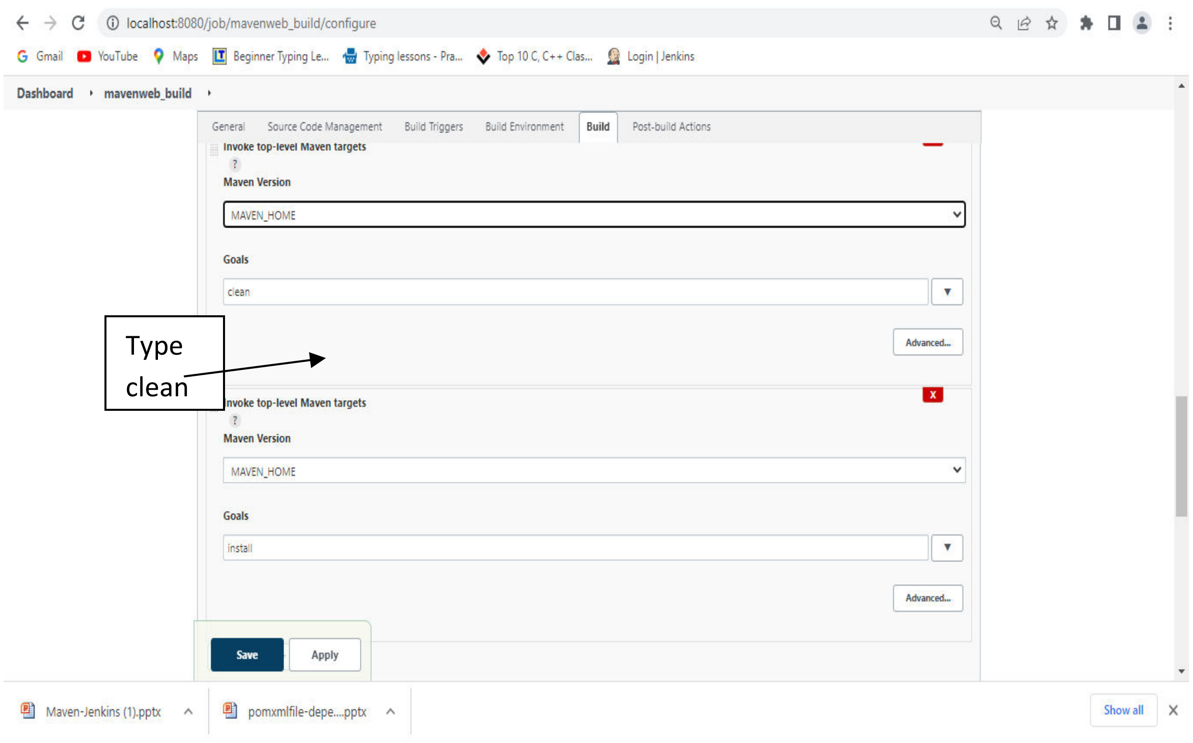
6. Now in Build, Invoke top-level Maven targets


localhost:8080/job/mavenweb_build/configure
Dashboard › mavenweb_build ›
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Invoke top-level Maven targets
Copy artifacts from another project
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Provide Configuration files
Run with timeout
Set build status to "pending" on GitHub commit
Trigger/call builds on other projects
Add build step
Post-build Actions
Archive the artifacts
Advanced...
Save
Apply

MAVEN_HOME

Goals: clean

7. Select the Maven path which is already set in the global credentials in Manage Jenkins
8. Set Goals field to clean as done in eclipse



Invoke top-level Maven targets

Maven Version: MAVEN_HOME

Goals: clean

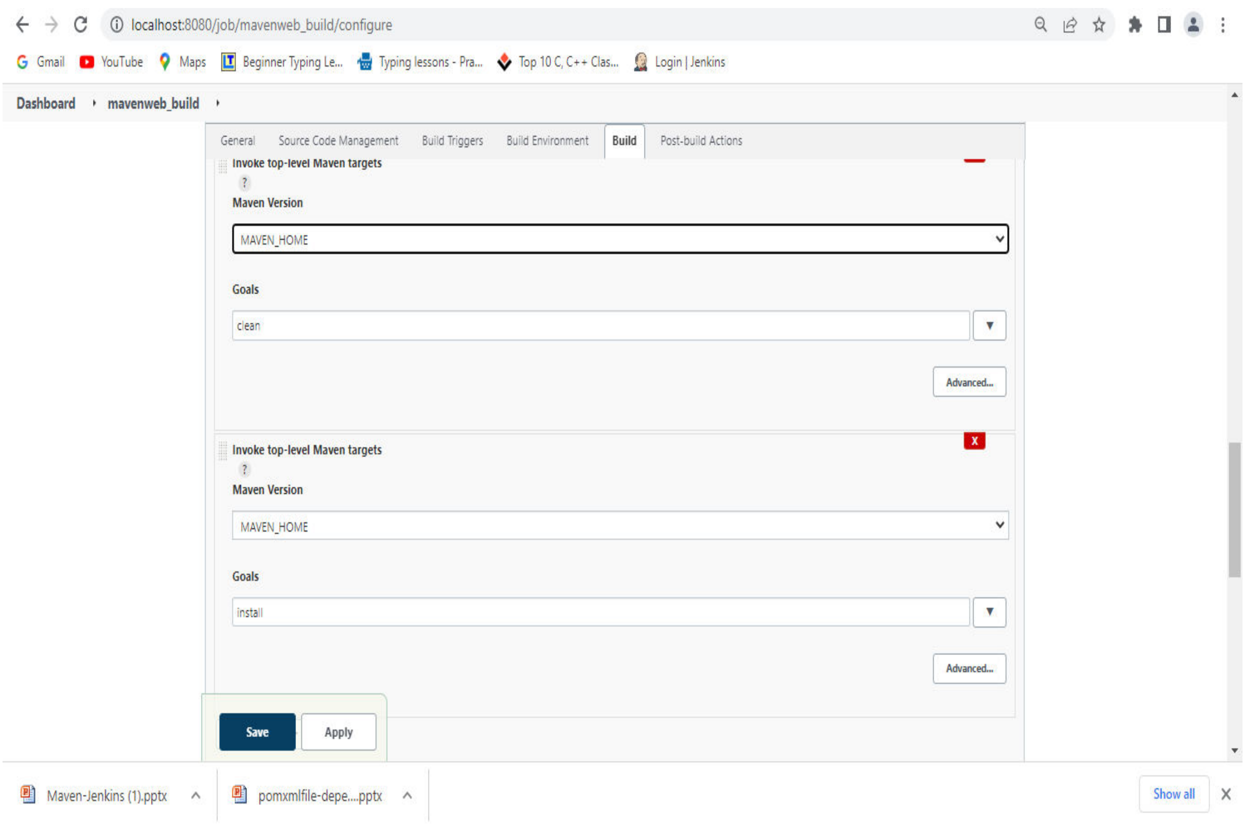
Invoke top-level Maven targets

Maven Version: MAVEN_HOME

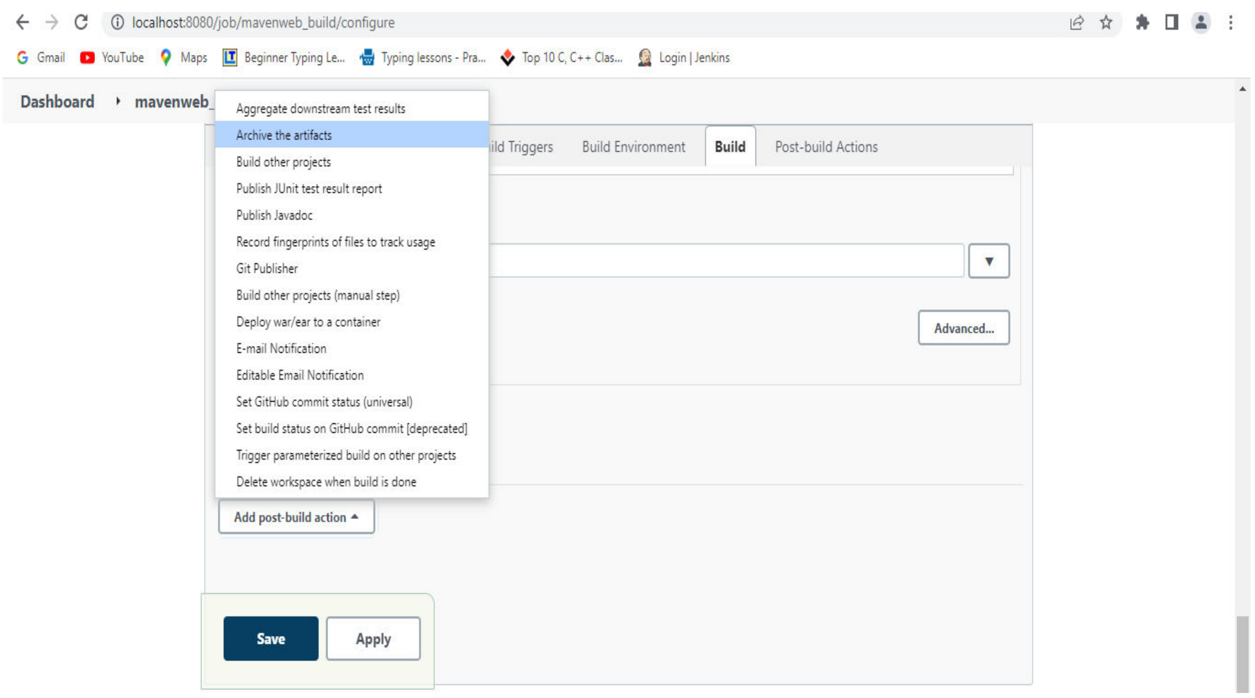
Goals: install

9. Again, in Build, select the “Invoke top-level Maven targets”

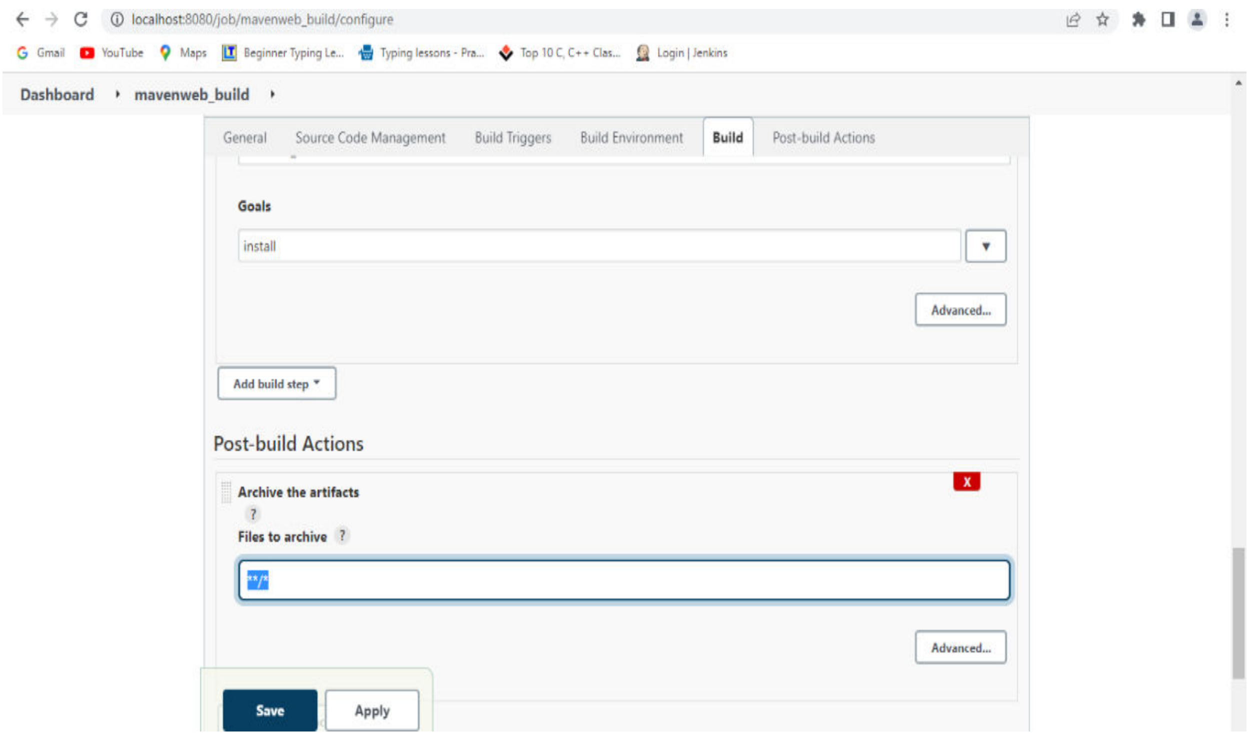
10. Select the Maven path which is already set in the global credentials in Manage Jenkins (eg MAVEN_HOME)
11. Set Goals field to **Install** as done in eclipse



12. Now in post build actions-> select "**Archive the artifacts**", to send the output of build project to the testing team


localhost:8080/job/mavenweb_build/configure
Dashboard
mavenweb_
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Git Publisher
Build other projects (manual step)
Deploy war/ear to a container
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Trigger parameterized build on other projects
Delete workspace when build is done
Add post-build action
ild Triggers
Build Environment
Build
Post-build Actions
Advanced...
Save
Apply

13. If we want to archive all the artifacts type **/* in Files to Archive



Goals

install

Post-build Actions

Archive the artifacts

Files to archive

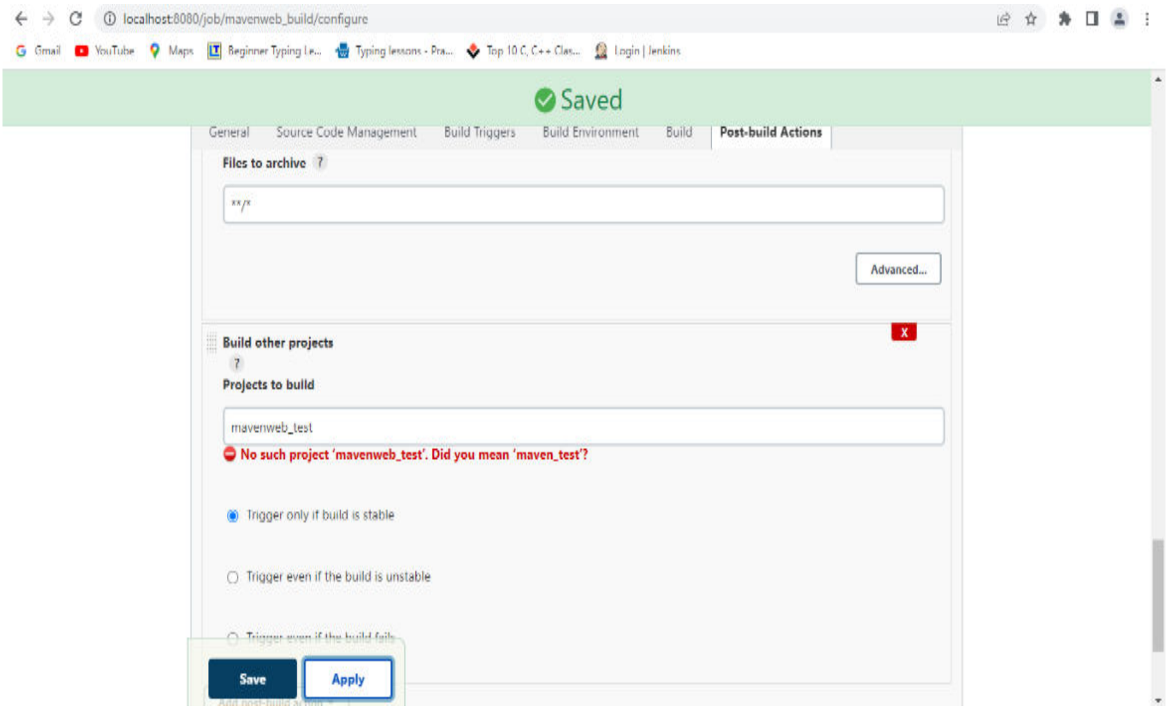
**/*

14. Now the next step is to build other projects, where we will create a test project which will be triggered by the build project.

For this **in <u>Add Post build Action</u> select "Build other projects"**

15. In "Projects to build" enter the next project name as "**mavenweb_test**"

16. Next press on Apply and Save



Files to archive

**/*

Build other projects

Projects to build

mavenweb_test

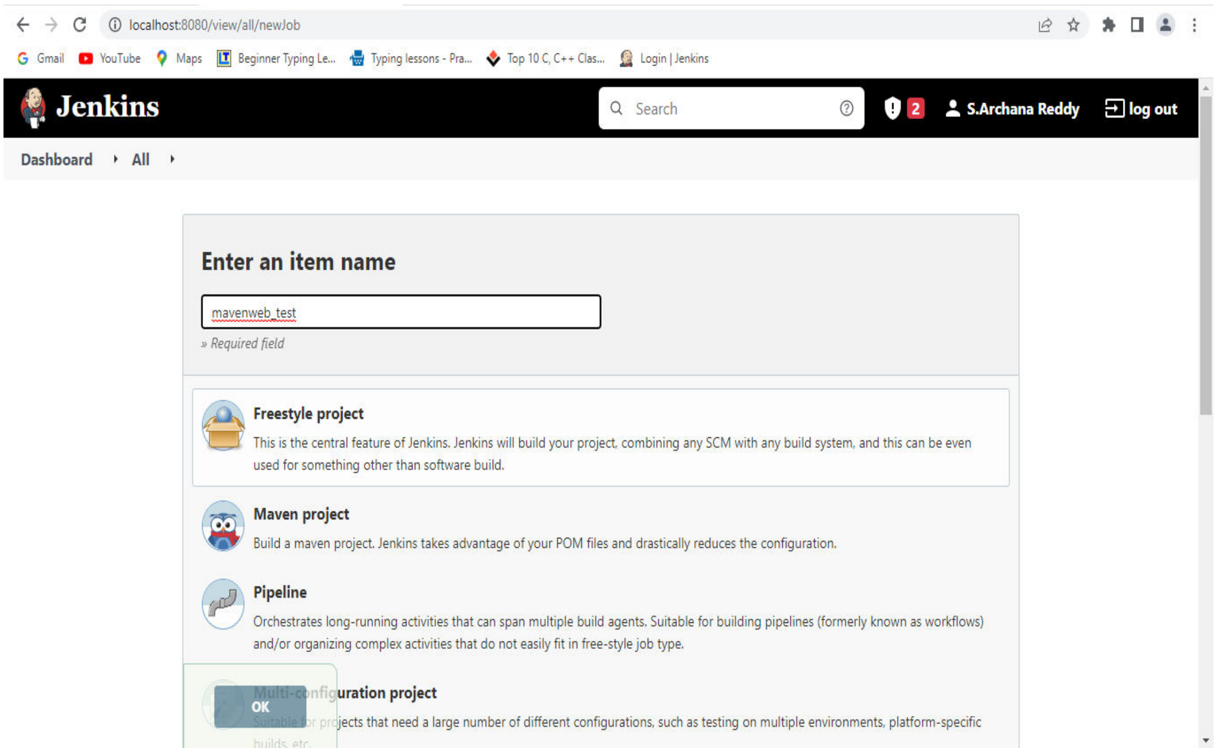
No such project 'mavenweb_test'. Did you mean 'maven_test'?

(●) Trigger only if build is stable

( ) Trigger even if the build is unstable

( ) Trigger even if the build fails

17. Go to dashboard -> New item-> Freestyle Project, and then give next project name as **MavenJavaProject_test**, then press on OK



Enter an item name

mavenweb_test

» Required field

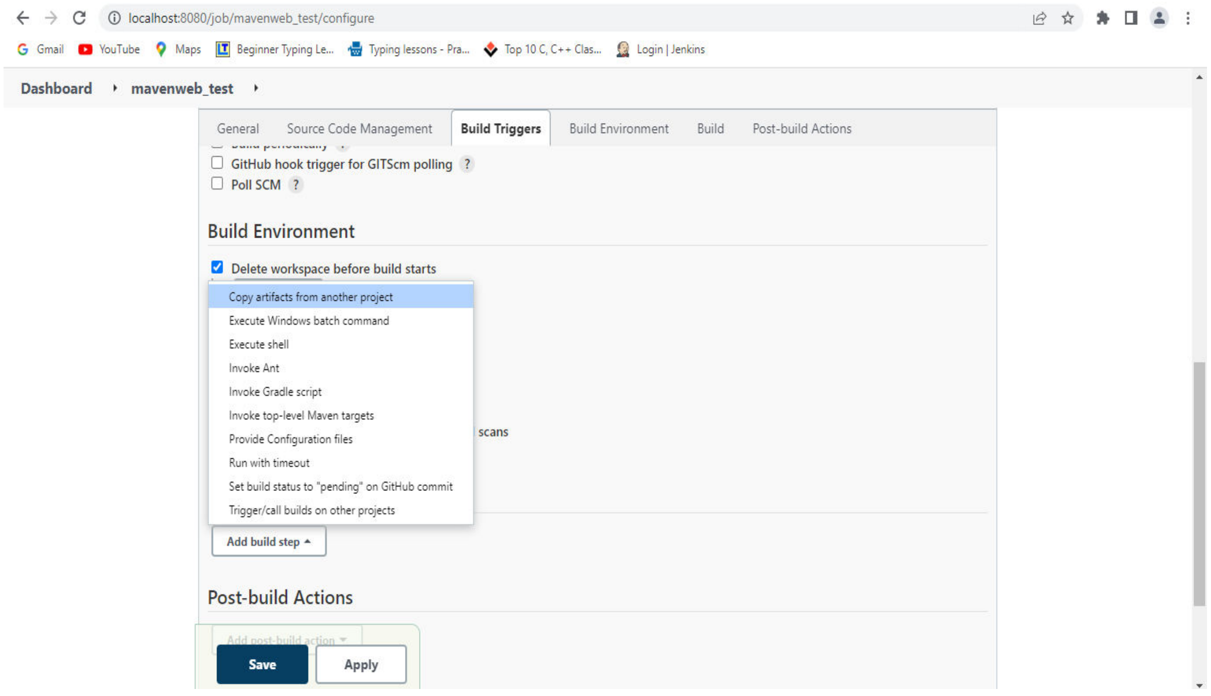
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

18. In description type eg. Test demo
19. In Build environment, check the box with name “Delete the workspace before build starts”. This is to discard old builds
20. In Build select “**copy the artifacts from another project**” to forward the artifacts of the previous project to the current test project i.e. mavenweb_test



- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

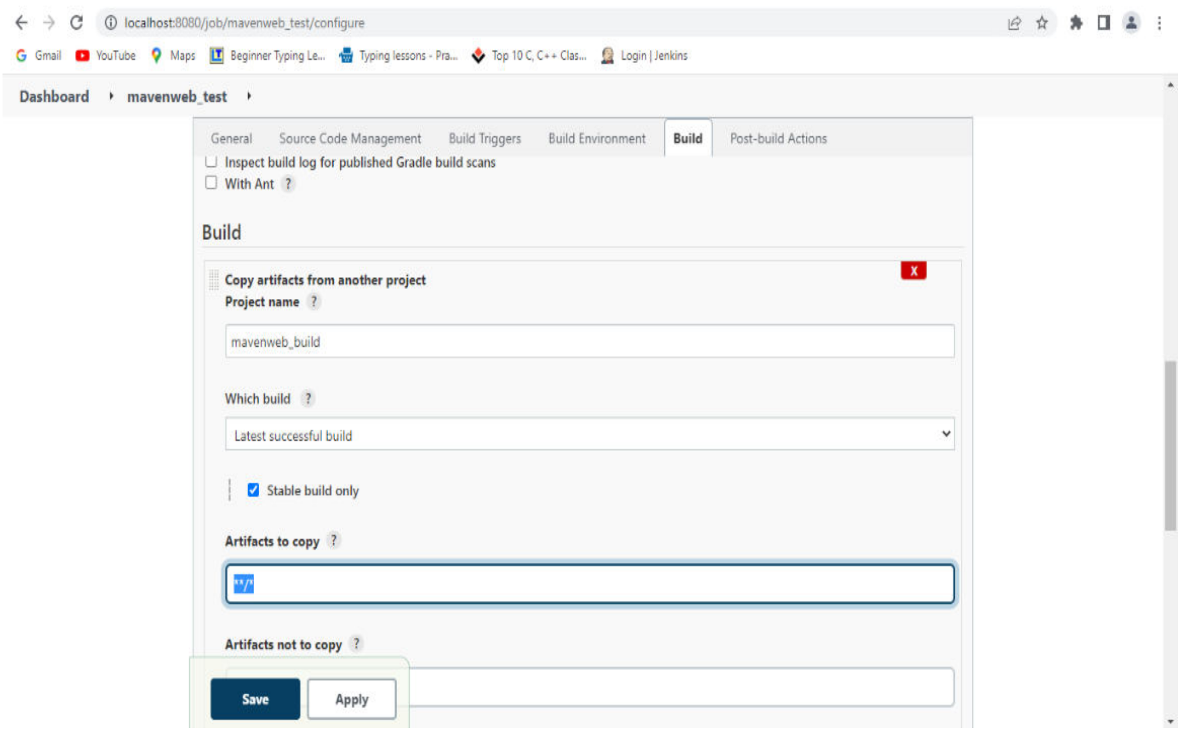
Build Environment

- [x] Delete workspace before build starts

Copy artifacts from another project
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Provide Configuration files
Run with timeout
Set build status to "pending" on GitHub commit
Trigger/call builds on other projects

Post-build Actions

21. Give the name of the project from which we want to copy the artifacts (eg. Type mavenweb_build) and check the box ->stable build only->to copy all the artifacts type **/*



localhost:8080/job/mavenweb_test/configure

Dashboard › mavenweb_test ›

Build

Copy artifacts from another project

Project name: mavenweb_build

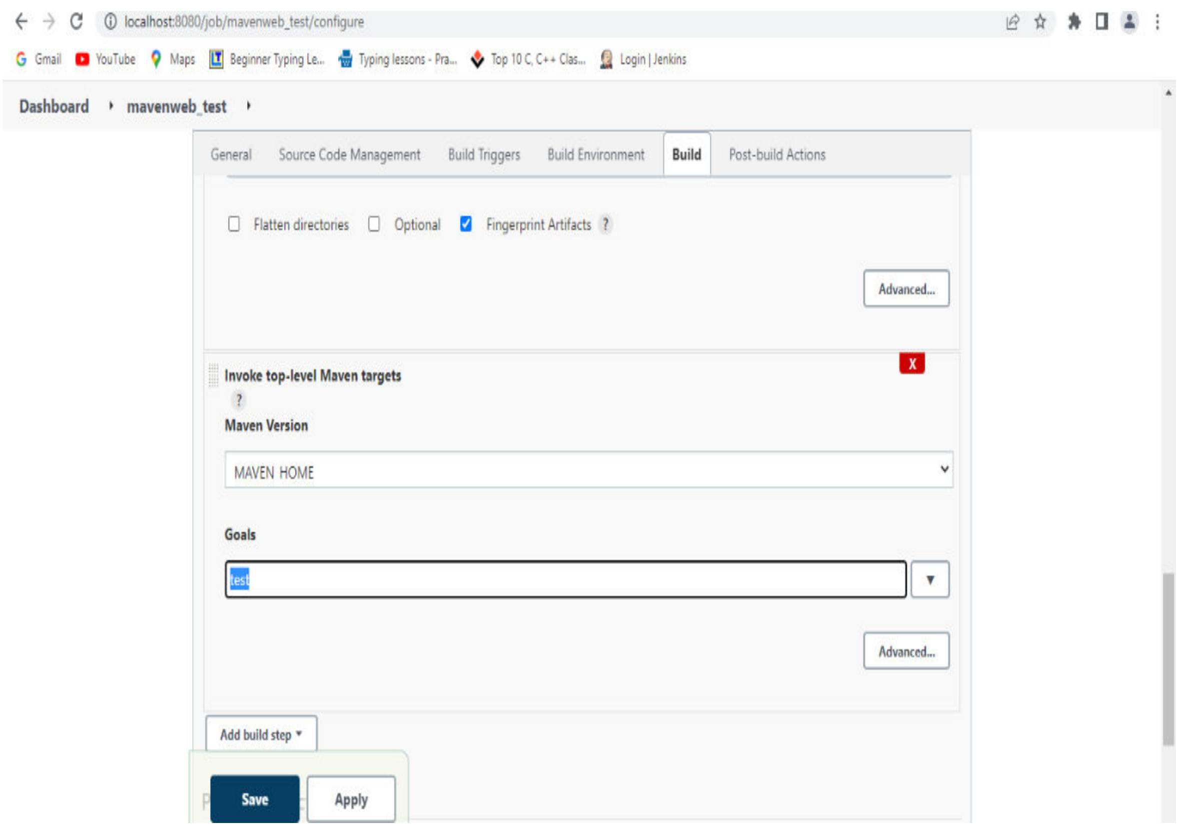
Which build: Latest successful build

☑ Stable build only

Artifacts to copy: **/*

Artifacts not to copy

22. Now select Invoke top-level Maven targets in build
23. This time give the goal as test after selecting the Maven version



localhost:8080/job/mavenweb_test/configure

Dashboard › mavenweb_test ›

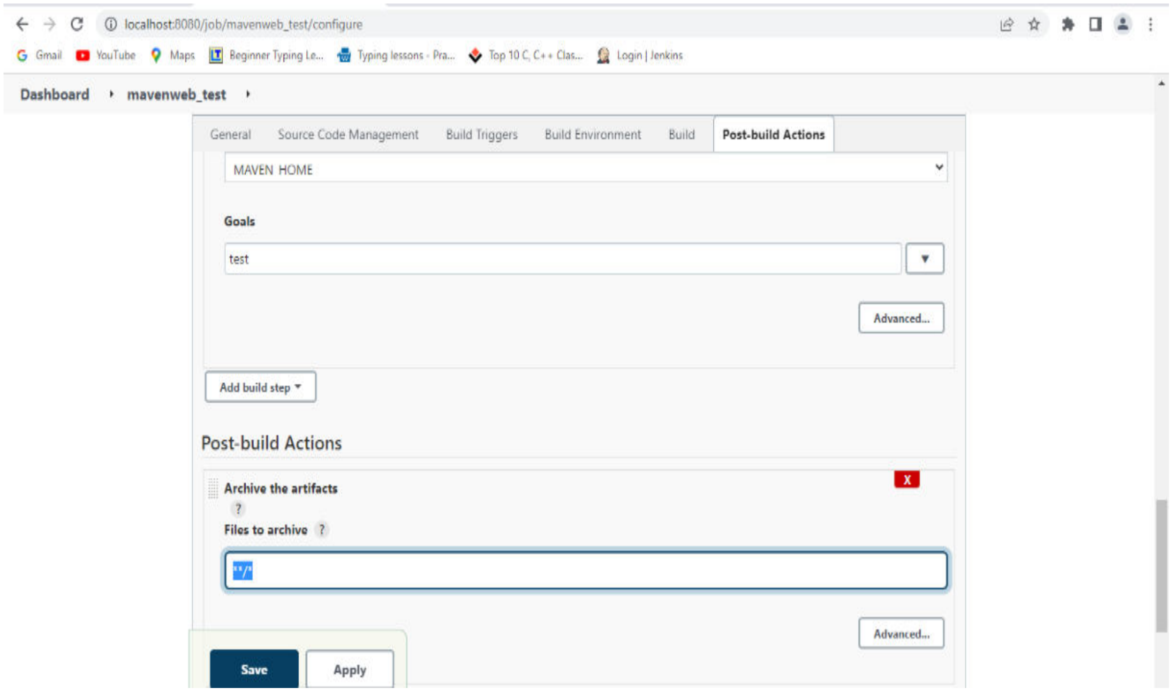
☐ Flatten directories ☐ Optional ☑ Fingerprint Artifacts

Invoke top-level Maven targets

Maven Version: MAVEN HOME

Goals: test

24. In post-build actions->select Archive the artifacts
25. To save all the artifacts->type **/* and Apply->Save



MAVEN HOME

Goals

test

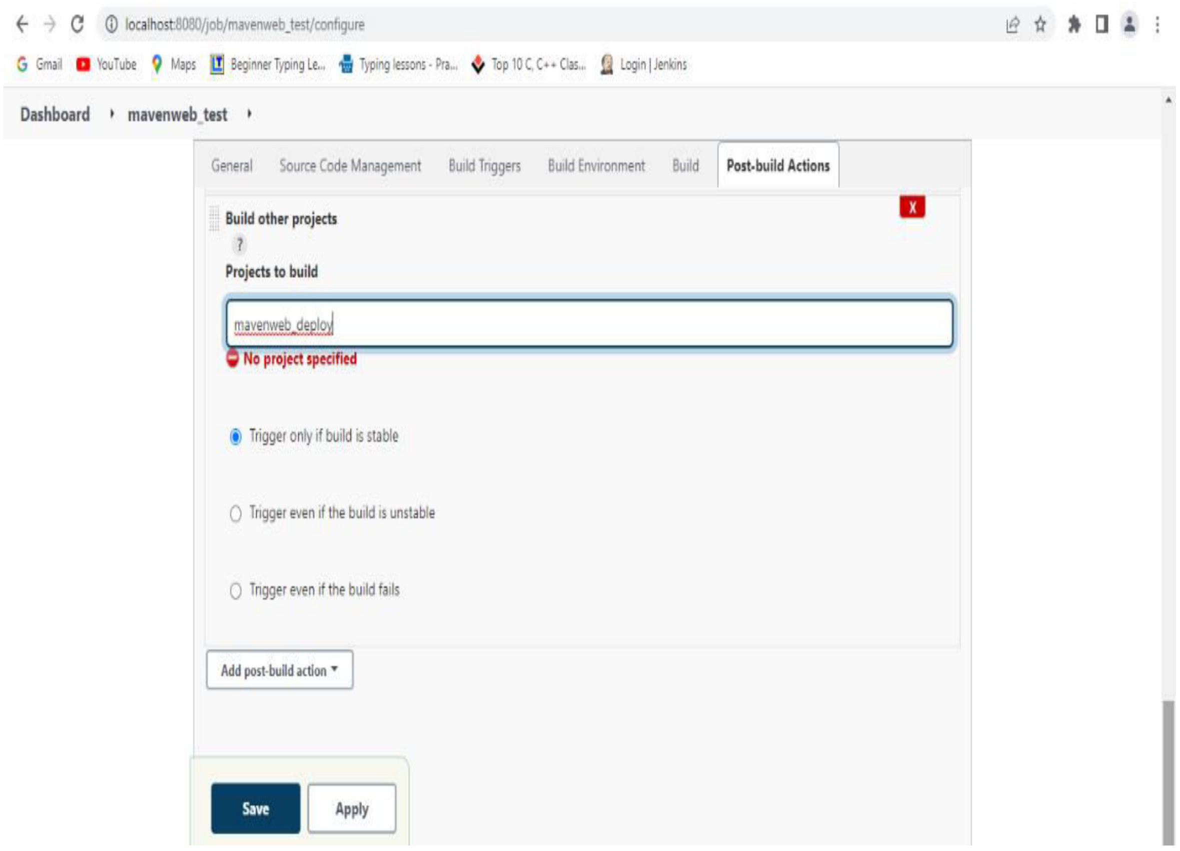
Post-build Actions

Archive the artifacts

Files to archive

**/*

26. Now we here select the build other projects, where we will create a deploy project which will be triggered by the test project, click Apply and Save



Build other projects

Projects to build

mavenweb_deploy

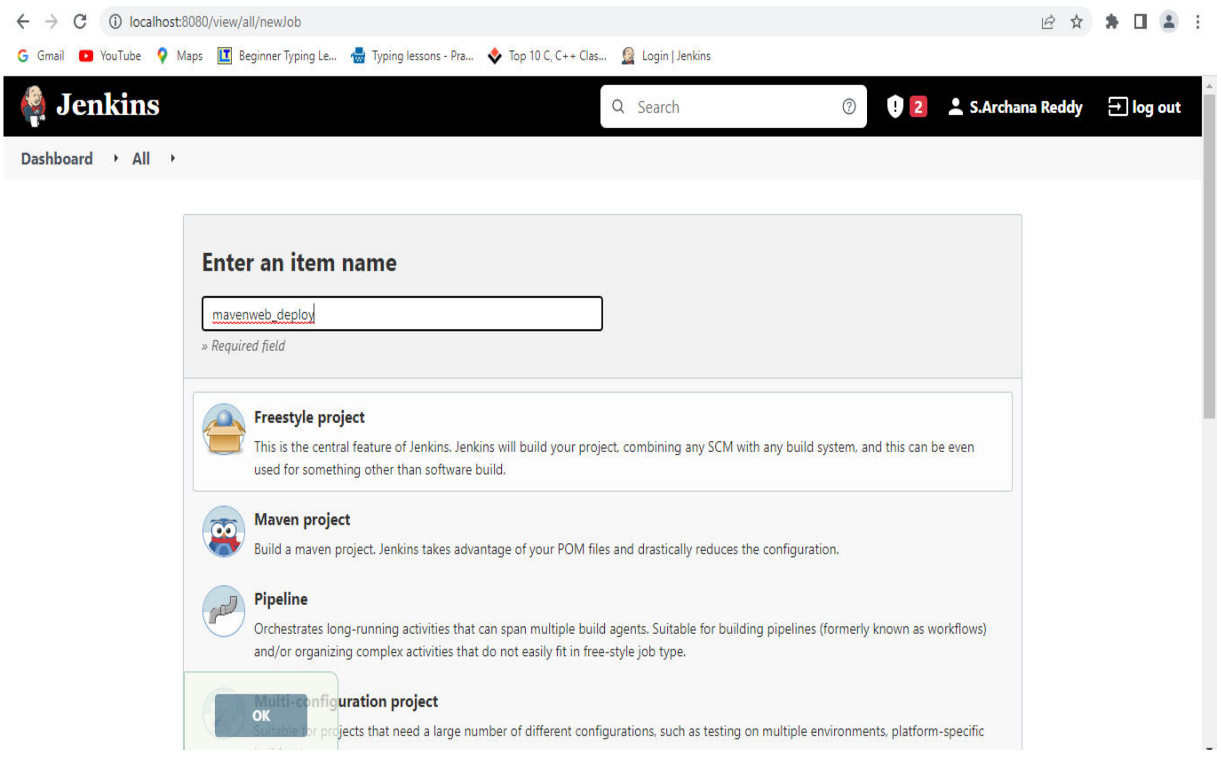
No project specified

(●) Trigger only if build is stable

( ) Trigger even if the build is unstable

( ) Trigger even if the build fails

27. Create a new freestyle project test as shown and click ok


localhost:8080/view/all/newJob

Dashboard › All ›

Enter an item name

mavenweb_deploy

» Required field

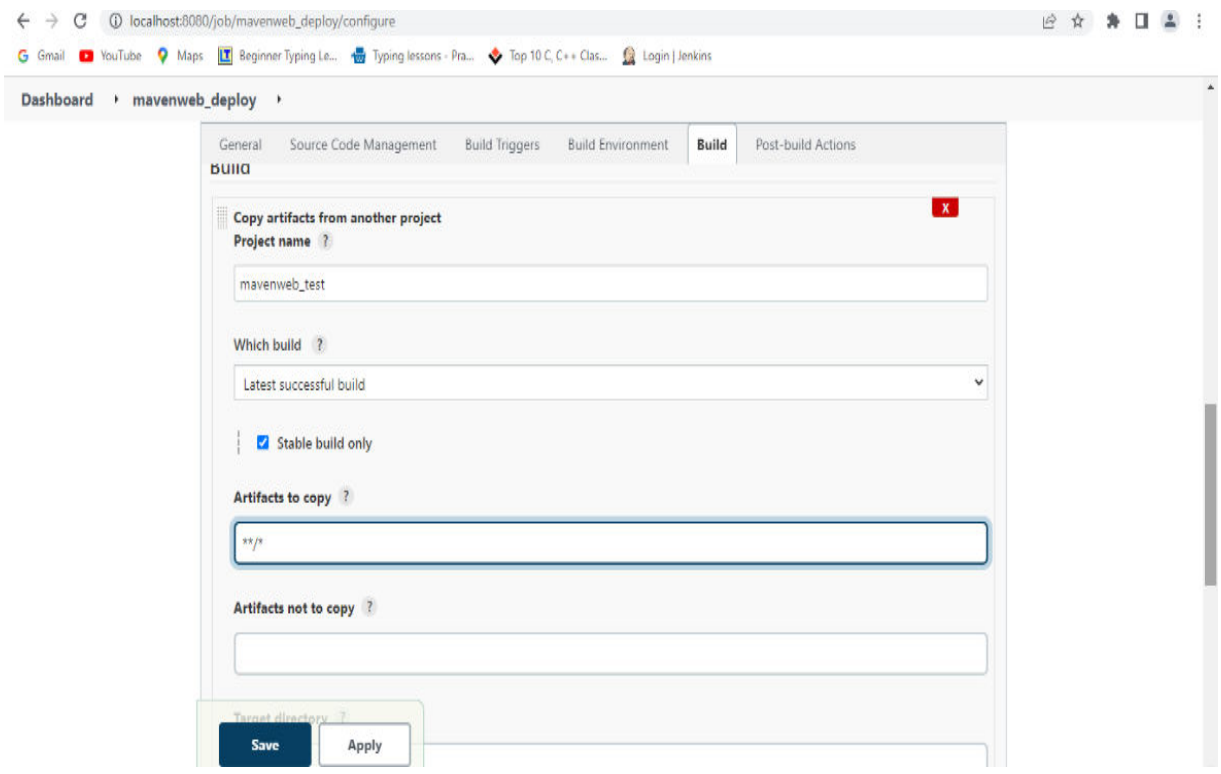
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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

28. Give the name of the project from which we want to copy the artifacts and check the box ->stable build only->to copy all the artifacts type **/*


localhost:8080/job/mavenweb_deploy/configure

Dashboard › mavenweb_deploy ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Build

Copy artifacts from another project

Project name: mavenweb_test

Which build: Latest successful build

[✓] Stable build only

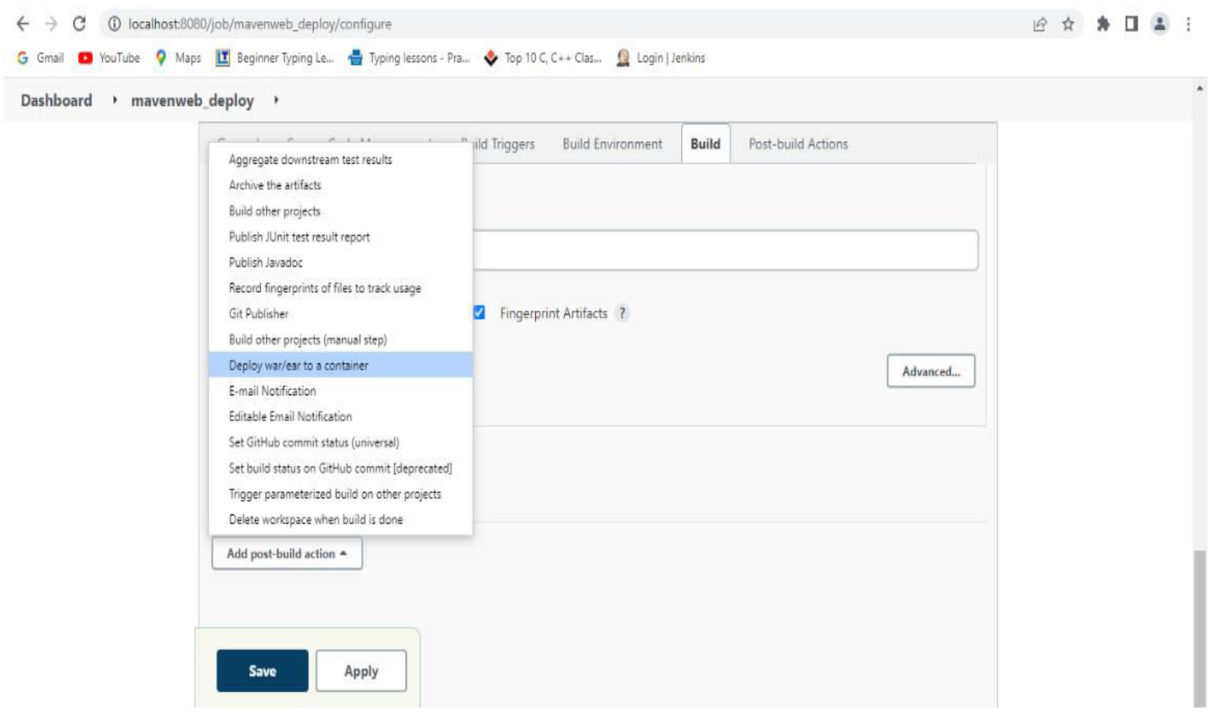
Artifacts to copy: **/*

Artifacts not to copy:

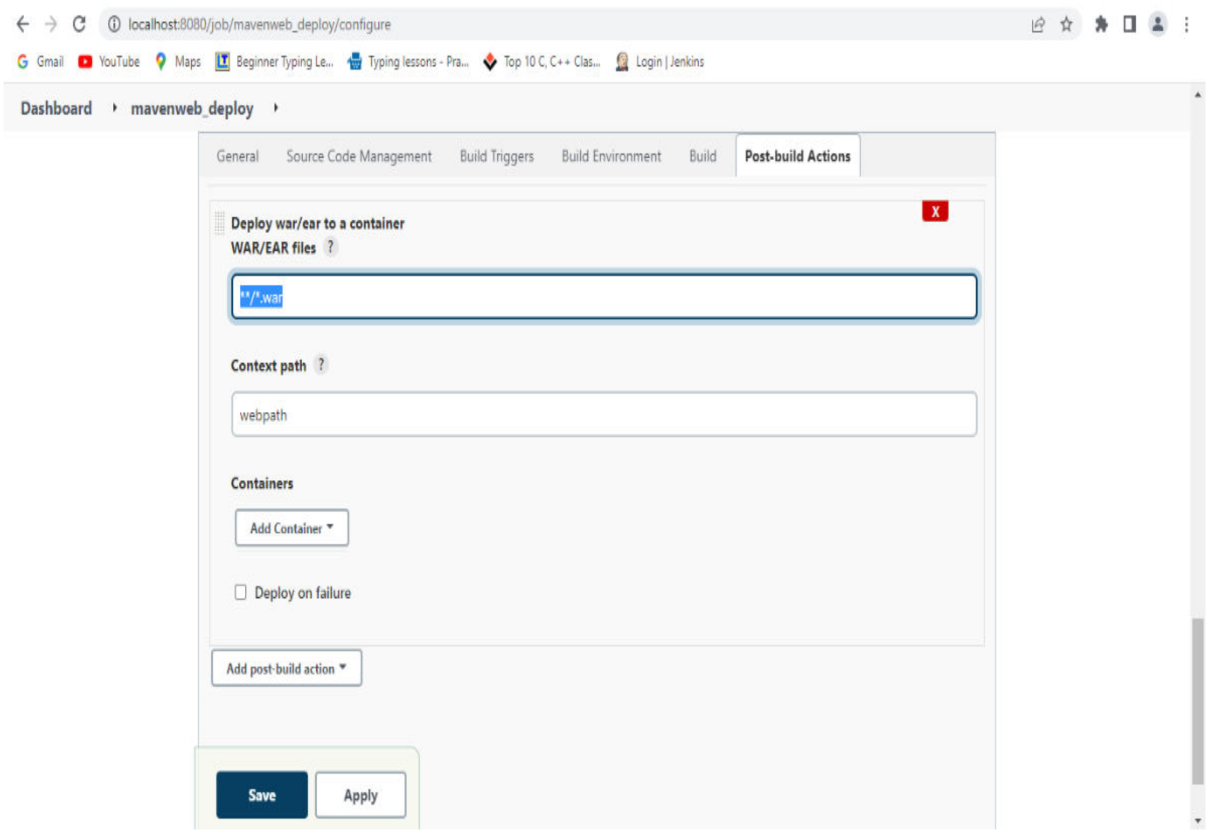
Target directory

29. Now here we go for Add post –build action where we select the Deploy war/ear to a container.

    This **plugin** takes a **war/ear** file and deploys that to a running remote application server at the end of a build.


localhost:8080/job/mavenweb_deploy/configure
Dashboard
mavenweb_deploy
Build Triggers
Build Environment
Build
Post-build Actions
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Git Publisher
Build other projects (manual step)
Deploy war/ear to a container
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Trigger parameterized build on other projects
Delete workspace when build is done
Fingerprint Artifacts
Advanced...
Add post-build action
Save
Apply

30. Deploy war/ear to a container takes the artifacts as **/*.war.


localhost:8080/job/mavenweb_deploy/configure

Dashboard > mavenweb_deploy

Deploy war/ear to a container

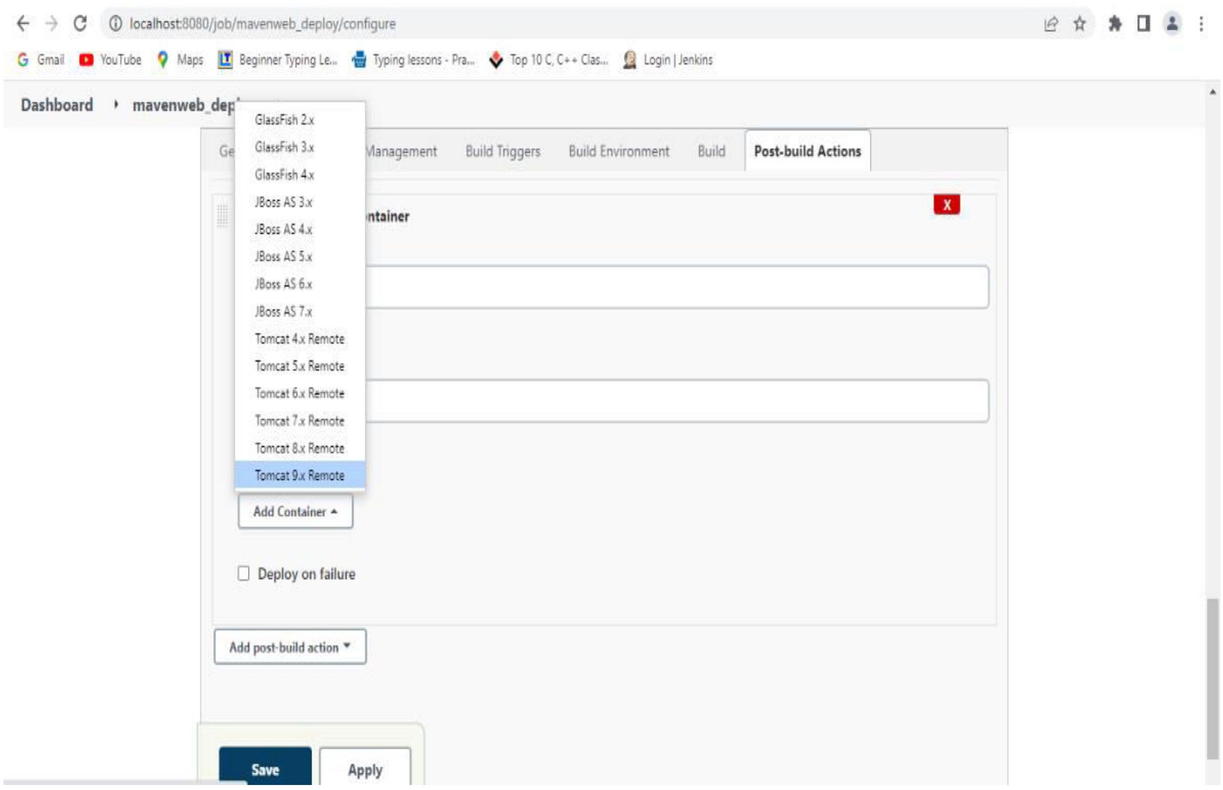
WAR/EAR files: **/*.war

Context path: webpath

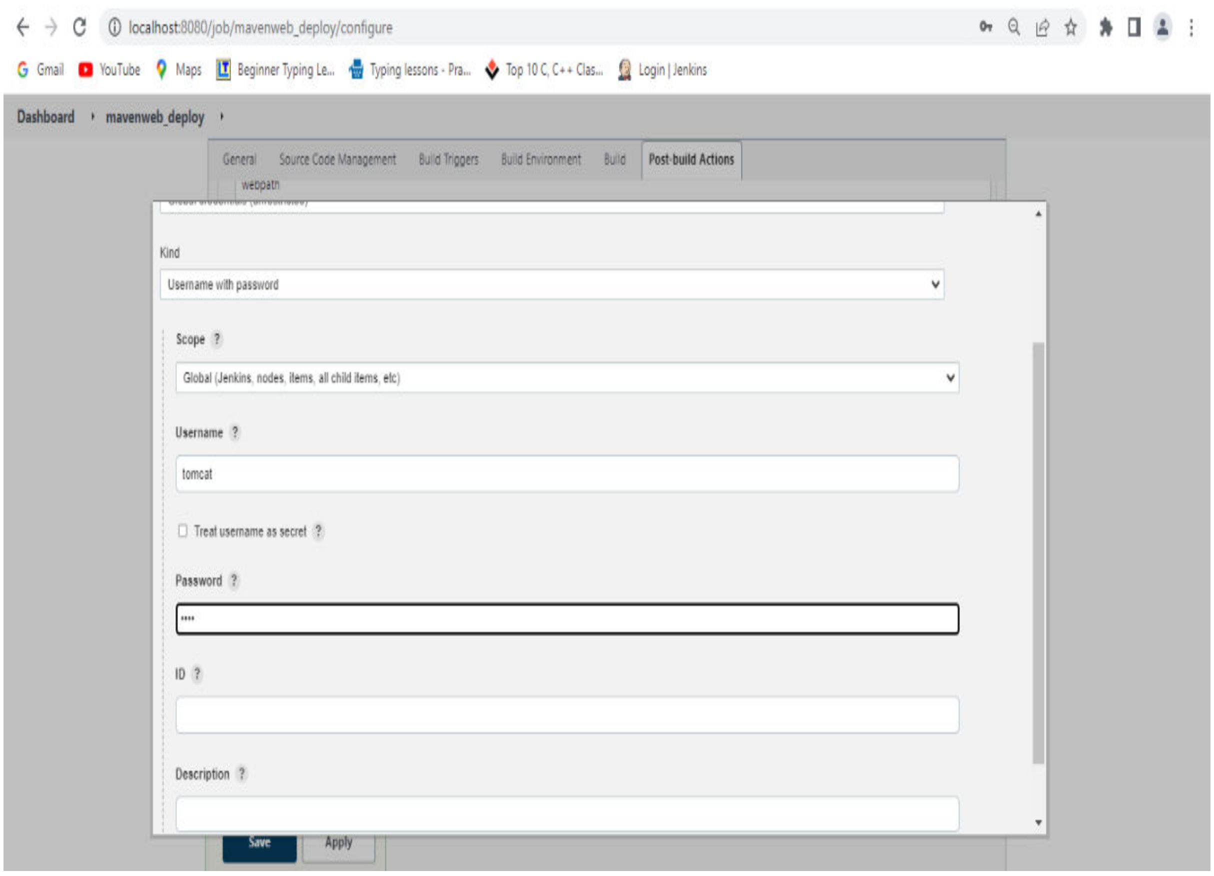
Containers

Deploy on failure

31. Select the tomcat version.


localhost:8080/job/mavenweb_deploy/configure

32. Here we add the credentials of tomcat


localhost:8080/job/mavenweb_deploy/configure

Dashboard › mavenweb_deploy ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: tomcat

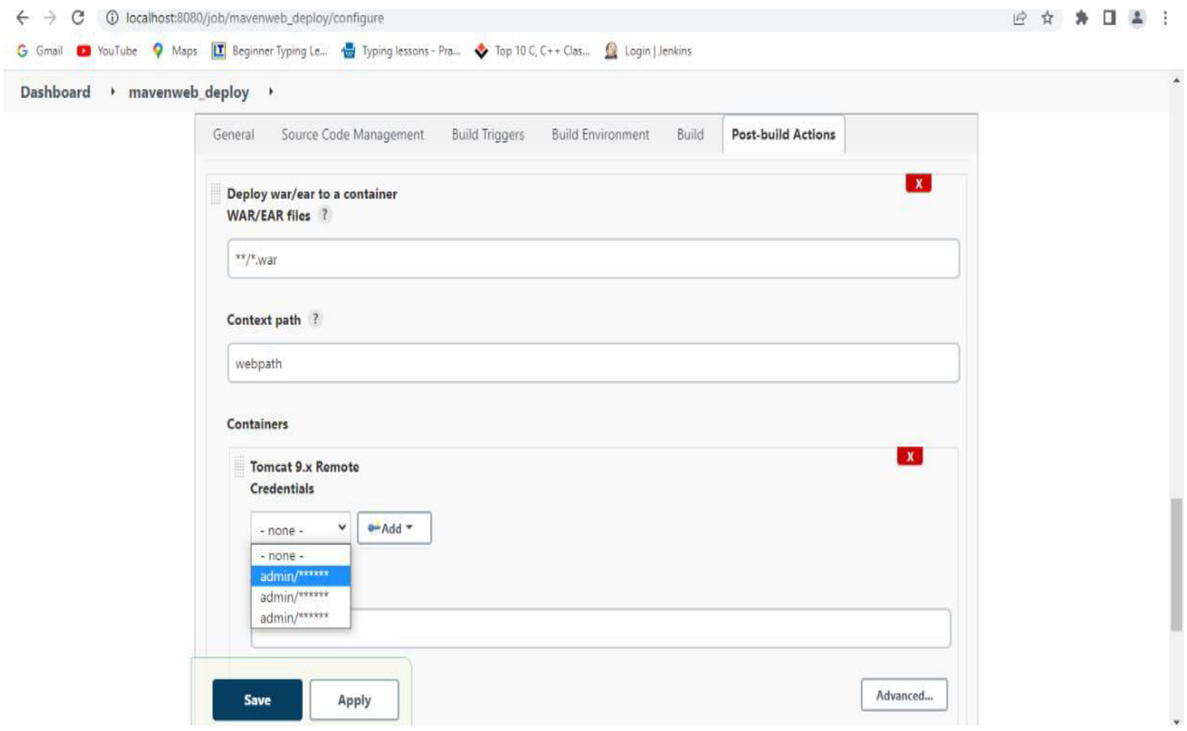
☐ Treat username as secret

Password: ****

ID:

Description:

33. Here we added the credentials of tomcat and tomcat URL also


localhost:8080/job/mavenweb_deploy/configure

Dashboard › mavenweb_deploy ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Deploy war/ear to a container

WAR/EAR files: **/*.war

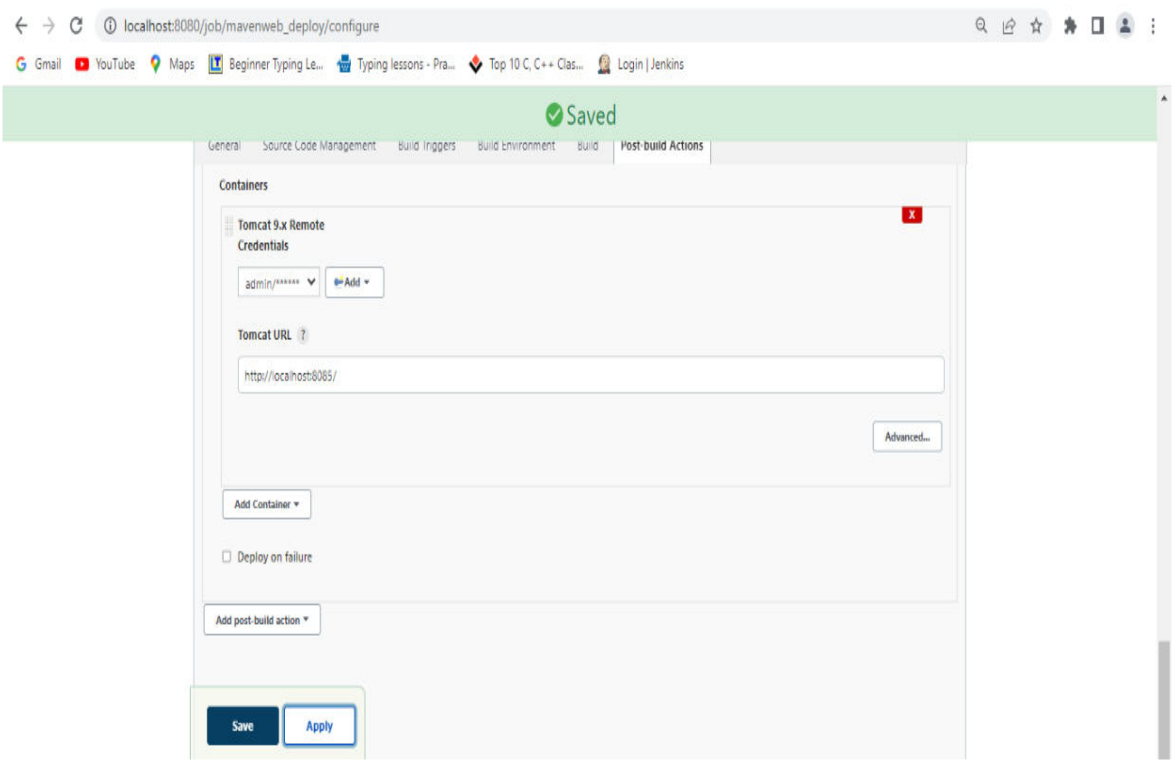
Context path: webpath

Containers

Tomcat 9.x Remote

Credentials: - none - / admin/****** / admin/****** / admin/******

34. Click on Apply and Save.



localhost:8080/job/mavenweb_deploy/configure

Saved

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Containers

Tomcat 9.x Remote

Credentials: admin/****** | Add

Tomcat URL: http://localhost:8085/

Advanced...

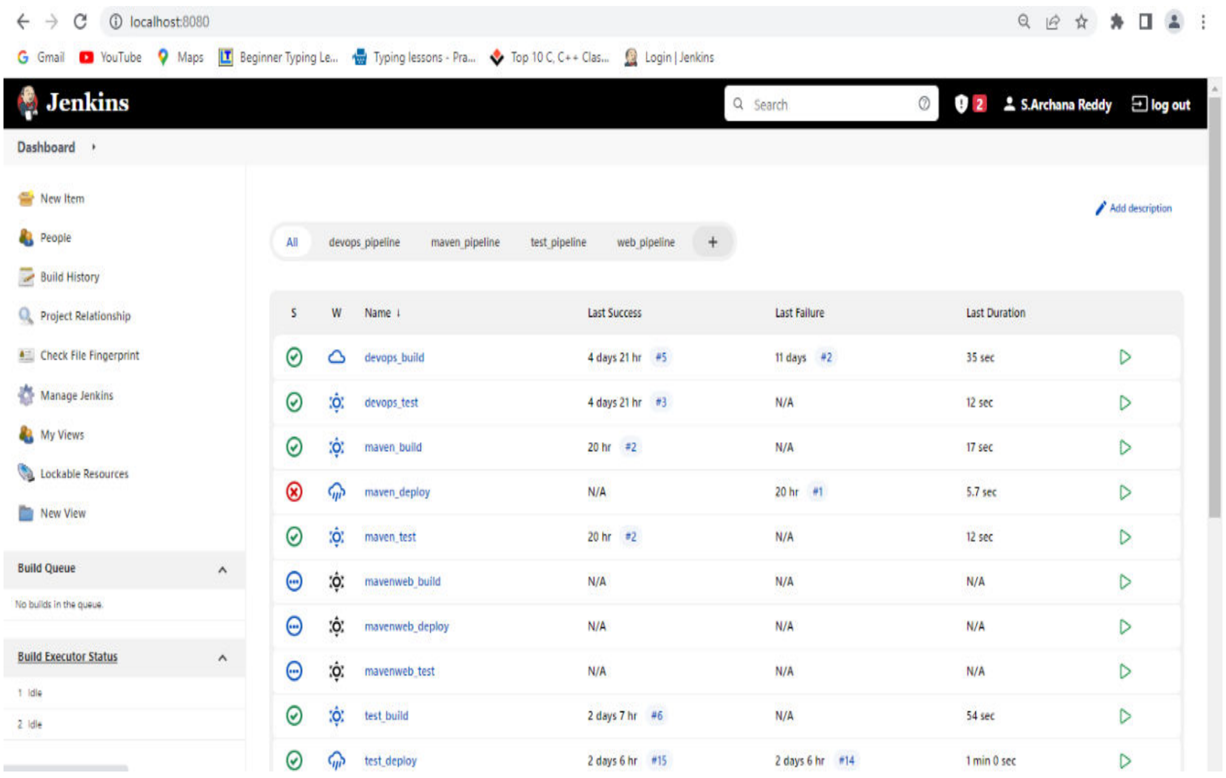
Add Container

☐ Deploy on failure

Add post-build action

Save | Apply

35. We Create a pipeline by clicking on + symbol in the dashboard ->a pipeline is a collection of events or jobs which are interlinked with one another in a sequence.



localhost:8080

Jenkins — S.Archana Reddy — log out

Dashboard

New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, New View

Build Queue: No builds in the queue.

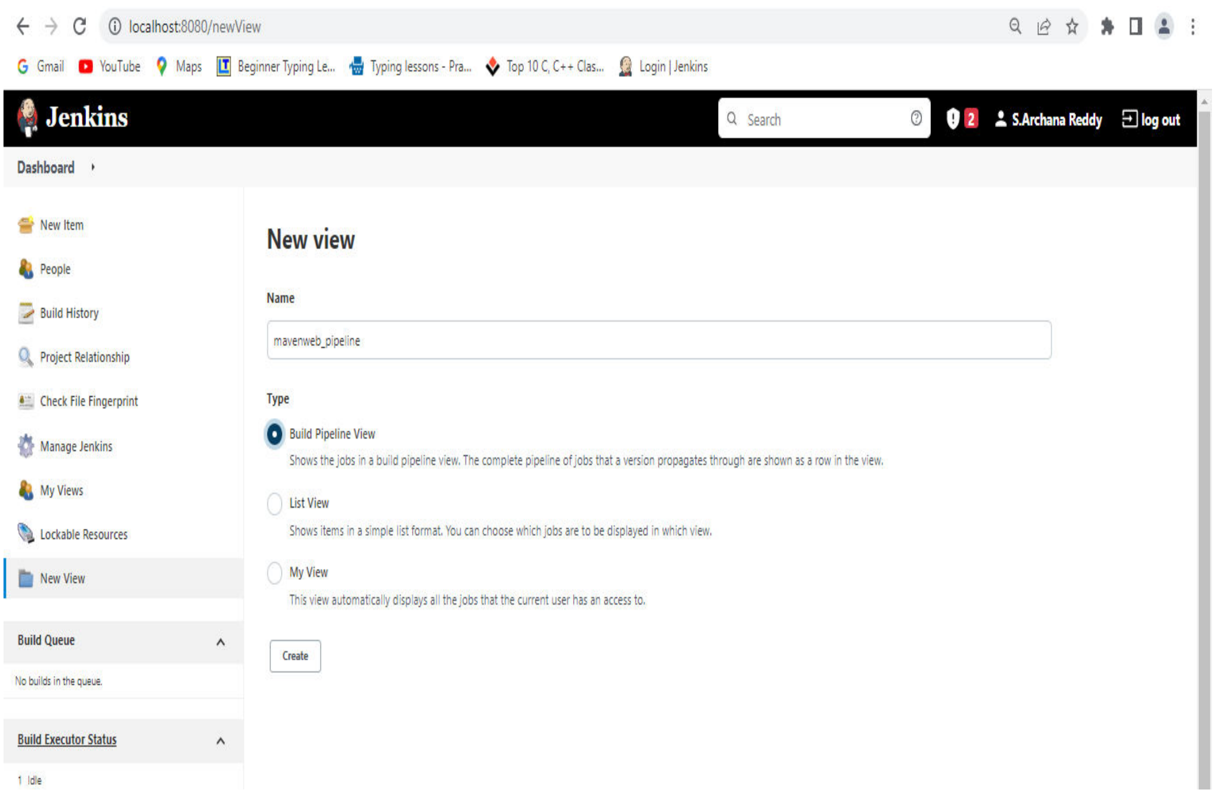
Build Executor Status: 1 Idle, 2 Idle

Add description

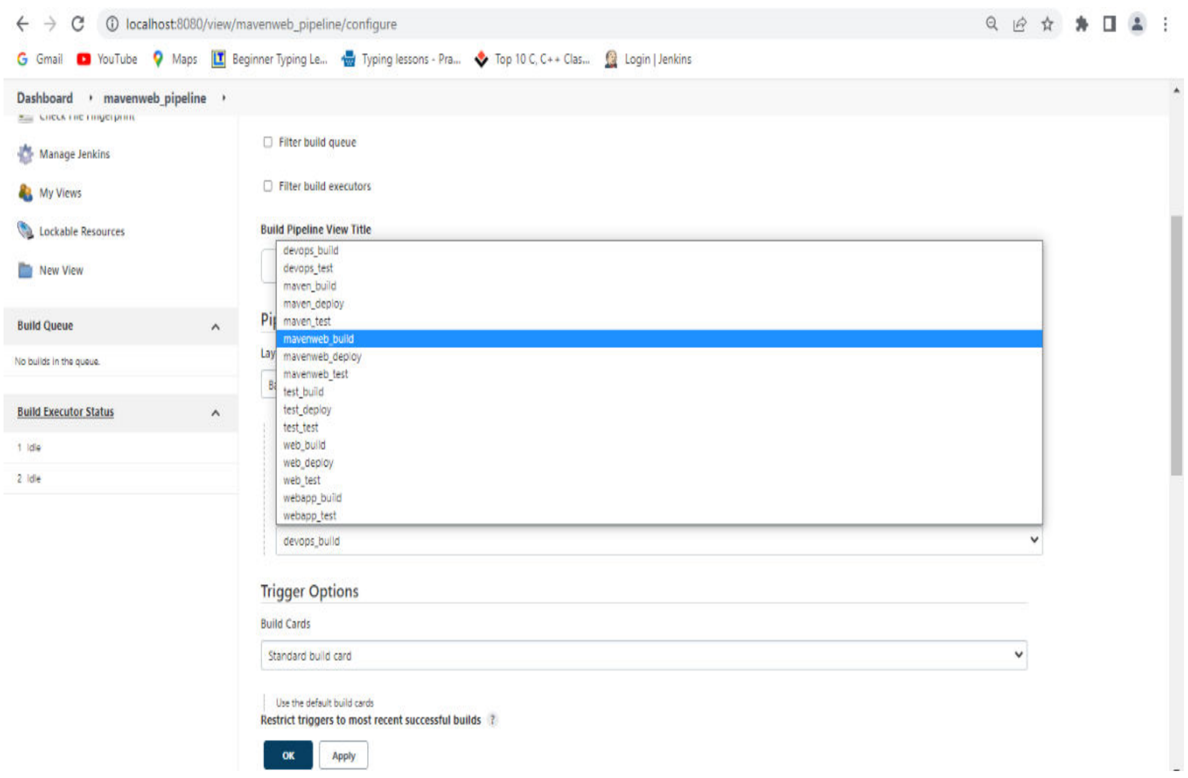
All | devops_pipeline | maven_pipeline | test_pipeline | web_pipeline | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		devops_build	4 days 21 hr #5	11 days #2	35 sec
✓		devops_test	4 days 21 hr #3	N/A	12 sec
✓		maven_build	20 hr #2	N/A	17 sec
✗		maven_deploy	N/A	20 hr #1	5.7 sec
✓		maven_test	20 hr #2	N/A	12 sec
⊖		mavenweb_build	N/A	N/A	N/A
⊖		mavenweb_deploy	N/A	N/A	N/A
⊖		mavenweb_test	N/A	N/A	N/A
✓		test_build	2 days 7 hr #6	N/A	54 sec
✓		test_deploy	2 days 6 hr #15	2 days 6 hr #14	1 min 0 sec

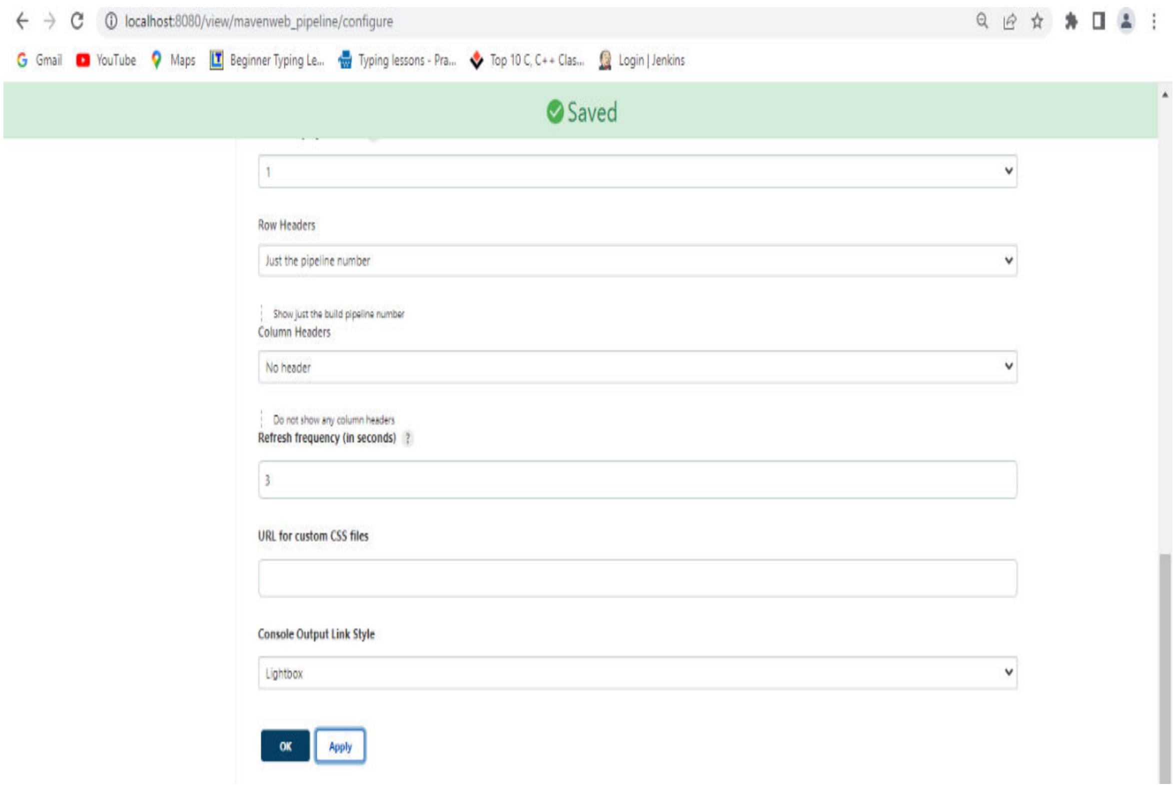
36. Give a name to the pipeline->select Build Pipeline View->create


localhost:8080/newView
Jenkins
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Lockable Resources
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
New view
Name
mavenweb_pipeline
Type
Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.
List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.
My View
This view automatically displays all the jobs that the current user has an access to.
Create

37. Select the first project to trigger the execution->build session of your project



38. Apply and Save


localhost:8080/view/mavenweb_pipeline/configure

Saved

1

Row Headers

Just the pipeline number

Show just the build pipeline number

Column Headers

No header

Do not show any column headers

Refresh frequency (in seconds)

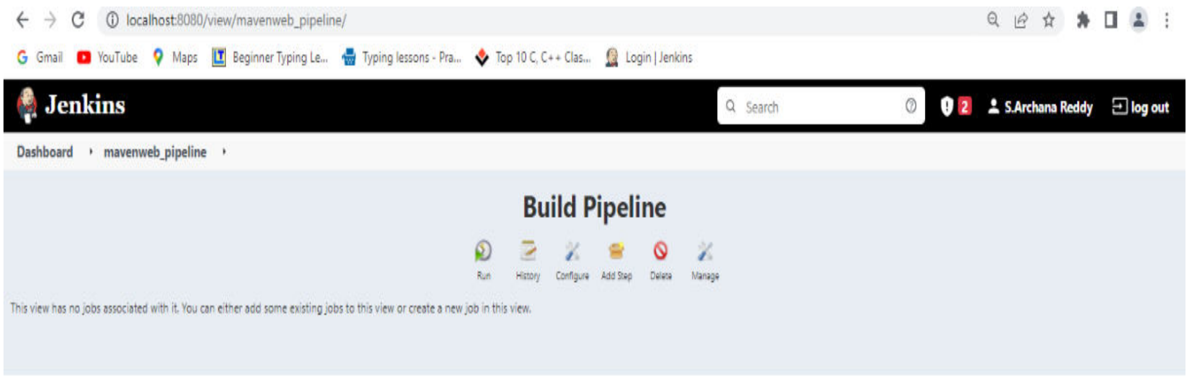
3

URL for custom CSS files

Console Output Link Style

Lightbox

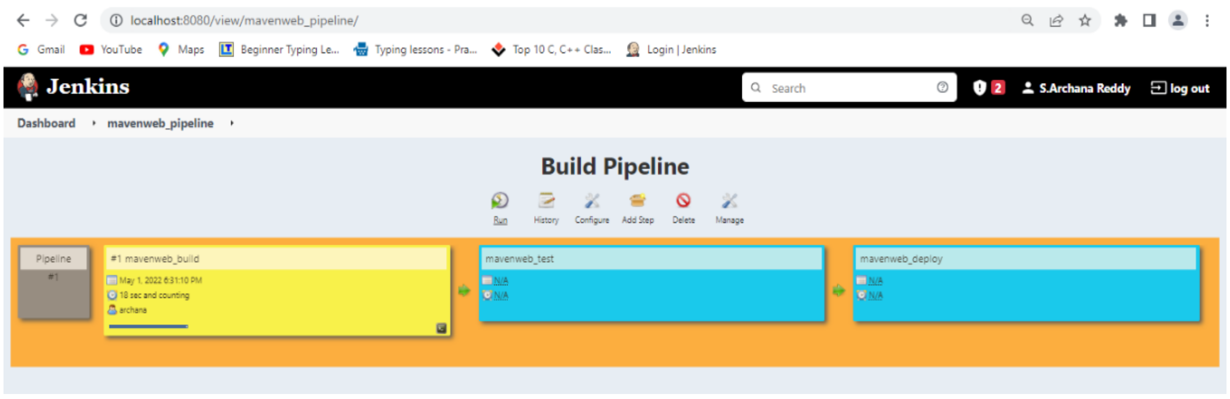
39. This is the console after save. Now click on run.


localhost:8080/view/mavenweb_pipeline/

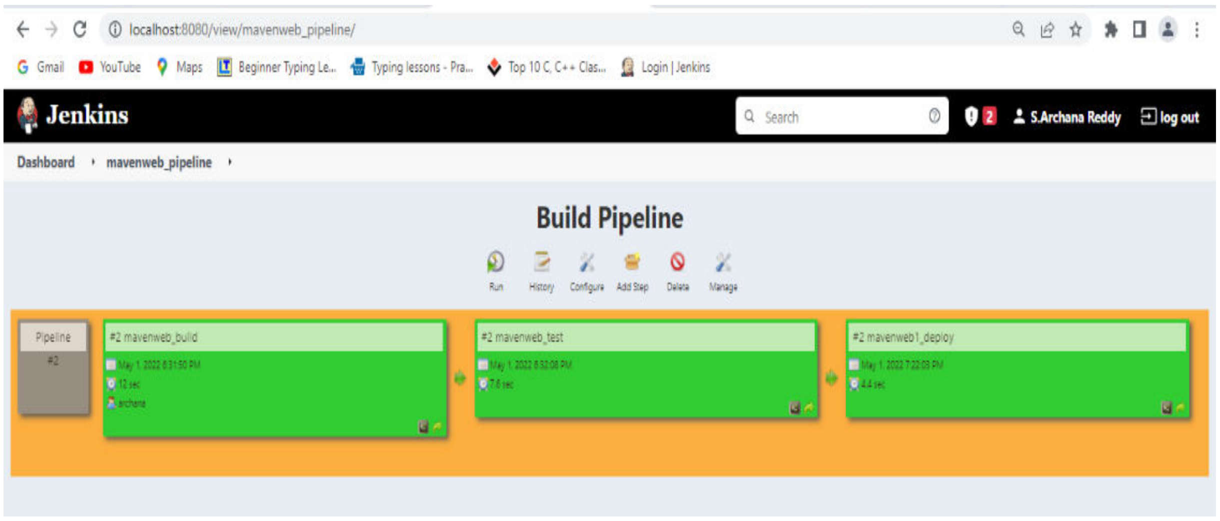
Build Pipeline

This view has no jobs associated with it. You can either add some existing jobs to this view or create a new job in this view.

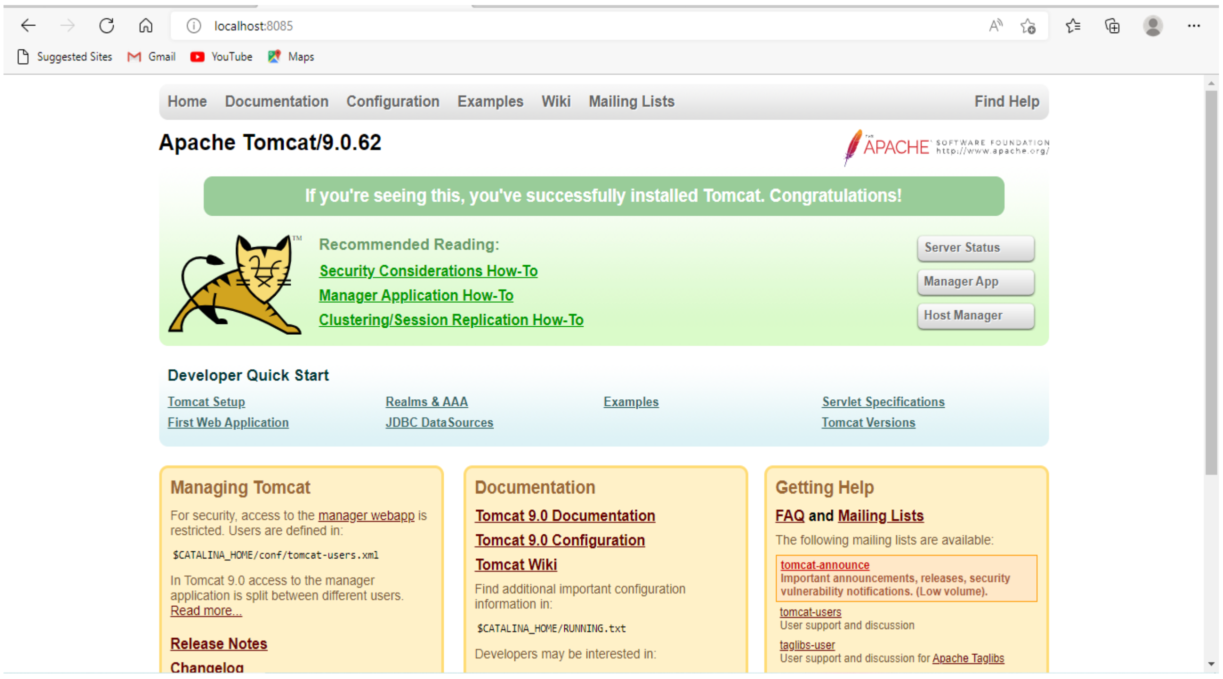
40. After Click on Run -> click on the small black box to open the console to check if the build is success



41. Now we see all the build has success.



42. Now we can run the local host of tomcat, click -> manager App



localhost:8085

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.62

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Server Status
Manager App
Host Manager

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Getting Help

FAQ and Mailing Lists

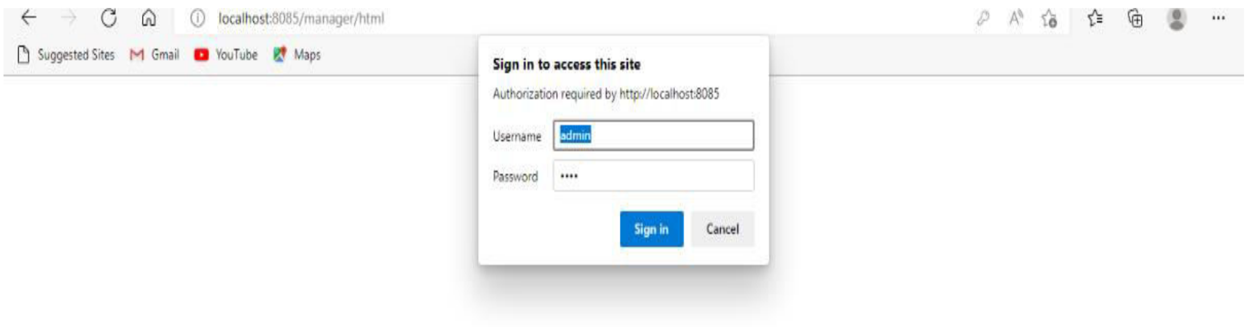
The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

43. It ask for user credentials for login ,provide the credentials of tomcat.



Sign in to access this site

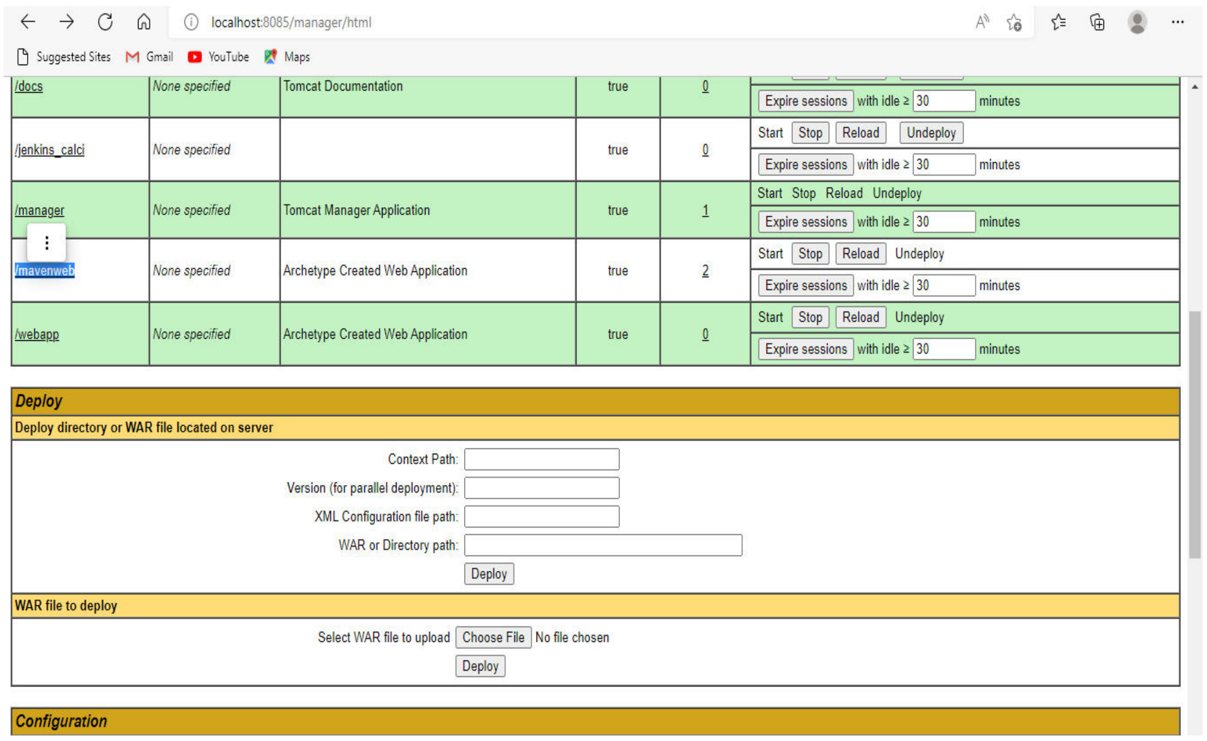
Authorization required by http://localhost:8085

Username: admin

Password: ••••

Sign in Cancel

44. It provide the page without project name which is highlighted.



/docs	None specified	Tomcat Documentation	true	0	Expire sessions with idle ≥ 30 minutes
/jenkins_calci	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/mavenweb	None specified	Archetype Created Web Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/webapp	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

Deploy

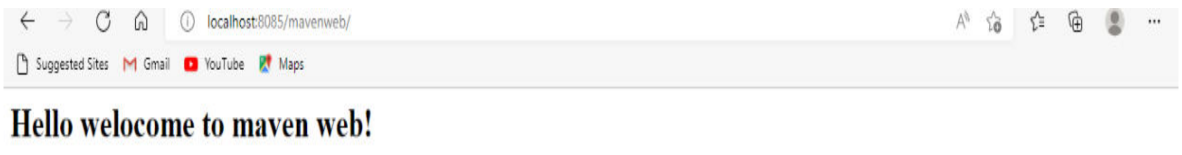
WAR file to deploy

Select WAR file to upload Choose File No file chosen

Deploy

Configuration

45. After clicking on our project, we can see our output here.



Hello welocome to maven web!