

## Module 3: Git And Jenkins

---

### Assignment - 2 Solution

edureka!

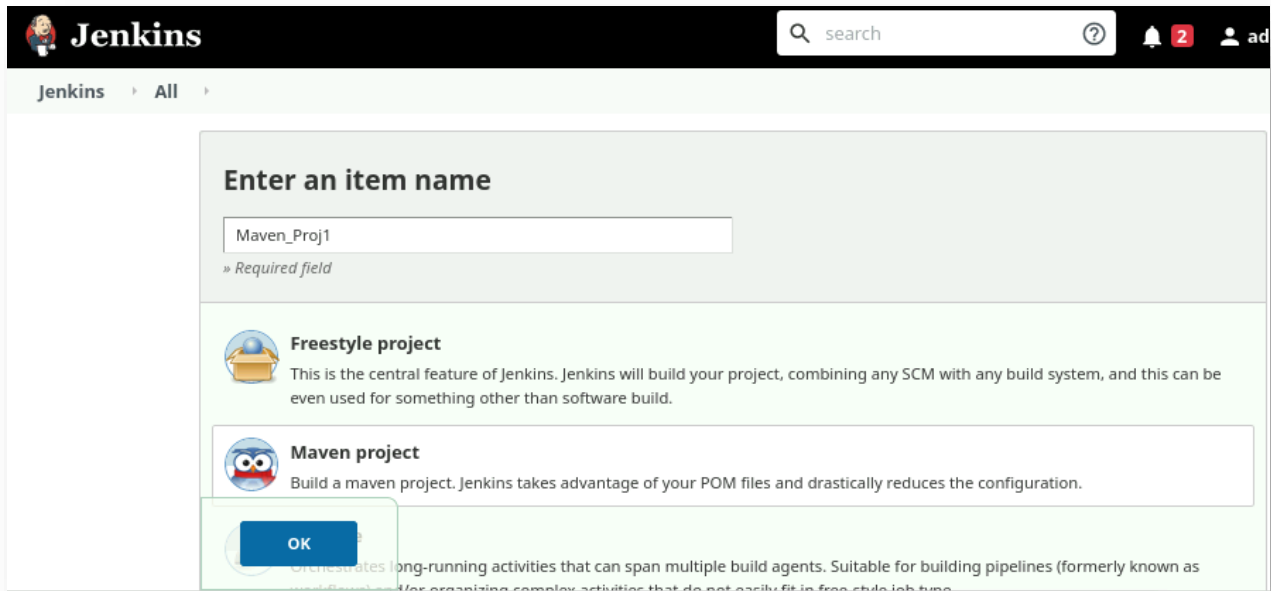
**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Assignment 2 - Solution

Step 1: Go to Jenkins Dashboard and click on the new item.

Step 2: Enter the project name and select Maven Project.

The image shows the Jenkins 'Enter an item name' dialog. At the top, there's a search bar and a user profile icon. Below the header, the breadcrumb 'Jenkins > All' is visible. The main section is titled 'Enter an item name'. A text input field contains 'Maven\_Proj1'. Below the field, it says '» Required field'. There are two options: 'Freestyle project' with a description '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.' and 'Maven project' with a description 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.' An 'OK' button is at the bottom left. A large 'edureka!' watermark is overlaid on the image.

Jenkins

search

Jenkins > All

**Enter an item name**

Maven\_Proj1

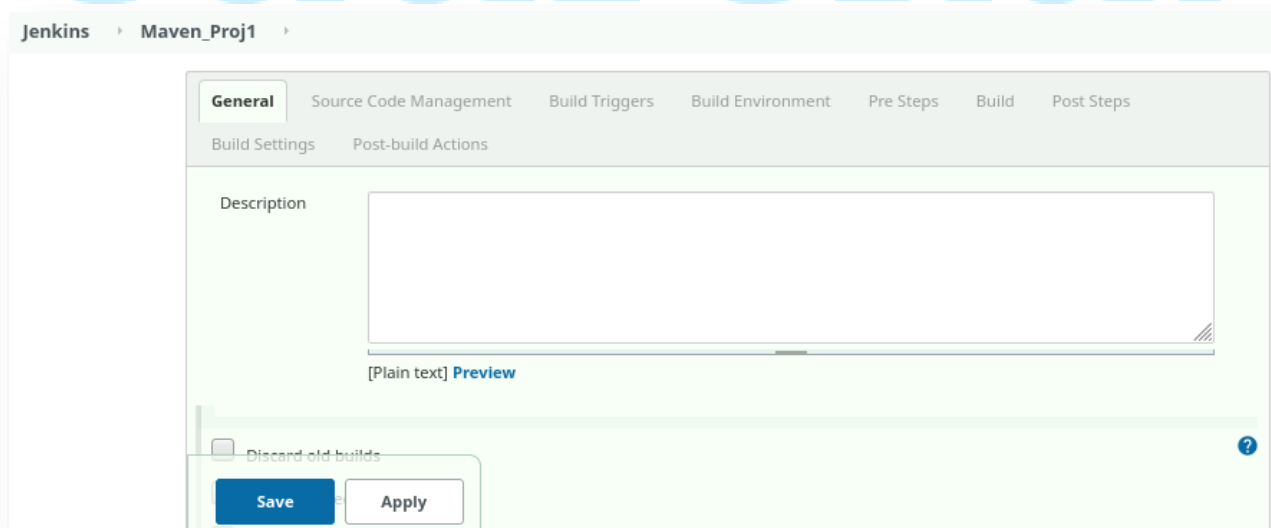
» Required field

**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.

OK

Step 3: Go to configure and start configuring your job.

The image shows the Jenkins 'Configure' page for the job 'Maven\_Proj1'. The breadcrumb is 'Jenkins > Maven\_Proj1'. The 'General' tab is selected, showing a 'Description' text area and a 'Discard old builds' checkbox. At the bottom, there are 'Save' and 'Apply' buttons. Other tabs like 'Source Code Management', 'Build Triggers', 'Build Environment', 'Pre Steps', 'Build', and 'Post Steps' are visible. A large 'edureka!' watermark is overlaid on the image.

Jenkins > Maven\_Proj1

**General** Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps

Build Settings Post-build Actions

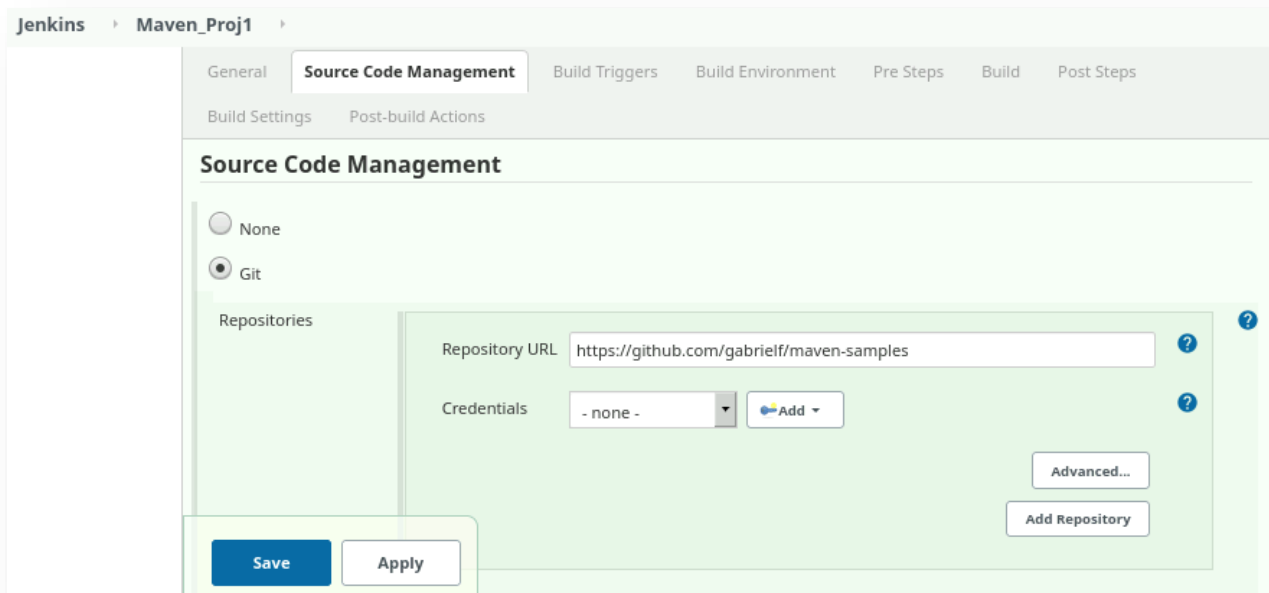
Description

[Plain text] Preview

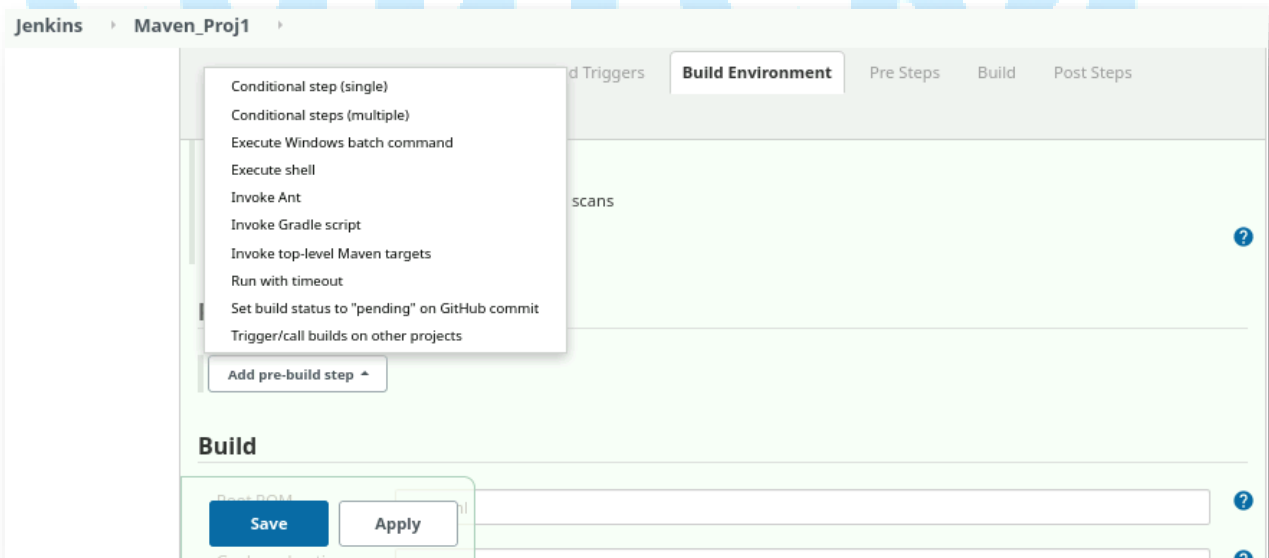
Discard old builds

Save Apply

Step 4: Go to Source Code Management and select the git option and provide the repository path from where you are pulling the code.



Step 5: Click on invoke top-level Maven targets and define the maven goal for your job. If you have not downloaded the Maven plugin, then go to manage plugin → Global tool configuration, and download the Maven plugin for your Jenkins.



Jenkins

Global Tool Configuration

List of Ant installations on this system

Maven

Maven installations

Add Maven

Maven

Name

My\_Maven

☒ Install automatically

Install from Apache

Version

3.8.1

Delete Installer

Add Installer

Save

Apply

Jenkins

Maven\_Proj1

General

Source Code Management

Build Triggers

Build Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Pre Steps

Invoke top-level Maven targets

Maven Version

My Maven

Goals

compile

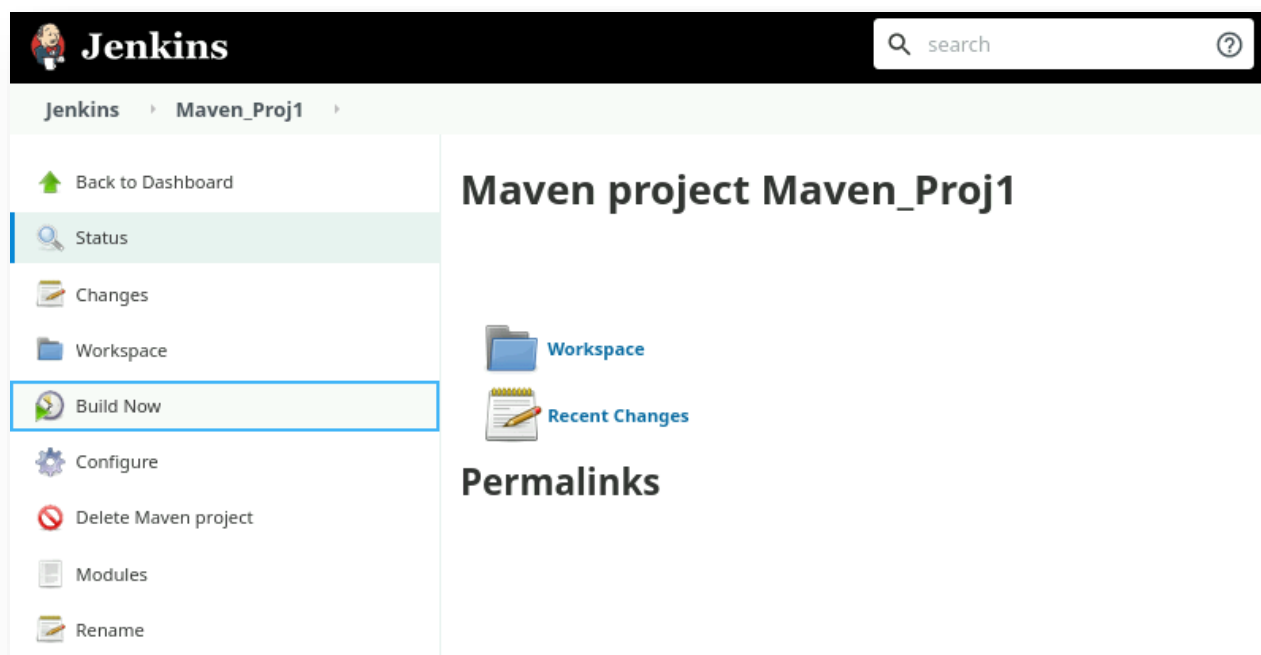
Advanced...

Add pre-build step

Save

Apply

Step 6: Go to build now and then check your console output by clicking on the blue or red dot, whichever comes after the build. Blue is a sign for a successful build, whereas red is for build failure.



Jenkins › Maven\_Proj1 ›

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

**Build History** trend ^

find X

#1 May 17, 2021 4:29 AM

Workspace

Recent Changes

## Permalinks

Jenkins › Maven\_Proj1 › #1

- Back to Project
- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete build '#1'
- Git Build Data
- Redeploy Artifacts
- Test Result

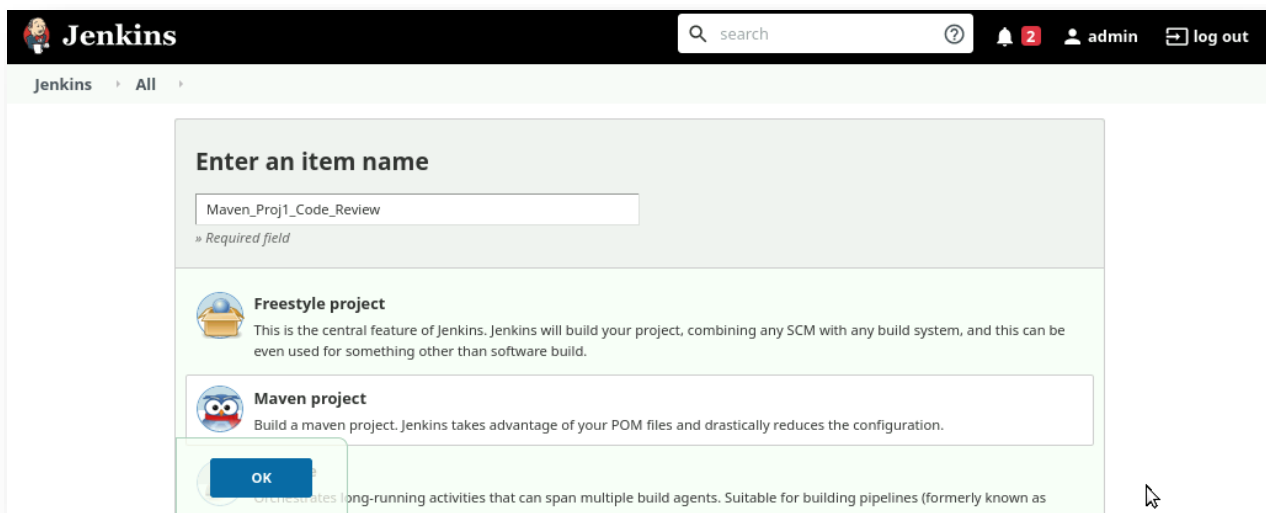
## Console Output

Started by user [admin](#)  
Running as SYSTEM  
Building on master in workspace /var/lib/jenkins/workspace/Maven\_Proj1  
The recommended git tool is: NONE  
No credentials specified  
Cloning the remote Git repository  
Cloning repository <https://github.com/gabrielf/maven-samples>  
> git init /var/lib/jenkins/workspace/Maven\_Proj1 # timeout=10  
Fetching upstream changes from <https://github.com/gabrielf/maven-samples>  
> git --version # timeout=10  
> git --version # 'git version 2.17.1'  
> git fetch --tags --progress -- <https://github.com/gabrielf/maven-samples> +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git config remote.origin.url <https://github.com/gabrielf/maven-samples> # timeout=10  
> git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
Avoid second fetch  
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10  
Checking out Revision 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b (refs/remotes/origin/master)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b # timeout=10

```
Jenkins  ▸ Maven_Proj1  ▸ #1

[INFO] Total time: 18.424 s
[INFO] Finished at: 2021-05-17T04:29:52Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/single-module/pom.xml to com.example.maven-
samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/single-module/target/single-module-project.jar to
com.example.maven-samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/multi-module/pom.xml to com.example.maven-
samples/multi-module-parent/1.0-SNAPSHOT/multi-module-parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/pom.xml to com.example.maven-samples/parent/1.0-
SNAPSHOT/parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/multi-module/webapp/pom.xml to com.example.maven-
samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/multi-module/webapp/target/webapp.war to
com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/multi-module/server/pom.xml to com.example.maven-
samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1/multi-module/server/target/server.jar to
com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

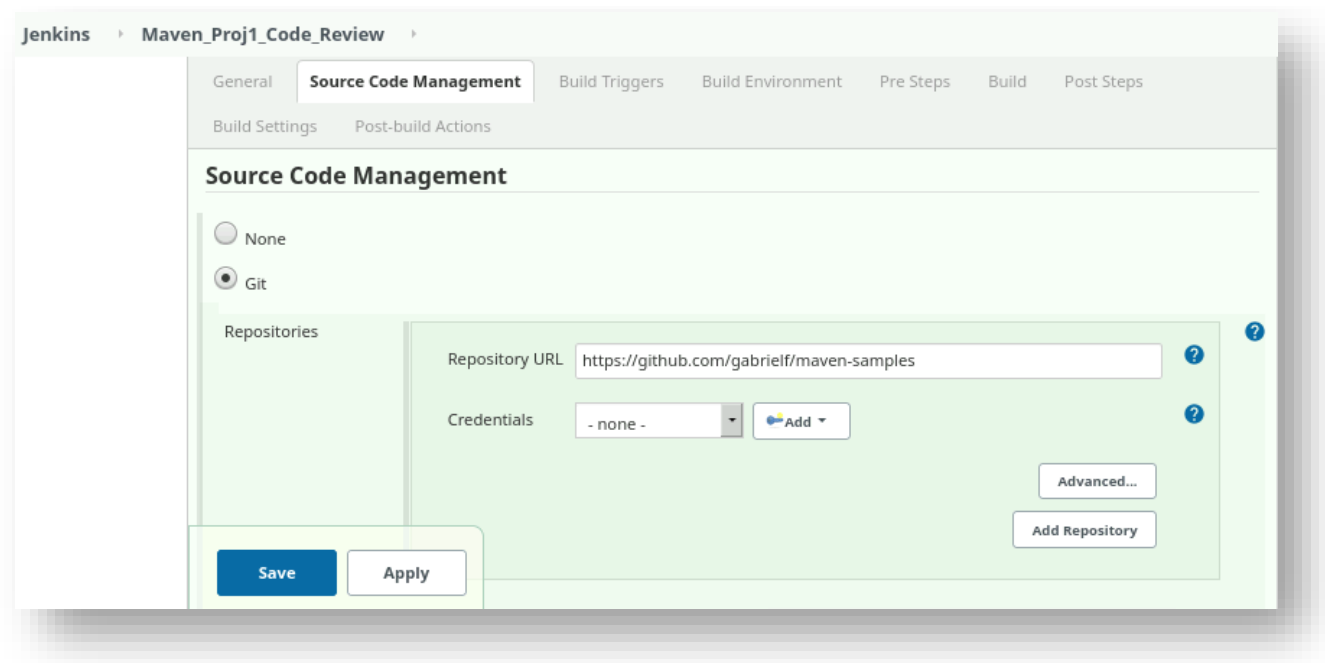
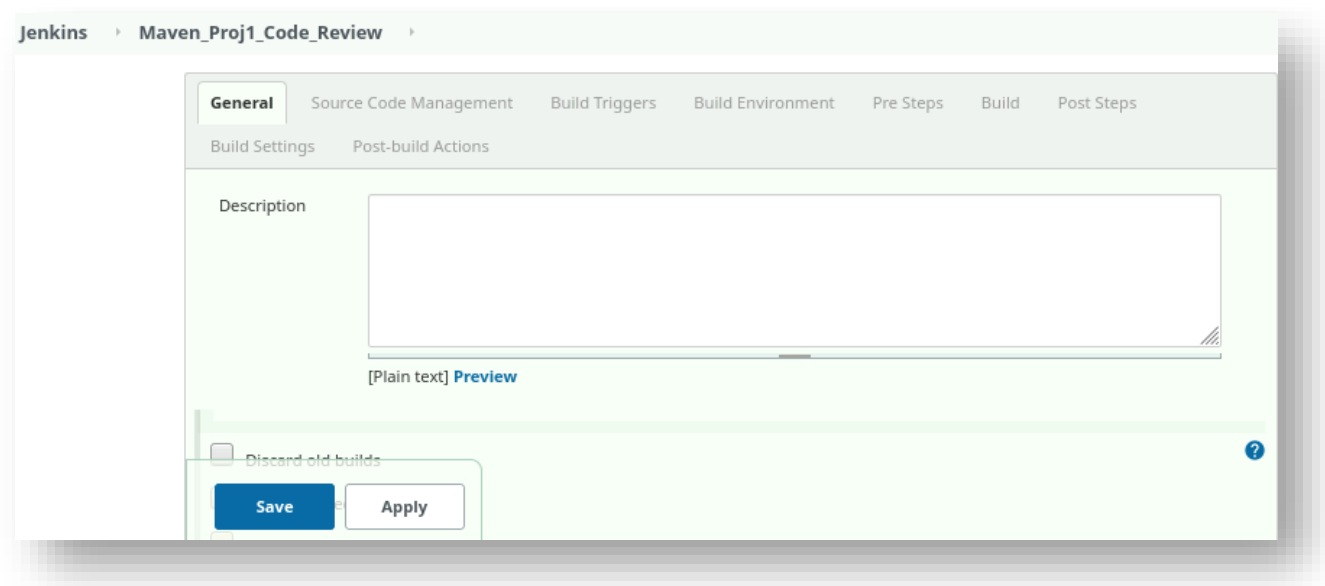
Step 7: Now, create a new job, and it will also be a Maven project.



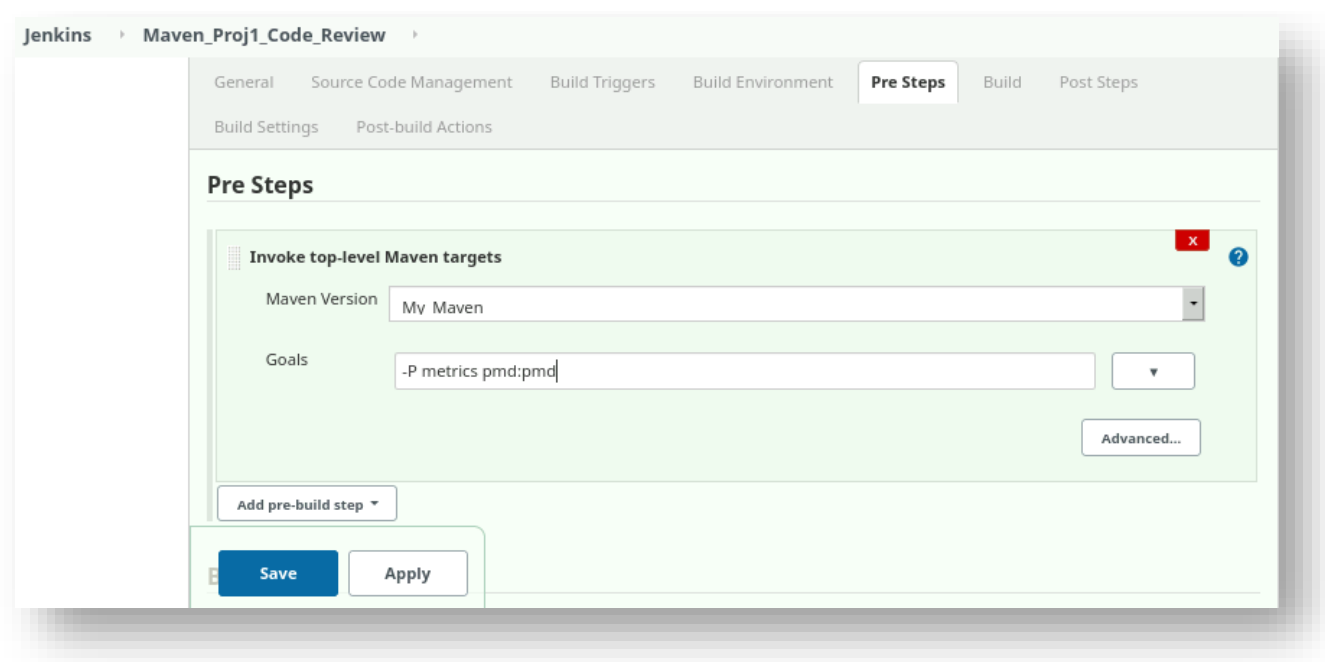
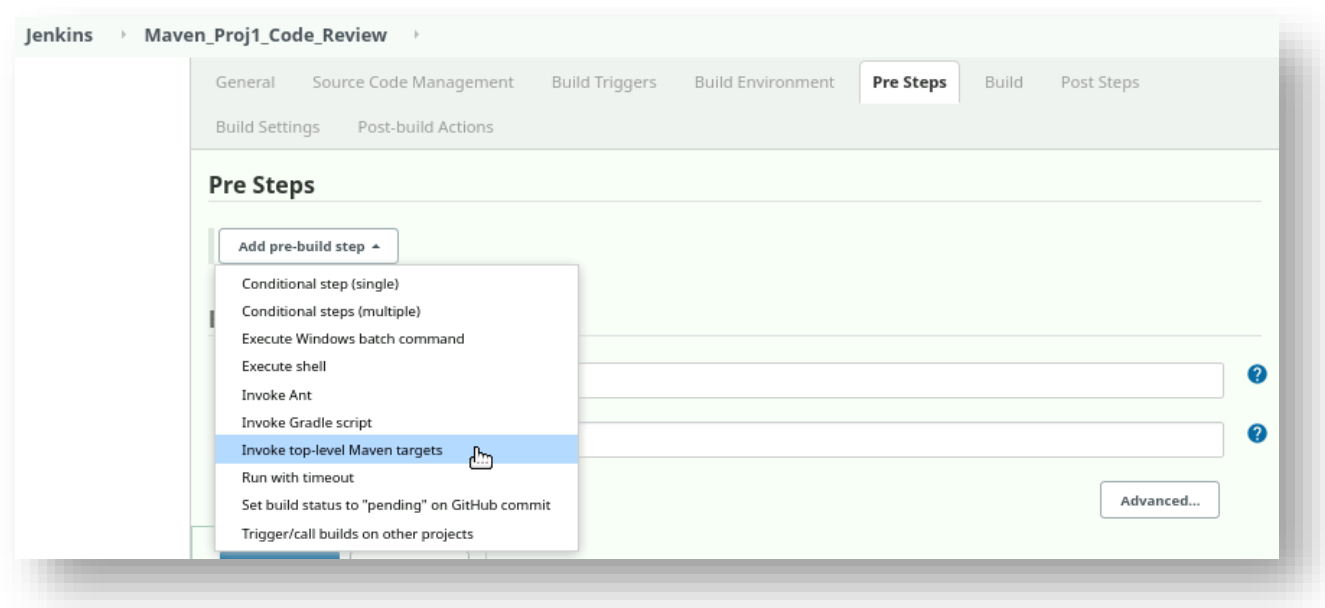
Step 8: Configure the job by providing the git path and defining the job goals.

Below is the link to the Github repository path.

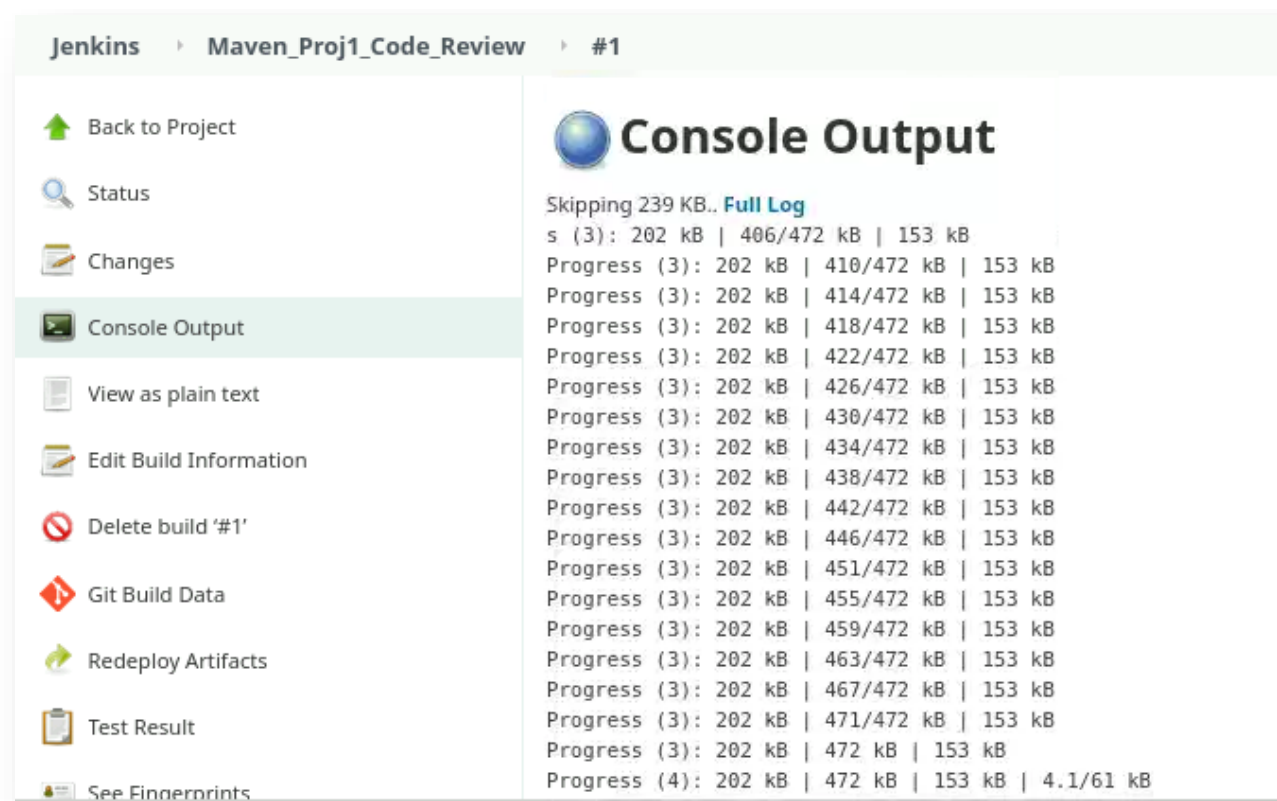
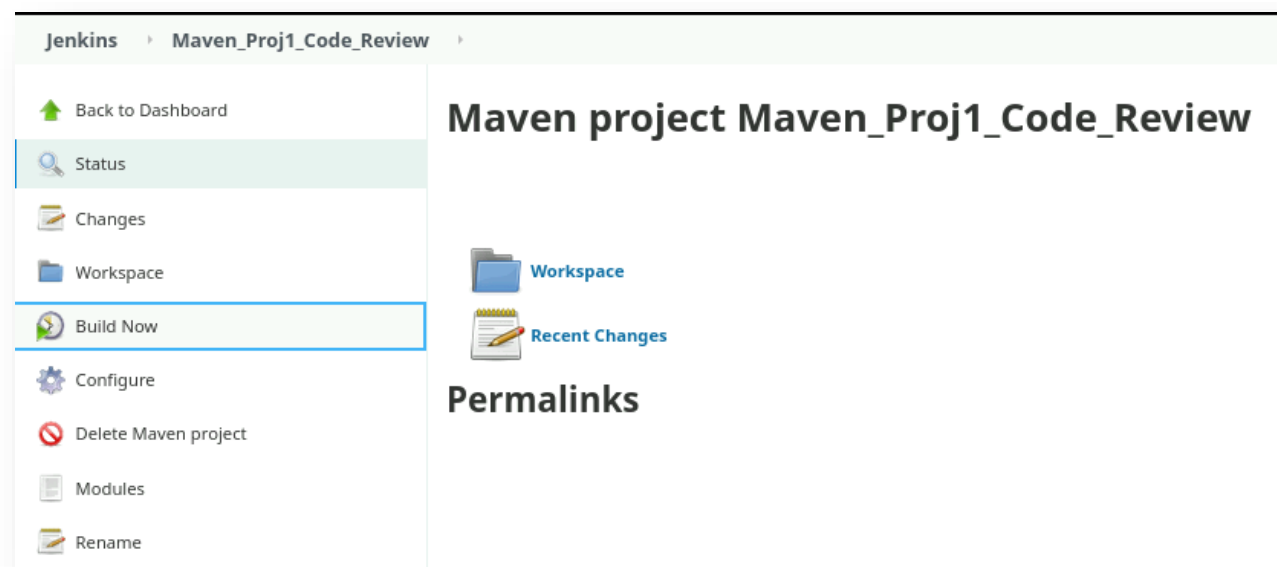
<https://github.com/gabrielf/maven-samples>







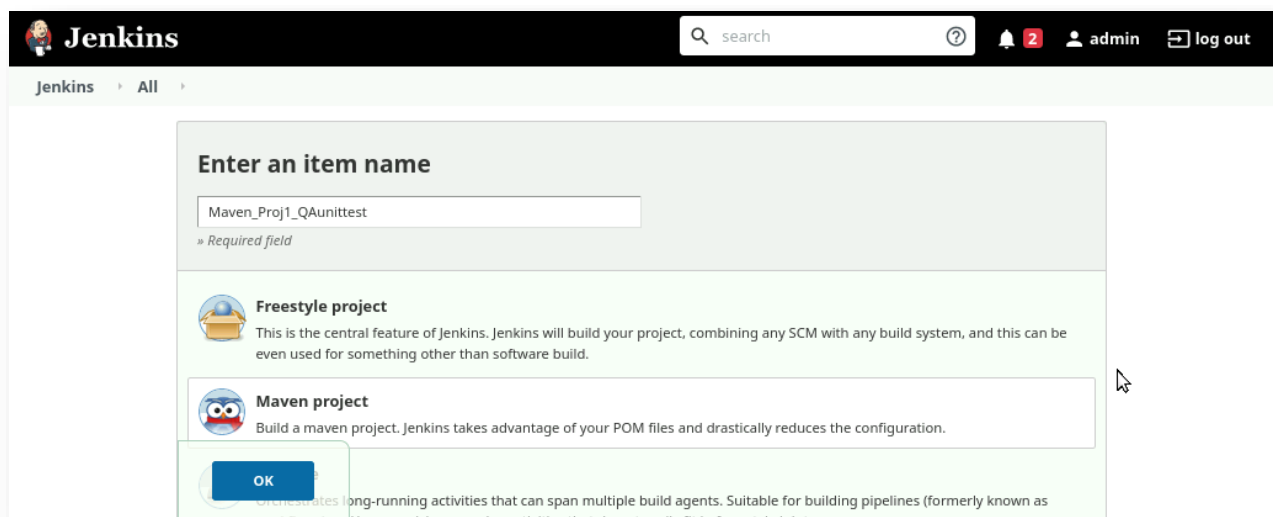
Step 9: Click on build now and check the console output.



```
Jenkins > Maven_Proj1_Code_Review > #1

Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/single-module/pom.xml to com.example.maven-samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/single-module/target/single-module-project.jar to com.example.maven-samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/multi-module/pom.xml to com.example.maven-samples/multi-module-parent/1.0-SNAPSHOT/multi-module-parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/pom.xml to com.example.maven-samples/parent/1.0-SNAPSHOT/parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/multi-module/webapp/pom.xml to com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/multi-module/webapp/target/webapp.war to com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/multi-module/server/pom.xml to com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_Code_Review/multi-module/server/target/server.jar to com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

Step 10: Now, create a new job which will also be a Maven project.



Step 11: Configure the job by providing the git path and defining the job goals.

Jenkins › Maven\_Proj1\_QAunittest ›

General Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps

Build Settings Post-build Actions

Description

[Plain text] [Preview](#)

Discard old builds

Save Apply

Jenkins › Maven\_Proj1\_QAunittest ›

General **Source Code Management** Build Triggers Build Environment Pre Steps Build Post Steps

Build Settings Post-build Actions

**Source Code Management**

☐ None

☒ Git

Repositories

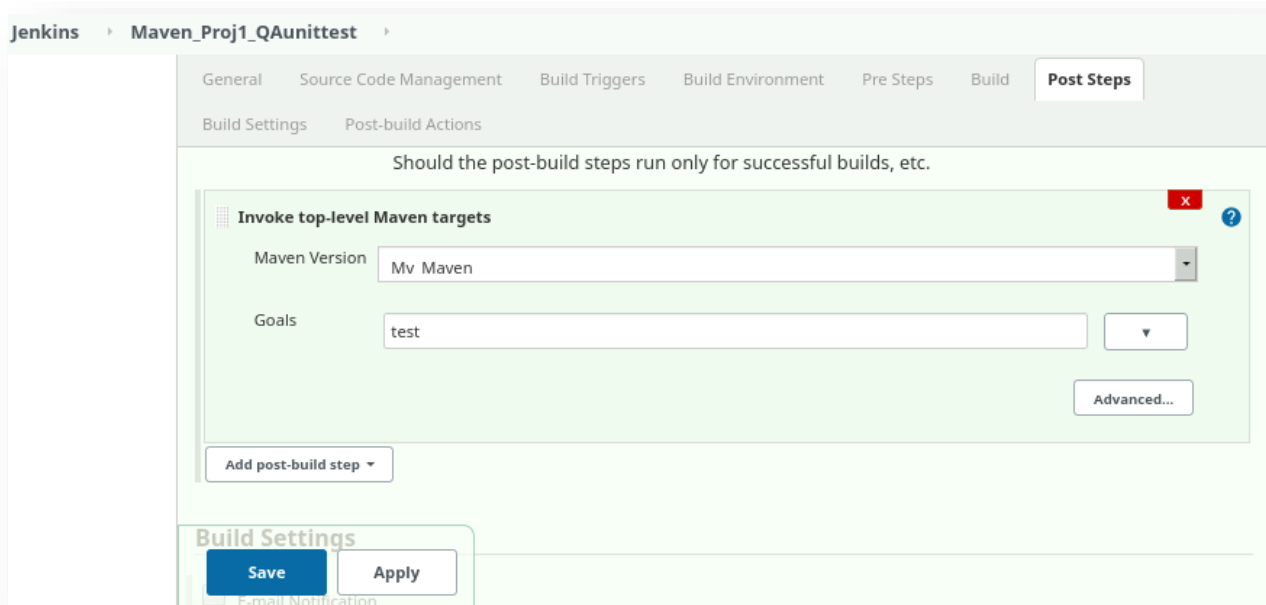
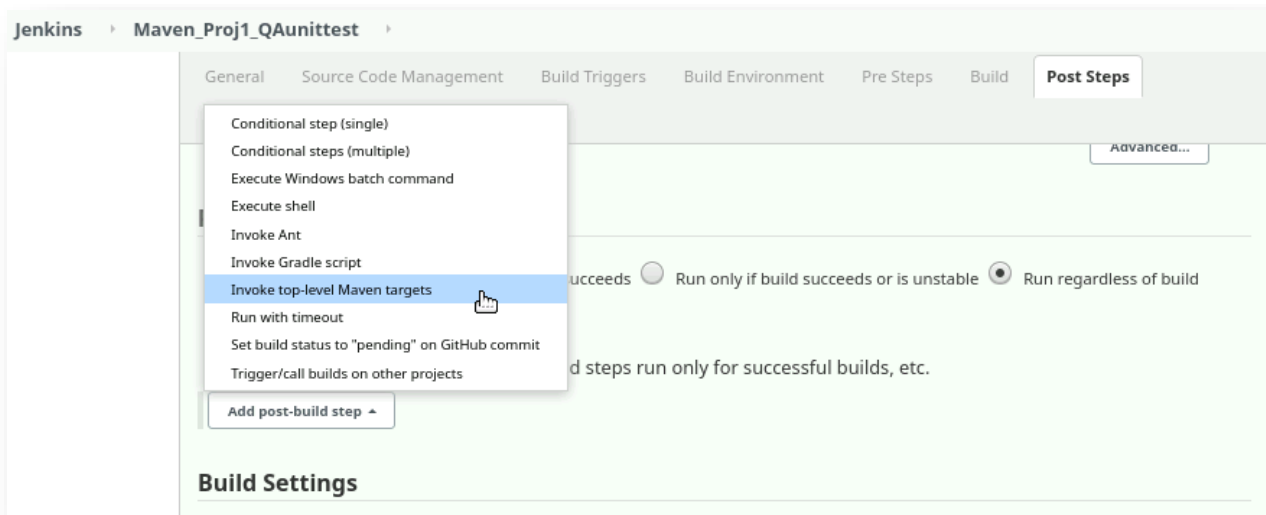
Repository URL

Credentials  [Add](#)

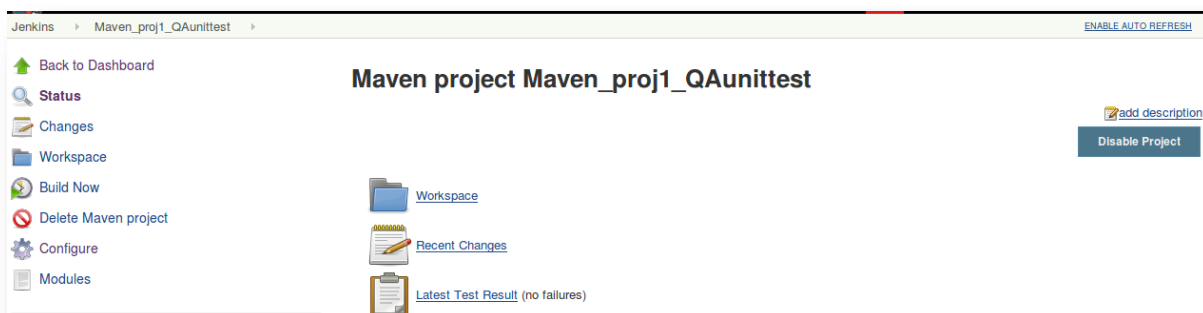
[Advanced...](#)

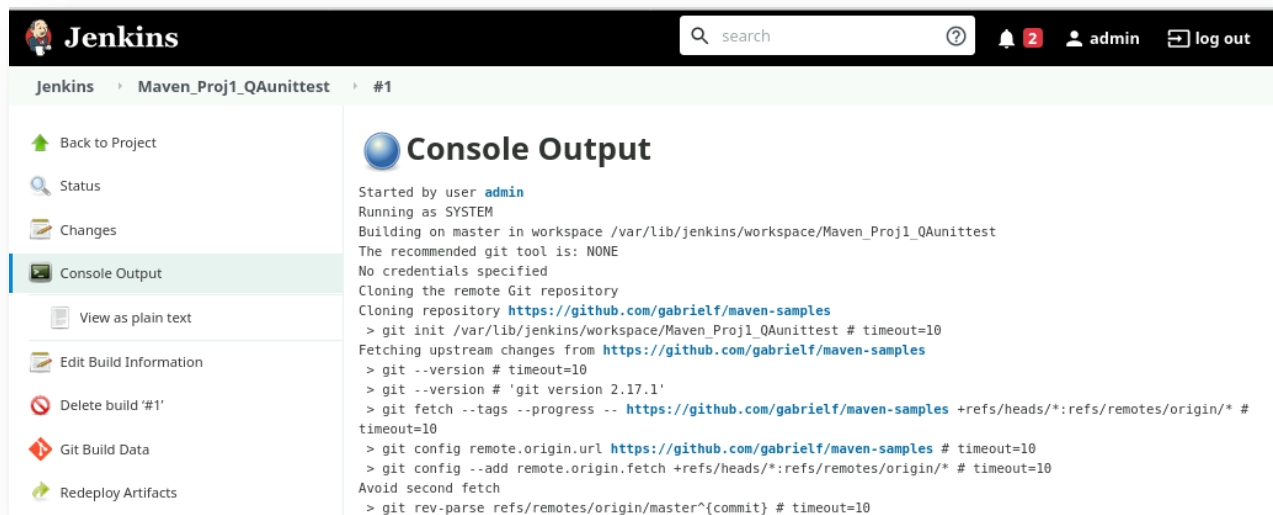
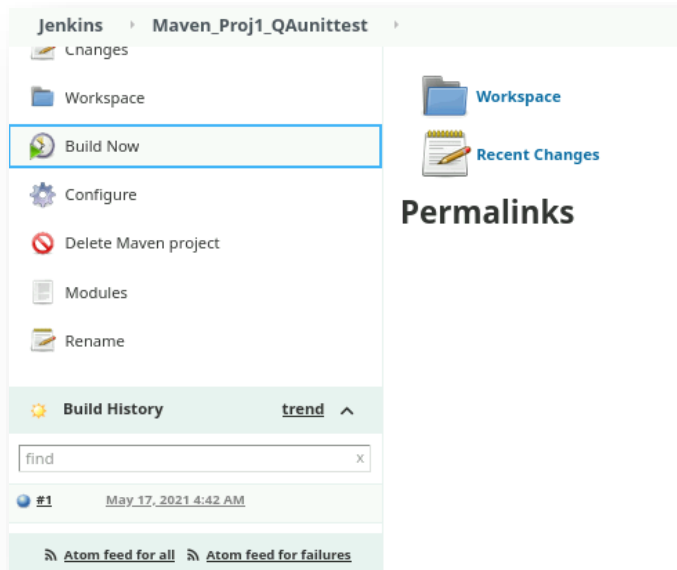
[Add Repository](#)

Save Apply



Step 12: Click on build now and go on console output after the job is built.



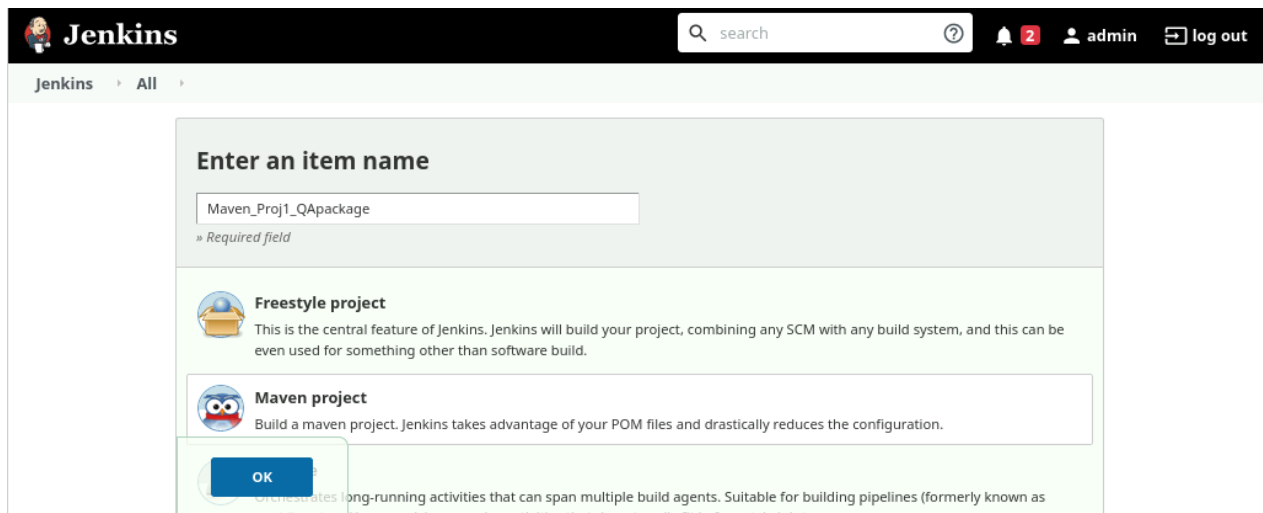


```
Jenkins  ▸ Maven_Proj1_QAunittest  ▸ #1

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

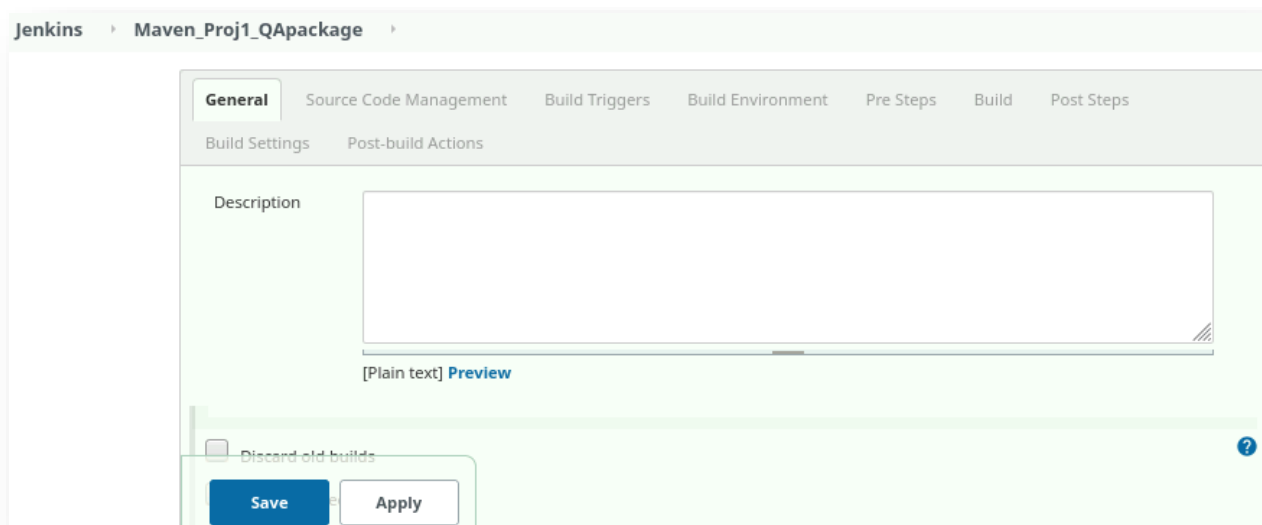
[INFO]
[INFO] -----< com.example.maven-samples:parent >-----
[INFO] Building Parent 1.0-SNAPSHOT [5/5]
[INFO] -----[ pom ]-----
[INFO] -----
[INFO] Reactor Summary for Parent 1.0-SNAPSHOT:
[INFO]
[INFO] Multi Module Project Parent ..... SUCCESS [ 0.004 s]
[INFO] Server ..... SUCCESS [ 1.575 s]
[INFO] Webapp ..... SUCCESS [ 0.294 s]
[INFO] A Single Maven Module ..... SUCCESS [ 0.372 s]
[INFO] Parent ..... SUCCESS [ 0.001 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.461 s
[INFO] Finished at: 2021-05-17T04:43:19Z
[INFO] -----
Finished: SUCCESS
```

Step 13: Now, create a new job, and it will also be a Maven project.



The image shows the Jenkins 'New Item' configuration page. At the top, the Jenkins logo and navigation bar are visible. Below the header, the breadcrumb 'Jenkins ▸ All ▸' is shown. The main section is titled 'Enter an item name' and contains a text input field with the value 'Maven\_Proj1\_QApackage'. Below the input field is a small note '» Required field'. Underneath, there are two selectable options: 'Freestyle project' with a brief description, and 'Maven project' with a brief description. The 'Maven project' option is highlighted with a blue border. At the bottom left, there is a blue 'OK' button. The bottom of the page shows a partially visible description for 'Pipeline'.

Step 14: Click on configure and configure your project.



The screenshot shows the Jenkins configuration page for a project named 'Maven\_Proj1\_QApackage'. The 'General' tab is selected. It features a 'Description' text area, a 'Discard old builds' checkbox, and 'Save' and 'Apply' buttons at the bottom.

Jenkins > Maven\_Proj1\_QApackage >

**General** Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps

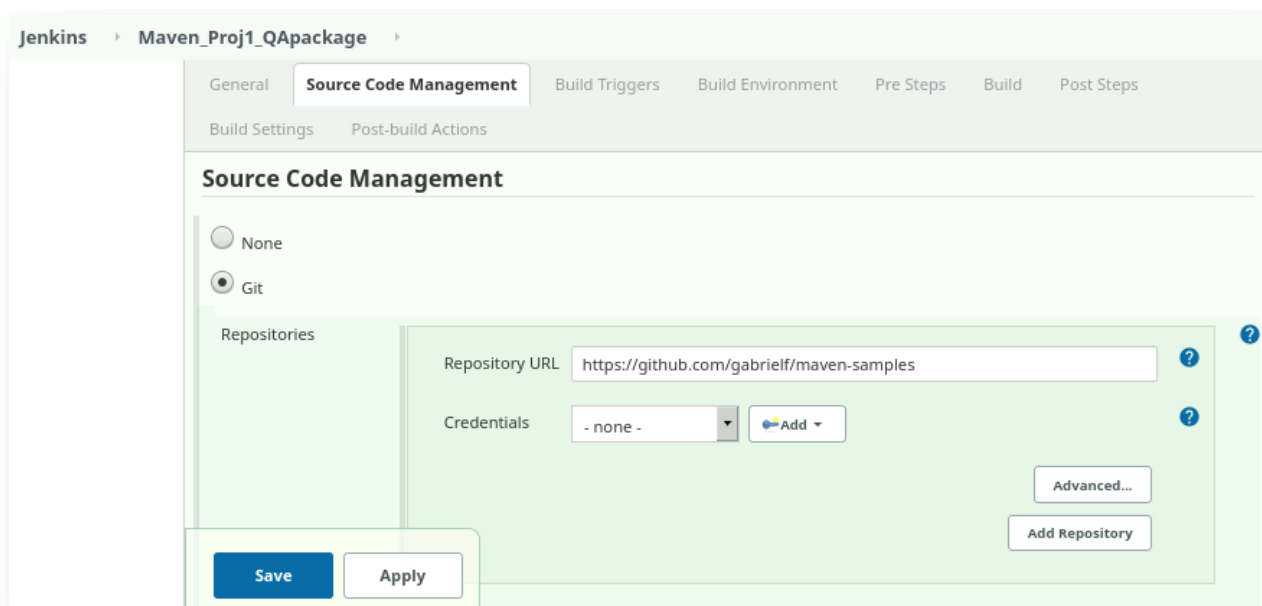
Build Settings Post-build Actions

Description

[Plain text] [Preview](#)

☐ Discard old builds

[Save](#) [Apply](#)



The screenshot shows the Jenkins configuration page for the same project, but with the 'Source Code Management' tab selected. It shows options for 'None' or 'Git', a 'Repository URL' field with the value 'https://github.com/gabrielf/maven-samples', a 'Credentials' dropdown set to '- none -', and buttons for 'Advanced...', 'Add Repository', 'Save', and 'Apply'.

Jenkins > Maven\_Proj1\_QApackage >

General **Source Code Management** Build Triggers Build Environment Pre Steps Build Post Steps

Build Settings Post-build Actions

**Source Code Management**

☐ None

☒ Git

Repositories

Repository URL

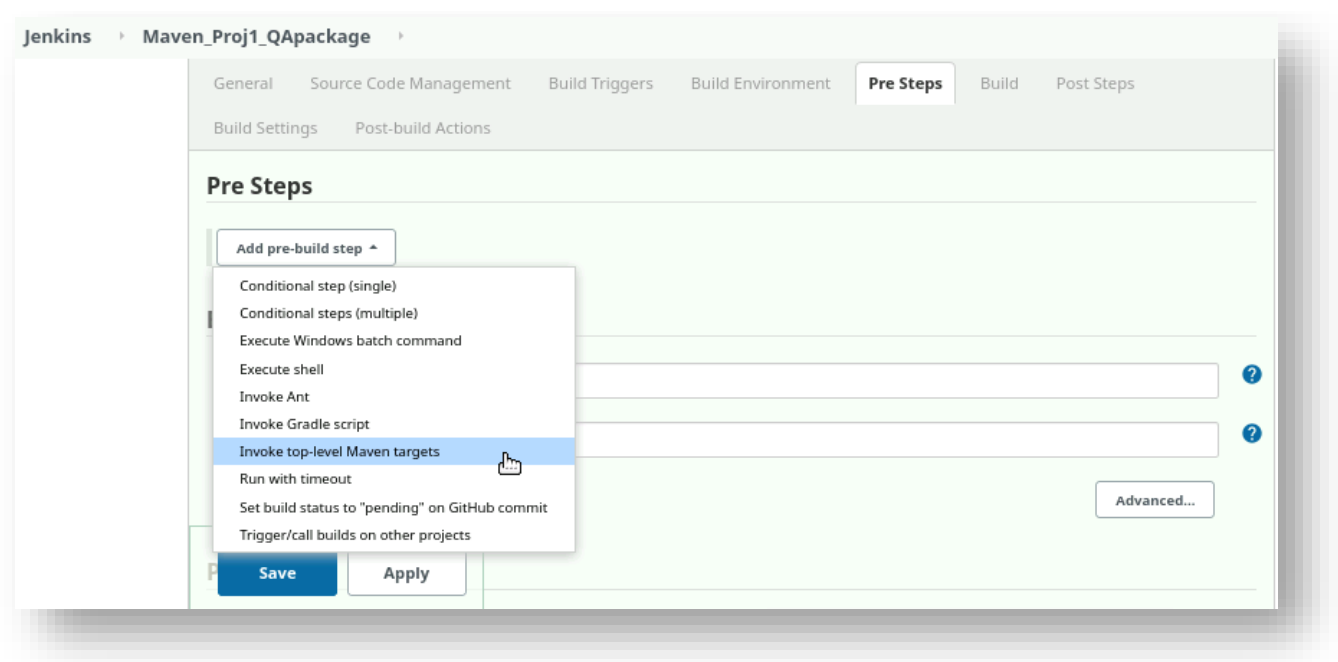
Credentials  [Add](#)

[Advanced...](#)

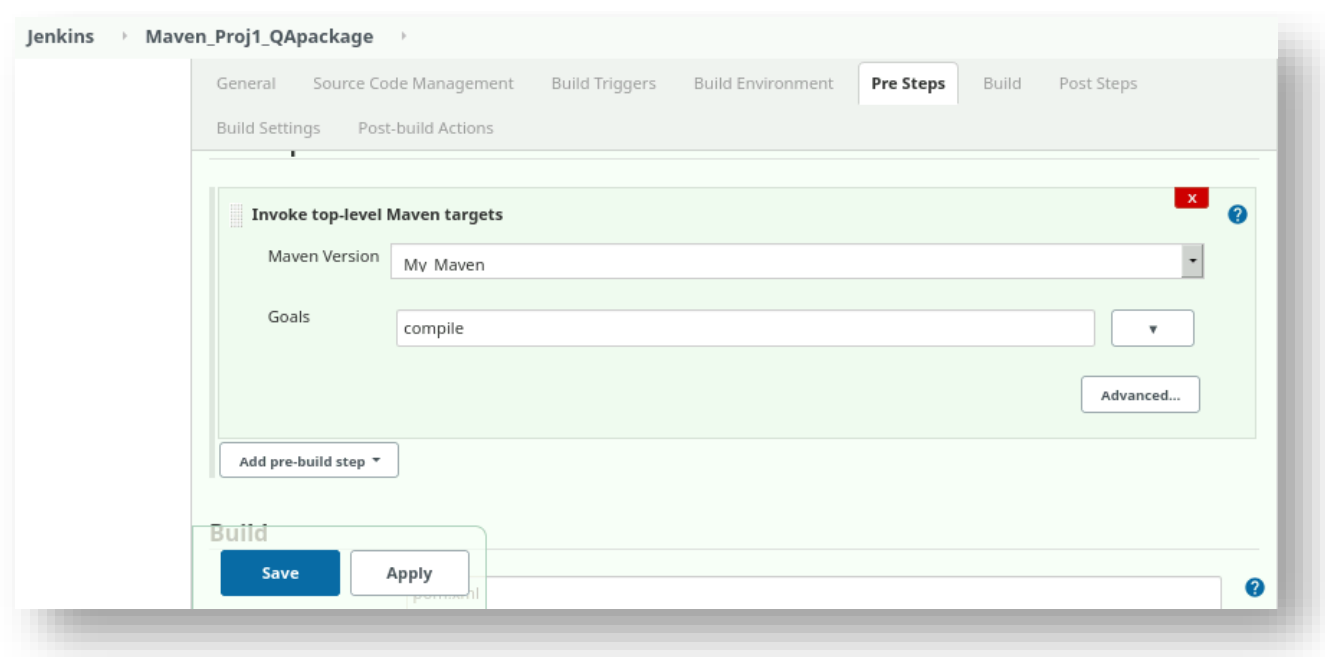
[Add Repository](#)

[Save](#) [Apply](#)

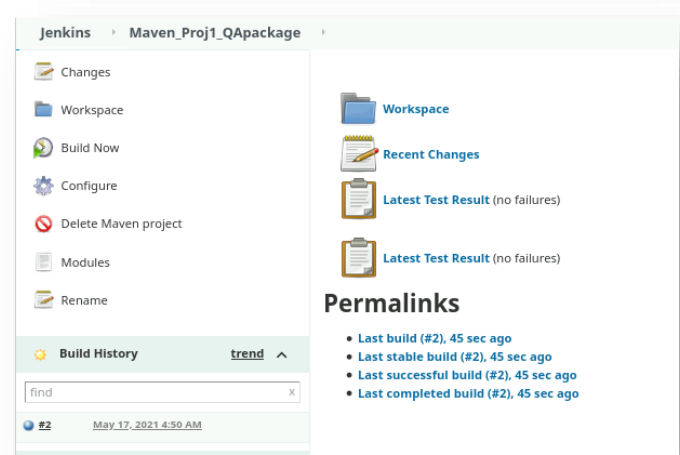
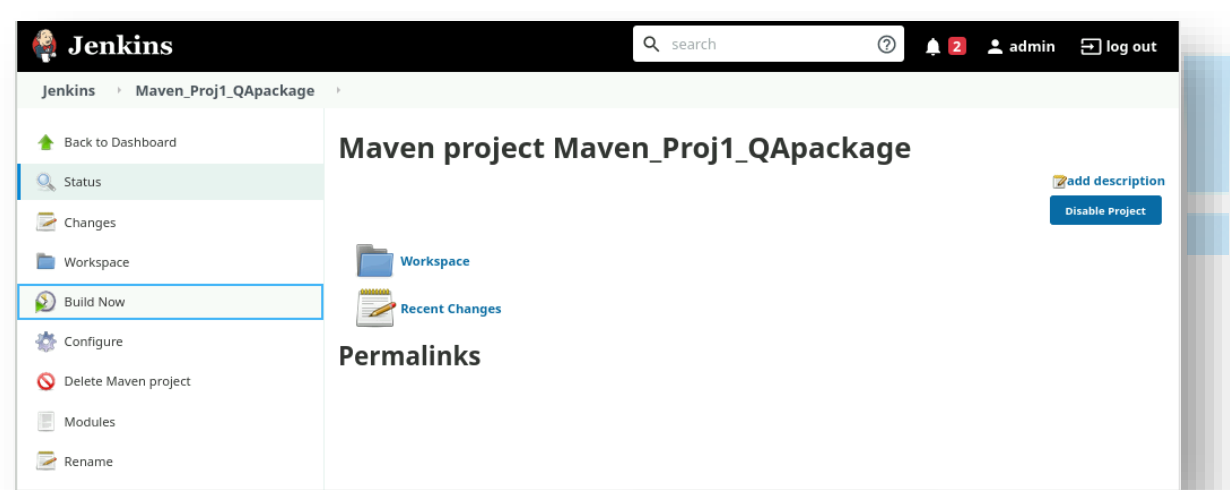




edureka!



Step 15: After configuring your job, click on build now and then check the console output.



```

Jenkins  Maven_Proj1_QApackage  #2
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#2'
Git Build Data
Redeploy Artifacts
Test Result
See Fingerprints

Building on master in workspace /var/lib/jenkins/workspace/Maven_Proj1_QApackage
The recommended git tool is: NONE
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/gabrielf/maven-samples # timeout=10
Fetching upstream changes from https://github.com/gabrielf/maven-samples
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/gabrielf/maven-samples +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b # timeout=10
Commit message: "Merge pull request #1 from narendra2244/patch-1"
> git rev-list --no-walk 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b # timeout=10
[Maven_Proj1_QApackage] $ /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/My_Maven/bin/mvn compile
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for com.example.maven-
samples:server:jar:1.0-SNAPSHOT
[WARNING] Reporting configuration should be done in <reporting> section, not in maven-site-plugin <configuration>
as reportPlugins parameter.

```

```

Jenkins  Maven_Proj1_QApackage  #2

[INFO] Total time: 9.865 s
[INFO] Finished at: 2021-05-17T04:50:41Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/single-module/pom.xml to com.example.maven-
samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/single-module/target/single-module-
project.jar to com.example.maven-samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/pom.xml to com.example.maven-
samples/multi-module-parent/1.0-SNAPSHOT/multi-module-parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/pom.xml to com.example.maven-samples/parent
/1.0-SNAPSHOT/parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/pom.xml to
com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target/webapp.war to
com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/server/pom.xml to
com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/server/target/server.jar to
com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS

```

Step 16: If the job is successfully built, go to the below path and find your webapp.war file.

```
$ cd /var/lib/Jenkins/workspace/Maven_proj1_QApackage/multi-module/webapp/target
```

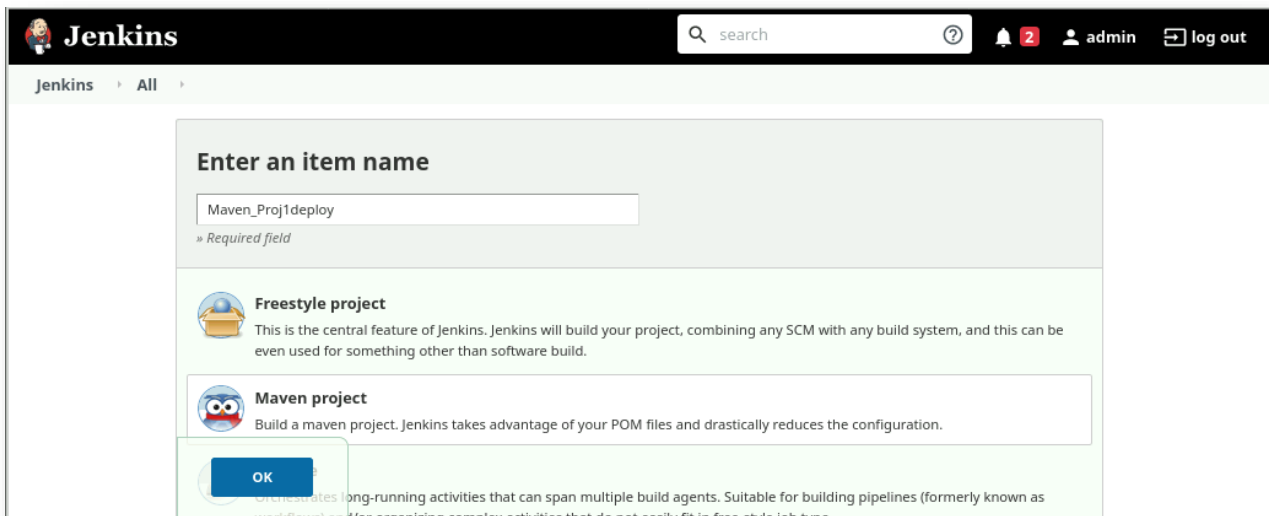
```
$ ls
```

```

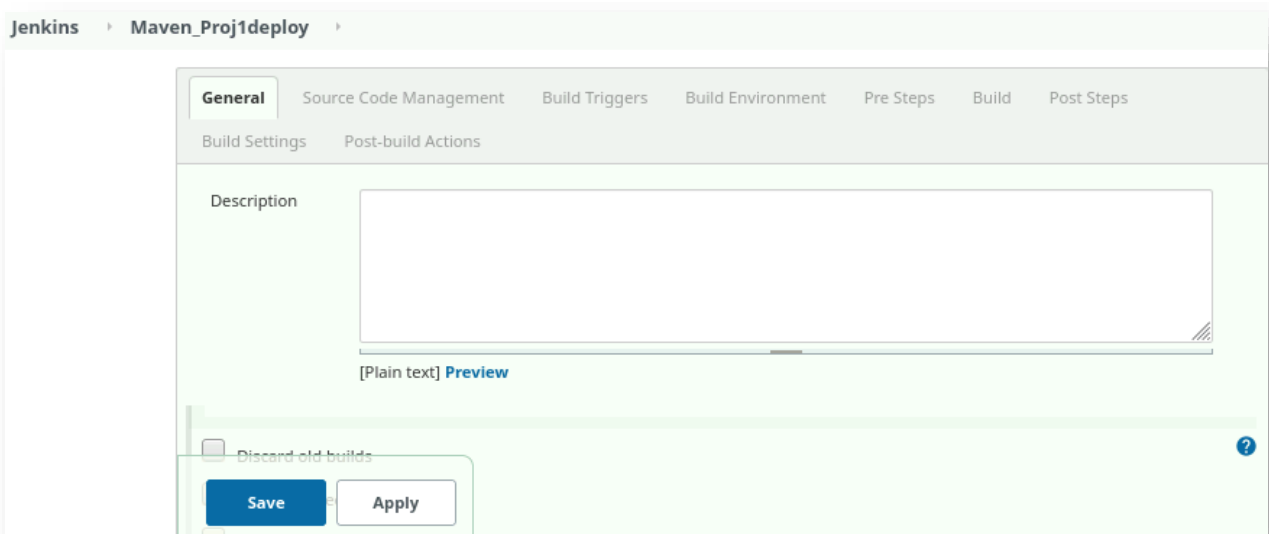
edureka@kmaster:~$ cd /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target/
edureka@kmaster:/var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target$ ls
edureka@kmaster:/var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target$

```

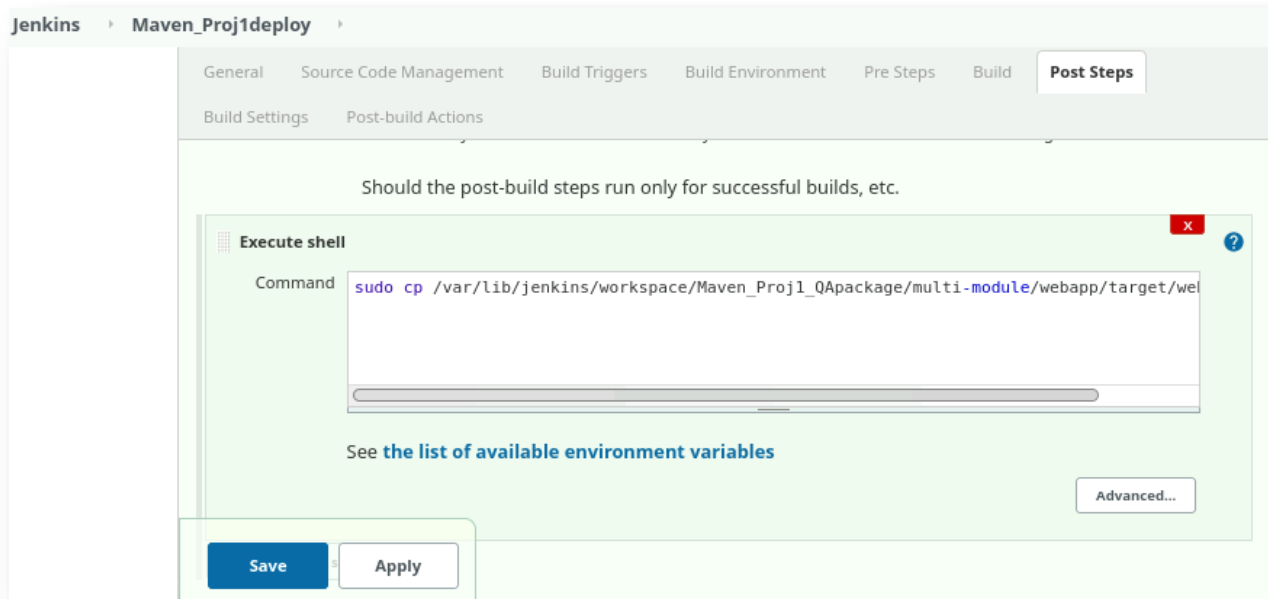
Step 17: After building the war file, create another job to deploy the war file over the tomcat server.



The image shows the Jenkins 'Enter an item name' dialog box. At the top, there's a search bar and user information (admin, log out). Below the title 'Enter an item name', there's a text input field containing 'Maven\_Proj1deploy'. A small message below the field says '» Required field'. Underneath, there are two project type options: 'Freestyle project' with a description '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.' and 'Maven project' with a description 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.' At the bottom, there's an 'OK' button.

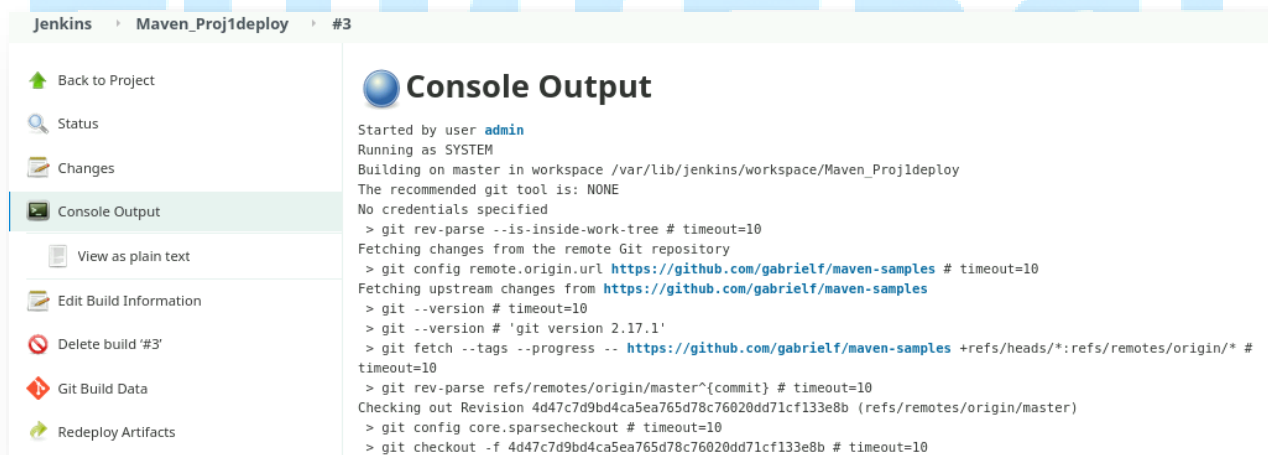


The image shows the Jenkins configuration page for a job named 'Maven\_Proj1deploy'. The page has a breadcrumb 'Jenkins > Maven\_Proj1deploy'. Below the breadcrumb is a tabbed interface with tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Pre Steps', 'Build', and 'Post Steps'. The 'General' tab is selected. Under the 'General' tab, there are sub-tabs for 'Build Settings' and 'Post-build Actions'. The 'Description' field is a large text area. Below it, there's a '[Plain text] Preview' link. At the bottom, there's a 'Discard old builds' checkbox and two buttons: 'Save' and 'Apply'.



Copy the given code and paste it in the Execute shell command:

`/var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target/webapp.war /opt/tomcat/webapps/`



```
Jenkins  ▸ Maven_Proj1deploy  ▸ #3

Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/single-module/pom.xml to com.example.maven-
samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/single-module/target/single-module-project.jar to
com.example.maven-samples/single-module-project/1.0-SNAPSHOT/single-module-project-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/multi-module/pom.xml to com.example.maven-
samples/multi-module-parent/1.0-SNAPSHOT/multi-module-parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/pom.xml to com.example.maven-samples/parent/1.0-
SNAPSHOT/parent-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/multi-module/webapp/pom.xml to com.example.maven-
samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/multi-module/webapp/target/webapp.war to
com.example.maven-samples/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/multi-module/server/pom.xml to com.example.maven-
samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Maven_Proj1deploy/multi-module/server/target/server.jar to
com.example.maven-samples/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
channel stopped
[Maven_Proj1deploy] $ /bin/sh -xe /tmp/jenkins3314760899362451375.sh
+ sudo cp /var/lib/jenkins/workspace/Maven_Proj1_QApackage/multi-module/webapp/target/webapp.war /opt/tomcat
/webapps/
Finished: SUCCESS
```

Step 18: After building the job, go to the address <http://localhost:8090/webapp> and you can see the below output.

