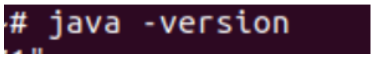
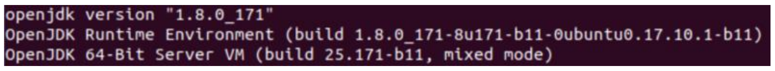
Jenkins:

Step 1: You should have java 7 or Java 8 installed in your system before starting the installation of Jenkins. Check if installed by using below command

java -version



```
# java -version
```



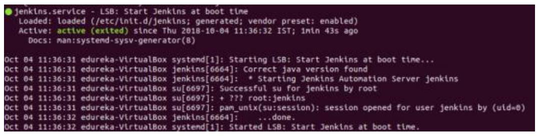
```
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-8u171-b11-0ubuntu0.17.10.1-b11)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

Step 3:

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key
add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \
    /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
.
```

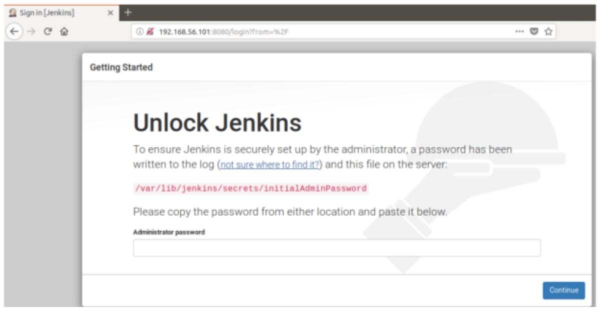
Step 6:Now you need to start the Jenkins service and you check the status of Jenkins if its running or not. Use the command given below:`service jenkins start`

```
Systemctl start Jenkins
```



```
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated; vendor preset: enabled)
   Active: active (exited) since Thu 2018-10-04 11:36:32 IST; 1min 43s ago
     Docs: man:systemd-sysv-generator(8)

Oct 04 11:36:31 edureka-VirtualBox systemd[1]: Starting LSB: Start Jenkins at boot time...
Oct 04 11:36:31 edureka-VirtualBox jenkins[6664]: Correct java version found
Oct 04 11:36:31 edureka-VirtualBox jenkins[6664]:  * Starting Jenkins Automation Server jenkins
Oct 04 11:36:31 edureka-VirtualBox su[6697]: Successful su for jenkins by root
Oct 04 11:36:31 edureka-VirtualBox su[6697]: + ??? root:jenkins
Oct 04 11:36:31 edureka-VirtualBox su[6697]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Oct 04 11:36:32 edureka-VirtualBox jenkins[6664]:    ...done.
Oct 04 11:36:32 edureka-VirtualBox systemd[1]: Started LSB: Start Jenkins at boot time.
```

Step 7: Go to your browser and go to Ipaddress:8080. In my case it's 192.168.56.101:8080. The port on which Jenkins is running is 8080.



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

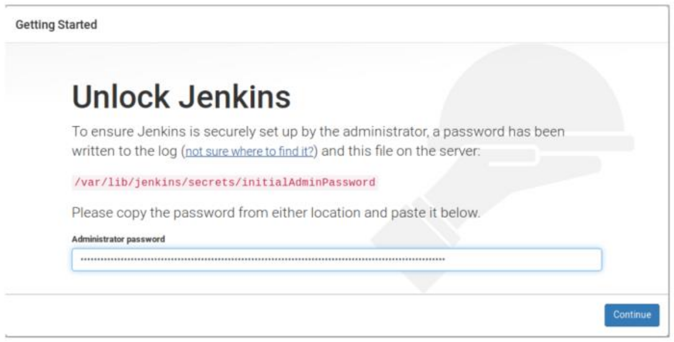
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Step 8: Now use the below command to get the password, copy it and paste it within the box shown above.

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```


Getting Started

Unlock Jenkins

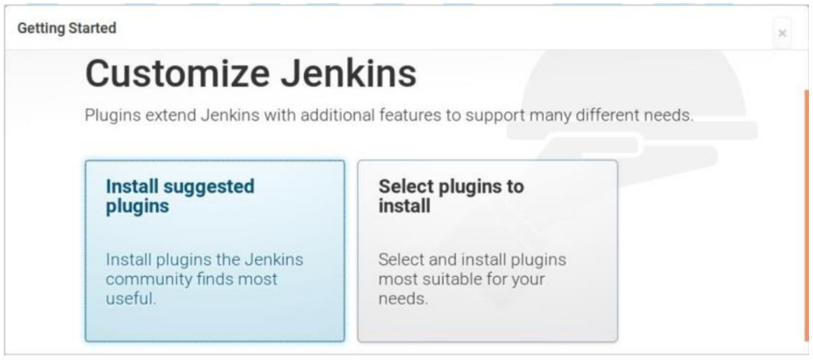
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

••


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

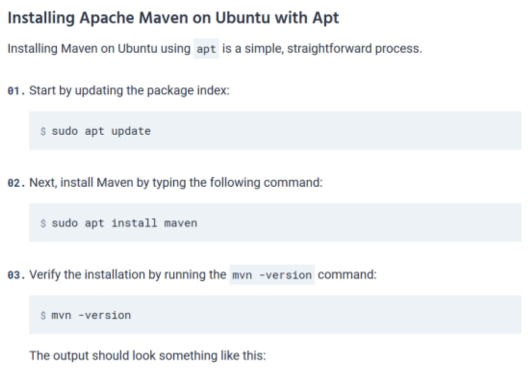
Select and install plugins most suitable for your needs.

Now you can access the jenkins through your public IP(in AWS environment) or private IP in your local

Environment with port 8080 **Error! Hyperlink reference not valid.** address>:8080

*****&****&********************&***

Maven Installation on Ubuntu


Installing Apache Maven on Ubuntu with Apt

Installing Maven on Ubuntu using `apt` is a simple, straightforward process.

01. Start by updating the package index:

```
$ sudo apt update
```

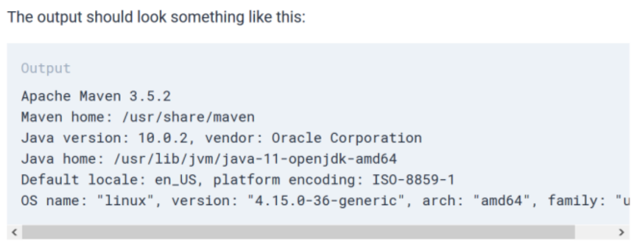
02. Next, install Maven by typing the following command:

```
$ sudo apt install maven
```

03. Verify the installation by running the `mvn -version` command:

```
$ mvn -version
```

The output should look something like this:

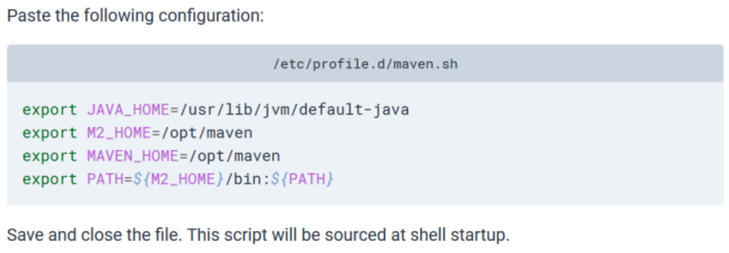

The output should look something like this:

```
Output
Apache Maven 3.5.2
Maven home: /usr/share/maven
Java version: 10.0.2, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_US, platform encoding: ISO-8859-1
OS name: "linux", version: "4.15.0-36-generic", arch: "amd64", family: "u
```

Installation of Latest Apache Mavenpt update

```
apt update
apt install default-jdk
java -version
wget
https://www-us.apache.org/dist/maven/maven-3/3.6.0/binaries/apache-maven-3.6.
0-bin.tar.gz -P /tmp
tar xf /tmp/apache-maven-*.tar.gz -C /opt
ln -s /opt/apache-maven-3.6.0 /opt/maven
```

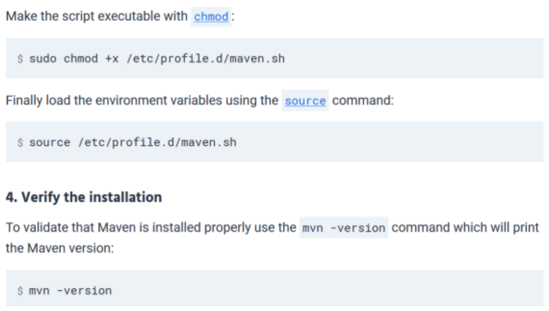
```
Setting up environment variable
```



Paste the following configuration:

```
/etc/profile.d/maven.sh

export JAVA_HOME=/usr/lib/jvm/default-java
export M2_HOME=/opt/maven
export MAVEN_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

Save and close the file. This script will be sourced at shell startup.



Make the script executable with `chmod`:

```
$ sudo chmod +x /etc/profile.d/maven.sh
```

Finally load the environment variables using the `source` command:

```
$ source /etc/profile.d/maven.sh
```

4. Verify the installation

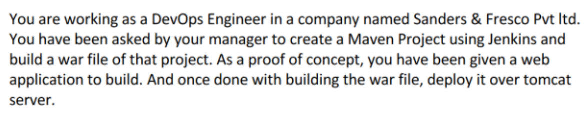
To validate that Maven is installed properly use the `mvn -version` command which will print the Maven version:

```
$ mvn -version
```

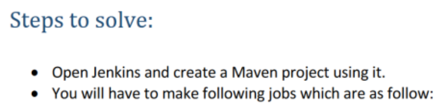
*************&*********&************&**********&*************

Jenkin Lab/Maven Lab

Scenario:



You are working as a DevOps Engineer in a company named Sanders & Fresco Pvt ltd. You have been asked by your manager to create a Maven Project using Jenkins and build a war file of that project. As a proof of concept, you have been given a web application to build. And once done with building the war file, deploy it over tomcat server.



Steps to solve:

- Open Jenkins and create a Maven project using it.
- You will have to make following jobs which are as follow:

1. Compile
2. Code review

3. Unit Test

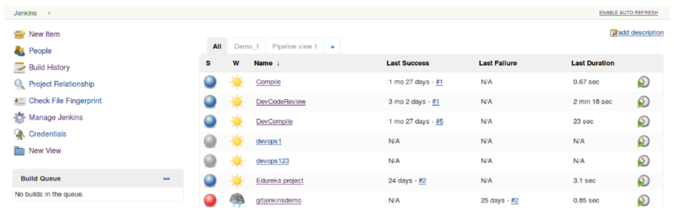
4. Package

5. Deploy

Before proceeding please make sure you are able to login to Jenkins and you have installed all

Suggested plugins:

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



Jenkins › ENABLE AUTO REFRESH

New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
Credentials
New View

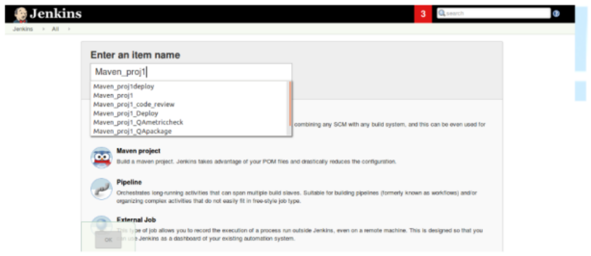
Build Queue
No builds in the queue.

add description

All | Demo_1 | Pipeline view 1 | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Compile	1 mo 27 days - #1	N/A	0.67 sec
●	☀	DevCodeReview	3 mo 2 days - #1	N/A	2 min 18 sec
●	☀	DevCompile	1 mo 27 days - #5	N/A	23 sec
●	☀	devops1	N/A	N/A	N/A
●	☀	devops123	N/A	N/A	N/A
●	☀	Edureka project	24 days - #2	N/A	3.1 sec
●	⛈	gitjenkinsdemo	N/A	25 days - #2	0.85 sec

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



Jenkins 3 search

Jenkins › All ›

Enter an item name

Maven_proj1

Maven_proj1deploy
Maven_proj1
Maven_proj1_code_review
Maven_proj1_Deploy
Maven_proj1_QAmetriccheck
Maven_proj1_QApackage

combining any SCM with any build system, and this can be even used for

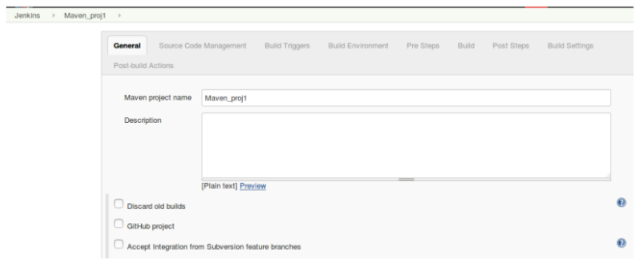
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

OK

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



Jenkins › Maven_proj1 ›

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

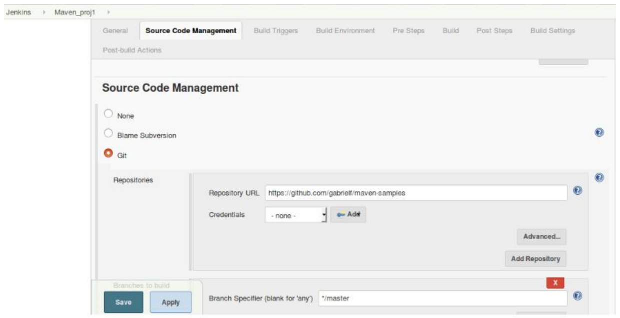
Maven project name: Maven_proj1

Description:

[Plain text] Preview

☐ Discard old builds
☐ GitHub project
☐ Accept Integration from Subversion feature branches

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



Jenkins › Maven_proj1 ›

Source Code Management

None [✓ Git] Blame Subversion

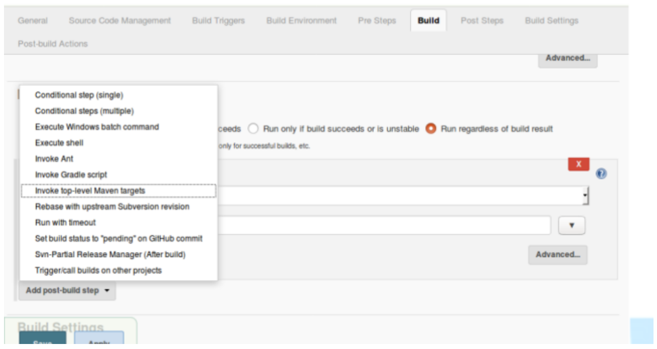
Repositories

Repository URL: https://github.com/gabrielf/maven-samples

Credentials: - none -

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

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 Maven plugin for your Jenkins.

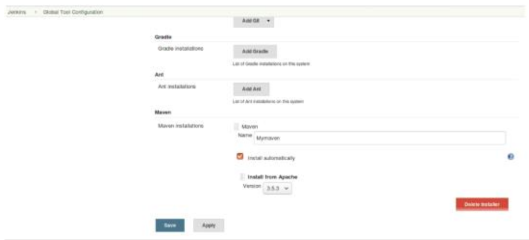


Run only if build succeeds or is unstable [✓ Run regardless of build result]

only for successful builds, etc.

General
Source Code Management
Build Triggers
Build Environment
Pre Steps
Build
Post Steps
Build Settings
Post-build Actions
Advanced...
Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Rebase with upstream Subversion revision
Run with timeout
Set build status to "pending" on GitHub commit
Svn-Partial Release Manager (After build)
Trigger/call builds on other projects
Add post-build step
X
Advanced...
Build Settings
Save
Apply

Go to Global Configuration and selection maven and get ready with your pom.xml file in /var/lib/Jenkins folder



Jenkins › Global Tool Configuration

Gradle

Gradle installations

List of Gradle installations on this system

Ant

Ant installations

List of Ant installations on this system

Maven

Maven installations

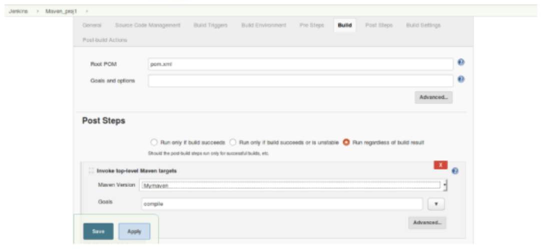
Maven

Name: Mymaven

[✓ Install automatically]

Install from Apache

Version: 3.5.3



Jenkins › Maven_proj1 ›

Root POM: pom.xml

Goals and options:

Post Steps

Run only if build succeeds Run only if build succeeds or is unstable [✓ Run regardless of build result]

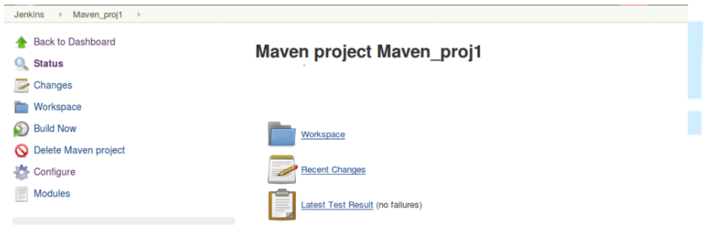
Should the post-build steps run only for successful builds, etc.

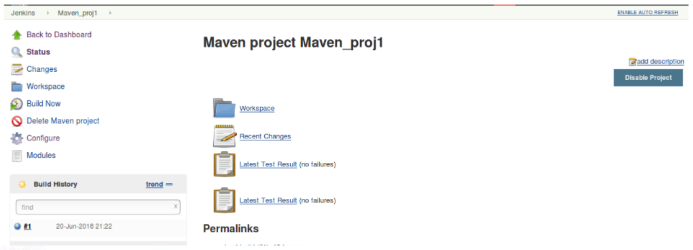
Invoke top-level Maven targets

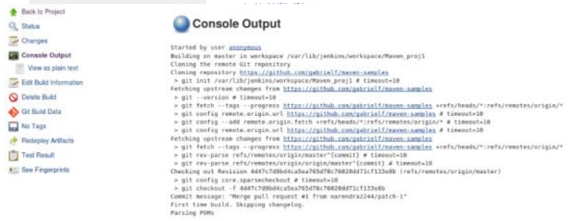
Maven Version: Mymaven

Goals: compile

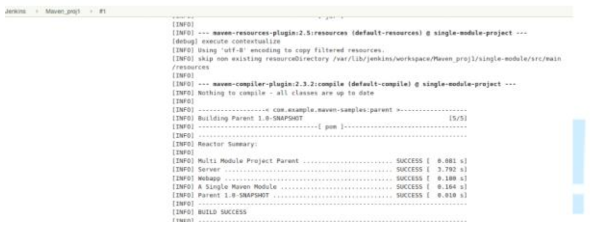
Step 6: Go to build now and then check your console output by clicking on the blue or red dot whicheve comes after the build. Blue is sign for successful build where as red is for build failure.





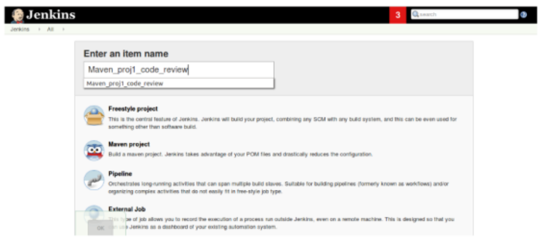


```
Started by user anonymous
Building on master in workspace /var/lib/jenkins/workspace/Maven_proj1
Cloning the remote Git repository
Cloning repository https://github.com/gabrielf/maven-samples
 > git init /var/lib/jenkins/workspace/Maven_proj1 # timeout=10
Fetching upstream changes from https://github.com/gabrielf/maven-samples
 > git --version # timeout=10
 > git fetch --tags --progress https://github.com/gabrielf/maven-samples +refs/heads/*:refs/remotes/origin/*
 > 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
 > git config remote.origin.url https://github.com/gabrielf/maven-samples # timeout=10
Fetching upstream changes from https://github.com/gabrielf/maven-samples
 > git fetch --tags --progress https://github.com/gabrielf/maven-samples +refs/heads/*:refs/remotes/origin/*
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b
Commit message: "Merge pull request #1 from narendra2244/patch-1"
First time build. Skipping changelog.
Parsing POMs
```



```
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ single-module-project ---
[debug] execute contextualize
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/Maven_proj1/single-module/src/main
/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ single-module-project ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] -----------------< com.example.maven-samples:parent >-----------------
[INFO] Building Parent 1.0-SNAPSHOT                                         [5/5]
[INFO] --------------------------------[ pom ]---------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Multi Module Project Parent ........................ SUCCESS [  0.081 s]
[INFO] Server ............................................. SUCCESS [  3.792 s]
[INFO] Webapp ............................................. SUCCESS [  0.100 s]
[INFO] A Single Maven Module .............................. SUCCESS [  0.164 s]
[INFO] Parent 1.0-SNAPSHOT ................................ SUCCESS [  0.010 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

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



Enter an item name

Maven_proj1_code_review

Maven_proj1_code_review

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.

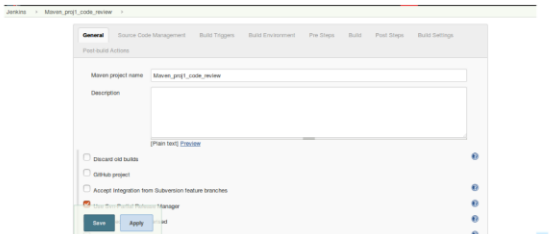
Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

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


Jenkins › Maven_proj1_code_review ›

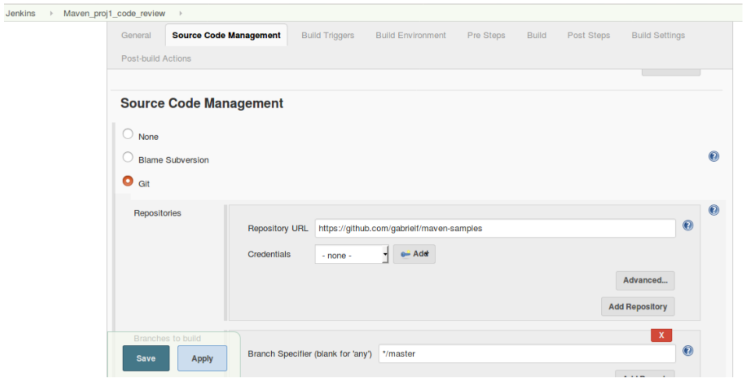
General | Source Code Management | Build Triggers | Build Environment | Pre Steps | Build | Post Steps | Build Settings | Post-build Actions

Maven project name: Maven_proj1_code_review

Description:

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] Accept integration from Subversion feature branches
- [✓] Manager


Jenkins › Maven_proj1_code_review ›

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

Source Code Management

None [✓ Git] Blame Subversion

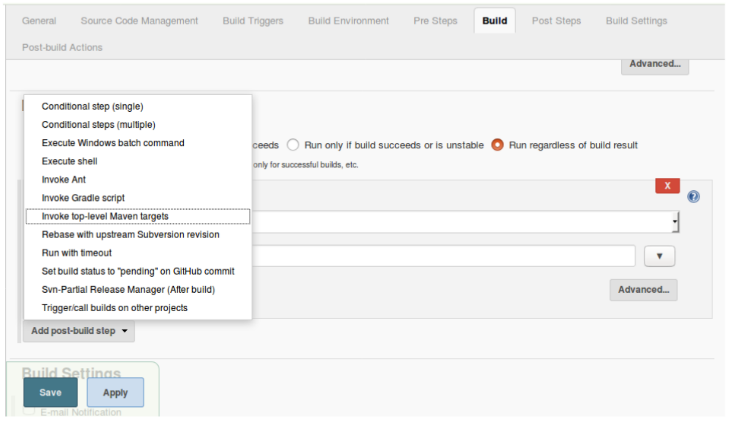
Repositories

Repository URL: https://github.com/gabrielf/maven-samples

Credentials: - none -

Branches to build

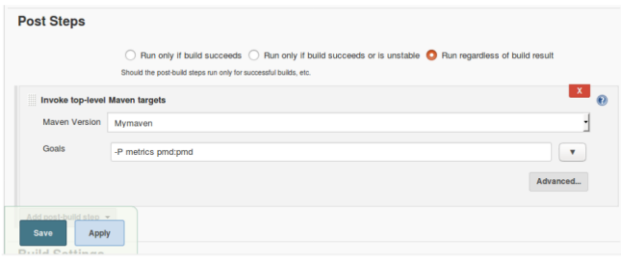
Branch Specifier (blank for 'any'): */master


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

ceeds Run only if build succeeds or is unstable [✓ Run regardless of build result]

only for successful builds, etc.

Advanced...
Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Rebase with upstream Subversion revision
Run with timeout
Set build status to "pending" on GitHub commit
Svn-Partial Release Manager (After build)
Trigger/call builds on other projects
Add post-build step
Build Settings
E-mail Notification
Save
Apply


Post Steps

Run only if build succeeds | Run only if build succeeds or is unstable | [✓ Run regardless of build result]

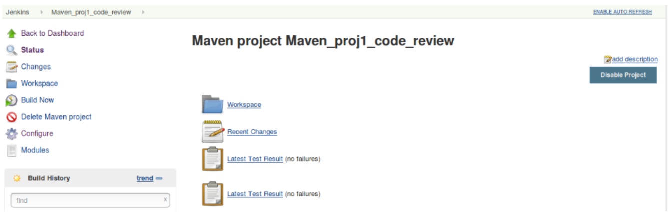
Should the post-build steps run only for successful builds, etc.

Invoke top-level Maven targets

Maven Version: Mymaven

Goals: -P metrics pmd:pmd

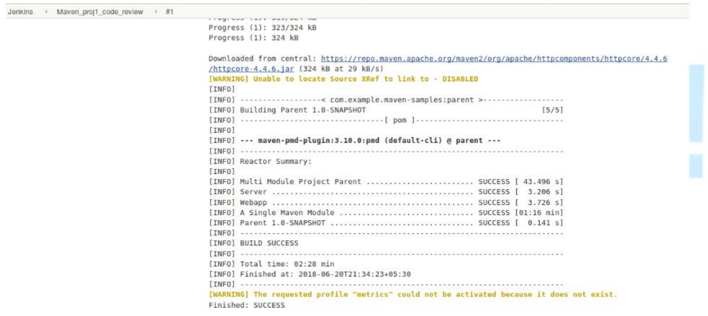
Step 9: Click on build now and check the console output.



Maven project Maven_proj1_code_review

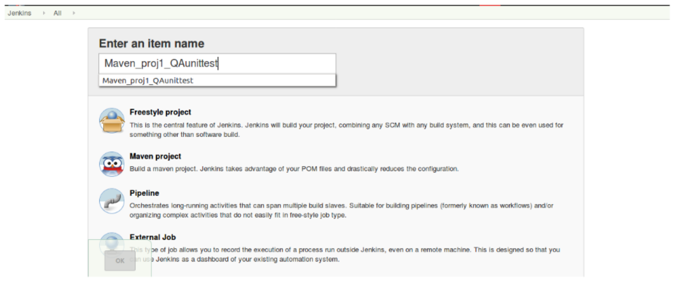
Latest Test Result (no failures)

Latest Test Result (no failures)



```
Progress (1): 323/324 kB
Progress (1): 324 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpcore/4.4.6
/httpcore-4.4.6.jar (324 kB at 29 kB/s)
[WARNING] Unable to locate Source XRef to link to - DISABLED
[INFO]
[INFO] -----------------< com.example.maven-samples:parent >-----------------
[INFO] Building Parent 1.0-SNAPSHOT                                       [5/5]
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] --- maven-pmd-plugin:3.10.0:pmd (default-cli) @ parent ---
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Multi Module Project Parent ........................ SUCCESS [ 43.496 s]
[INFO] Server ............................................. SUCCESS [  3.206 s]
[INFO] Webapp ............................................. SUCCESS [  3.726 s]
[INFO] A Single Maven Module .............................. SUCCESS [01:16 min]
[INFO] Parent 1.0-SNAPSHOT ................................ SUCCESS [  0.141 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 02:28 min
[INFO] Finished at: 2018-06-20T21:34:23+05:30
[INFO] ------------------------------------------------------------------------
[WARNING] The requested profile "metrics" could not be activated because it does not exist.
Finished: SUCCESS
```

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



Enter an item name

Maven_proj1_QAunittest

Maven_proj1_QAunittest

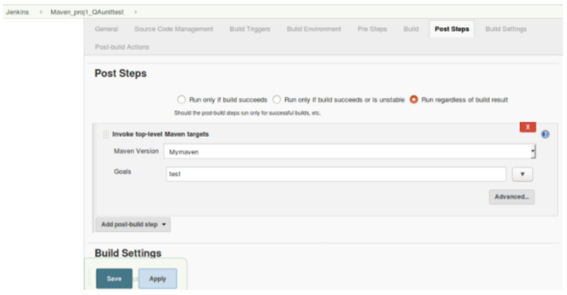
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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 slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

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



Post Steps

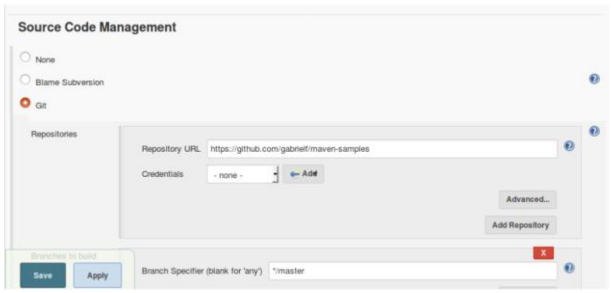
Run only if build succeeds [ ] Run only if build succeeds or is unstable [ ] [✓ Run regardless of build result]

Should the post-build steps run only for successful builds, etc.

Invoke top-level Maven targets

Maven Version: Mymaven

Goals: test



Source Code Management

None

Blame Subversion

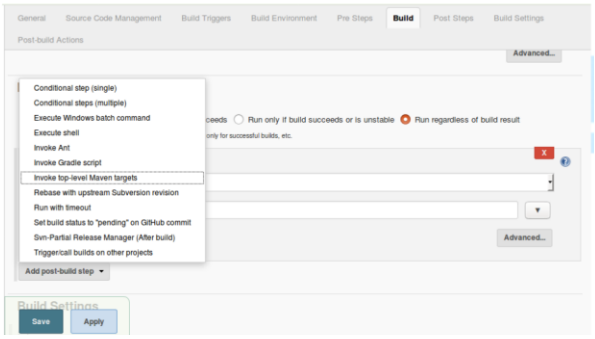
[✓ Git]

Repositories

Repository URL: https://github.com/gabrielf/maven-samples

Credentials: - none -

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

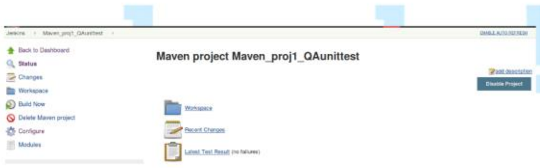


General
Source Code Management
Build Triggers
Build Environment
Pre Steps
Build
Post Steps
Build Settings
Post-build Actions
Advanced...
Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Rebase with upstream Subversion revision
Run with timeout
Set build status to "pending" on GitHub commit
Svn-Partial Release Manager (After build)
Trigger/call builds on other projects
Add post-build step
X
Advanced...
Build Settings
Save
Apply

…ceeds ○ Run only if build succeeds or is unstable [✓ Run regardless of build result]

…only for successful builds, etc.

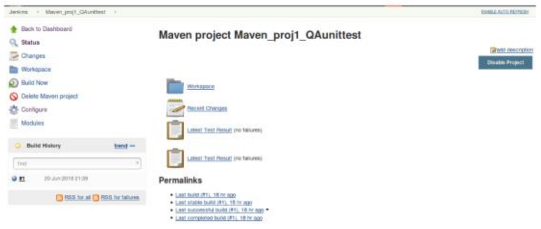
Step 12: Click on build now and go on console output after the job is built.



Jenkins › Maven_proj1_QAunittest ›

ENABLE AUTO REFRESH

Maven project Maven_proj1_QAunittest



Jenkins › Maven_proj1_QAunittest ›

ENABLE AUTO REFRESH

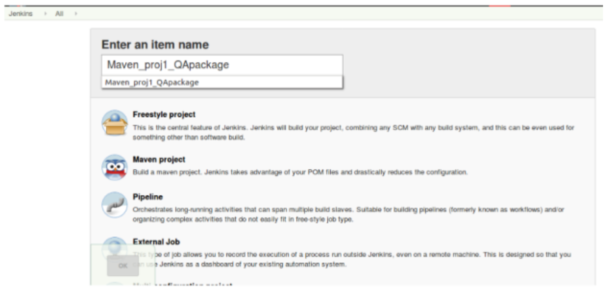
Maven project Maven_proj1_QAunittest

#1 20-Jun-2018 21:39

Permalinks

- Last build (#1), 18 hr ago
- Last stable build (#1), 18 hr ago
- Last successful build (#1), 18 hr ago
- Last completed build (#1), 18 hr ago

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



Jenkins › All ›

Enter an item name

Maven_proj1_QApackage

Maven_proj1_QApackage

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.

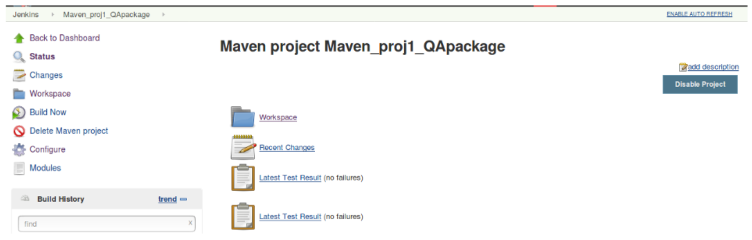
Pipeline

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

External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Step 14: Click on configure and configure your project.


Jenkins › Maven_proj1_QApackage › ENABLE AUTO REFRESH

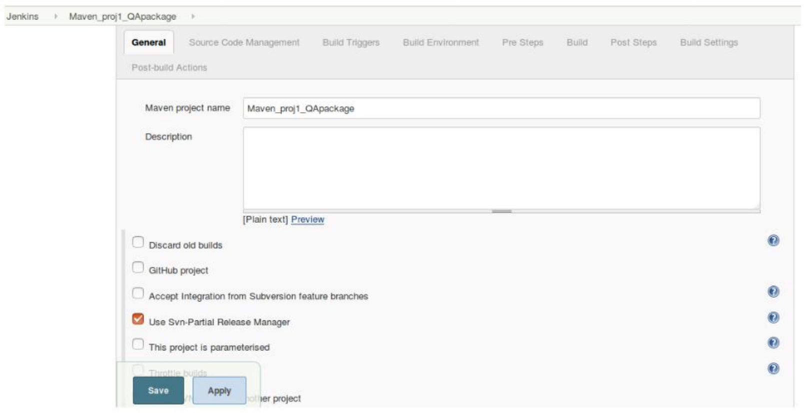
Back to Dashboard
Status
Changes
Workspace
Build Now
Delete Maven project
Configure
Modules

Build History trend
find

Maven project Maven_proj1_QApackage

add description
Disable Project

Workspace
Recent Changes
Latest Test Result (no failures)
Latest Test Result (no failures)


Jenkins › Maven_proj1_QApackage ›

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

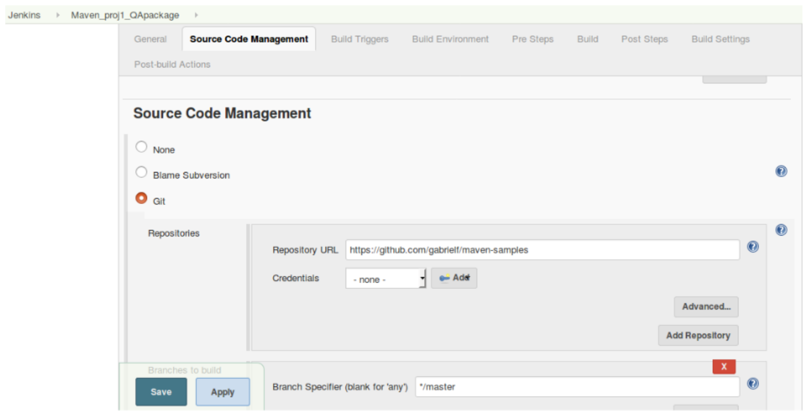
Maven project name: Maven_proj1_QApackage

Description:

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] Accept Integration from Subversion feature branches
- [x] Use Svn-Partial Release Manager
- [ ] This project is parameterised
- [ ] Throttle builds

Save Apply


Jenkins › Maven_proj1_QApackage ›

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

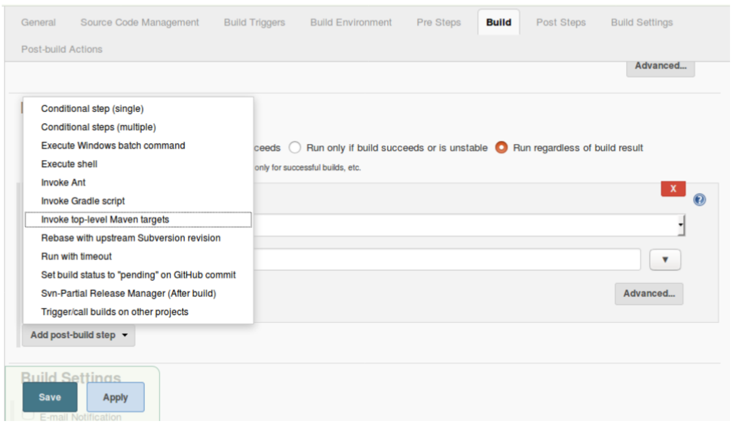
Source Code Management

None [✓ Git] Blame Subversion

Repositories
Repository URL: https://github.com/gabrielf/maven-samples
Credentials: - none - Add
Advanced...
Add Repository

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

Save Apply


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

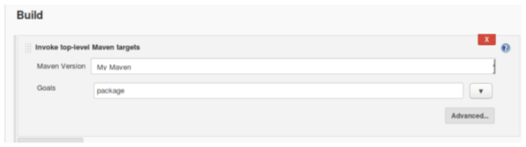
Advanced...

Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Rebase with upstream Subversion revision
Run with timeout
Set build status to "pending" on GitHub commit
Svn-Partial Release Manager (After build)
Trigger/call builds on other projects
Add post-build step

...ceeds Run only if build succeeds or is unstable [✓ Run regardless of build result]
...only for successful builds, etc.

Advanced...

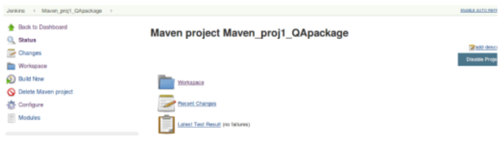
Build Settings
Save Apply
E-mail Notification

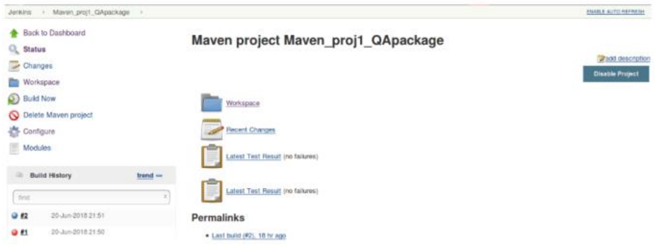


```
Maven Version   My Maven
Goals           package
```

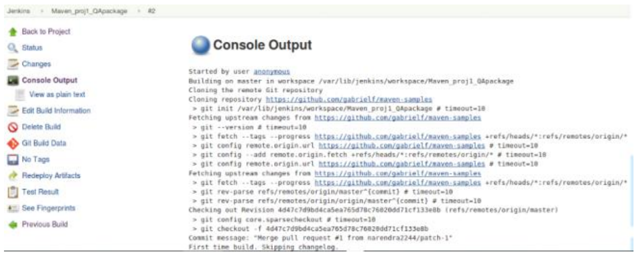
Define the maven goals for your project.

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

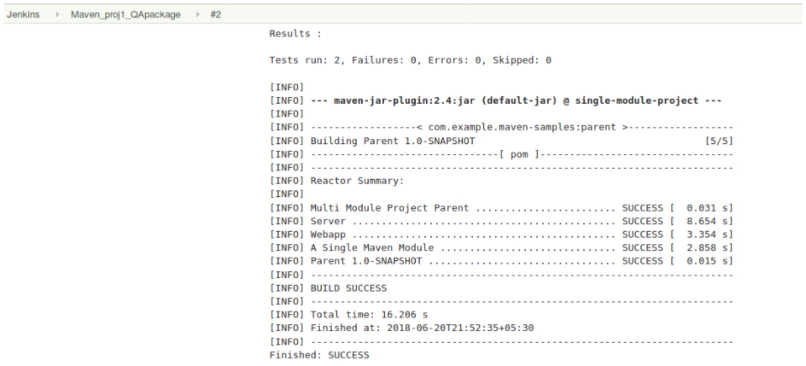




```
#2    20-Jun-2018 21:51
#1    20-Jun-2018 21:50
```



```
Started by user anonymous
Building on master in workspace /var/lib/jenkins/workspace/Maven_proj1_QApackage
Cloning the remote Git repository
Cloning repository https://github.com/gabrielf/maven-samples
 > git init /var/lib/jenkins/workspace/Maven_proj1_QApackage # timeout=10
Fetching upstream changes from https://github.com/gabrielf/maven-samples
 > git --version # timeout=10
 > git fetch --tags --progress https://github.com/gabrielf/maven-samples +refs/heads/*:refs/remotes/origin/*
 > 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
 > git config remote.origin.url https://github.com/gabrielf/maven-samples # timeout=10
Fetching upstream changes from https://github.com/gabrielf/maven-samples
 > git fetch --tags --progress https://github.com/gabrielf/maven-samples +refs/heads/*:refs/remotes/origin/*
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 4d47c7d9bd4ca5ea765d78c76020dd71cf133e8b
Commit message: "Merge pull request #1 from narendra2244/patch-1"
First time build. Skipping changelog.
```

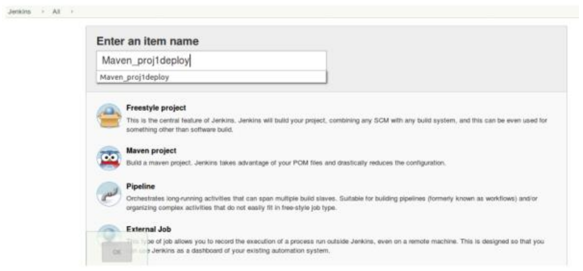


```
Results :

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

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ single-module-project ---
[INFO]
[INFO] -----------------< com.example.maven-samples:parent >-----------------
[INFO] Building Parent 1.0-SNAPSHOT                                     [5/5]
[INFO] --------------------------------[ pom ]---------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Multi Module Project Parent ........................ SUCCESS [  0.031 s]
[INFO] Server ............................................. SUCCESS [  8.654 s]
[INFO] Webapp ............................................. SUCCESS [  3.354 s]
[INFO] A Single Maven Module .............................. SUCCESS [  2.858 s]
[INFO] Parent 1.0-SNAPSHOT ................................ SUCCESS [  0.015 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 16.206 s
[INFO] Finished at: 2018-06-20T21:52:35+05:30
[INFO] ------------------------------------------------------------------------
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
```

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



Enter an item name

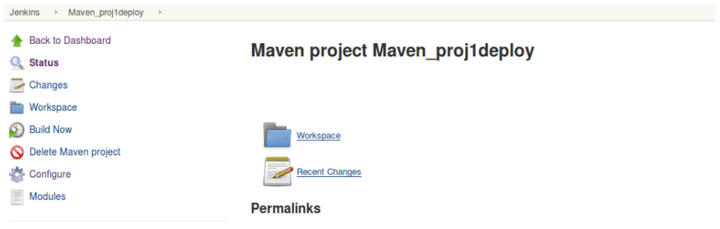
Maven_proj1deploy

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.

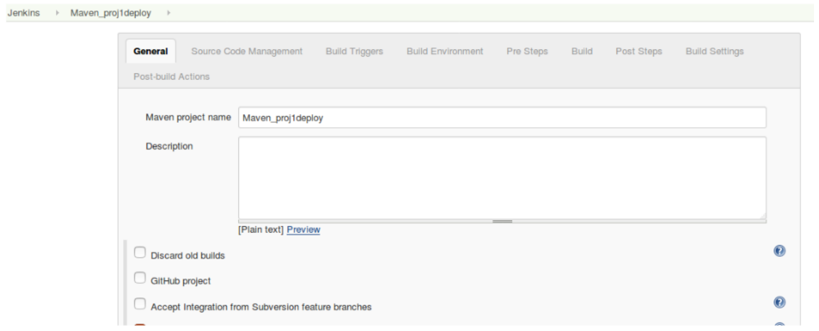
Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



Maven project Maven_proj1deploy

Permalinks



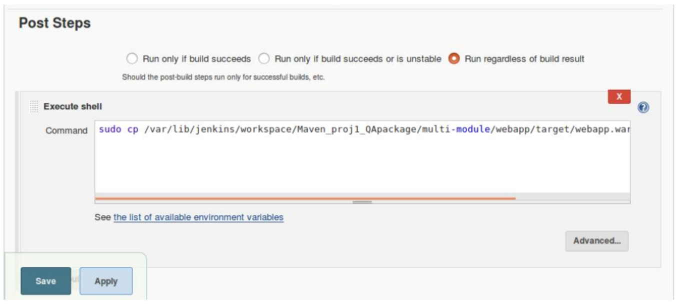
Maven project name: Maven_proj1deploy

Description:

☐ Discard old builds

☐ GitHub project

☐ Accept Integration from Subversion feature branches



Post Steps

Run only if build succeeds ( ) Run only if build succeeds or is unstable ( ) [✓ Run regardless of build result]

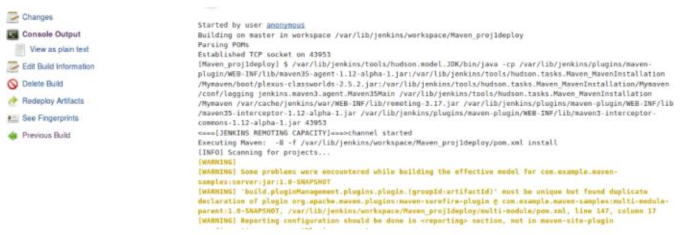
Should the post-build steps run only for successful builds, etc.

Execute shell

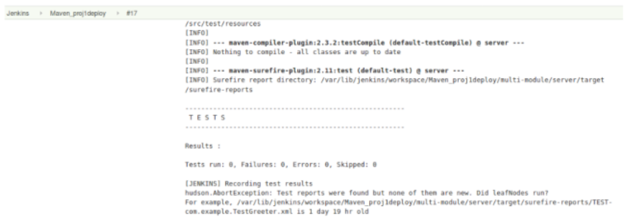
Command:

```
sudo cp /var/lib/jenkins/workspace/Maven_proj1_QApackage/multi-module/webapp/target/webapp.war
```

See the list of available environment variables

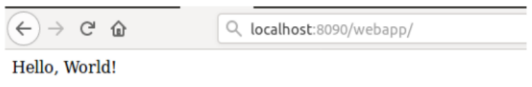


```
Started by user anonymous
Building on master in workspace /var/lib/jenkins/workspace/Maven_proj1deploy
Parsing POMs
Established TCP socket on 43953
[Maven_proj1deploy] $ /var/lib/jenkins/tools/hudson.model.JDK/bin/java -cp /var/lib/jenkins/plugins/maven-
plugin/WEB-INF/lib/maven35-agent-1.12-alpha-1.jar:/var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation
/Mymaven/boot/plexus-classworlds-2.5.2.jar:/var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/Mymaven
/conf/logging jenkins.maven3.agent.Maven35Main /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation
/Mymaven /var/cache/jenkins/war/WEB-INF/lib/remoting-3.17.jar /var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib
/maven35-interceptor-1.12-alpha-1.jar /var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven3-interceptor-
commons-1.12-alpha-1.jar 43953
<===[JENKINS REMOTING CAPACITY]===>channel started
Executing Maven:  -B -f /var/lib/jenkins/workspace/Maven_proj1deploy/pom.xml install
[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] 'build.pluginManagement.plugins.plugin.(groupId:artifactId)' must be unique but found duplicate
declaration of plugin org.apache.maven.plugins:maven-surefire-plugin @ com.example.maven-samples:multi-module-
parent:1.0-SNAPSHOT, /var/lib/jenkins/workspace/Maven_proj1deploy/multi-module/pom.xml, line 147, column 17
[WARNING] Reporting configuration should be done in <reporting> section, not in maven-site-plugin
```


Jenkins › Maven_proj1deploy › #17

```
/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ server ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.11:test (default-test) @ server ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/Maven_proj1deploy/multi-module/server/target
/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------

Results :

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

[JENKINS] Recording test results
hudson.AbortException: Test reports were found but none of them are new. Did leafNodes run?
For example, /var/lib/jenkins/workspace/Maven_proj1deploy/multi-module/server/target/surefire-reports/TEST-
com.example.TestGreeter.xml is 1 day 19 hr old
```

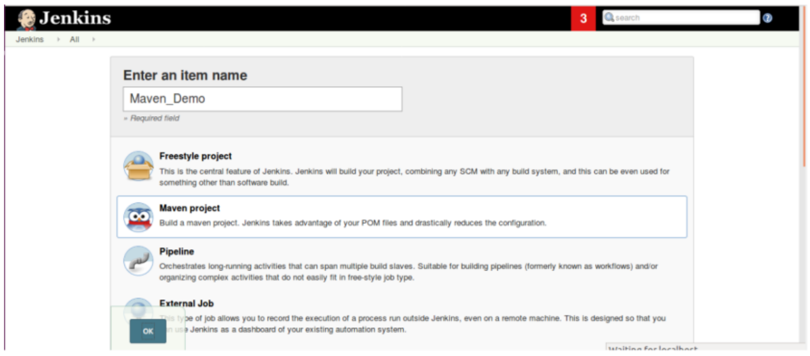
Step 18: After building the job, go to the address http://localhost:8090/webapp and you can see the below output.


localhost:8090/webapp/

Hello, World!

*********&*&***************&**********&**************

Lab: Create Maven Project using Jenkins.


Enter an item name

Maven_Demo

» Required field

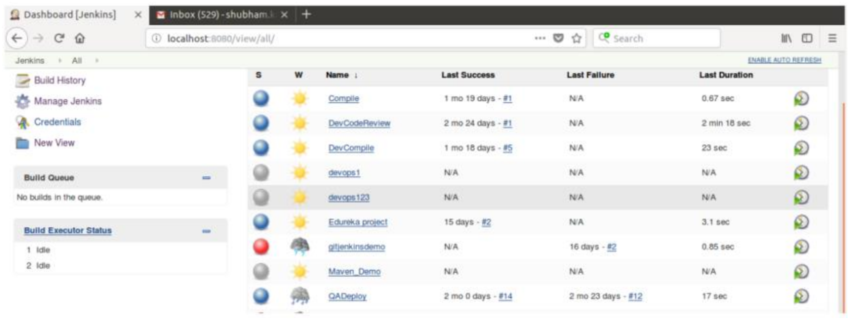
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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 slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Step 2: Open the Jenkins dashboard to view your project.



Build Queue

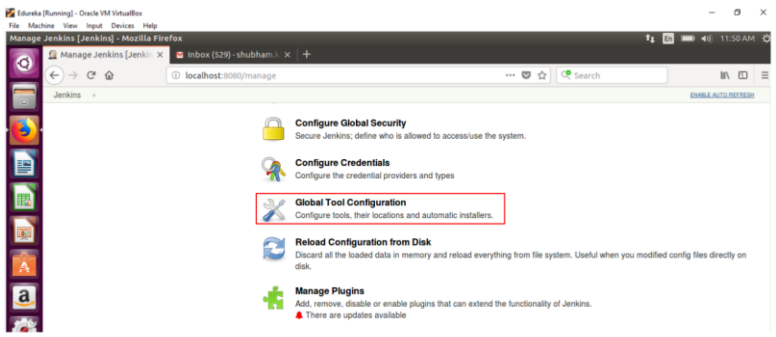
No builds in the queue.

Build Executor Status

1 Idle
2 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
🔵	☀	Compile	1 mo 19 days - #1	N/A	0.67 sec
🔵	☀	DevCodeReview	2 mo 24 days - #1	N/A	2 min 18 sec
🔵	☀	DevCompile	1 mo 18 days - #5	N/A	23 sec
⚪	☀	devops1	N/A	N/A	N/A
⚪	☀	devops123	N/A	N/A	N/A
🔵	☀	Edureka project	15 days - #2	N/A	3.1 sec
🔴	🌧	gitjenkinsdemo	N/A	16 days - #2	0.85 sec
⚪	☀	Maven_Demo	N/A	N/A	N/A
🔵	🌧	QADeploy	2 mo 0 days - #14	2 mo 23 days - #12	17 sec

Step 3: Go to manage Jenkins and then on Global Tool Configuration.


Edureka [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Manage Jenkins [Jenkins] - Mozilla Firefox
Manage Jenkins [Jenkins] × | Inbox (529) - shubham × | +
localhost:8080/manage
Jenkins › ENABLE AUTO REFRESH

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

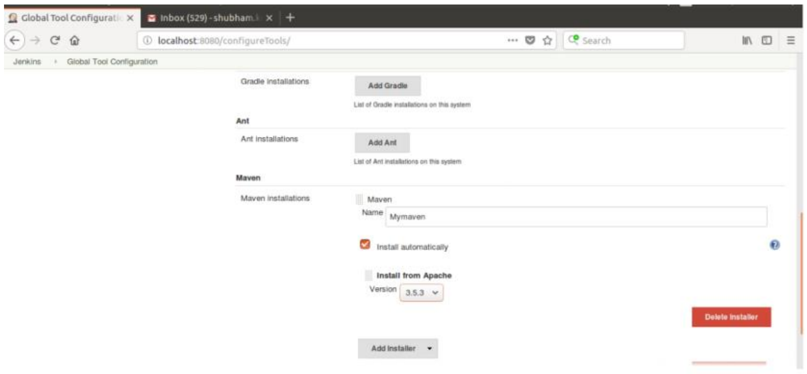
Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available

.Step 4: Install Maven plugin with the latest version available in the dropdown menuof Apache


Global Tool Configuration × | Inbox (529) - shubham × | +
localhost:8080/configureTools/
Jenkins › Global Tool Configuration

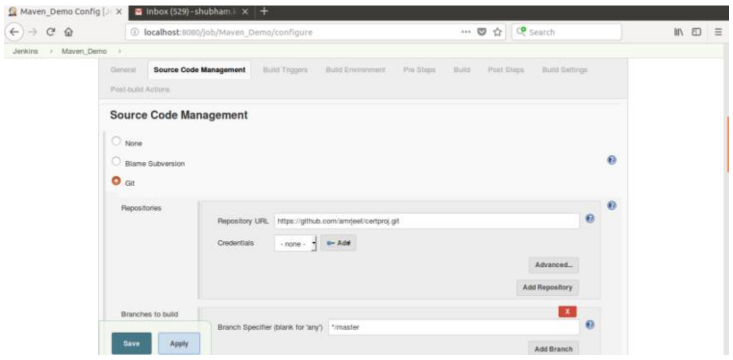
Gradle installations — Add Gradle
List of Gradle installations on this system

Ant
Ant installations — Add Ant
List of Ant installations on this system

Maven
Maven installations
Maven
Name: Mymaven
☑ Install automatically
Install from Apache
Version: 3.5.3
Delete Installer
Add Installer

Now Jenkins will install the maven automatically**.**

Step 5: Provide the repository path to your project which is to be built**.**


Maven_Demo Config × | Inbox (529) - shubham × | +
localhost:8080/job/Maven_Demo/configure
Jenkins › Maven_Demo ›

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

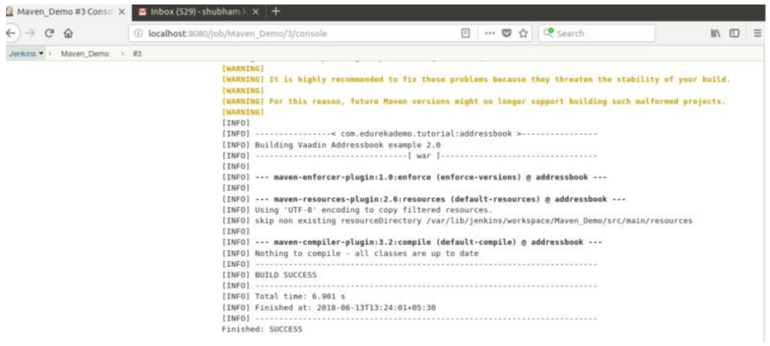
Source Code Management
○ None
○ Blame Subversion
● Git

Repositories
Repository URL: https://github.com/amrjeet/certproj.git
Credentials: - none - | Add
Advanced... | Add Repository

Branches to build
Branch Specifier (blank for 'any'): */master
Add Branch

Save | Apply

Step 6: Go to the build and check the console output of your project.



```
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] ----------------< com.edurekademo.tutorial:addressbook >----------------
[INFO] Building Vaadin Addressbook example 2.0
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- maven-enforcer-plugin:1.0:enforce (enforce-versions) @ addressbook ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ addressbook ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/Maven_Demo/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.2:compile (default-compile) @ addressbook ---
[INFO] Nothing to compile - all classes are up to date
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.901 s
[INFO] Finished at: 2018-06-13T13:24:01+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

*******************&***********&******

Lab for Jenkins Pipeline.

Module 4: Problem Statement

- Create a freestyle project with the name QA_UNIT_TEST in Jenkins that is driven from job DEVELOPER_CODE_REVIEW and performs unit testing

  Take a screenshot of the console output showing successful build of unit testing

- Create a freestyle project with the name QA_ METRICS _CHECK in Jenkins to check the test cases.

  Make sure ***cobertura*** plugin is installed in Jenkins

  Take a screenshot of the metrics from the dashboard of the project.

- Create a freestyle project with the name QA_ PACKAGE in Jenkins to create an executable jar/war file.

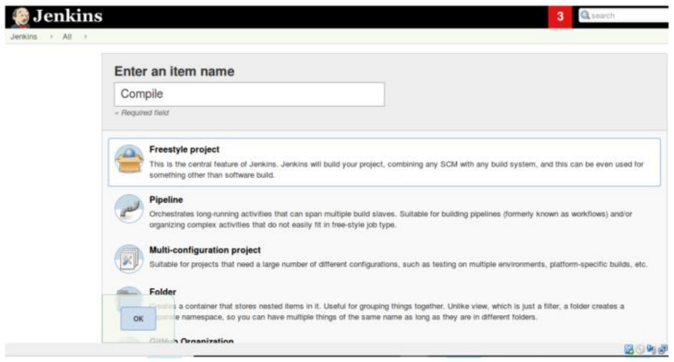
  Take a screenshot of the target folder created in workspace.

- Create a pipeline named SAMPLE_COMPILE_VIEW with **Build Pipeline View** option, select DEVELOPER_COMPILE project under layout section and run the pipeline to check the console output

Take a screenshot of the pipeline dashboard showing the status of the projects
• The pipelines can also be extended to running web tests and Load tests. Explain how you would do the same using Jenkins?

Solution:

Demo – 1: Create Pipeline view using DevCompile and QAUnitTest



Enter an item name

Compile

» Required field

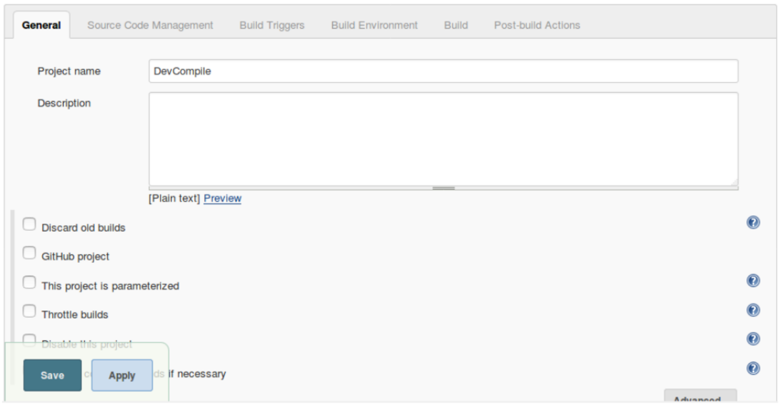
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. 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
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization

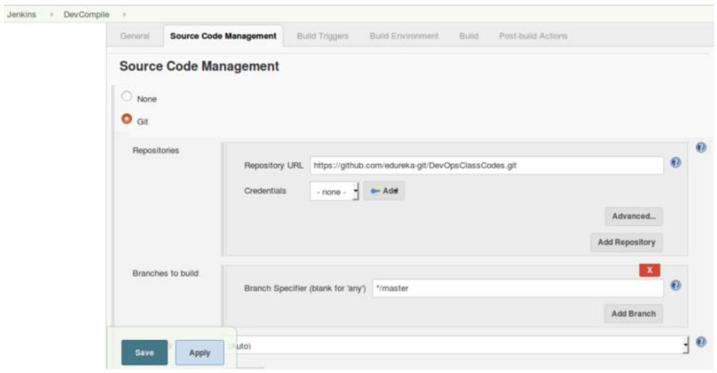


Project name: DevCompile

Description:

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project



Jenkins › DevCompile ›

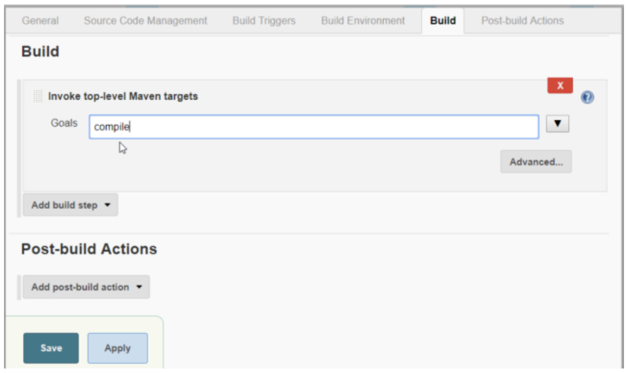
Source Code Management

( ) None
(●) Git

Repositories
Repository URL: https://github.com/edureka-git/DevOpsClassCodes.git
Credentials: - none -

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

(Auto)

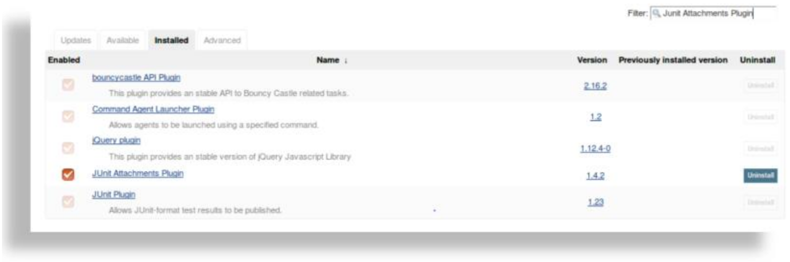


Build

Invoke top-level Maven targets
Goals: compile

Post-build Actions

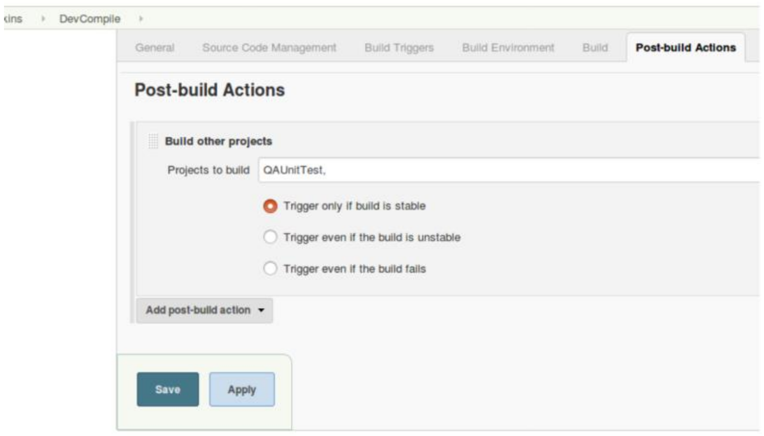
- Create QAunitTest project and provide the same SCM path and execute it after DevCompile Action by setting build.



Filter: Junit Attachments Plugin

Enabled	Name ↓	Version	Previously installed version	Uninstall
✓	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
✓	Command Agent Launcher Plugin Allows agents to be launched using a specified command.	1.2		Uninstall
✓	jQuery plugin This plugin provides an stable version of jQuery Javascript Library	1.12.4-0		Uninstall
✓	JUnit Attachments Plugin	1.4.2		Uninstall
✓	JUnit Plugin Allows JUnit-format test results to be published.	1.23		Uninstall

- Configure DevCompile project and set post build action as QAUnitTest



kins › DevCompile ›

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

Post-build Actions

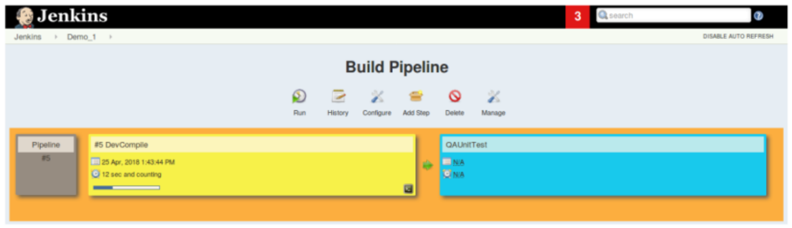
Build other projects

Projects to build: QAUnitTest,

[✓ Trigger only if build is stable]

Trigger even if the build is unstable

Trigger even if the build fails

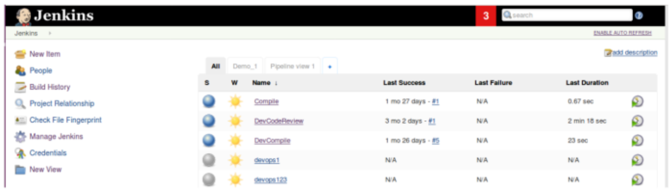


********&***********&***************&***********

Adding slave node to Jenkin:

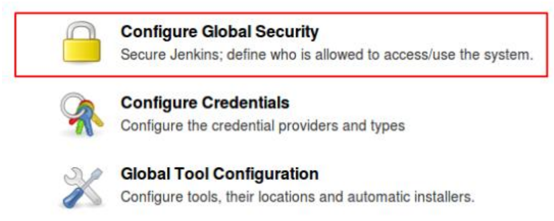
Demo 2: Add a slave node in your Jenkins.

Step 1: Open Jenkins Dashboard and click on manage Jenkins.



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Compile	1 mo 27 days - #1	N/A	0.67 sec
●	☀	DevCodeReview	3 mo 2 days - #1	N/A	2 min 18 sec
●	☀	DevCompile	1 mo 26 days - #5	N/A	23 sec
●	☀	devops1	N/A	N/A	N/A
●	☀	devops123	N/A	N/A	N/A

Step2:Click on Configure Global Security.

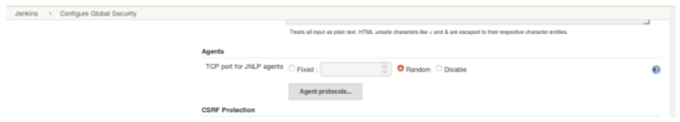


Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Step 3: Select random option for TCP ports for JLNP agents

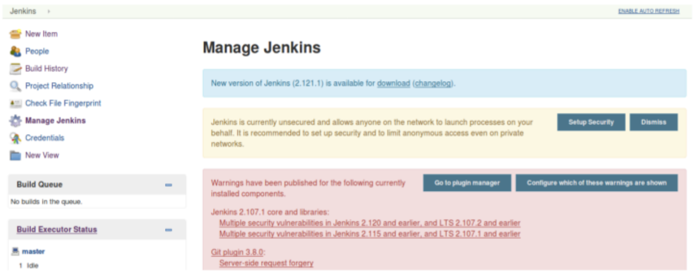


Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Agents

TCP port for JNLP agents: Fixed : [✓ Random] Disable

Step 4: Click on manage Jenkins.



No builds in the queue.

Manage Jenkins

New version of Jenkins (2.121.1) is available for download (changelog).

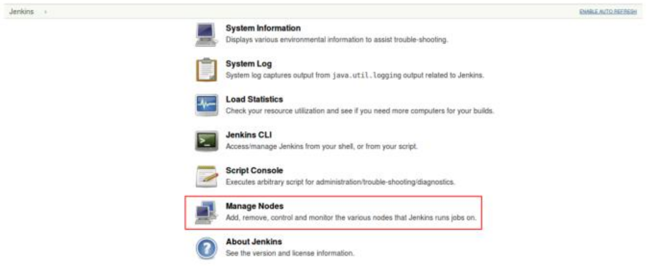
Jenkins is currently unsecured and allows anyone on the network to launch processes on your behalf. It is recommended to set up security and to limit anonymous access even on private networks.

Warnings have been published for the following currently installed components.

Jenkins 2.107.1 core and libraries:
- Multiple security vulnerabilities in Jenkins 2.120 and earlier, and LTS 2.107.2 and earlier
- Multiple security vulnerabilities in Jenkins 2.115 and earlier, and LTS 2.107.1 and earlier

Git plugin 3.8.0:
- Server-side request forgery

Step 5: Click on Manage Nodes option.



System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

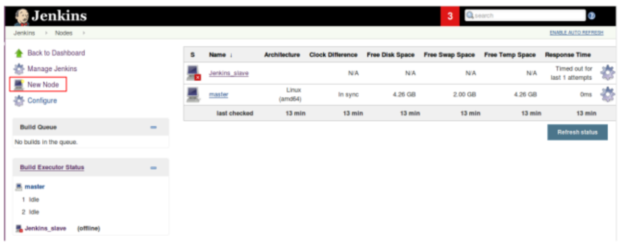
Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

Manage Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

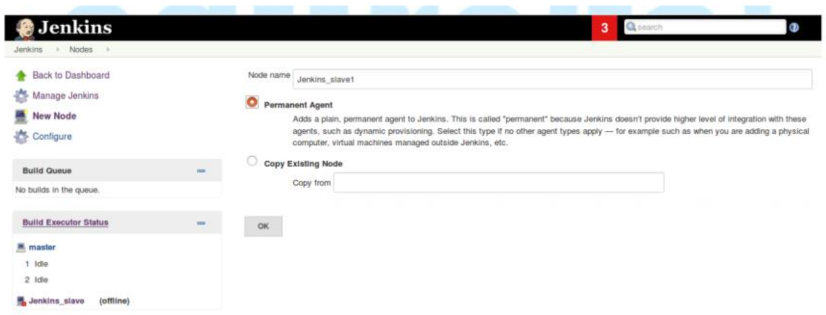
About Jenkins
See the version and license information.

Step 6: Click on new node and then give your node a name and make it a permanent agent.



No builds in the queue.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Jenkins_slave		N/A	N/A	N/A	N/A	Timed out for last 1 attempts
	master	Linux (amd64)	In sync	4.26 GB	2.00 GB	4.26 GB	0ms
	last checked	13 min	13 min	13 min	13 min	13 min	13 min



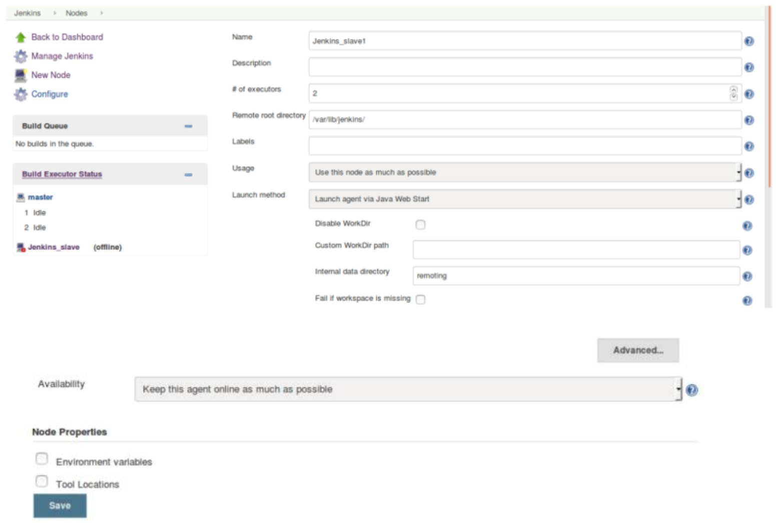
No builds in the queue.

Node name: Jenkins_slave1

[✓ **Permanent Agent**]
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Copy Existing Node
Copy from:

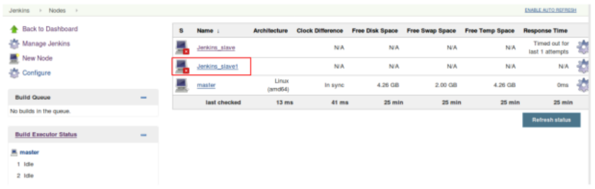
Step 7: Select the launch method as shown in the screen shot and then set the availability as shown in the screenshot and then click on the save button.



Name: Jenkins_slave1
Description:
# of executors: 2
Remote root directory: /var/lib/jenkins/
Labels:
Usage: Use this node as much as possible
Launch method: Launch agent via Java Web Start
Disable WorkDir: ☐
Custom WorkDir path:
Internal data directory: remoting
Fail if workspace is missing: ☐
Availability: Keep this agent online as much as possible

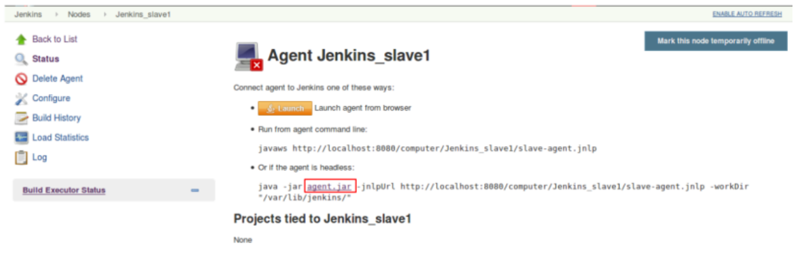
Node Properties
☐ Environment variables
☐ Tool Locations

Step 8: Click on the name you have given to your slave node.



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Jenkins_slave		N/A	N/A	N/A	N/A	Timed out for last 1 attempts
	Jenkins_slave1		N/A	N/A	N/A	N/A	N/A
	master	Linux (amd64)	In sync	4.26 GB	2.00 GB	4.26 GB	0ms
	last checked	13 ms	41 ms	25 min	25 min	25 min	25 min

Step 9: Click on the agent.jar and download the file, also it should be present on the node which you want to add as slave node.



Agent Jenkins_slave1

Connect agent to Jenkins one of these ways:

- Launch agent from browser
- Run from agent command line:

```
javaws http://localhost:8080/computer/Jenkins_slave1/slave-agent.jnlp
```

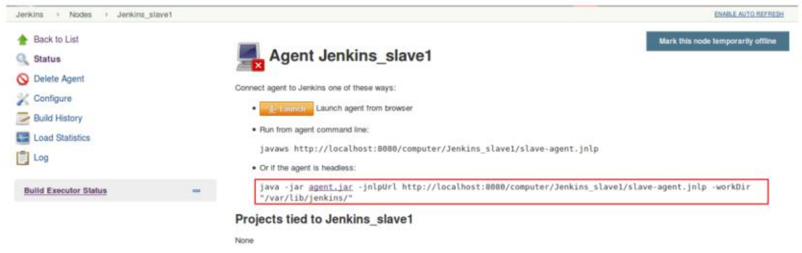
- Or if the agent is headless:

```
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/Jenkins_slave1/slave-agent.jnlp -workDir
"/var/lib/jenkins/"
```

Projects tied to Jenkins_slave1

None

Step 10: Copy the whole link and run it on the slave node which you want to add.



Agent Jenkins_slave1

Connect agent to Jenkins one of these ways:

- Launch agent from browser
- Run from agent command line:

```
javaws http://localhost:8080/computer/Jenkins_slave1/slave-agent.jnlp
```

- Or if the agent is headless:

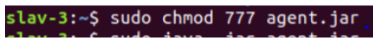
```
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/Jenkins_slave1/slave-agent.jnlp -workDir
"/var/lib/jenkins/"
```

Projects tied to Jenkins_slave1

None

Step 11: Go to the slave node and download that agent.jar file on this node. Change the permission of that file.

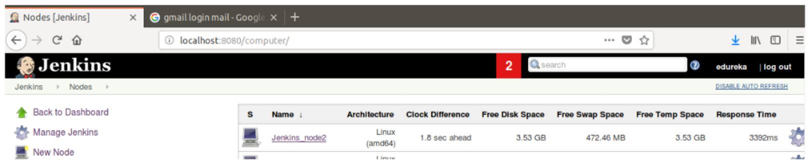
Then copied link from Jenkins master should be pasted here and replace the local host with the IP address of the Jenkins master node. You will get the below output.



```
slav-3:~$ sudo chmod 777 agent.jar
```

Execute the above java string on slave and mention ip address of master node.

Step 12: Go to your Jenkins master and check if your slave node is connected or not.



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Jenkins_node2	Linux (amd64)	1.8 sec ahead	3.53 GB	472.46 MB	3.53 GB	3392ms