

Aim :

To learn how to build different jobs with Jenkins and then integrate them to create Continuous Integration and Continuous Deployment Pipeline.

Problem Statement :

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.

Steps to solve:

- Open Jenkins and create a Maven project using it.
- You will have to create the following jobs, which are as follows:
 - O Compile
 - O Code Review
 - O Unit test
 - O Package
 - O Deploy

Solution : 3.1**1. Start the Jenkins**

```
durgeshmandge@durgeshmandge-VirtualBox:~$ sudo systemctl enable jenkins
[sudo] password for durgeshmandge:
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
durgeshmandge@durgeshmandge-VirtualBox:~$ sudo systemctl start jenkins
```

```

durgeshmandge@durgeshmandge-VirtualBox: ~$ sudo systemctl start jenkins
durgeshmandge@durgeshmandge-VirtualBox: ~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-02-02 10:07:22 IST; 2min 31s ago
     Main PID: 638 (java)
       Tasks: 43 (limit: 4599)
      Memory: 342.8M
         CPU: 15.744s
    CGroup: /system.slice/jenkins.service
            └─638 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/

Feb 02 10:07:19 durgeshmandge-VirtualBox jenkins[638]: WARNING: All illegal access operations will be denied in a future release
Feb 02 10:07:20 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:20.175+0000 [id=28] INFO h.p.b.g.GlobalTimeoutConfiguration#load: global timeout not set
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.233+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.710+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.776+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.885+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.895+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
Feb 02 10:07:22 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:22.002+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
Feb 02 10:07:22 durgeshmandge-VirtualBox system[1]: Started Jenkins Continuous Integration Server.
lines 1-20/20 (END) ...skipping...
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-02-02 10:07:22 IST; 2min 31s ago
     Main PID: 638 (java)
       Tasks: 43 (limit: 4599)
      Memory: 342.8M
         CPU: 15.744s
    CGroup: /system.slice/jenkins.service
            └─638 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Feb 02 10:07:19 durgeshmandge-VirtualBox jenkins[638]: WARNING: All illegal access operations will be denied in a future release
Feb 02 10:07:20 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:20.175+0000 [id=28] INFO h.p.b.g.GlobalTimeoutConfiguration#load: global timeout not set
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.233+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.710+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.776+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.885+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
Feb 02 10:07:21 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:21.895+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
Feb 02 10:07:22 durgeshmandge-VirtualBox jenkins[638]: 2024-02-02 04:37:22.002+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
Feb 02 10:07:22 durgeshmandge-VirtualBox system[1]: Started Jenkins Continuous Integration Server.
~

```

2. Make a job for Compile

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/edureka-git/DevOpsClassCodes.git

Credentials ?

- none - ▾

+ Add ▾

Advanced ▾

Add Repository

Save

Apply

3.

3. Select the name and Goal of job - compile

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

≡ Invoke top-level Maven targets ?

Maven Version

mymaven ▾

Goals

compile ▾

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save

Apply

4. Click on build and navigate to workspace and following location to see class files.**Output**

Dashboard > compile > Workspace

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History

Workspace of compile on Built-In Node

compile / target / classes / com / edureka-demo / utilities /

CaseInsensitiveComparator.class	Feb 2, 2024, 12:11:21 PM	4.00 KB	
GenericComparator.class	Feb 2, 2024, 12:11:21 PM	5.90 KB	
GenericComparator\$1.class	Feb 2, 2024, 12:11:21 PM	918 B	
GenericComparator\$CompareMode.class	Feb 2, 2024, 12:11:21 PM	1.32 KB	
HexAsciiConvertor.class	Feb 2, 2024, 12:11:21 PM	1.69 KB	
LoggerStackTraceUtil.class	Feb 2, 2024, 12:11:22 PM	2.32 KB	
PrepareTargetMethod.class	Feb 2, 2024, 12:11:22 PM	945 B	
PropertyHelper.class	Feb 2, 2024, 12:11:21 PM	2.33 KB	
PropertyLoader.class	Feb 2, 2024, 12:11:21 PM	2.52 KB	
StringUtilities.class	Feb 2, 2024, 12:11:22 PM	4.95 KB	

(all files in zip)

CodeReview

- Initialise the repository and select branch

Source Code Management

☐ None
 ☒ Git

Repositories

Repository URL

https://github.com/edureka-git/DevOpsClassCodes.git

Credentials

- none -

+ Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Save

Apply

2. Select goal as pmd:pmd and postbuild action as pmd it will generate the redable version of XML file

Build Steps

≡ Invoke top-level Maven targets ?

Maven Version

mymaven

Goals

pmd:pmd

Advanced

Add build step

Post-build Actions

≡ Report Violations ?

XML filename pattern ?

checkstyle

10

999

999

codenarc

10

999

999

jslint

10

999

999

pep8

10

999

999

perlritic

10

999

999

pmd

10

999

999

target/pmd.xml

pylint

10

999

999

simian

10

999

999

Output : pmd.xml

/orkspace

Workspace of codeReview on Built-In Node

codeReview / target / →

- 📁 pmd/rulesets
- 📁 site
- 📄 pmd.xml Feb 2, 2024, 12:44:53 PM 5.25 KB 👁


📄 (all files in zip)

```

<!--
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.
-->
<ruleset name="Default Maven PMD Plugin Ruleset" xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0 http://pmd.sourceforge.net/ruleset_2_0_0.xsd">
  <description>
    The default ruleset used by the Maven PMD Plugin, when no other ruleset is specified. It contains the rules of the old (pre PMD 6.0.0) rulesets java-basic, java-empty, java-imports, java-unnecessary, java-
    might be used as a starting point for an own customized ruleset [0]. [0] https://pmd.github.io/latest/pmd_userdocs_making_rulesets.html
  </description>
  <rule ref="category/java/bestpractices.xml/AvoidUsingHardCodedIP"/>
  <rule ref="category/java/bestpractices.xml/CheckResultSet"/>
  <rule ref="category/java/bestpractices.xml/PrimitiveWrapperInstantiation"/>
  <rule ref="category/java/bestpractices.xml/UnusedFormalParameter"/>
  <rule ref="category/java/bestpractices.xml/UnusedLocalVariable"/>
  <rule ref="category/java/bestpractices.xml/UnusedPrivateField"/>
  <rule ref="category/java/bestpractices.xml/UnusedPrivateMethod"/>
  <rule ref="category/java/codestyle.xml/EmptyControlStatement"/>
  <rule ref="category/java/codestyle.xml/ExtendsObject"/>
  <rule ref="category/java/codestyle.xml/ForLoopShouldBeWhileLoop"/>
  <rule ref="category/java/codestyle.xml/TooManyStaticImports"/>
  <rule ref="category/java/codestyle.xml/UnnecessaryFullyQualifiedName"/>
  <rule ref="category/java/codestyle.xml/UnnecessaryImport"/>
  <rule ref="category/java/codestyle.xml/UnnecessaryModifier"/>
  <rule ref="category/java/codestyle.xml/UnnecessaryReturn"/>
  <rule ref="category/java/codestyle.xml/UnnecessarySemicolon"/>
  <rule ref="category/java/codestyle.xml/UselessParentheses"/>
  <rule ref="category/java/codestyle.xml/UselessQualifiedThis"/>
  <rule ref="category/java/design.xml/CollapsibleIfStatements"/>
  <rule ref="category/java/design.xml/SimplifiedTernary"/>
  <rule ref="category/java/design.xml/UselessOverridingMethod"/>
  <rule ref="category/java/errorprone.xml/AvoidBranchingStatementAsLastInLoop"/>
</ruleset>

```



Violations Report for build 2

Type	Violations	Files in violation
pmd	12	6

nd

Count

12

10

8

6

4

2

0

Low

Medium

High

filename

src/main/java/com/edurekademo/utilities/StringUtilities.java

src/main/java/com/edurekademo/utilities/PropertyLoader.java

src/main/java/com/edurekademo/utilities/PropertyHelper.java

src/main/java/com/edurekademo/tutorial/addressbook/ContactForm.java

src/main/java/com/edurekademo/tutorial/addressbook/backend/ContactService.java

src/main/java/com/edurekademo/utilities/GenericComparator.java

Test

1. Initialise the repository and select the branch

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ? ✕

`https://github.com/edureka-git/DevOpsClassCodes.git`

Credentials ?

- none -

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? ✕

`*/master`

Add Branch

Save Apply

2. In post build steps select the target.xml and click on save

Post-build Actions

≡ Publish JUnit test result report ? ✕

Test report XMLs

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is [the workspace root](#).

target/surefire-reports/TEST-com.edurekademo.utilities.TestLogger.xml

Test output retention ?

All tests

☐ keep all the properties

Health report amplification factor ?

1.0

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results ?

☐ Do not fail the build on empty test results

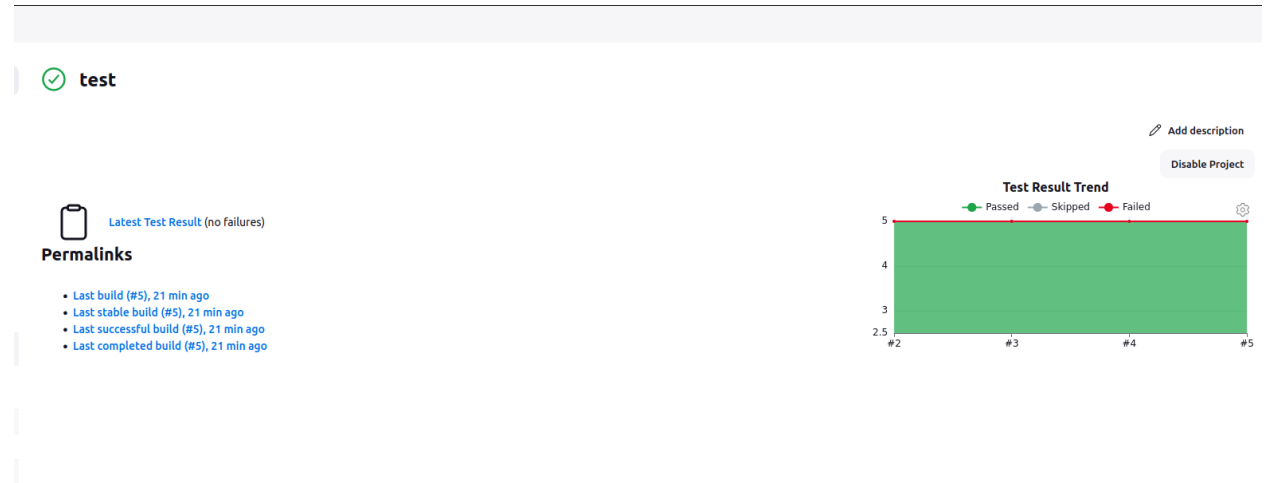
Skip publishing checks ?

☐ If unchecked, then issues will be published to SCM provider platforms

Checks name ?

Save Apply

Output:



Metric check

1. Select the name and branch for metric check

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/edureka-gilt/DevOpsClassCodes.git

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Save Apply

2. Select the goal as package and in post build actions add “**/**.exec” name

Build Steps

≡

Invoke top-level Maven targets ?

×

Maven Version

mymaven

▼

Goals

package

▼

Advanced

▼

Add build step ▼

Post-build Actions

≡

Record JaCoCo coverage report ?

×

Path to exec files (e.g.: **/target/**/*.exec, **/jacoco.exec)

Path to class directories (e.g.: **/target/classDir, **/classes)

Inclusions (e.g.: **/*.class)

Exclusions (e.g.: **/*Test*.class)

**/*.exec

**/classes

Save

Apply

3. Go to Post Build Actions, select Record Jacoco coverage report.

Path to class directories (e.g.: **/target/classDir, **/classes)

**/classes

Path to source directories (e.g.: **/mySourceFiles)

Path to source directories (e.g.: **/mySourceFiles)

Inclusions (e.g.: **/*.java, **/*.groovy, **/*.gs)

Exclusions (e.g.: generated/**/*.java)

**/src/main/java

**/*.java, **/*.groovy, **/*.kt, **/*.kts

☐ Disable display of source files for coverage ?

☐ Change build status according to the defined thresholds ?

☐ Always run coverage collection, even if build is FAILED or ABORTED ?

☀

0

☁

0

Instruction	% Branch	% Complexity	% Line	% Method	% Class
0	0	0	0	0	0
0	0	0	0	0	0

☐ Fail the build if coverage degrades more than the delta thresholds ?

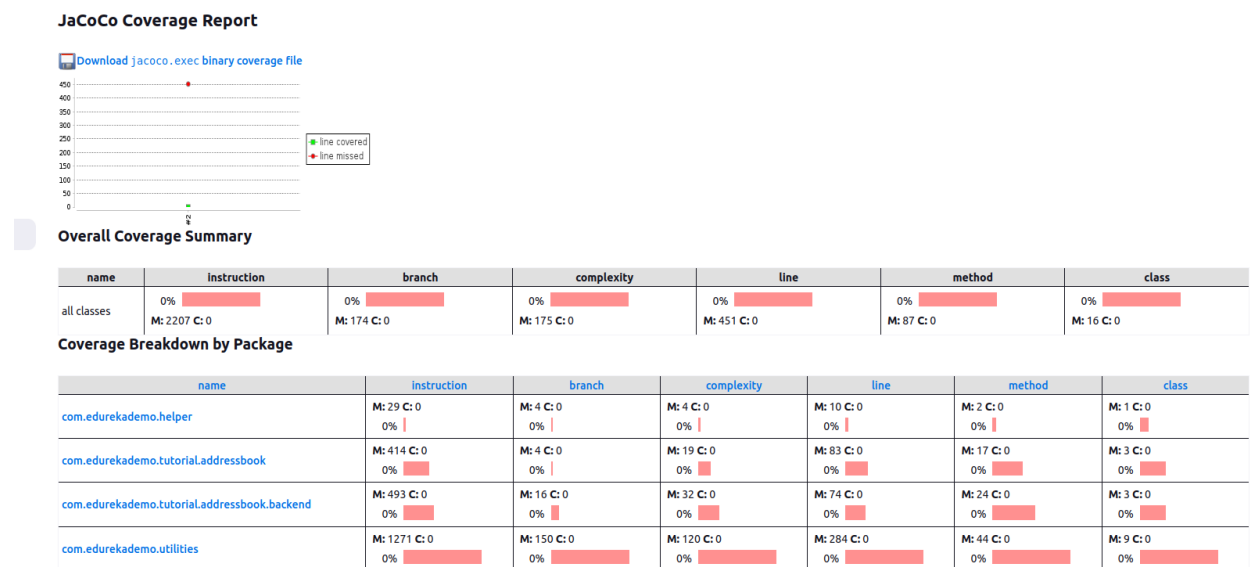
Instruction	% Branch	% Complexity	% Line	% Method	% Class
0	0	0	0	0	0

Add post-build action ▼

Save

Apply

4. Save and build your project.



Package:

Create a new freestyle project named package. In source code management select GIT and add the url of your GIT repository. Go to Build-steps, select Invoke top level maven targets, select mymaven in maven target and set goal as package

Git ?

Repositories ?

Repository URL ?

https://github.com/edureka-git/DevOpsClassCodes.git

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Save Apply

Go to Post Build Actions, select Archive the artifacts. In files to archive, type target/*.war

Build Steps

≡ Invoke top-level Maven targets ?

Maven Version

mymaven

Goals

package

Advanced

Add build step

Post-build Actions

≡ Archive the artifacts ?

Files to archive ?

target/*.war

Advanced

Add post-build action

Save Apply

Save and build your project

board > package >

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename


Build History


trend ▾

Filter builds... /

Feb 2, 2024, 3:25 PM

✓ package

 Last Successful Artifacts

 addressbook.war 15.77 MB [view](#)

Permalinks

- [Last build \(#2\), 35 sec ago](#)
- [Last stable build \(#2\), 35 sec ago](#)
- [Last successful build \(#2\), 35 sec ago](#)
- [Last completed build \(#2\), 35 sec ago](#)

Problem Statement (3.2):

- 1) 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 a successful build of unit testing
- 2) Create a freestyle project with the name QA_METRICS_CHECK in Jenkins to check the test cases. Make sure the Cobertura plugin is installed in Jenkins. Take a screenshot of the metrics from the dashboard of the project.
- 3) 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 the workspace.
- 4) Create a pipeline named SAMPLE_COMPILE_VIEW with Build Pipeline View option, select DEVELOPER_CODE_REVIEW 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
- 5) The pipelines can also be extended to running web tests and load tests. Explain how you would do the same using Jenkins?

Solution (3.2):

Click on New View, select Build pipeline view and give a name for your view. In select initial job, select Compile

Go to post build actions, select Build other projects and choose codeReview

The screenshot shows the Jenkins Configuration page for a job named 'compile'. The browser address bar indicates the URL is 'localhost:8080/job/compile/configure'. The left sidebar contains a 'Configure' section with a list of tabs: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' tab is currently selected. The main content area is divided into two sections. The top section, 'Configure', contains a text input field with 'mymaven', a 'Goals' section with a text input field containing 'compile', and an 'Advanced' dropdown menu. The bottom section, 'Post-build Actions', features a dashed border and contains a 'Build other projects' action with a help icon. Below this, the 'Projects to build' section has a text input field containing 'codeReview'. Three radio buttons are present: 'Trigger only if build is stable' (which is selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. At the bottom of the 'Post-build Actions' section is an 'Add post-build action' dropdown. At the very bottom of the configuration page are two buttons: 'Save' and 'Apply'.

← → ↻ localhost:8080/job/compile/configure

Dashboard > compile > Configuration

Configure

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

mymaven

Goals

compile

Advanced ▾

Add build step ▾

Post-build Actions

≡ Build other projects ?

Projects to build

codeReview

☒ 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

Then go to your Compile project, go to Configure – build triggers, select PollSCM, in schedule add * * * * *.

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 ?

Schedule ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H*****" to poll once per hour
Would last have run at Tuesday, 6 February, 2024 at 11:00:31 PM India Standard Time; would next run at Tuesday, 6
February, 2024 at 11:00:31 PM India Standard Time.


- ☐ Ignore post-commit hooks ?

←


→


↺


localhost:8080/job/codeReview/


 **Jenkins**


Dashboard > codeReview >


 Status


 Changes


 Workspace


 Build Now


 Configure

 Delete Project


 Violations

 Rename


 **Build History** trend ▾

 Filter builds...


/

 **#3**


[Feb 6, 2024, 9:52 AM](#)


 **#2**


[Feb 2, 2024, 12:44 PM](#)

 **#1**


[Feb 2, 2024, 12:24 PM](#)

 [Atom feed for all](#)


 [Atom feed for failures](#)

 **codeReview**

Upstream Projects

 [compile](#)

Downstream Projects

 [test](#)

Permalinks

- [Last build \(#3\), 13 min ago](#)
- [Last stable build \(#3\), 13 min ago](#)
- [Last successful build \(#3\), 13 min ago](#)
- [Last completed build \(#3\), 13 min ago](#)

Similarly for codeReview, go to Build triggers, select Build after other projects are built and choose Compile

Go to post build actions, select Build other projects and choose test

Repeat the same procedure for test, metricCheck and package with following details:

Test – build triggers – codeReview, post build actions – metricCheck
metricCheck – build triggers – test, post build actions – package
package – build triggers – metricCheck

The screenshot shows the Jenkins configuration page for a job named 'codeReview'. The browser address bar indicates the URL is 'localhost:8080/job/codeReview/configure'. The breadcrumb navigation shows 'Dashboard > codeReview > Configuration'. On the left, the 'Configure' sidebar has several options: 'General', 'Source Code Management', 'Build Triggers' (which is selected and highlighted), 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main content area is titled 'Build Triggers' and contains several configuration options:

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☒ Build after other projects are built ?
- Projects to watch:
- ☒ Trigger only if build is stable
- ☐ Trigger even if the build is unstable
- ☐ Trigger even if the build fails
- ☐ Always trigger, even if the build is aborted
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

←

→

↺

localhost:8080/job/codeReview/configure

Dashboard > codeReview > Configuration

Add build step ▾

Configure

⚙️ General

🔑 Source Code Management

🕒 Build Triggers

🌐 Build Environment

☰ Build Steps

📦 Post-build Actions

Post-build Actions

☰ Build other projects ?

Projects to build

test


☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails


☰ Report Violations ?


☀️ ☁️ 🟡 XML filename pattern ?





Jenkins


Dashboard > test >


 Status


 Changes


 Workspace

 Build Now

 Configure

 Delete Project

 Rename




 Build History


trend


▼

Filter builds...


/

 #6	Feb 6, 2024, 9:53 AM
 #5	Feb 2, 2024, 2:43 PM
 #4	Feb 2, 2024, 2:42 PM


 test

 Latest Test Result (no failures)

Upstream Projects

 [codeReview](#)

Downstream Projects






 [metriccheck](#)

Permalinks

- [Last build \(#6\), 16 min ago](#)
- [Last stable build \(#6\), 16 min ago](#)
- [Last successful build \(#6\), 16 min ago](#)

Dashboard > test > Configuration

Configure

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

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?☒ Build after other projects are built ?

Projects to watch






codeReview,

☒ Trigger only if build is stable☐ Trigger even if the build is unstable☐ Trigger even if the build fails☐ Always trigger, even if the build is aborted☐ Build periodically ?☐ GitHub hook trigger for GITScm polling ?☐ Poll SCM ?


Build Environment

Dashboard > test > Configuration

Configure

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


Post-build Actions

 **Build other projects** ?


Projects to build

metriccheck


☒ Trigger only if build is stable☐ Trigger even if the build is unstable☐ Trigger even if the build fails


 **Jenkins**


Dashboard > metriccheck >


 **Status**


</> Changes


 Workspace


 Build Now

 Configure


 Delete Project

 Rename


 Coverage Trend

 **Build History** **trend** ▾


Dashboard > metriccheck > Configuration

 **metriccheck**

Upstream Projects


 test


Downstream Projects


 package


Permalinks


- [Last build \(#3\), 17 min ago](#)
- [Last stable build \(#3\), 17 min ago](#)


 General

 Source Code Management

 **Build Triggers**

 Build Environment

 Build Steps

 Post-build Actions

Configure

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Projects to watch

test,

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Always trigger, even if the build is aborted

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Dashboard > metriccheck > Configuration

Configure

☐ With Ant ?

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

Build Steps

Invoke top-level Maven targets ?

Maven Version

mymaven

Goals

package

Advanced ▾

Add build step ▾

Activities Firefox Web Browser Feb 6, 09:53

samplePipeline [jenkins]

localhost:8080/view/samplePipeline/

Jenkins

Dashboard > samplePipeline >

Build Pipeline

Run History Configure Add Step Delete Manage

Pipeline #3

#3 compile 26 Feb 2024 9:52:27 AM 19 sec admin

#3 codeReview 26 Feb 2024 9:52:45 AM 16 sec

#6 test 26 Feb 2024 9:53:10 AM 1 sec

#3 metriccheck 26 Feb 2024 9:53:20 AM 13 sec

#3 package 26 Feb 2024 9:53:40 AM 7.8 sec

5. The pipelines can also be extended to running web tests and load tests. Explain how you would do the same using Jenkins?

Ans.

Web tests can be done by using libraries like Selenium Java, and JUnit.

In Jenkins we can automate this testing by using **Selenium Plugin**.

For web tests with Selenium, you might have scripts written in languages like Java, Python, or JavaScript. For JUnit tests, you'd execute your test classes. Similarly, for load tests with JMeter or Gatling, you'd invoke the test scripts or scenarios.

Observation:

In this lab practical I learned how to integrate all the jobs in Jenkins and trigger them one after another and execute entire CI/CD pipeline seamlessly.

