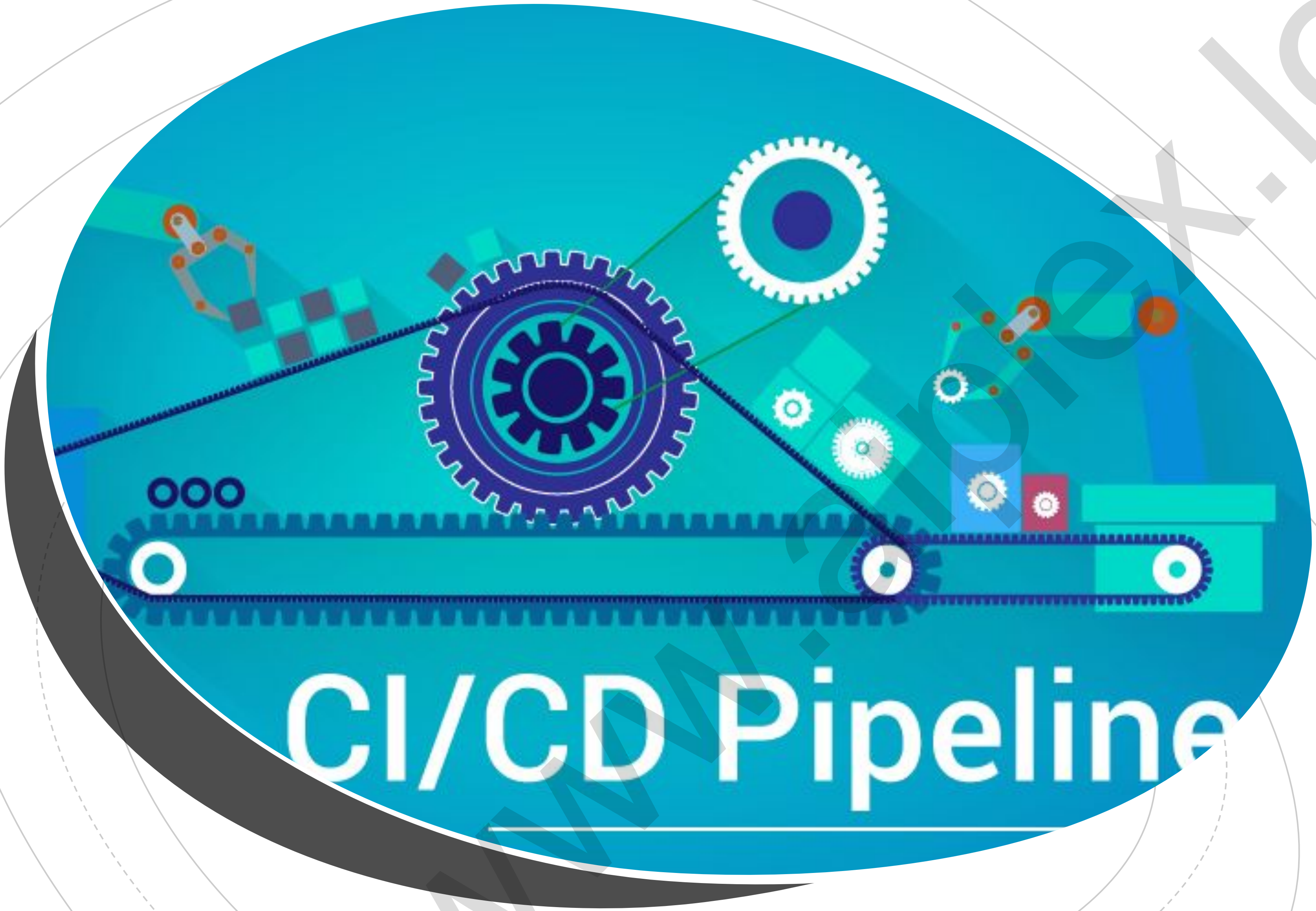
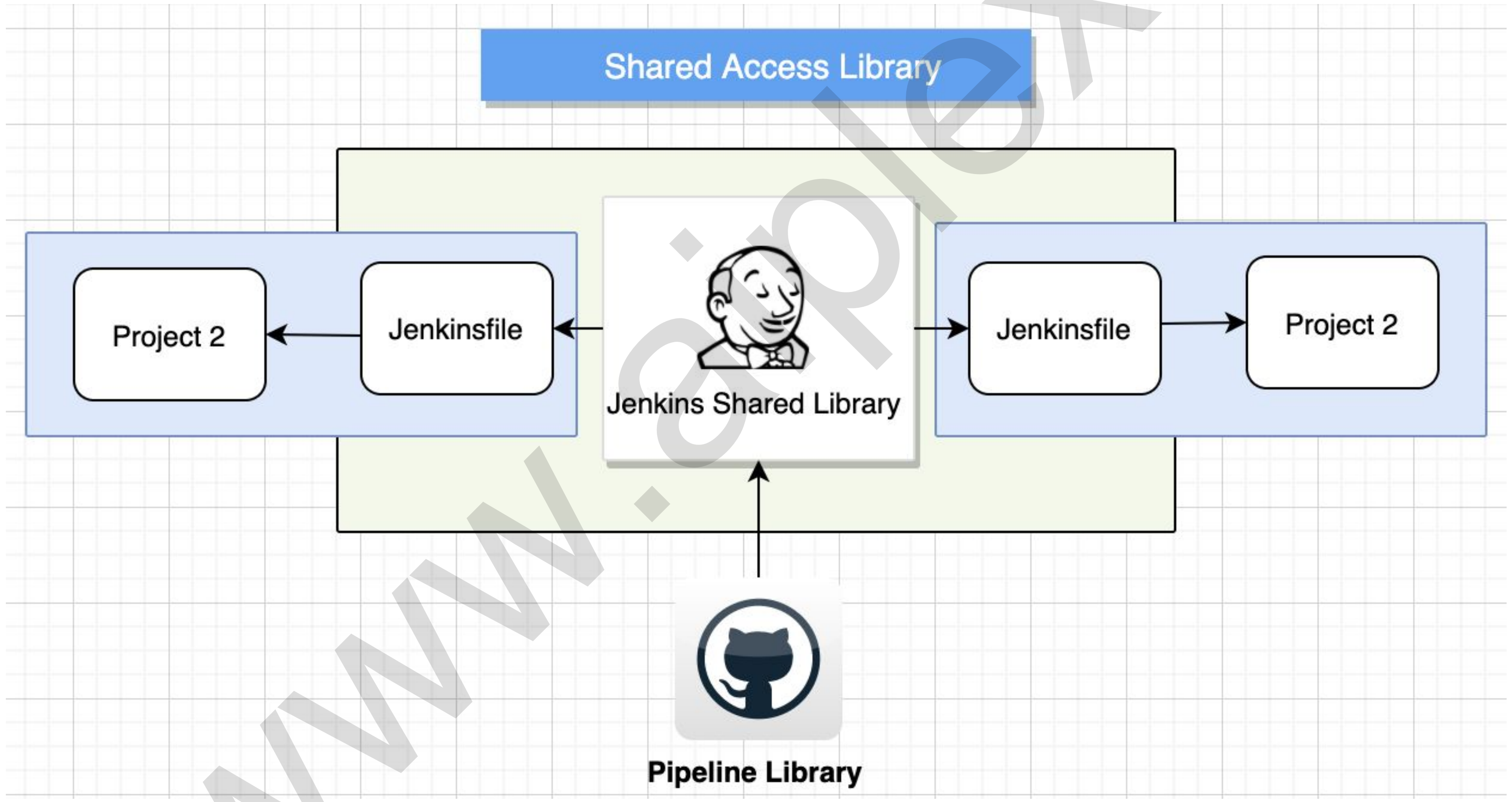


CICD Session-2

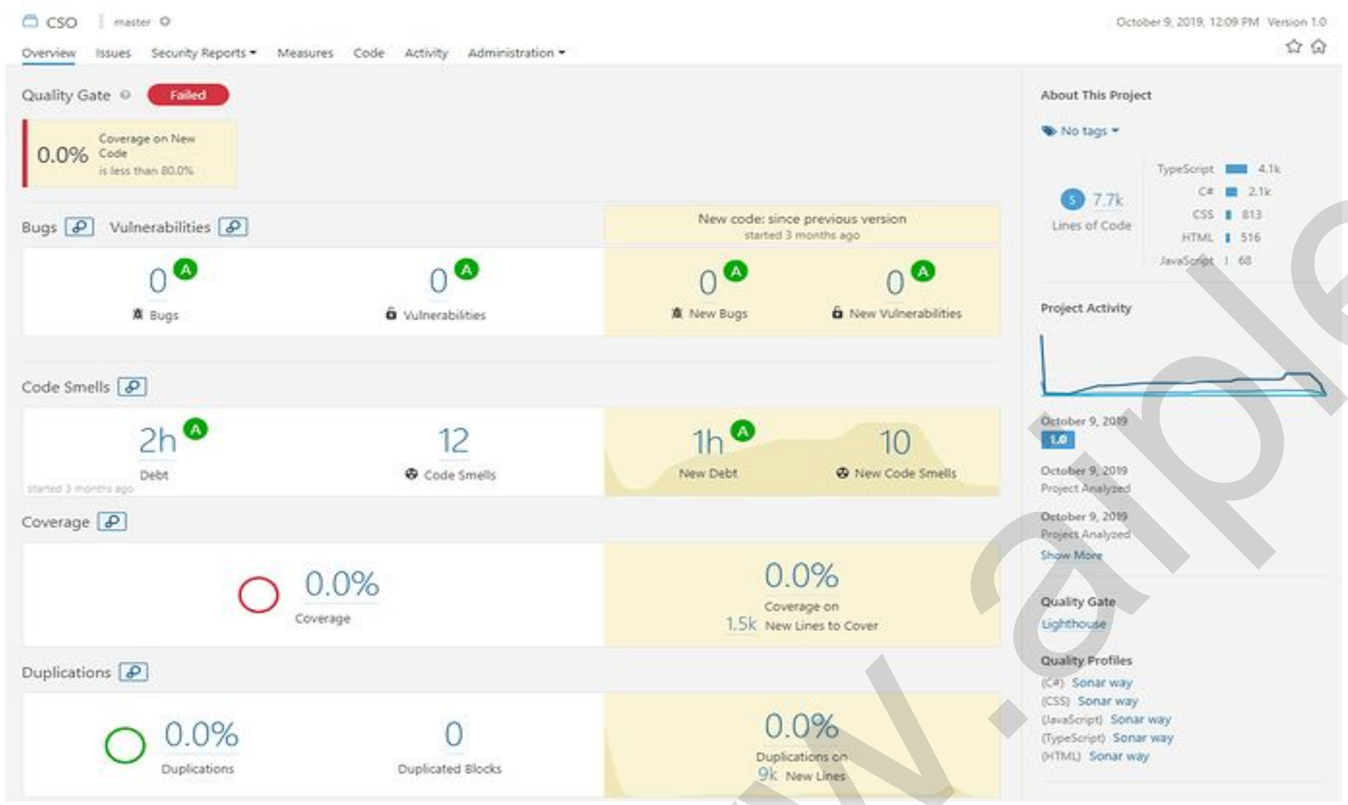
By Praveen Singampall

curl –x PUT –u admin:Test1234 –T kubernetes-configmap-reload-0.0.1-SNAPSHOT.jar http://54.23

Jenkins Shared library



SonarQube



CSO | master

October 9, 2019, 12:09 PM Version 1.0

Overview Issues Security Reports ▾ Measures Code Activity Administration ▾

Quality Gate Failed

0.0% Coverage on New Code is less than 80.0%

Bugs Vulnerabilities

New code: since previous version started 3 months ago

0 A Bugs | 0 A Vulnerabilities | 0 A New Bugs | 0 A New Vulnerabilities

Code Smells

2h A Debt | 12 Code Smells | 1h A New Debt | 10 New Code Smells

started 3 months ago

Coverage

0.0% Coverage | 0.0% Coverage on 1.5k New Lines to Cover

Duplications

0.0% Duplications | 0 Duplicated Blocks | 0.0% Duplications on 9k New Lines

About This Project

No tags ▾

7.7k Lines of Code

TypeScript	4.1k
C#	2.1k
CSS	813
HTML	516
JavaScript	68

Project Activity

October 9, 2019 1.0

October 9, 2019 Project Analyzed

October 9, 2019 Project Analyzed

Show More

Quality Gate Lighthouse

Quality Profiles

(C#) Sonar way
(CSS) Sonar way
(JavaScript) Sonar way
(TypeScript) Sonar way
(HTML) Sonar way

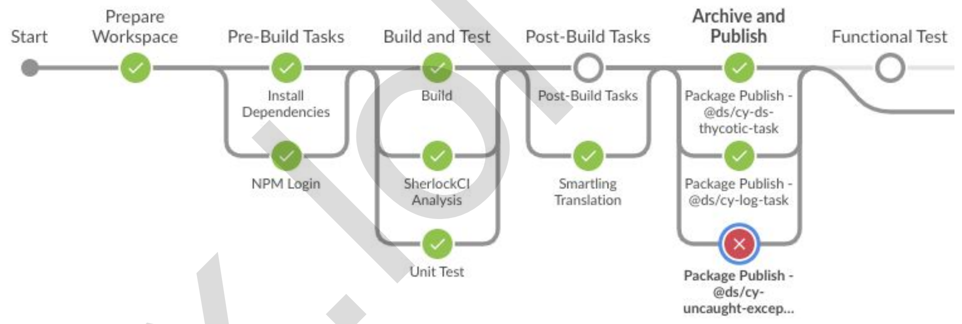
- SonarQube is an open-source platform developed by Sonar Source for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages.

Quality profiles & Quality Gates

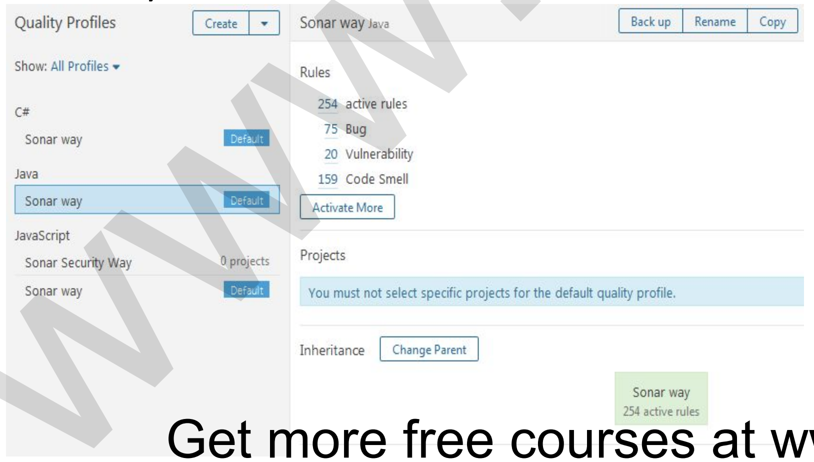
- **Quality Profiles** are a core component of SonarQube where you define sets of **Rules** that, when violated, raise issues on your codebase

SonarQube executes rules on source code to generate issues. There are four types of rules:

- Code Smell (Maintainability domain)
- Bug (Reliability domain)
- Vulnerability (Security domain)
- Security Hotspot (Security domain)



- Quality Gates can be defined as a set of threshold measures set on your project. Few conditions that can be in included are listed below.
- Code Coverage > certain value
- Number of Blocker issues >certain value
- Security Rating / Unit Test Pass Rate etc..


Quality Profiles Create

Show: All Profiles

C#
Sonar way Default

Java
Sonar way Default

JavaScript
Sonar Security Way 0 projects
Sonar way Default

Sonar way Java Back up Rename Copy

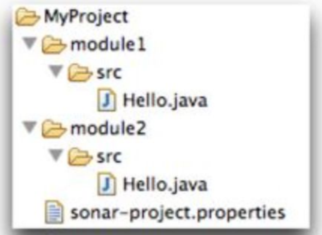
Rules
254 active rules
75 Bug
20 Vulnerability
159 Code Smell
Activate More

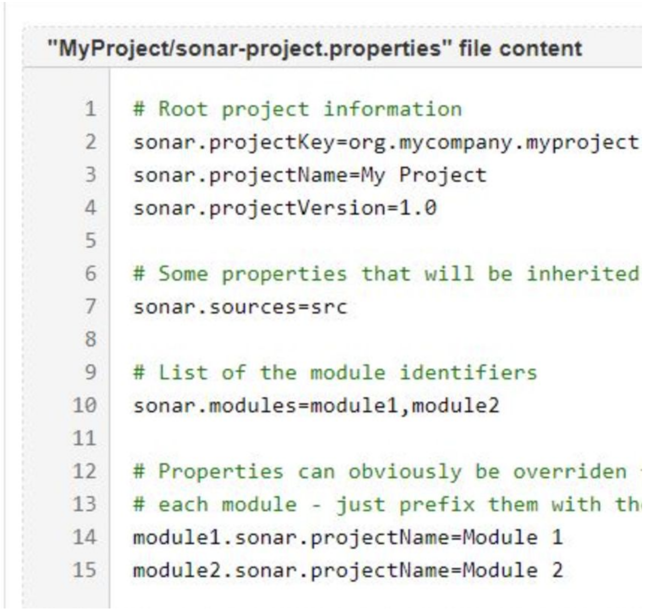
Projects
You must not select specific projects for the default quality profile.

Inheritance Change Parent
Sonar way
254 active rules

Sonar Code Coverage

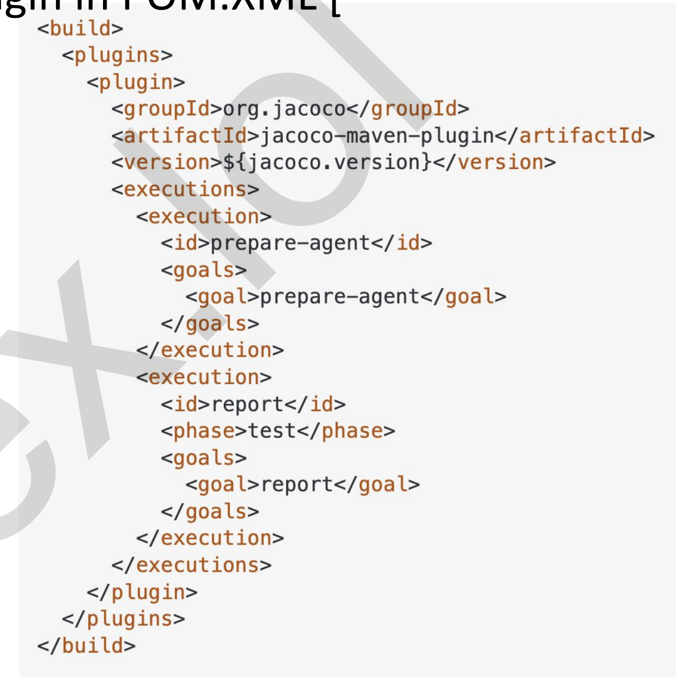
1) Add the sonar.properties file




"MyProject/sonar-project.properties" file content

```
# Root project information
sonar.projectKey=org.mycompany.myproject
sonar.projectName=My Project
sonar.projectVersion=1.0

# Some properties that will be inherited
sonar.sources=src

# List of the module identifiers
sonar.modules=module1,module2

# Properties can obviously be overriden
# each module - just prefix them with th
module1.sonar.projectName=Module 1
module2.sonar.projectName=Module 2
```

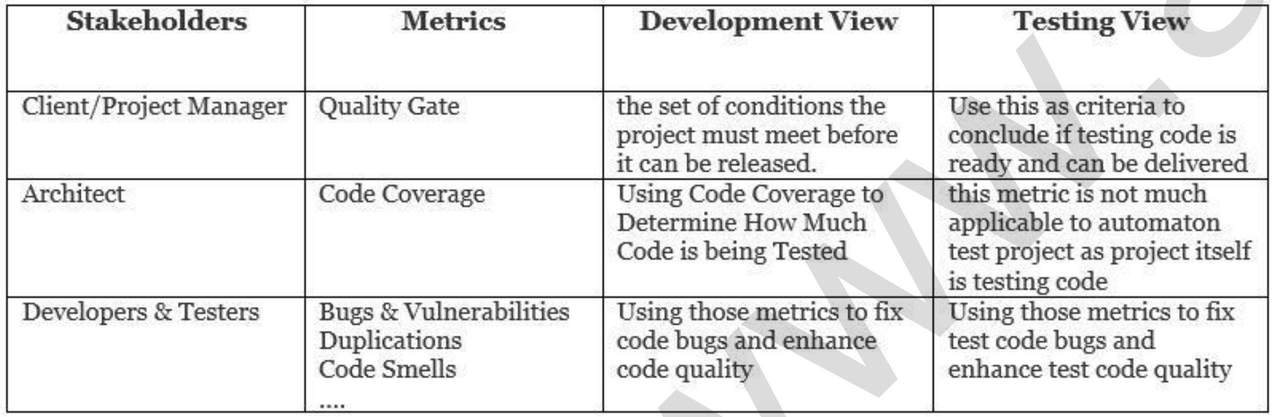
2) Add the Jacoco plugin in POM.XML [heart of java]



```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>${jacoco.version}</version>
      <executions>
        <execution>
          <id>prepare-agent</id>
          <goals>
            <goal>prepare-agent</goal>
          </goals>
        </execution>
        <execution>
          <id>report</id>
          <phase>test</phase>
          <goals>
            <goal>report</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

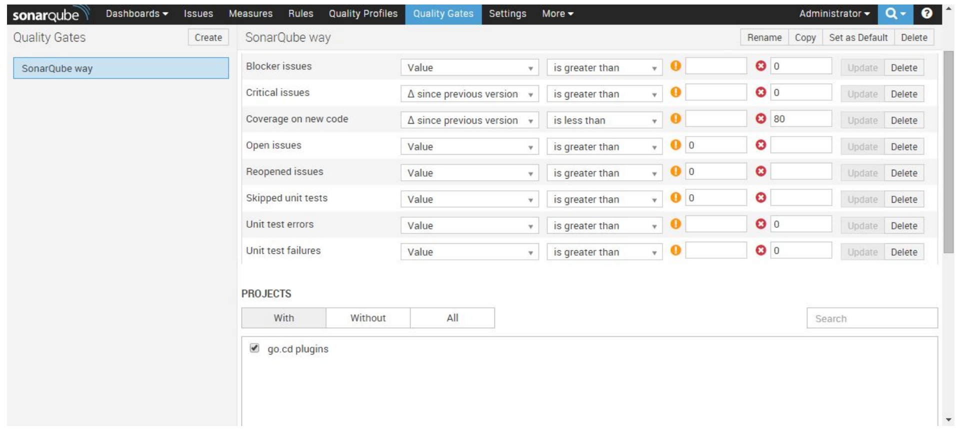
The reports can then be found in `target/site/jacoco/`

3) QG/CC/B&V



Stakeholders	Metrics	Development View	Testing View
Client/Project Manager	Quality Gate	the set of conditions the project must meet before it can be released.	Use this as criteria to conclude if testing code is ready and can be delivered
Architect	Code Coverage	Using Code Coverage to Determine How Much Code is being Tested	this metric is not much applicable to automaton test project as project itself is testing code
Developers & Testers	Bugs & Vulnerabilities Duplications Code Smells	Using those metrics to fix code bugs and enhance code quality	Using those metrics to fix test code bugs and enhance test code quality

4)QG threshold setup in Sonar



SonarQube way

Blocker issues	Value	is greater than		0	Update	Delete
Critical issues	Δ since previous version	is greater than		0	Update	Delete
Coverage on new code	Δ since previous version	is less than		80	Update	Delete
Open issues	Value	is greater than	0		Update	Delete
Reopened issues	Value	is greater than	0		Update	Delete
Skipped unit tests	Value	is greater than	0		Update	Delete
Unit test errors	Value	is greater than		0	Update	Delete
Unit test failures	Value	is greater than		0	Update	Delete

PROJECTS

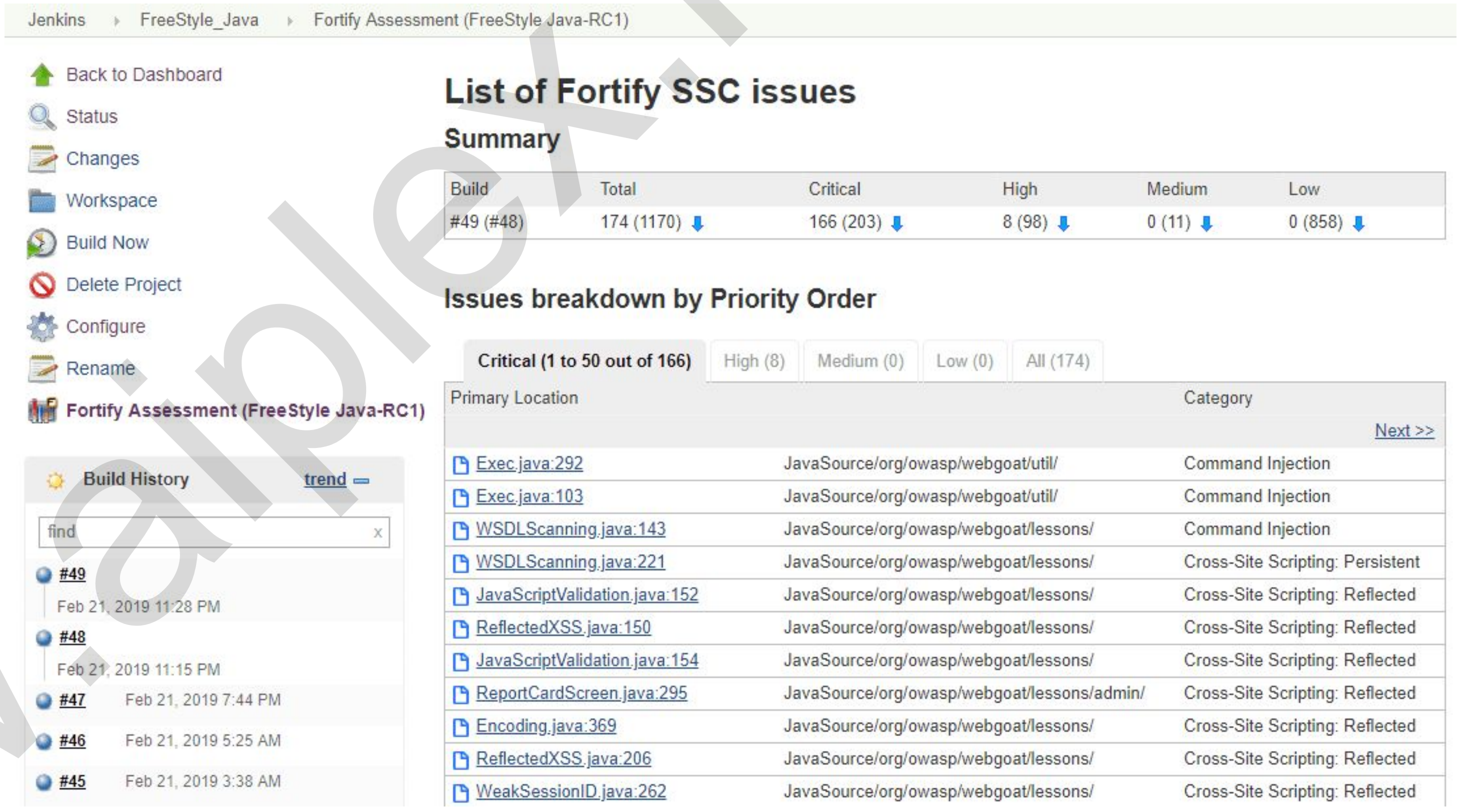
With | Without | All

Search

✓ go.cd plugins

Fortify(Security Tool)

- Static Analysis, also known as Static Application Security Testing (SAST), available from Fortify Static Code Analyzer (SCA).
- Detects more types of potential vulnerabilities than any other detection method
- Pinpoints the root cause of vulnerabilities with line-of-code detail
- Helps you identify critical issues during development when they are easiest and least expensive to fix


Jenkins › FreeStyle_Java › Fortify Assessment (FreeStyle Java-RC1)

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Rename
- Fortify Assessment (FreeStyle Java-RC1)

Build History trend

find

#49	Feb 21, 2019 11:28 PM
#48	Feb 21, 2019 11:15 PM
#47	Feb 21, 2019 7:44 PM
#46	Feb 21, 2019 5:25 AM
#45	Feb 21, 2019 3:38 AM

List of Fortify SSC issues

Summary

Build	Total	Critical	High	Medium	Low
#49 (#48)	174 (1170)	166 (203)	8 (98)	0 (11)	0 (858)

Issues breakdown by Priority Order

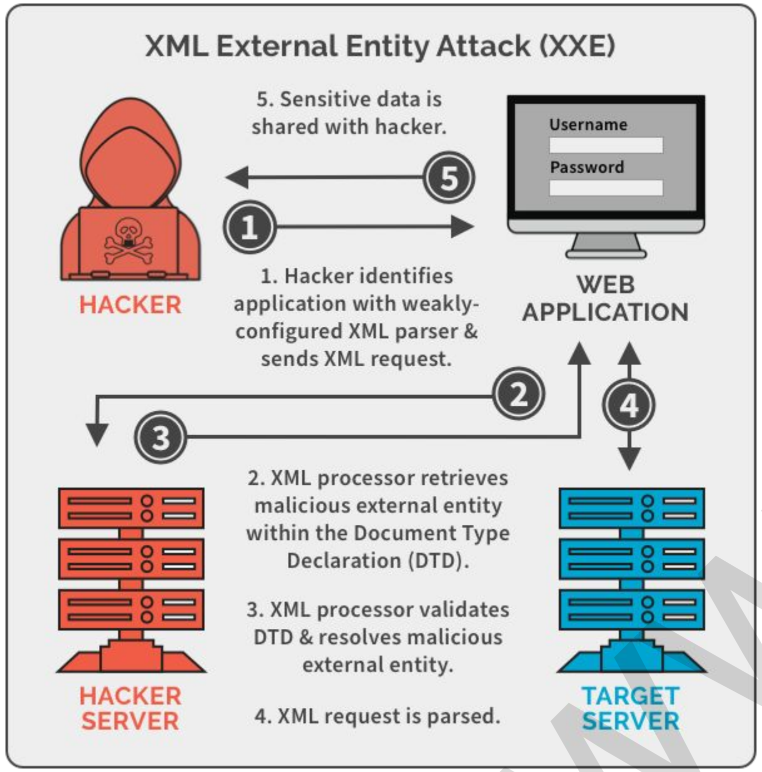
Critical (1 to 50 out of 166) | High (8) | Medium (0) | Low (0) | All (174)

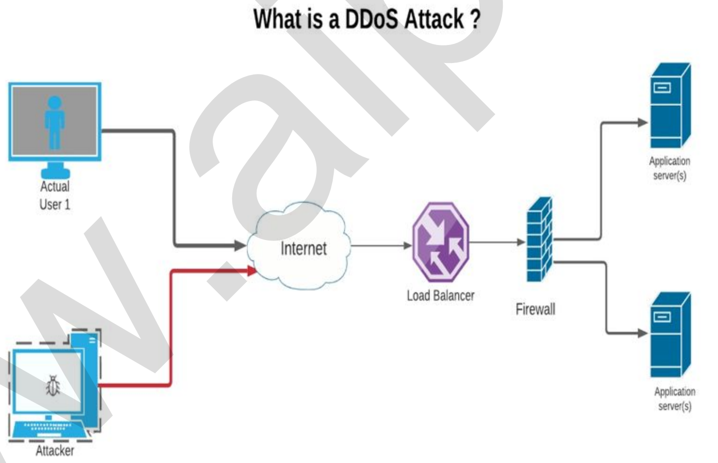
Next >>

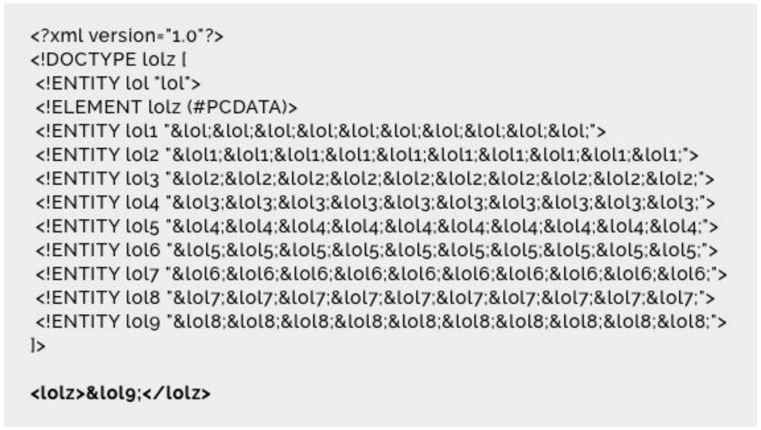
Primary	Location	Category
Exec.java:292	JavaSource/org/owasp/webgoat/util/	Command Injection
Exec.java:103	JavaSource/org/owasp/webgoat/util/	Command Injection
WSDLScanning.java:143	JavaSource/org/owasp/webgoat/lessons/	Command Injection
WSDLScanning.java:221	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Persistent
JavaScriptValidation.java:152	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected
ReflectedXSS.java:150	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected
JavaScriptValidation.java:154	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected
ReportCardScreen.java:295	JavaSource/org/owasp/webgoat/lessons/admin/	Cross-Site Scripting: Reflected
Encoding.java:369	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected
ReflectedXSS.java:206	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected
WeakSessionID.java:262	JavaSource/org/owasp/webgoat/lessons/	Cross-Site Scripting: Reflected

Fortify issues

1) Denial of service attack – Do not allow untrusted data to be used as regex like in headers (*)
2) XML external Entity attack - XML should be configured securely(For example you upload adhaar in portal then the doc should not bring any vulnerabilities into live environment)
3) Values shading – Not to take data from HTTP request instead take from cookies
4) Password management/Hardcode password


XML External Entity Attack (XXE)
5. Sensitive data is shared with hacker.
Username
Password
HACKER
WEB APPLICATION
1. Hacker identifies application with weakly-configured XML parser & sends XML request.
1
2
3
4
5
2. XML processor retrieves malicious external entity within the Document Type Declaration (DTD).
3. XML processor validates DTD & resolves malicious external entity.
4. XML request is parsed.
HACKER SERVER
TARGET SERVER

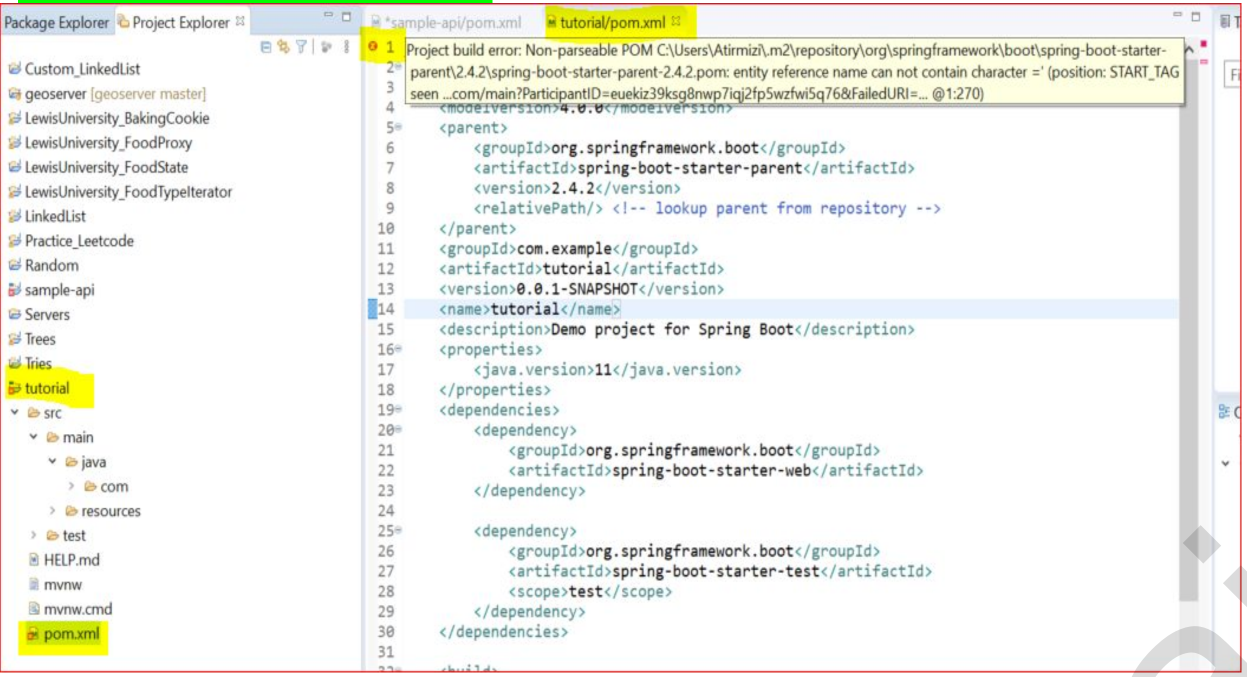




```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>

<lolz>&lol9;</lolz>
```

Maven E2E

1) POM.XML


Package Explorer | Project Explorer

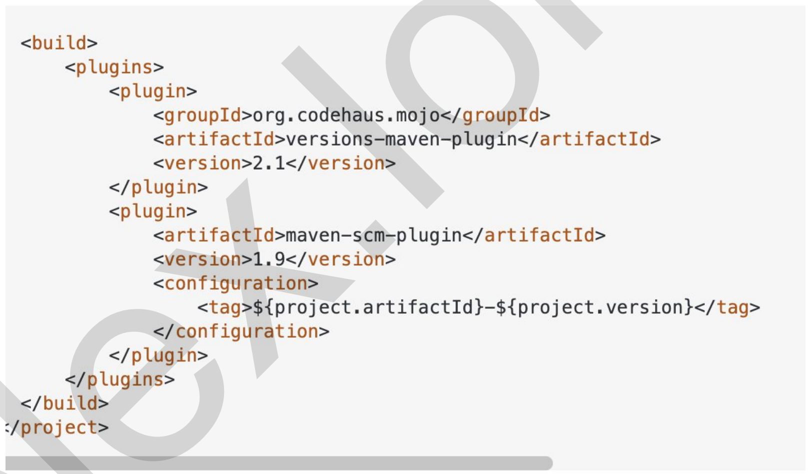
- Custom_LinkedList
- geoserver [geoserver master]
- LewisUniversity_BakingCookie
- LewisUniversity_FoodProxy
- LewisUniversity_FoodState
- LewisUniversity_FoodTypeIterator
- LinkedList
- Practice_Leetcode
- Random
- sample-api
- Servers
- Trees
- Tries
- tutorial
  - src
    - main
      - java
        - com
      - resources
    - test
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

*sample-api/pom.xml | tutorial/pom.xml

```
Project build error: Non-parseable POM C:\Users\Atirmizi\.m2\repository\org\springframework\boot\spring-boot-starter-parent\2.4.2\spring-boot-starter-parent-2.4.2.pom: entity reference name can not contain character =' (position: START_TAG seen ...com/main?ParticipantID=euekiz39ksg8nwp7iqj2fp5wzfwi5q76&FailedURI=... @1:270)
```

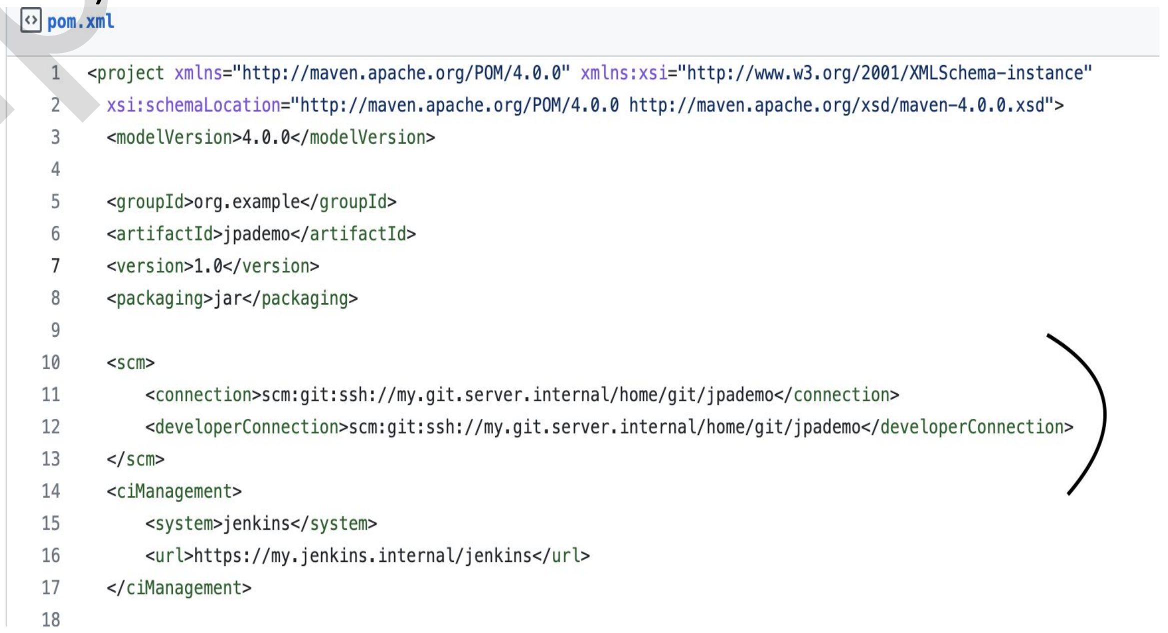
```xml
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.4.2</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>tutorial</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>tutorial</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>11</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

```

2) SCM PLUGIN ADD IN POM.XML



```xml
<build>
    <plugins>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>versions-maven-plugin</artifactId>
            <version>2.1</version>
        </plugin>
        <plugin>
            <artifactId>maven-scm-plugin</artifactId>
            <version>1.9</version>
            <configuration>
                <tag>${project.artifactId}-${project.version}</tag>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```

3) SCM TAG IN POM.XML

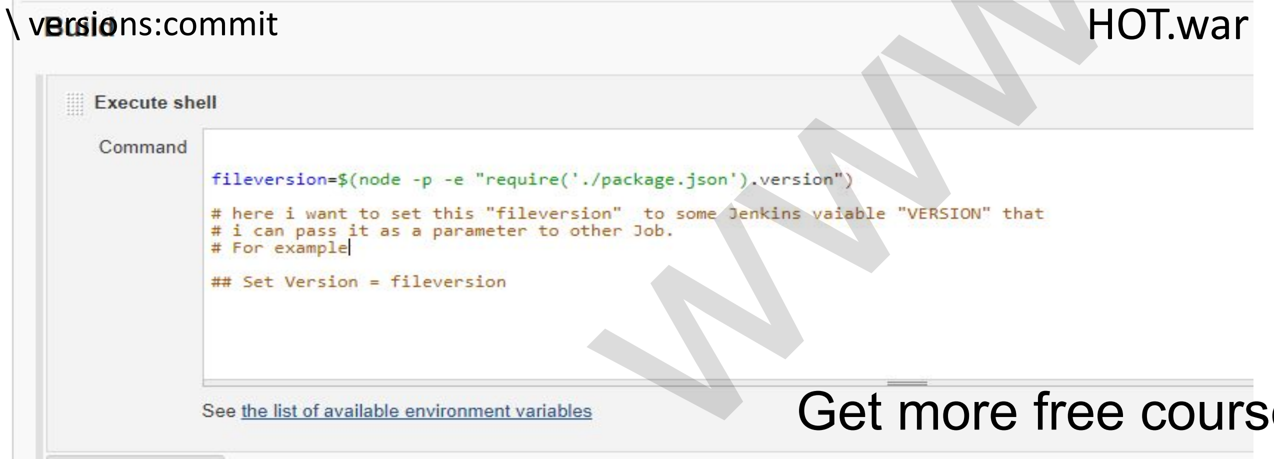

pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>jpademo</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <scm>
      <connection>scm:git:ssh://my.git.server.internal/home/git/jpademo</connection>
      <developerConnection>scm:git:ssh://my.git.server.internal/home/git/jpademo</developerConnection>
  </scm>
  <ciManagement>
      <system>jenkins</system>
      <url>https://my.jenkins.internal/jenkins</url>
  </ciManagement>

```

4) SHELL SCRIPT TO BE ADDED IN JENKINS TO INVOKE SCM

```
Mvn clean install
mvn build-helper:parse-version versions:set \
-DnewVersion=\${parsedVersion.nextMinorVersion}.0.1-SNAPSHOT
\ versions:commit
```

OUTPUT
Tutorial-1.0.1-SNAPSHOT.war


Build

Execute shell

Command

```
fileversion=$(node -p -e "require('./package.json').version")

# here i want to set this "fileversion"  to some Jenkins vaiable "VERSION" that
# i can pass it as a parameter to other Job.
# For example

## Set Version = fileversion
```

See the list of available environment variables

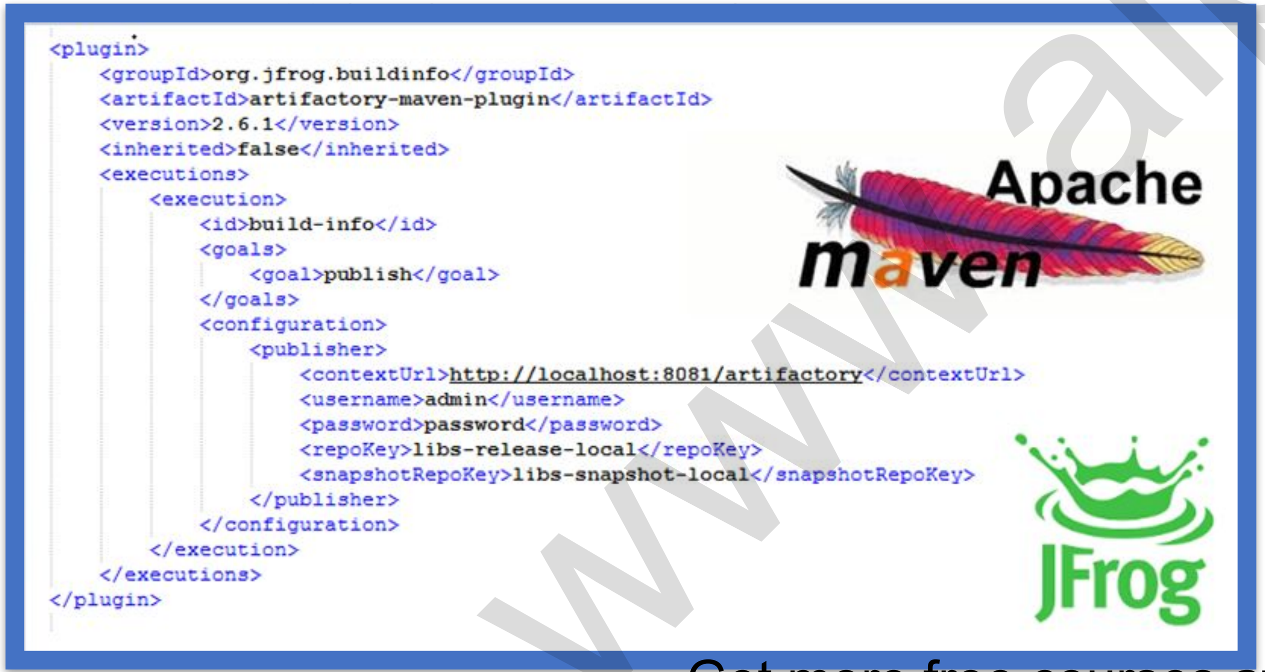
JFROG

- The Version which will be created in the above step is now sent to JFROG i.e. **Tutorial-1.0.1-SNAPSHOT.war**

How to upload the JAR/WAR or any binary into JFROG:

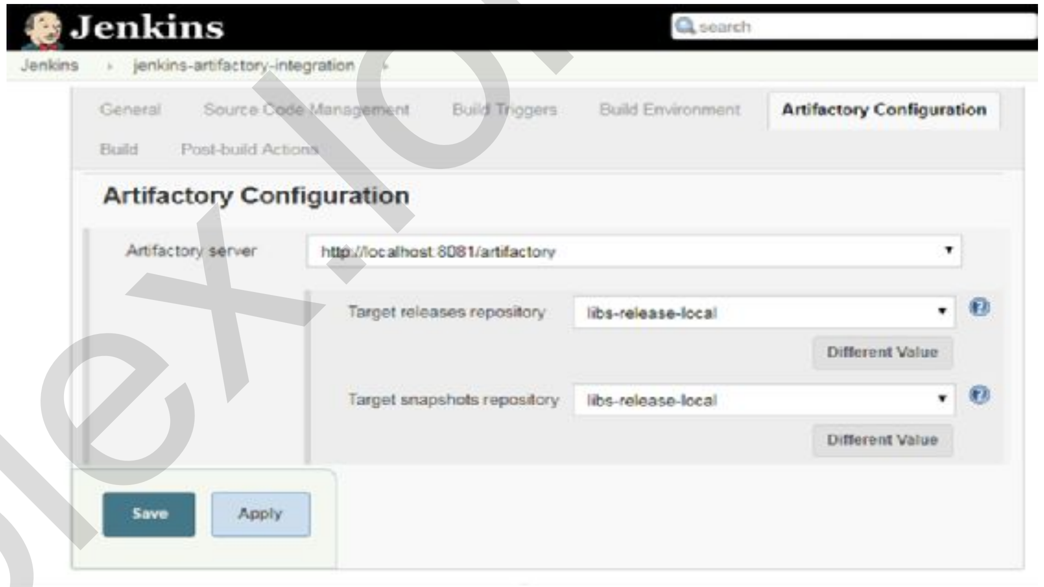
CASE 1 - Curl COMMAND to upload a artifact

```
curl -X PUT -u username:password –T
Tutorial-1.0.1-SNAPSHOT.war
http://localhost:8081/artifactory/libs-release-local/
Tutorial-1.0.1-SNAPSHOT.war
```



```
<plugin>
    <groupId>org.jfrog.buildinfo</groupId>
    <artifactId>artifactory-maven-plugin</artifactId>
    <version>2.6.1</version>
    <inherited>false</inherited>
    <executions>
        <execution>
            <id>build-info</id>
            <goals>
                <goal>publish</goal>
            </goals>
            <configuration>
                <publisher>
                    <contextUrl>http://localhost:8081/artifactory</contextUrl>
                    <username>admin</username>
                    <password>password</password>
                    <repoKey>libs-release-local</repoKey>
                    <snapshotRepoKey>libs-snapshot-local</snapshotRepoKey>
                </publisher>
            </configuration>
        </execution>
    </executions>
</plugin>
```

CASE 3 – Add Plugin in Jenkins



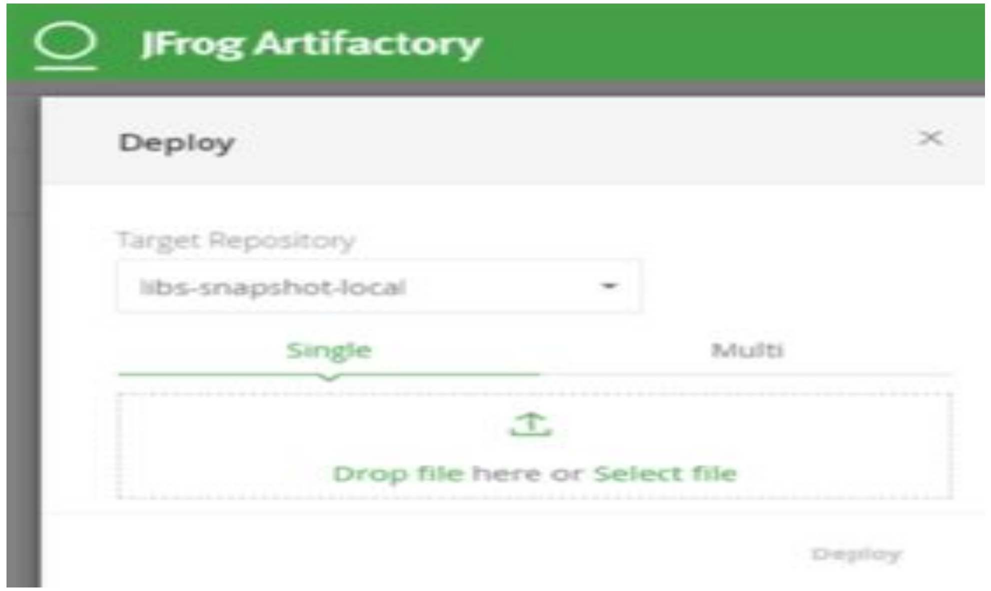
Artifactory Configuration

Artifactory server: http://localhost:8081/artifactory

Target releases repository: libs-release-local

Target snapshots repository: libs-release-local

CASE 4 – Upload to Jfrog repo manually



Target Repository: libs-snapshot-local