Anushka Shahane D15A 55

EXPERIMENT 8 - ADVANCE DEVOPS

Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web /

Theory:
What is SAST?
Static application security testing (SAST), or static analysis, is a testing methodology that
analyzes source code to find security vulnerabilities that make your organization's applications
susceptible to attack. SAST scans an application before the code is compiled. It's also known as
white box testing.
What problems does SAST solve?
SAST takes place very early in the software development life cycle (SDLC) as it does not
require a working application and can take place without code being executed. It helps
developers identify vulnerabilities in the initial stages of development and quickly resolve issues
without breaking builds or passing on vulnerabilities to the final release of the application.
SAST tools give developers real-time feedback as they code, helping them fix issues before they
pass the code to the next phase of the SDLC. This prevents security-related issues from being
considered an afterthought. SAST tools also provide graphical representations of the issues
found, from source to sink. These help you navigate the code easier. Some tools pointout the
exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth
guidance on how to fix issues and the best place in the code to fix them, without requiring deep
security domain expertise.
It's important to note that SAST tools must be run on the application on a regular basis, such as
during daily/monthly builds, every time code is checked in, or during a code release.
Why is SAST important?

Developers dramatically outnumber security staff. It can be challenging for anorganization to
find the resources to perform code reviews on even a fraction of its applications. A key strength
of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster
than manual secure code reviews performed by humans. These tools can scan millionsof lines of
code in a matter of minutes. SAST tools automatically identify criticalvulnerabilities—such as
buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.What is a CI/CD Pipeline?

.

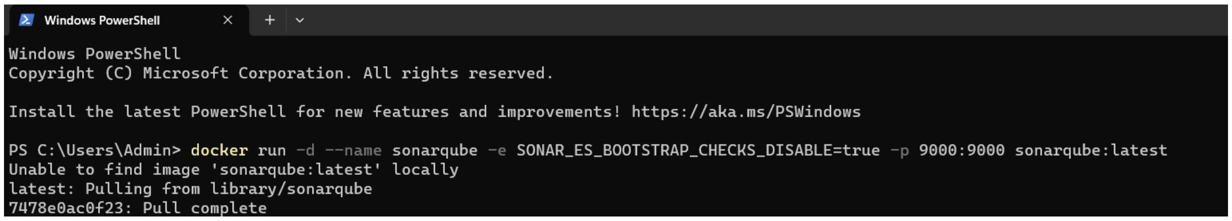
What is SonarQube

SonarQube is an open-source platform developed by SonarSource for continuous inspection of
code quality. Sonar does static code analysis, which provides a detailed report of bugs, code
smells, vulnerabilities, code duplications.

It supports 25+ major programming languages through built-in rulesets and can also be extended
with various plugins.

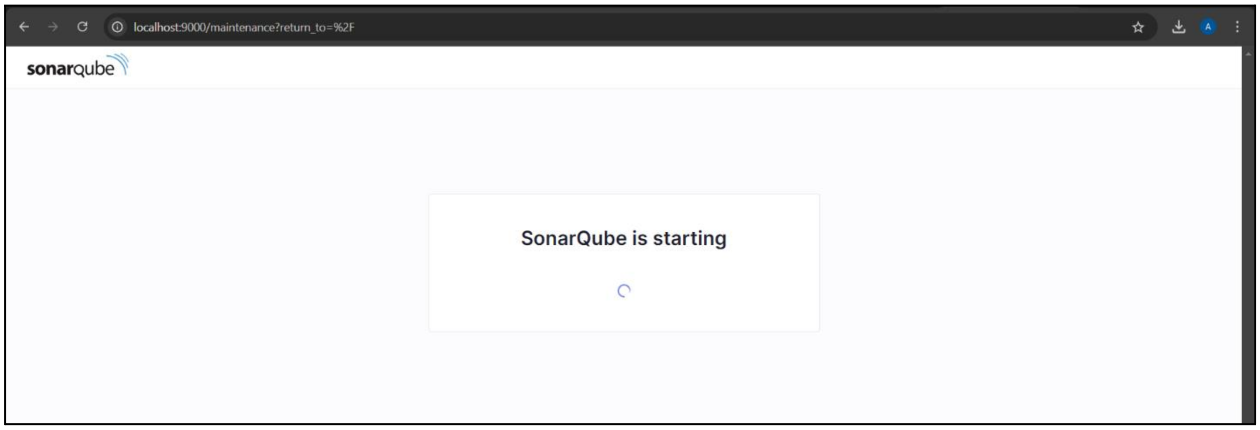
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -

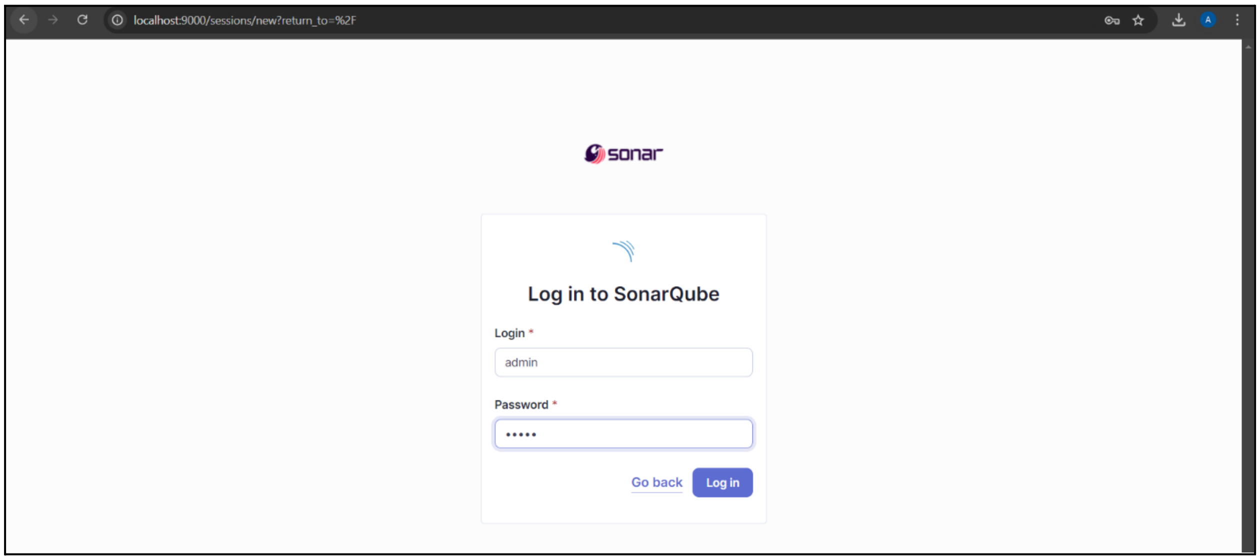


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
```

3. Once the container is up and running, you can check the status of SonarQube at localhost
port 9000.

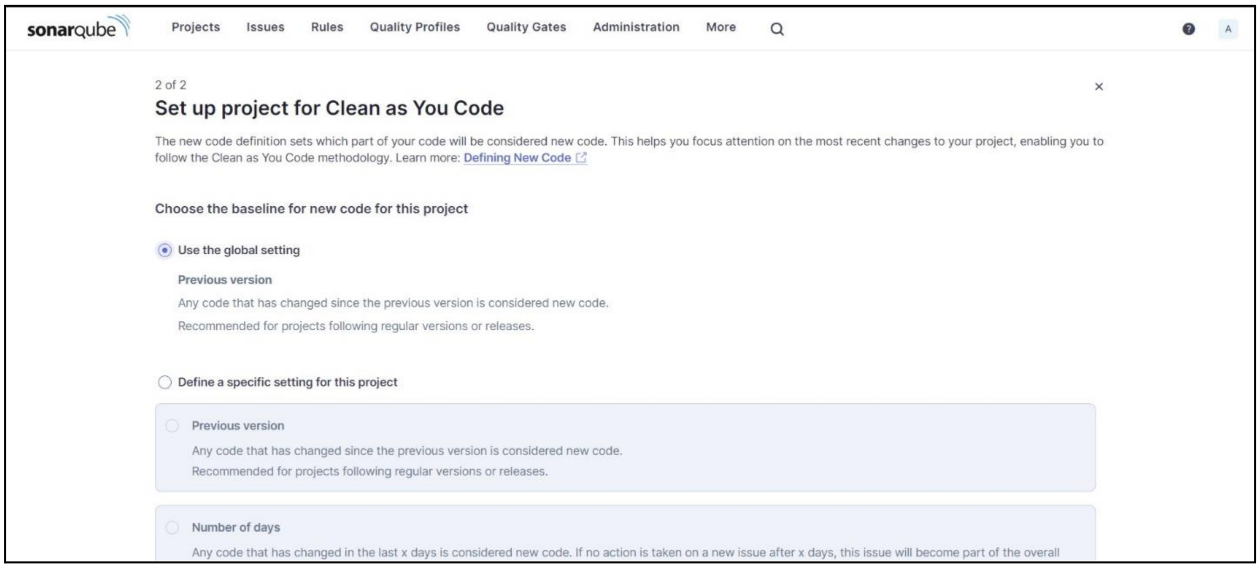


4. Login to SonarQube using username admin and password admin.



5. Create a manual project in SonarQube with the name sonarqube-test

Anushka Shahane D15A 55



2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: Defining New Code

Choose the baseline for new code for this project

[✓ Use the global setting]

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

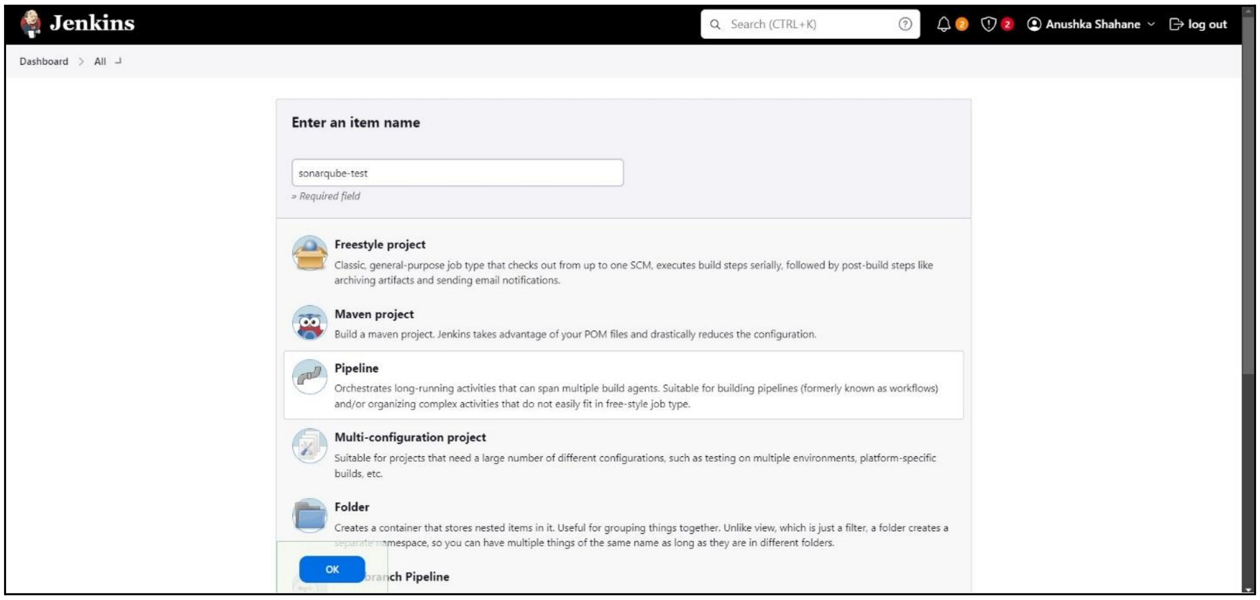
Define a specific setting for this project

Previous version

Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall



Enter an item name

sonarqube-test

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

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

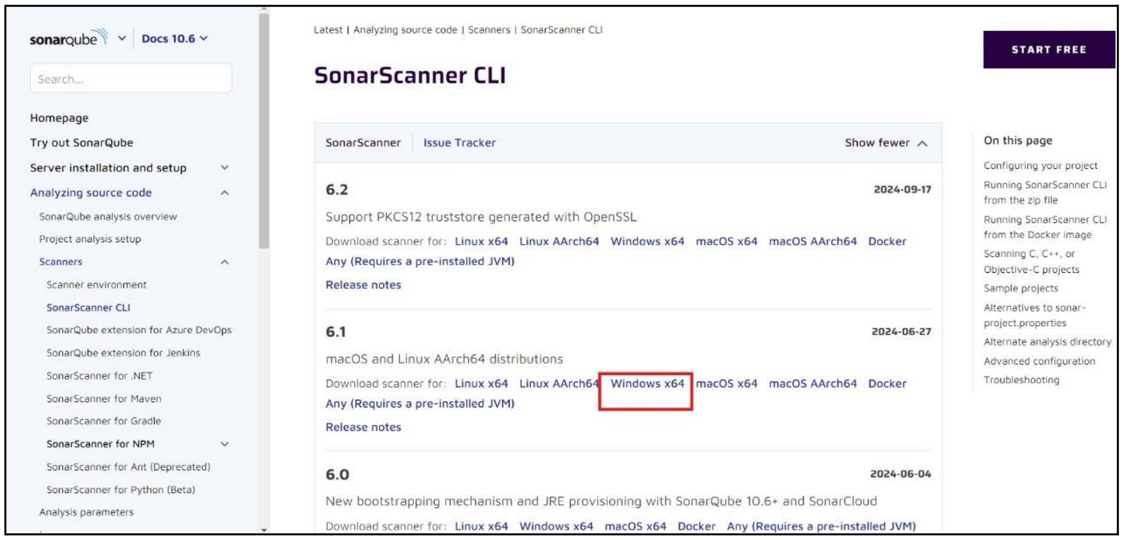
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
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.

ch Pipeline

Anushka Shahane D15A 55

Step 6 : Go to download sonarscanner to download sonar scanner


sonarqube
Docs 10.6
Search...
Homepage
Try out SonarQube
Server installation and setup
Analyzing source code
SonarQube analysis overview
Project analysis setup
Scanners
Scanner environment
SonarScanner CLI
SonarQube extension for Azure DevOps
SonarQube extension for Jenkins
SonarScanner for .NET
SonarScanner for Maven
SonarScanner for Gradle
SonarScanner for NPM
SonarScanner for Ant (Deprecated)
SonarScanner for Python (Beta)
Analysis parameters
Latest | Analyzing source code | Scanners | SonarScanner CLI
START FREE
On this page
Configuring your project
Running SonarScanner CLI from the zip file
Running SonarScanner CLI from the Docker image
Scanning C, C++, or Objective-C projects
Sample projects
Alternatives to sonar-project.properties
Alternate analysis directory
Advanced configuration
Troubleshooting

SonarScanner CLI

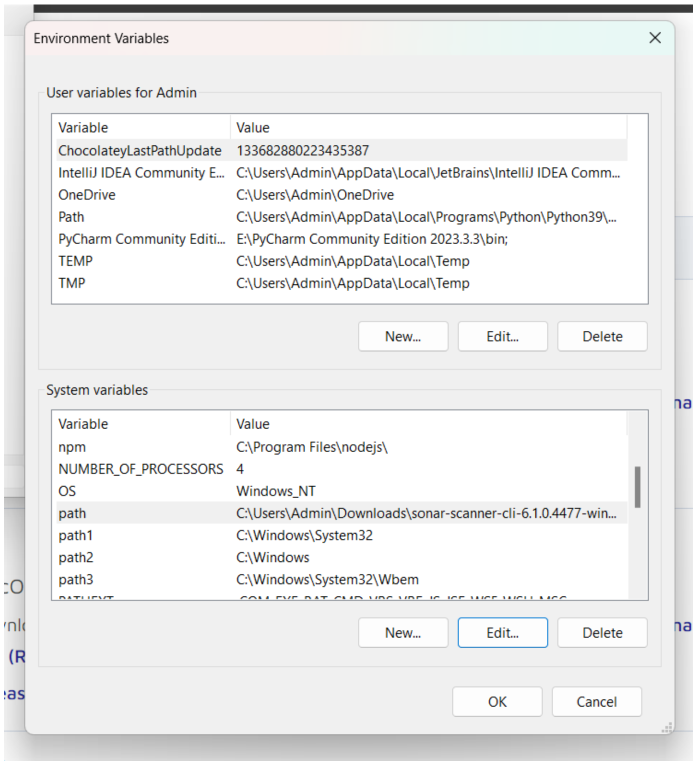
SonarScanner | Issue Tracker | Show fewer

6.2 — 2024-09-17
Support PKCS12 truststore generated with OpenSSL
Download scanner for: Linux x64 Linux AArch64 Windows x64 macOS x64 macOS AArch64 Docker Any (Requires a pre-installed JVM)
Release notes

6.1 — 2024-06-27
macOS and Linux AArch64 distributions
Download scanner for: Linux x64 Linux AArch64 Windows x64 macOS x64 macOS AArch64 Docker Any (Requires a pre-installed JVM)
Release notes

6.0 — 2024-06-04
New bootstrapping mechanism and JRE provisioning with SonarQube 10.6+ and SonarCloud
Download scanner for: Linux x64 Windows x64 macOS x64 Docker Any (Requires a pre-installed JVM)

Step 7: After the download is complete, extract the file and copy the path to bin folder
Go to environment variables, system variables and click on path
Add a new path, paste the path copied earlier.



User variables for Admin

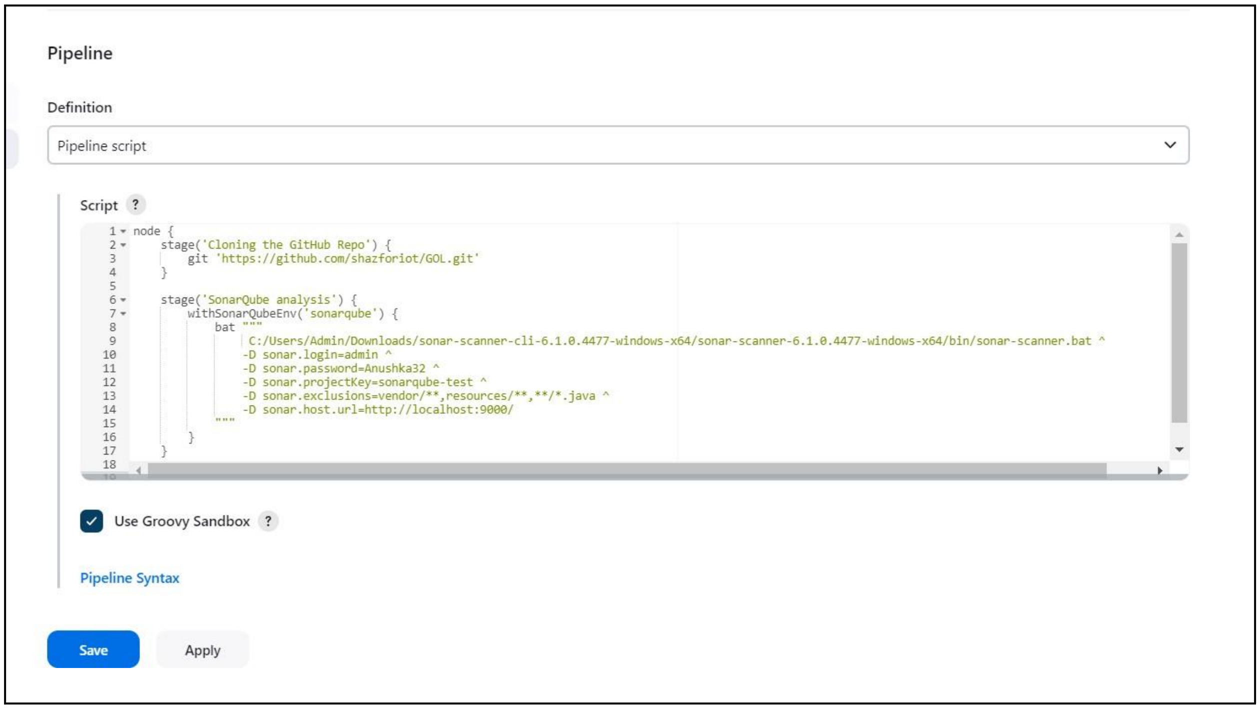
Variable	Value
ChocolateyLastPathUpdate	133682880223435387
IntelliJ IDEA Community E...	C:\Users\Admin\AppData\Local\JetBrains\IntelliJ IDEA Comm...
OneDrive	C:\Users\Admin\OneDrive
Path	C:\Users\Admin\AppData\Local\Programs\Python\Python39\...
PyCharm Community Editi...	E:\PyCharm Community Edition 2023.3.3\bin;
TEMP	C:\Users\Admin\AppData\Local\Temp
TMP	C:\Users\Admin\AppData\Local\Temp

System variables

Variable	Value
npm	C:\Program Files\nodejs\
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
path	C:\Users\Admin\Downloads\sonar-scanner-cli-6.1.0.4477-win...
path1	C:\Windows\System32
path2	C:\Windows
path3	C:\Windows\System32\Wbem

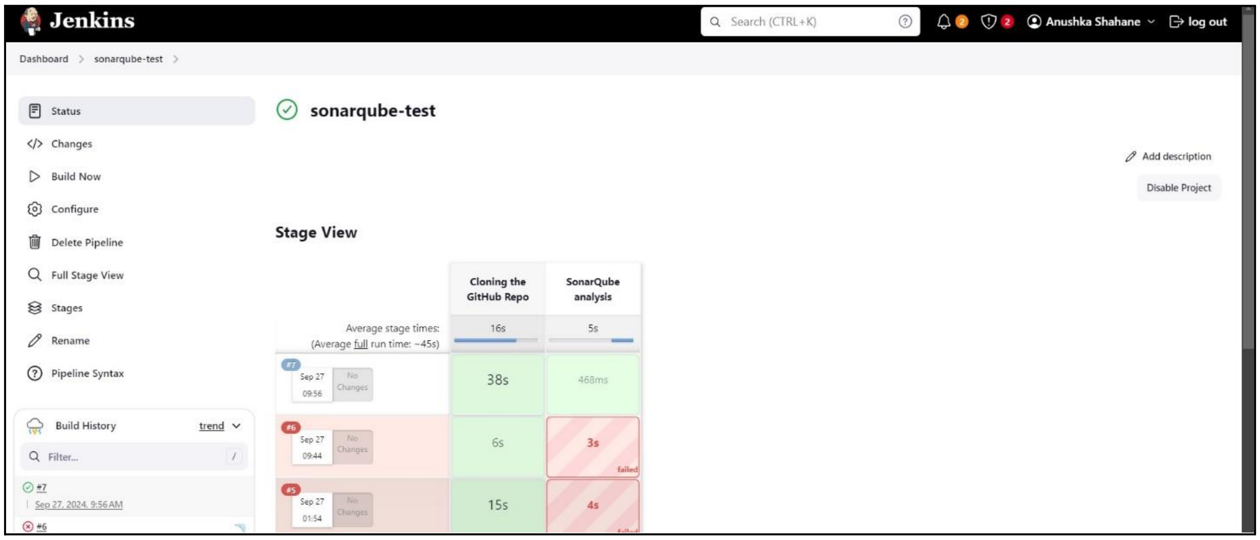
Anushka Shahane D15A 55

Step 8 : Save the pipeline and build it.



```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }

    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
             C:/Users/Admin/Downloads/sonar-scanner-cli-6.1.0.4477-windows-x64/sonar-scanner-6.1.0.4477-windows-x64/bin/sonar-scanner.bat ^
            -D sonar.login=admin ^
            -D sonar.password=Anushka32 ^
            -D sonar.projectKey=sonarqube-test ^
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
            -D sonar.host.url=http://localhost:9000/
            """
        }
    }
```

Output :



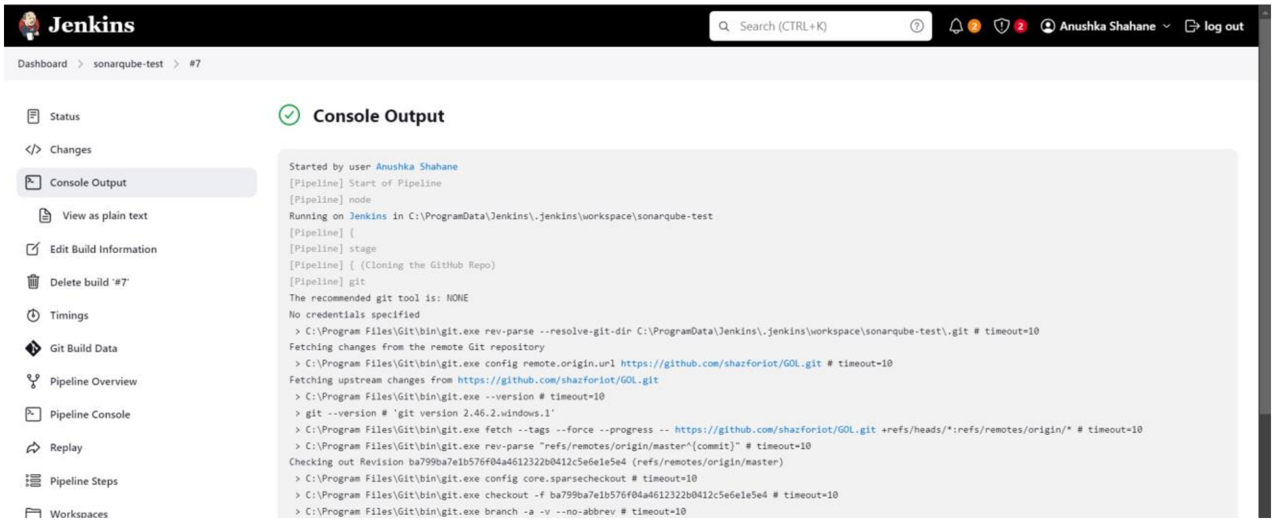
sonarqube-test

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~45s)	16s	5s
#7 Sep 27 09:56 No Changes	38s	468ms
#6 Sep 27 09:44 No Changes	6s	3s failed
#5 Sep 27 01:54 No Changes	15s	4s failed

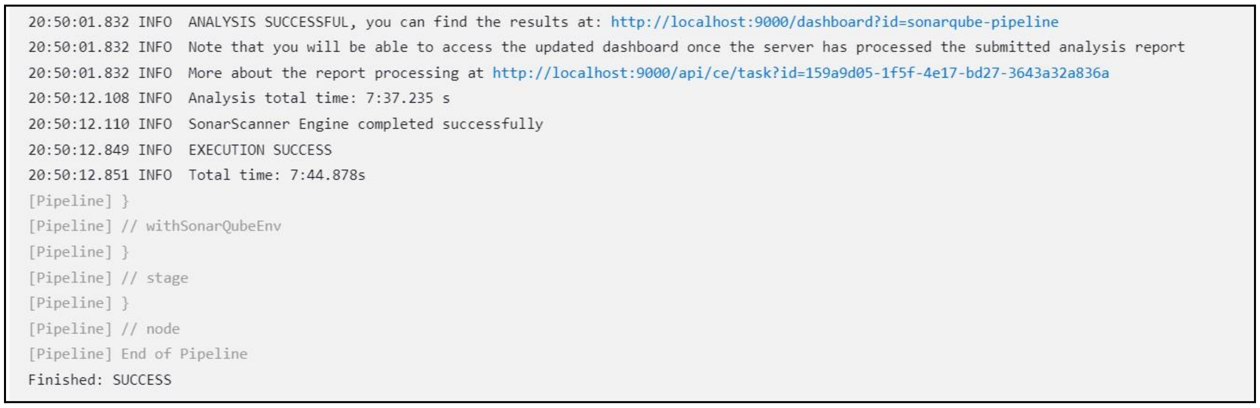
#7 Sep 27, 2024, 9:56 AM
#6

Anushka Shahane D15A 55



Console Output

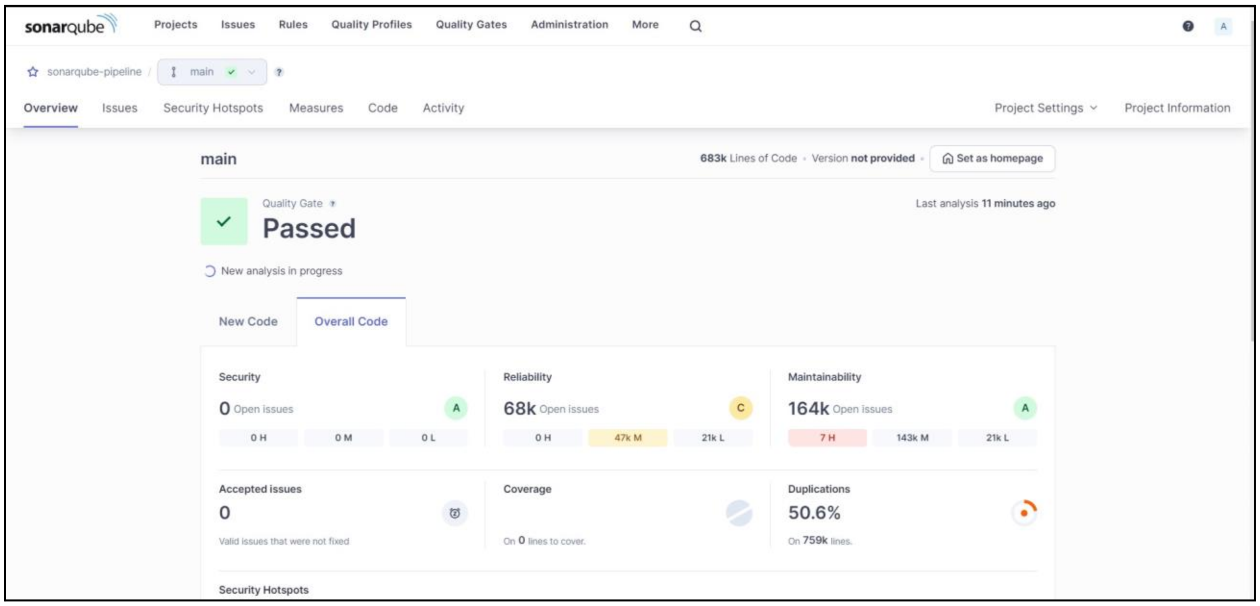
```
Started by user Anushka Shahane
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test\.git # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > git --version # 'git version 2.46.2.windows.1'
 > C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
 > C:\Program Files\Git\bin\git.exe branch -a -v --no-abbrev # timeout=10
```



```
20:50:01.832 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-pipeline
20:50:01.832 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:50:01.832 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=159a9d05-1f5f-4e17-bd27-3643a32a836a
20:50:12.108 INFO  Analysis total time: 7:37.235 s
20:50:12.110 INFO  SonarScanner Engine completed successfully
20:50:12.849 INFO  EXECUTION SUCCESS
20:50:12.851 INFO  Total time: 7:44.878s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Anushka Shahane D15A 55

Step 9 : Check the project in SonarQube



main

683k Lines of Code • Version not provided

Quality Gate
Passed

Last analysis 11 minutes ago

New analysis in progress

Security
0 Open issues A
0 H 0 M 0 L

Reliability
68k Open issues C
0 H 47k M 21k L

Maintainability
164k Open issues A
7 H 143k M 21k L

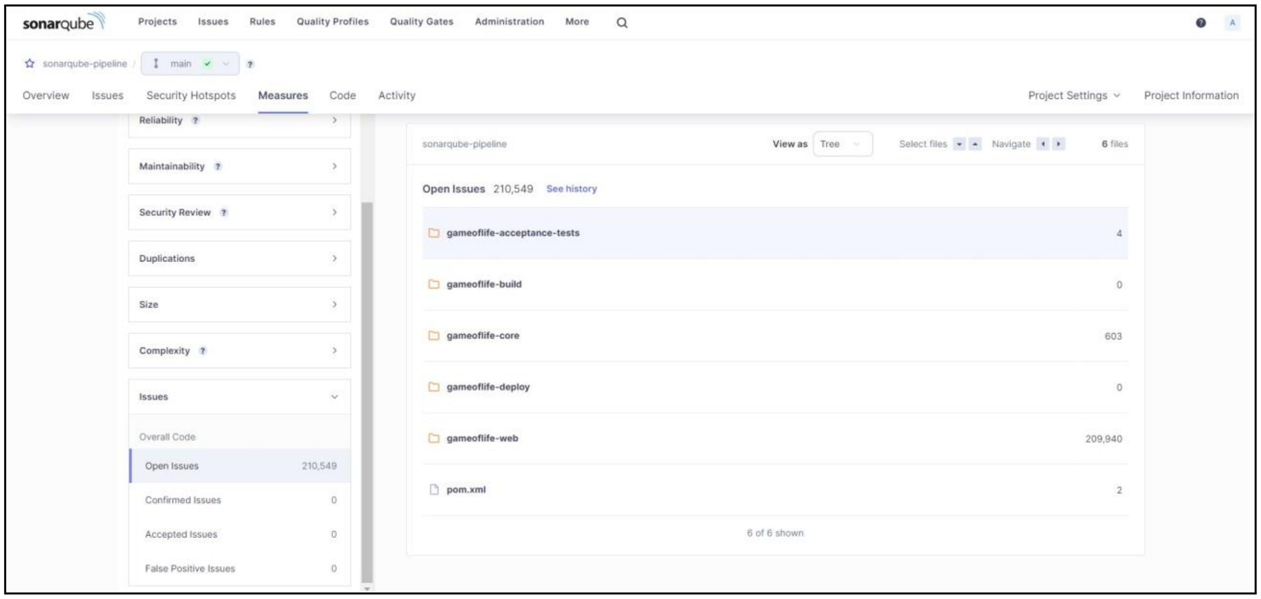
Accepted issues
0
Valid issues that were not fixed

Coverage
On 0 lines to cover.

Duplications
50.6%
On 759k lines.

Security Hotspots

Under different tabs, check all different issues with the code.



Overall Code	
Open Issues	210,549
Confirmed Issues	0
Accepted Issues	0
False Positive Issues	0

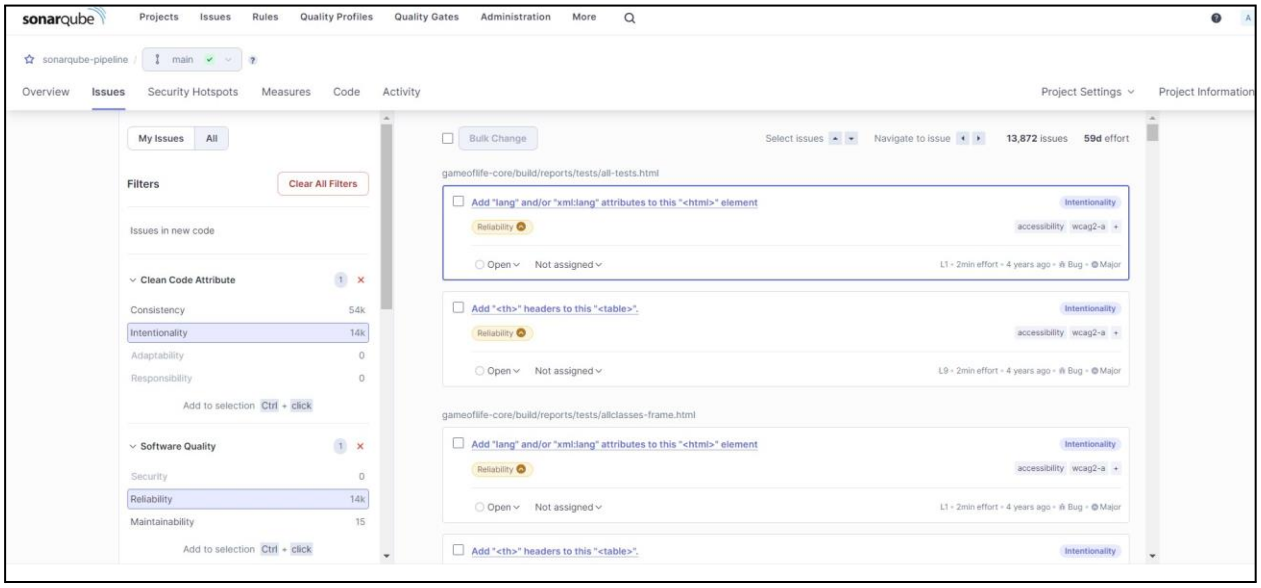
sonarqube-pipeline — 6 files

Open Issues 210,549

gameoflife-acceptance-tests	4
gameoflife-build	0
gameoflife-core	603
gameoflife-deploy	0
gameoflife-web	209,940
pom.xml	2

6 of 6 shown

Anushka Shahane D15A 55



13,872 issues 59d effort

Clean Code Attribute 1

Consistency	54k
Intentionality	14k
Adaptability	0
Responsibility	0

Software Quality 1

Security	0
Reliability	14k
Maintainability	15

gameoflife-core/build/reports/tests/all-tests.html

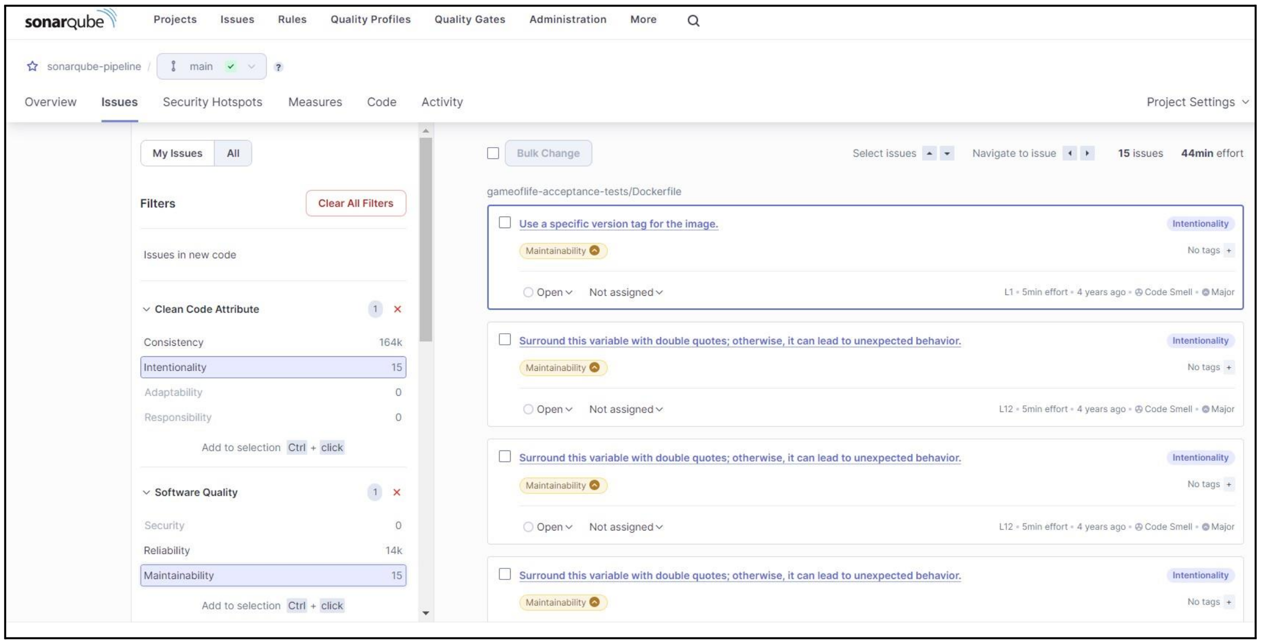
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L1 · 2min effort · 4 years ago · Bug · Major

Add "<th>" headers to this "<table>".
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L9 · 2min effort · 4 years ago · Bug · Major

gameoflife-core/build/reports/tests/allclasses-frame.html

Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L1 · 2min effort · 4 years ago · Bug · Major

Add "<th>" headers to this "<table>".
Intentionality



15 issues 44min effort

Clean Code Attribute 1

Consistency	164k
Intentionality	15
Adaptability	0
Responsibility	0

Software Quality 1

Security	0
Reliability	14k
Maintainability	15

gameoflife-acceptance-tests/Dockerfile

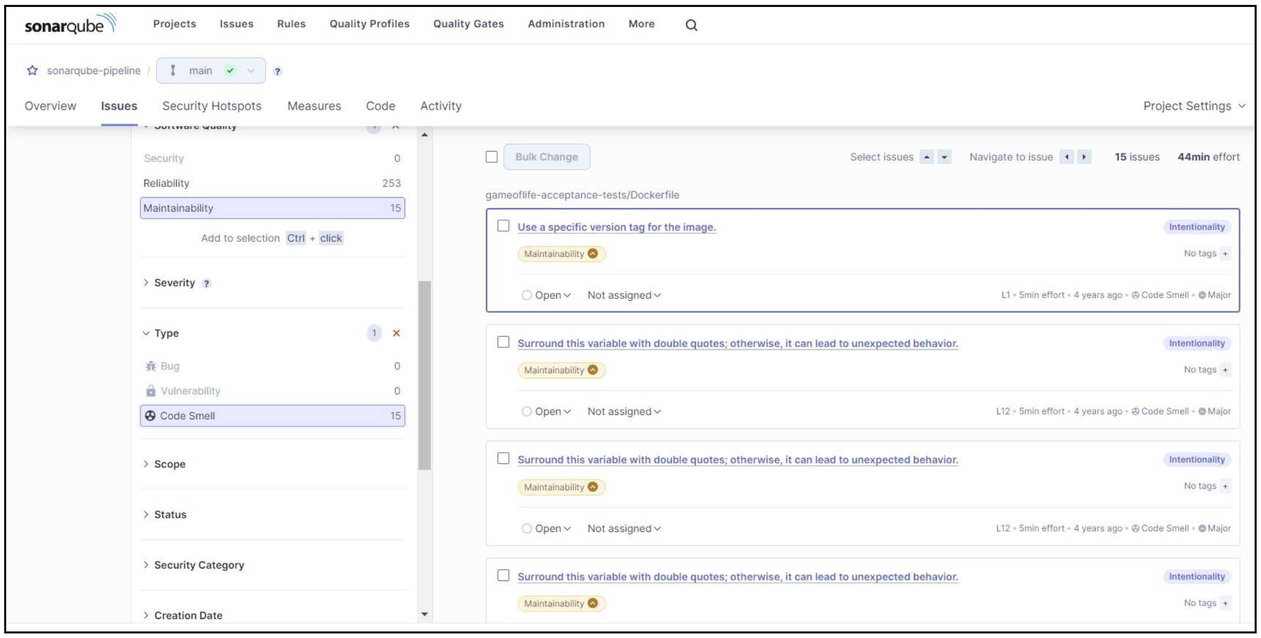
Use a specific version tag for the image.
Intentionality
Maintainability
No tags
Open
Not assigned
L1 · 5min effort · 4 years ago · Code Smell · Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 · 5min effort · 4 years ago · Code Smell · Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 · 5min effort · 4 years ago · Code Smell · Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags

Anushka Shahane D15A 55


sonarqube-pipeline / main

Software Quality

Security	0
Reliability	253
Maintainability	15

Add to selection Ctrl + click

Severity

Type

Bug	0
Vulnerability	0
Code Smell	15

Scope

Status

Security Category

Creation Date

Select issues · Navigate to issue · 15 issues · 44min effort

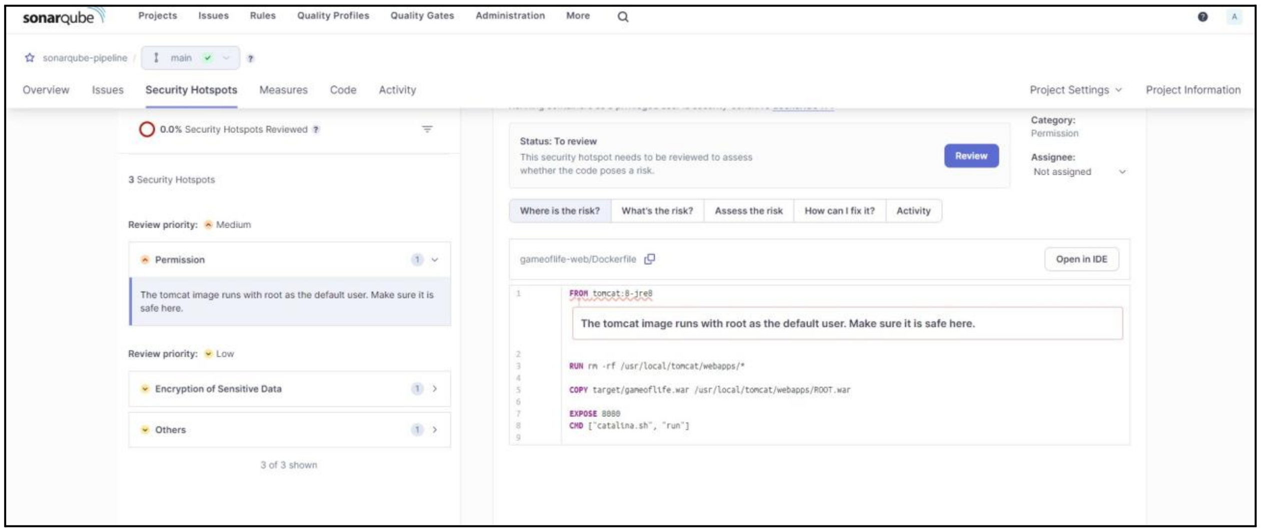
gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image.
Intentionality · Maintainability · No tags
Open · Not assigned · L1 • 5min effort • 4 years ago • Code Smell • Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality · Maintainability · No tags
Open · Not assigned · L12 • 5min effort • 4 years ago • Code Smell • Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality · Maintainability · No tags
Open · Not assigned · L12 • 5min effort • 4 years ago • Code Smell • Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality · Maintainability · No tags


sonarqube-pipeline / main

0.0% Security Hotspots Reviewed

3 Security Hotspots

Review priority: Medium

Permission 1

The tomcat image runs with root as the default user. Make sure it is safe here.

Review priority: Low

Encryption of Sensitive Data 1

Others 1

3 of 3 shown

Status: To review
This security hotspot needs to be reviewed to assess whether the code poses a risk.

Category: Permission
Assignee: Not assigned

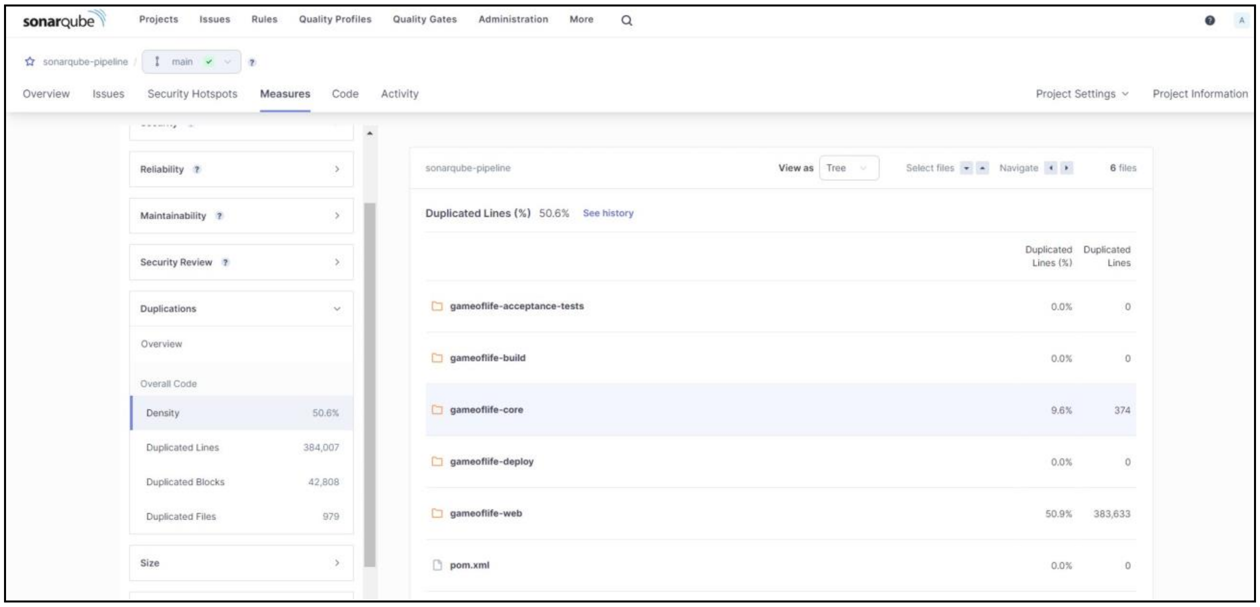
gameoflife-web/Dockerfile

```
FROM tomcat:8-jre8
```

The tomcat image runs with root as the default user. Make sure it is safe here.

```

RUN rm -rf /usr/local/tomcat/webapps/*

COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war

EXPOSE 8080
CMD ["catalina.sh", "run"]
```

Anushka Shahane D15A 55

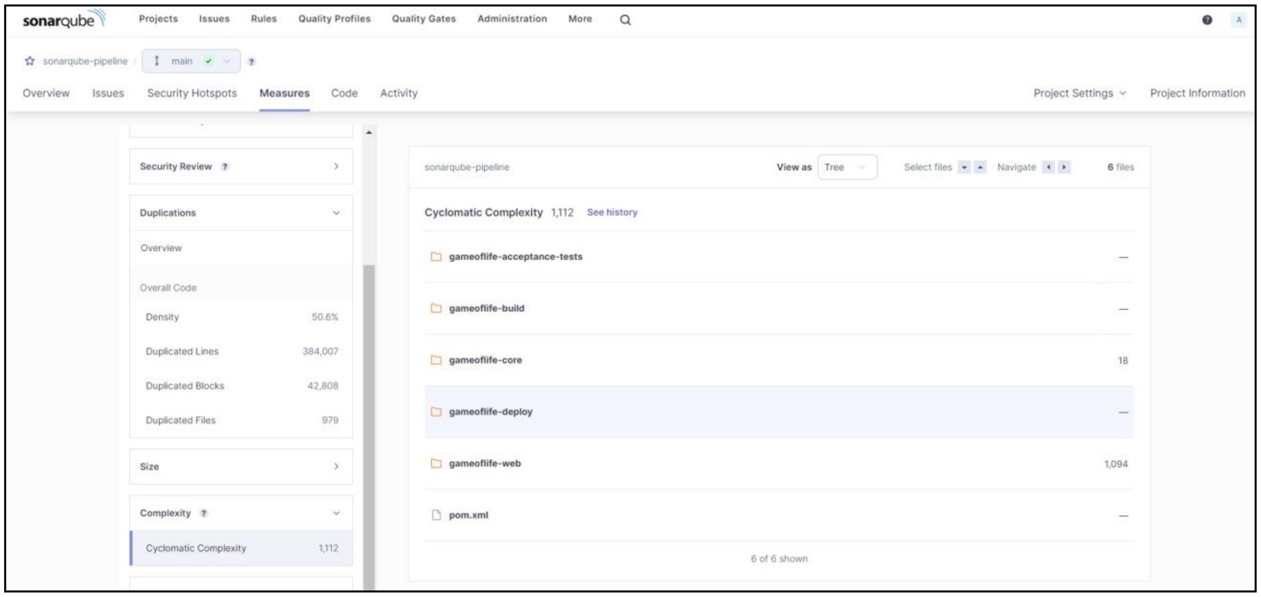

sonarqube
Projects Issues Rules Quality Profiles Quality Gates Administration More
sonarqube-pipeline / main
Overview Issues Security Hotspots Measures Code Activity
Project Settings Project Information
Reliability
Maintainability
Security Review
Duplications
Overview
Overall Code
Size
View as Tree
Select files Navigate 6 files

Overall Code	
Density	50.6%
Duplicated Lines	384,007
Duplicated Blocks	42,808
Duplicated Files	979

sonarqube-pipeline

Duplicated Lines (%) 50.6% See history

	Duplicated Lines (%)	Duplicated Lines
gameoflife-acceptance-tests	0.0%	0
gameoflife-build	0.0%	0
gameoflife-core	9.6%	374
gameoflife-deploy	0.0%	0
gameoflife-web	50.9%	383,633
pom.xml	0.0%	0


sonarqube
Projects Issues Rules Quality Profiles Quality Gates Administration More
sonarqube-pipeline / main
Overview Issues Security Hotspots Measures Code Activity
Project Settings Project Information
Security Review
Duplications
Overview
Overall Code
Size
Complexity
View as Tree
Select files Navigate 6 files

Overall Code	
Density	50.6%
Duplicated Lines	384,007
Duplicated Blocks	42,808
Duplicated Files	979
Cyclomatic Complexity	1,112

sonarqube-pipeline

Cyclomatic Complexity 1,112 See history

gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

6 of 6 shown