

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 / Java / Python application.

Prerequisites:

1. Download Sonar Scanner:

Access the SonarQube documentation and download the SonarQube scanner CLI from this link:

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>

The screenshot shows the SonarScanner CLI documentation page. The left sidebar contains a navigation menu with links like 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code', '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', and 'Language'. The main content area is titled 'SonarScanner CLI' and shows version '6.2' with a release date of '2024-09-17'. It mentions 'Support PKCS12 truststore generated with OpenSSL' and provides download links for various operating systems: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, and Docker. A note states 'Any (Requires a pre-installed JVM)'. Below this, there is a section 'Configuring your project' with a tip: 'The SonarScanners run on code that is checked out. See Verifying the code checkout step of your build.' On the right, there is a 'START FREE' button and a list of links under 'On this page' including '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', and 'Troubleshooting'.

2. After downloading, extract the zip file into a designated folder.

Install Docker:

Run the following command to verify Docker is installed:

```
C:\Users\91799>docker -v
Docker version 27.0.3, build 7d4bcd8
C:\Users\91799>
```

3 .Pull SonarQube Docker Image:

Install the SonarQube image by executing:

Copy code

```
docker pull sonarqube
```

```
PS C:\Users\91799> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

4. Ensure Jenkins is installed:

Confirm that Jenkins is installed and configured on your system.

Experiment Steps:**Step 1:**

Run the SonarQube Docker container by entering the command below:

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

```
PS C:\Users\91799> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
17782929ab2dbb01ea35d5bc93ed52fcffdce44ee8595a07b73a9269e0d39106
```

Step 2:

After SonarQube is running, open your browser and go to <http://localhost:9000>.

Step 3:

Log in to SonarQube using the default credentials:


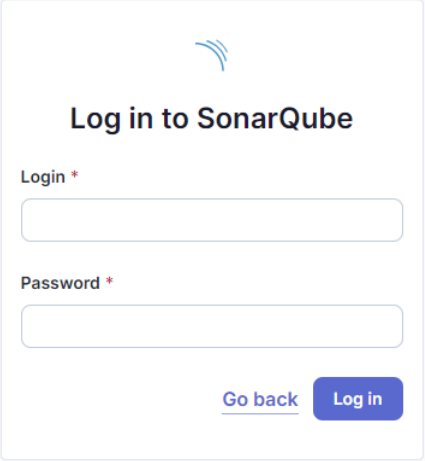
```
Username: admin Password: admin
```

Name:Pratik Patil

Div:D15C

Roll No:40

You will be asked to reset the password after logging in for the first time. Set a new password and remember it.


Log in to SonarQube

Login *

Password *

[Go back](#)

Update your password

 This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Step 4:

On the SonarQube dashboard, click **Create a Local Project**. Provide a project name and a unique project key.

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMoreQ

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

Import from Azure DevOpsSetup

Import from Bitbucket CloudSetup

Import from Bitbucket ServerSetup

Import from GitHubSetup

Import from GitLabSetup

Are you just testing or have an advanced use-case? Create a local project.

Create a local project

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

1 of 2

Create a local project

Project display name *

sonarqube-test

Project key *

sonarqube-test-

Main branch name *

main

The name of your project's default branch [Learn More](#)

Cancel

Next



Your project has been created. ✕

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 code.
Recommended for projects following continuous delivery.

☐ Reference branch

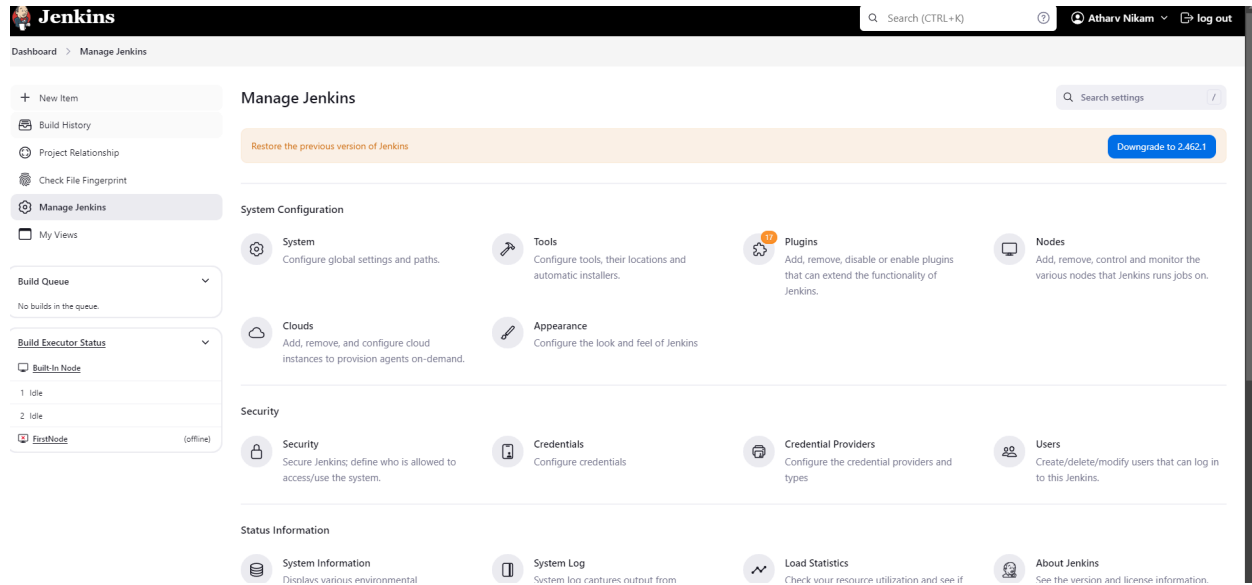
Choose a branch as the baseline for the new code.
Recommended for projects using feature branches.

Back

Create project

Step 6:

In Jenkins, go to **Manage Jenkins** → **Plugins** and search for **SonarQube Scanner for Jenkins**. Install the plugin.

**Step 7:**

Once installed, head to **Manage Jenkins** → **System**. Under **SonarQube Servers**, add your SonarQube server, and provide any necessary authentication tokens.




Step 8:

Next, under **Manage Jenkins** → **Tools**, navigate to **SonarQube Scanner** and configure it to automatically install the latest version.


Dashboard > Manage Jenkins > Tools

Add SonarScanner for MSBuild

SonarQube Scanner installations


SonarQube Scanner installations ^  Edited


Add SonarQube Scanner

≡ **SonarQube Scanner** 

Name


sonarqube-test


☒ Install automatically 

≡ **Install from Maven Central** 

Version

SonarQube Scanner 6.2.0.4584



Add Installer 

Add SonarQube Scanner

Step 9:

Create a new pipeline item in Jenkins

New Item

Enter an item name

sonarqube-test-

Select an item type

**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.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Step 10:

In the pipeline script section, input the following:

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
  
    stage('SonarQube Analysis') {  
        withSonarQubeEnv('sonarqube-test') {  
            bat """  
            <PATH_TO_SONARSCANNER_FOLDER>\\bin\\sonar-scanner.bat ^  
            -D sonar.login=<SONARQUBE_LOGIN> ^  
            -D sonar.password=<SONARQUBE_PASSWORD> ^  
            -D sonar.projectKey=<PROJECT_KEY> ^  
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^  
            -D sonar.host.url=http://localhost:9000/  
            """  
        }  
    }  
}
```


Pipeline

Definition

Pipeline script

Script ?

```
1 node {  
2   stage('Cloning the GitHub Repo') {  
3     git 'https://github.com/shazforiot/GOL.git'  
4   }  
5  
6   stage('SonarQube Analysis') {  
7     withSonarQubeEnv('sonarqube') {  
8       bat """  
9         C:\\Users\\91799\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat  
10        -D sonar.login=admin  
11        -D sonar.password=pratik  
12        -D sonar.projectKey=sonarqube-test  
13        -D sonar.exclusions=vendor/**,resources/**,**/*.java  
14        -D sonar.host.url=http://localhost:9000/  
15        """  
16     }  
17   }  
18 }
```

☒ Use Groovy Sandbox ?[Pipeline Syntax](#)

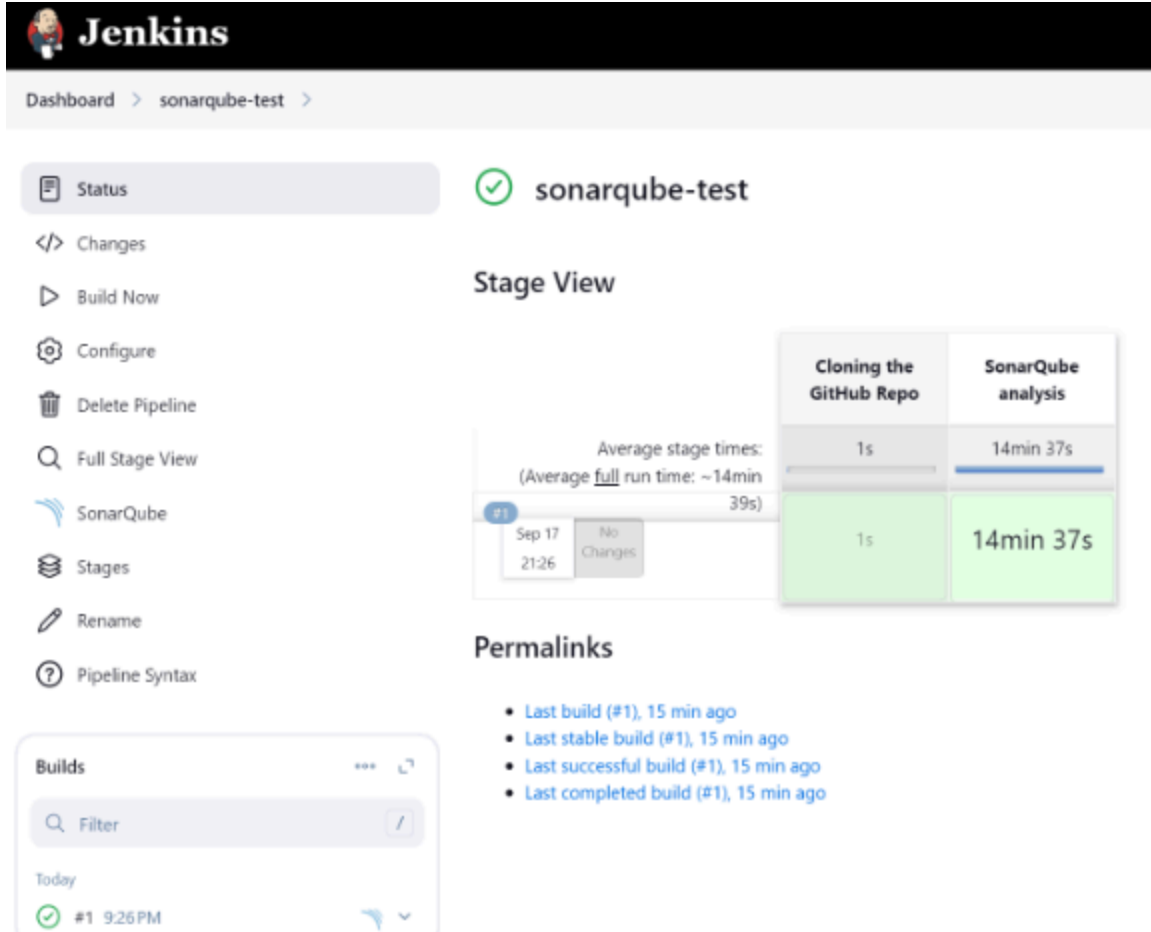
Save

Apply

This script clones a sample Java project from GitHub, which has several issues that SonarQube will detect.

Step 11:

Go back to Jenkins, select the job you just created, and click **Build Now** to run the pipeline.



The Jenkins dashboard for the pipeline 'sonarqube-test' shows a successful status with a green checkmark. The left sidebar contains navigation options: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. The main area displays the 'Stage View' for build #1, which completed on Sep 17 at 21:26 with no changes. A table shows stage durations: 'Cloning the GitHub Repo' (1s) and 'SonarQube analysis' (14min 37s). The 'Permalinks' section lists: Last build (#1), 15 min ago; Last stable build (#1), 15 min ago; Last successful build (#1), 15 min ago; and Last completed build (#1), 15 min ago. A 'Builds' panel at the bottom left shows a filter and a single build entry: #1 9:26 PM.

Stage	Duration
Cloning the GitHub Repo	1s
SonarQube analysis	14min 37s

```

Started by user Pratik Manish Patil
[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
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10

```

Step 12:

Once the build is complete, return to SonarQube to view the analysis of your project. Check for bugs, code smells, duplications, and other metrics related to the quality of your code.

The screenshot shows the SonarQube Overview page for a project named 'atharvsonarqubetest1'. The main status is 'Passed' with a green checkmark. A warning message states: 'The last analysis has warnings. See details'. The 'Last analysis' was 10 minutes ago. The page is divided into two tabs: 'New Code' and 'Overall Code'. The 'Overall Code' tab is active, displaying various quality metrics:

Metric	Value	Grade
Security	0 Open Issues	A
Reliability	68k Open Issues	C
Maintainability	164k Open Issues	A
Accepted Issues	0	
Coverage	On 0 lines to cover.	
Duplications	50.6% (On 759k lines)	

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

- Code Problems

The screenshot shows the SonarQube Measures page for the same project. The 'Measures' tab is active. On the left, there is a sidebar with a list of metrics: Size, Complexity, Issues, New Code, Open Issues, Accepted Issues, Overall Code, Open Issues, Confirmed Issues, Accepted Issues, and False Positive Issues. The 'Open Issues' metric is selected, showing 0 issues. The main area displays a list of 'New Open Issues' with 0 total. The list includes the following items:

Item	Value
gameoflife-acceptance-tests	0
gameoflife-build	0
gameoflife-core	0
gameoflife-deploy	0
gameoflife-web	0
pom.xml	0

At the bottom, it indicates '6 of 6 shown'.

- Consistency

☆ atharvsonarqubetest1 / ⓘ main ✓ ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings ▾ Project Inform

My Issues All

Filters [Clear All Filters](#)

Issues in new code

▼ Clean Code Attribute 1 ✕

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection [Ctrl + click](#)

▼ Software Quality

Security	0
Reliability	54k

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

☐ Bulk Change Select issues ▾ Navigate to issue ◀ ▶ 196,662 issues 3075d effort

☐ Insert a <IDOCYPE> declaration to before this <html> tag. Consistency user-experience ▾

Reliability ⓘ

Open ▾ Not assigned ▾ L1 • 5min effort • 4 years ago • ⓘ Bug • ⓘ Major

☐ Remove this deprecated "width" attribute. Consistency html5 obsolete ▾

Maintainability ⓘ

Open ▾ Not assigned ▾ L9 • 5min effort • 4 years ago • ⓘ Code Smell • ⓘ Major

☐ Remove this deprecated "align" attribute. Consistency html5 obsolete ▾

Maintainability ⓘ

Open ▾ Not assigned ▾ L11 • 5min effort • 4 years ago • ⓘ Code Smell • ⓘ Major

- Intentionality

☆ atharvsonarqubetest1 / ⓘ main ✓ ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings ▾ Project Info

My Issues All

Filters [Clear All Filters](#)

Issues in new code

▼ Clean Code Attribute 1 ✕

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection [Ctrl + click](#)

▼ Software Quality

Security	0
Reliability	14k

gameoflife-acceptance-tests/Dockerfile

☐ Bulk Change Select issues ▾ Navigate to issue ◀ ▶ 13,887 issues 59d effort

☐ Use a specific version tag for the image. Intentionality No tags ▾

Maintainability ⓘ

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 No tags ▾

Maintainability ⓘ

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 No tags ▾

Maintainability ⓘ

Open ▾ Not assigned ▾ L12 • 5min effort • 4 years ago • ⓘ Code Smell • ⓘ Major

Name:Pratik Patil

Div:D15C

Roll No:40

• Bugs

☆ atharvsonarqubetest1 / main ✓ ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings ▾ Project Informa

Maintainability 0

> Severity ?

▼ Type 1 ✕

- Bug 47k
- Vulnerability 0
- Code Smell 164k

Add to selection Ctrl + click

> Scope

> Status

> Security Category

> Creation Date

☐ Bulk Change

Select Issues ▾ Navigate to issue ⏪ ⏩ 46,515 issues 1426d effort

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

☐ Insert a <DOCTYPE> declaration to before this <html> tag. Consistency

Reliability Ⓢ

user-experience +

○ Open ▾ Not assigned ▾ L1 • 5min effort • 4 years ago • # Bug • @ Major

☐ 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

Embedded database should be used for evaluation purposes only

• Code Smells

☆ atharvsonarqubetest1 / main ✓ ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings ▾ Project Informa

Maintainability 164k

> Severity ?

▼ Type 1 ✕

- Bug
- Vulnerability 0
- Code Smell 164k

Add to selection Ctrl + click

> Scope

> Status

> Security Category

> Creation Date

☐ Bulk Change

Select Issues ▾ Navigate to issue ⏪ ⏩ 164,034 issues 1708d 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

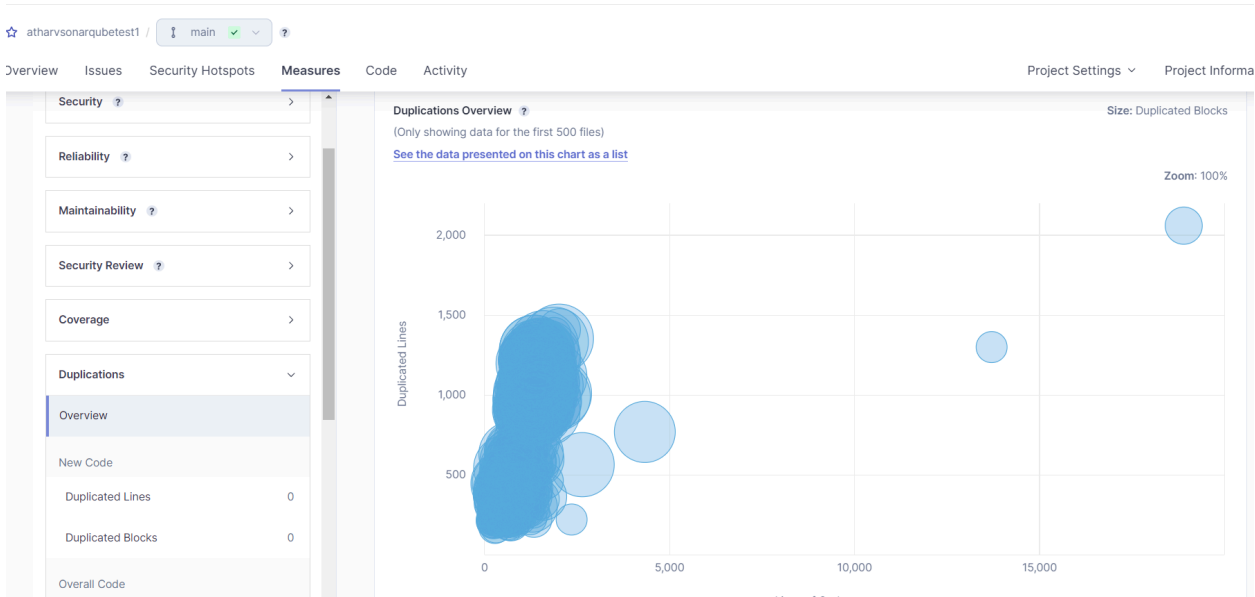
Maintainability Ⓢ

No tags +

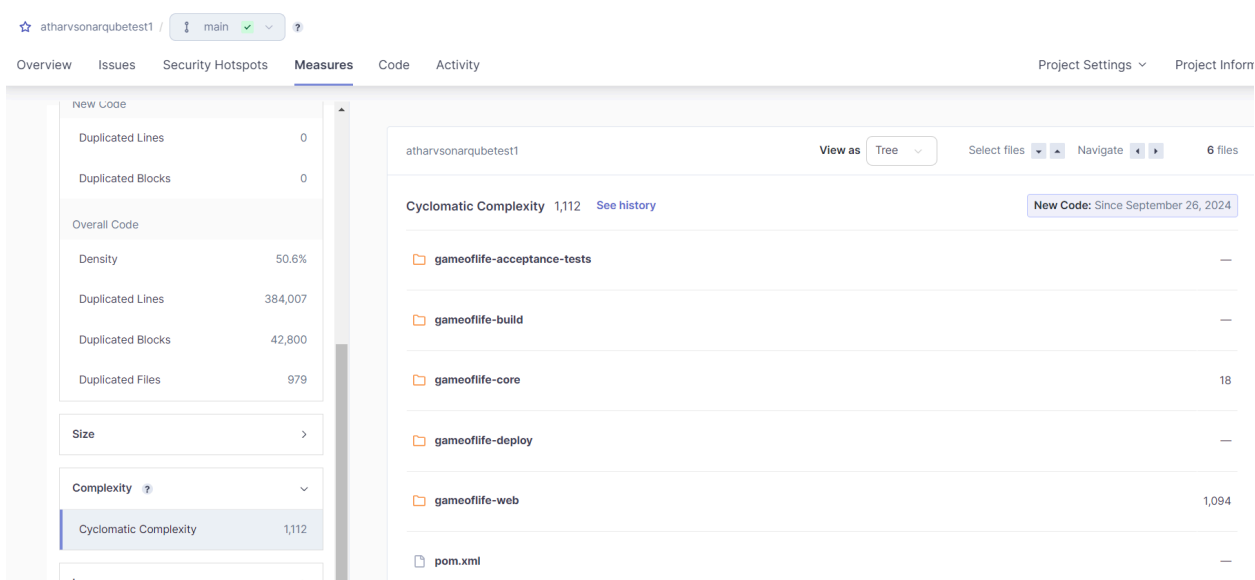
○ Open ▾ Not assigned ▾ L12 • 5min effort • 4 years ago • @ Code Smell • @ Major

Embedded database should be used for evaluation purposes only

• Duplications



• Cyclomatic Complexities



Conclusion:

This experiment allowed us to integrate Jenkins and SonarQube to set up a CI/CD pipeline capable of performing static analysis on Java code. Through this process, we automated the detection of common code issues such as bugs, code smells, and duplications. By leveraging Docker for SonarQube and the Jenkins pipeline, we streamlined the code scanning process, ensuring any issues were highlighted during the build phase. This integration demonstrates the importance of automated code quality checks in a continuous delivery environment