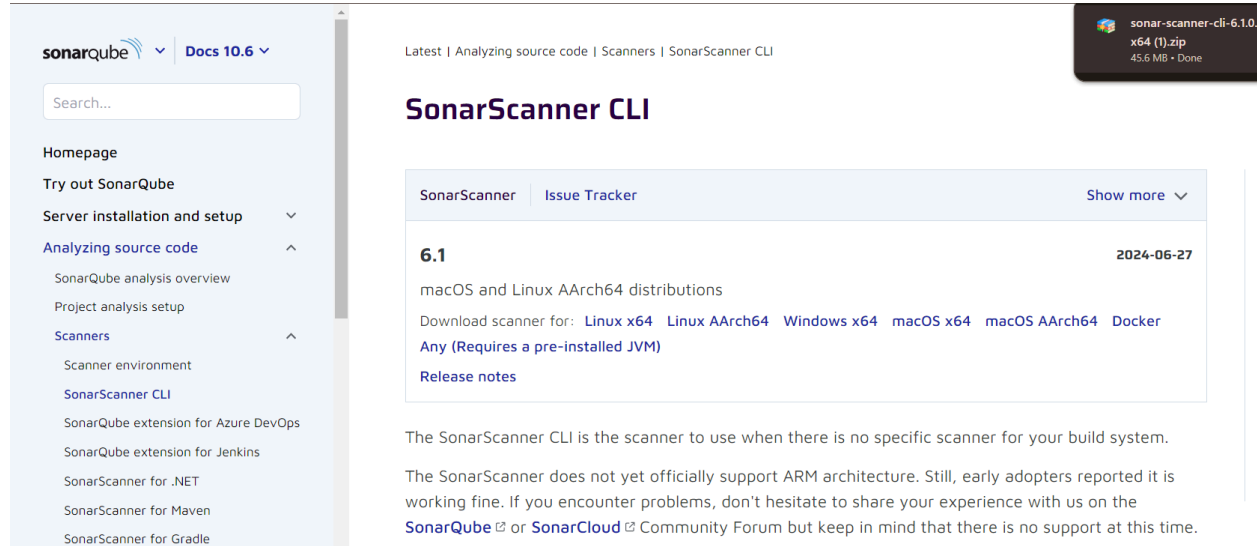


## 08 Advanced DevOps Lab

Aim: Create a Jenkins CI/CD 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.

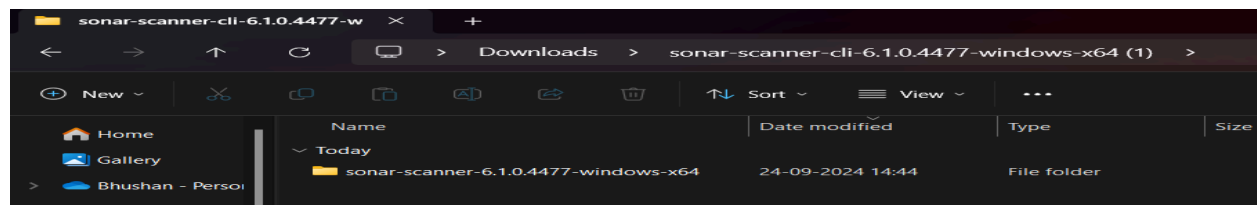
### Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>  
Visit this link and download the sonarqube scanner CLI.



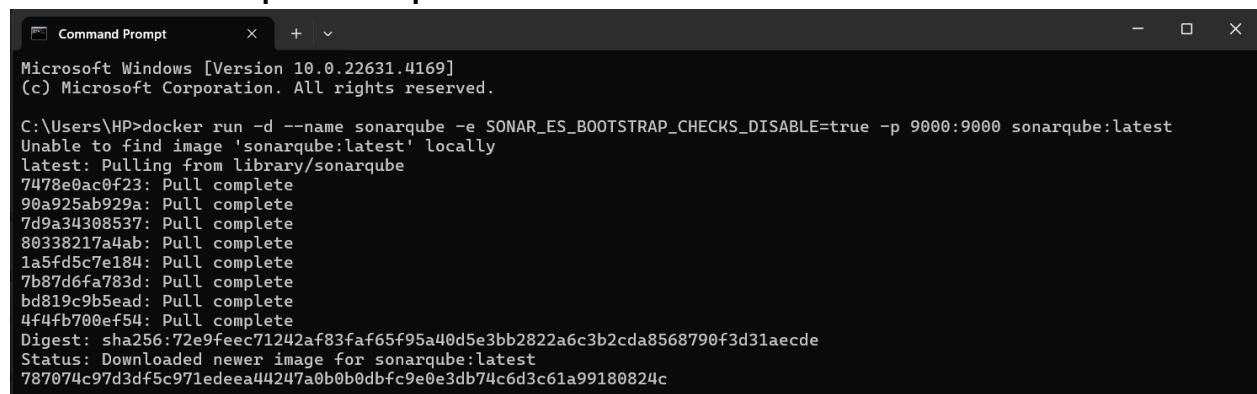
The screenshot shows the SonarScanner CLI download page. The page title is 'SonarScanner CLI'. The page content includes a download button for 'x64 (1).zip' (45.6 MB). The page also includes a sidebar with navigation links and a search bar.

Extract the downloaded zip file in a folder.

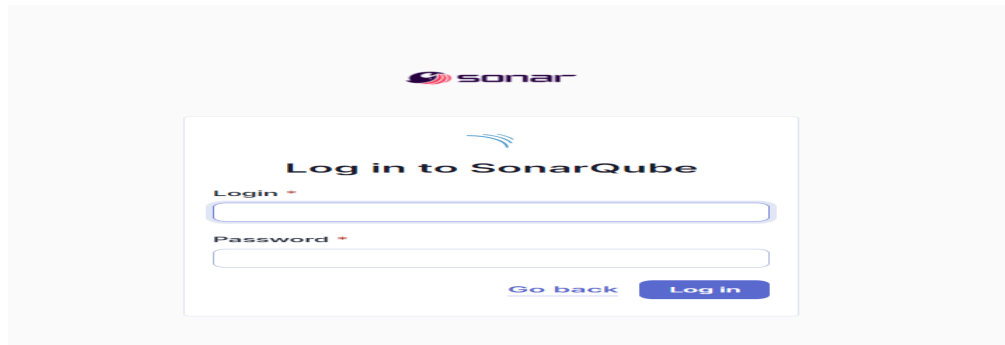


### 1. Install sonarqube image

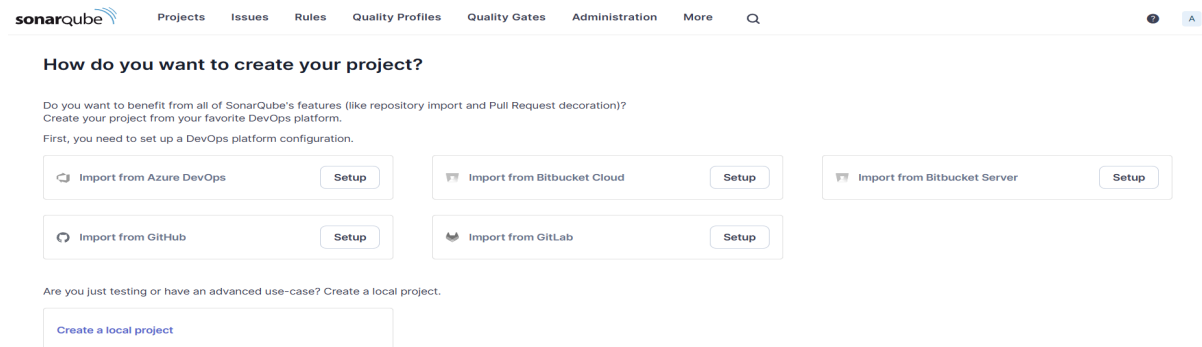
Command: **docker pull sonarqube**



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



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



4. Create a manual project in SonarQube with the name sonarqube

1 of 2

## Create a local project

Project display name \*

sonarqube



Project key \*

sonarqube



Main branch name \*

main

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

Cancel

Next

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 follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

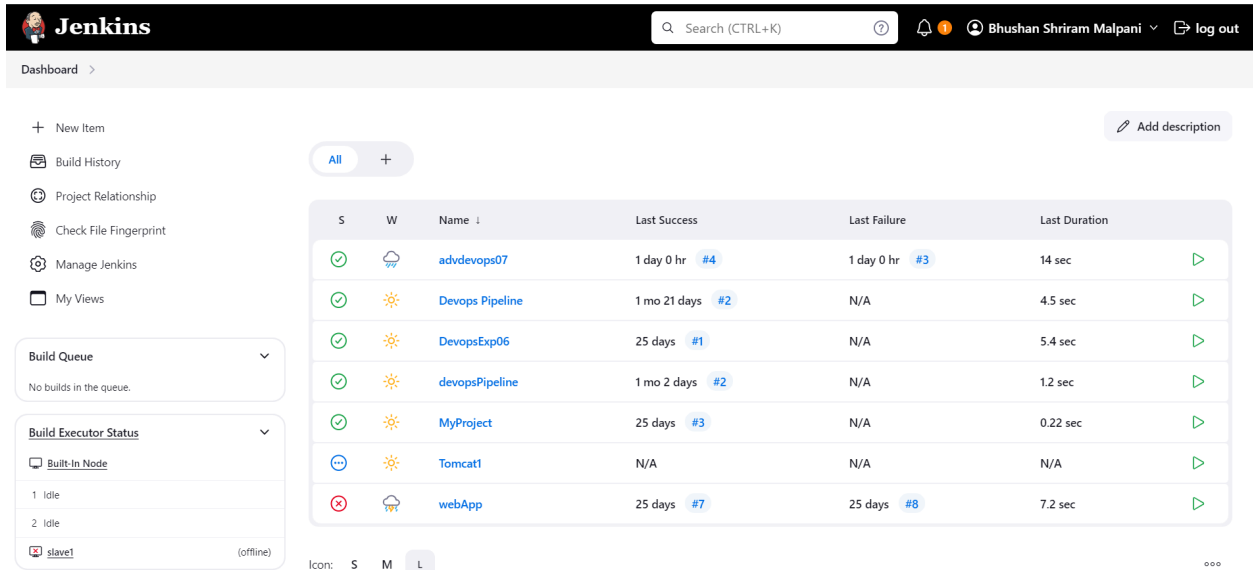
☒ Use the global setting

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

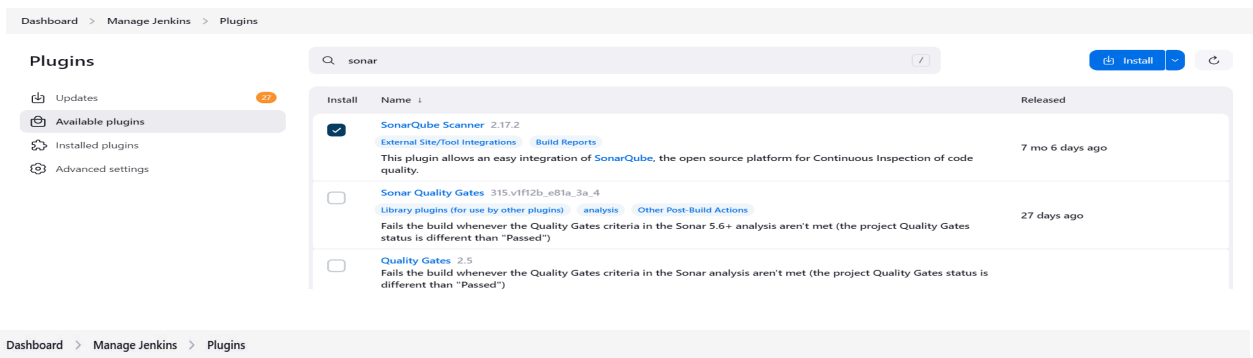
5. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. Below these are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 'Built-In Node' with 1 idle and 2 idle executors, and 'slave1' as offline). The main area displays a table of builds with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table lists several builds, including 'advdevops07', 'Devops Pipeline', 'DevopsExp06', 'devopsPipeline', 'MyProject', 'Tomcat1', and 'webApp'. A search bar at the top right contains 'sonar'.

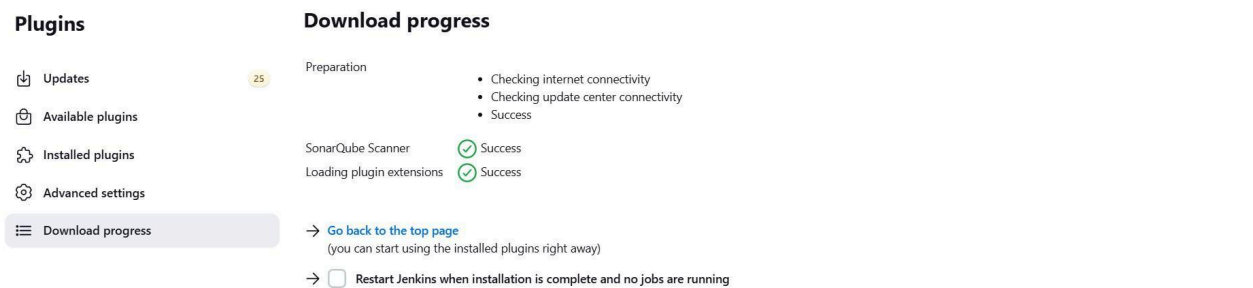
S	W	Name	Last Success	Last Failure	Last Duration
✓	☁	advdevops07	1 day 0 hr #4	1 day 0 hr #3	14 sec
✓	☀	Devops Pipeline	1 mo 21 days #2	N/A	4.5 sec
✓	☀	DevopsExp06	25 days #1	N/A	5.4 sec
✓	☀	devopsPipeline	1 mo 2 days #2	N/A	1.2 sec
✓	☀	MyProject	25 days #3	N/A	0.22 sec
⌛	☀	Tomcat1	N/A	N/A	N/A
✗	☁	webApp	25 days #7	25 days #8	7.2 sec

6. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



The screenshot shows the 'Manage Jenkins' page, specifically the 'Plugins' section. On the left, there's a sidebar with navigation links: Updates, Available plugins, Installed plugins, and Advanced settings. The main area displays a table of plugins with columns: Install, Name, and Released. The table lists three plugins: 'SonarQube Scanner' (version 2.17.2), 'Sonar Quality Gates' (version 3.15.v1f12b\_e81a\_3a\_4), and 'Quality Gates' (version 2.5). The 'SonarQube Scanner' plugin is highlighted, and its description is shown below the table. A search bar at the top right contains 'sonar'.

Install	Name	Released
<input checked="" type="checkbox"/>	<b>SonarQube Scanner</b> 2.17.2 External Site/Tool Integrations Build Reports This plugin allows an easy integration of <b>SonarQube</b> , the open source platform for Continuous Inspection of code quality.	7 mo 6 days ago
<input type="checkbox"/>	<b>Sonar Quality Gates</b> 3.15.v1f12b_e81a_3a_4 Library plugins (for use by other plugins) analysis Other Post-Build Actions Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")	27 days ago
<input type="checkbox"/>	<b>Quality Gates</b> 2.5 Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")	



The screenshot shows the 'Download progress' section of the Jenkins 'Manage Jenkins' page. On the left, there's a sidebar with navigation links: Updates, Available plugins, Installed plugins, and Advanced settings. The main area displays a table of download progress with columns: Name, Status, and Details. The table lists three items: 'Preparation', 'SonarQube Scanner', and 'Loading plugin extensions'. The 'SonarQube Scanner' item is highlighted, and its status is shown as 'Success'. A search bar at the top right contains 'sonar'.

Name	Status	Details
Preparation		<ul style="list-style-type: none"> <li>Checking internet connectivity</li> <li>Checking update center connectivity</li> <li>Success</li> </ul>
SonarQube Scanner	✓ Success	
Loading plugin extensions	✓ Success	

7. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **adv\_devops\_7\_sonarqube**

In **Server URL** Default is **http://localhost:9000**

#### SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

☐ Environment variables

#### SonarQube installations

List of SonarQube installations

Name

sonarqube

Server URL

Default is http://localhost:9000

https://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add

Save

Apply

8. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

**Dashboard > Manage Jenkins > Tools**

#### Gradle installations

Add Gradle

#### SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

#### SonarQube Scanner installations

SonarQube Scanner installations Edited

Check the “Install automatically” option. → Under name any name as identifier → Check the “Install automatically” option.

## SonarQube Scanner installations

SonarQube Scanner installations ^  Edited

Add SonarQube Scanner

 SonarQube Scanner 

Name

☒ Install automatically 

 Install from Maven Central 

Version

Add Installer



9. After configuration, create a New Item → choose a pipeline project.

### New Item

Enter an item name

advdevop\_exp08

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.

OK

10. Under Pipeline script, enter the following:

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

```

}

stage('SonarQube analysis') {
  withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
    sh """
      <PATH_TO_SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
      -D sonar.login=<SonarQube_USERNAME> \
      -D sonar.password=<SonarQube_PASSWORD> \
      -D sonar.projectKey=<Project_KEY> \
      -D sonar.exclusions=vendor/**,resources/**,*/*.java \
      -D sonar.host.url=<SonarQube_URL>(default: http://localhost:9000/)
    """
  }
}
}

```

It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

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\HP\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64 (1)\sonar-scanner-6.1.0.4477-windows-x64\bin\sona
10        -D sonar.login=admin ^
11        -D sonar.password=#bhushan45 ^
12        -D sonar.projectKey=sonarqube ^
13        -D sonar.exclusions=vendor/**,resources/**,*/*.java ^
14        -D sonar.host.url=http://localhost:9000
15        """
16      }
17    }
18  }

```

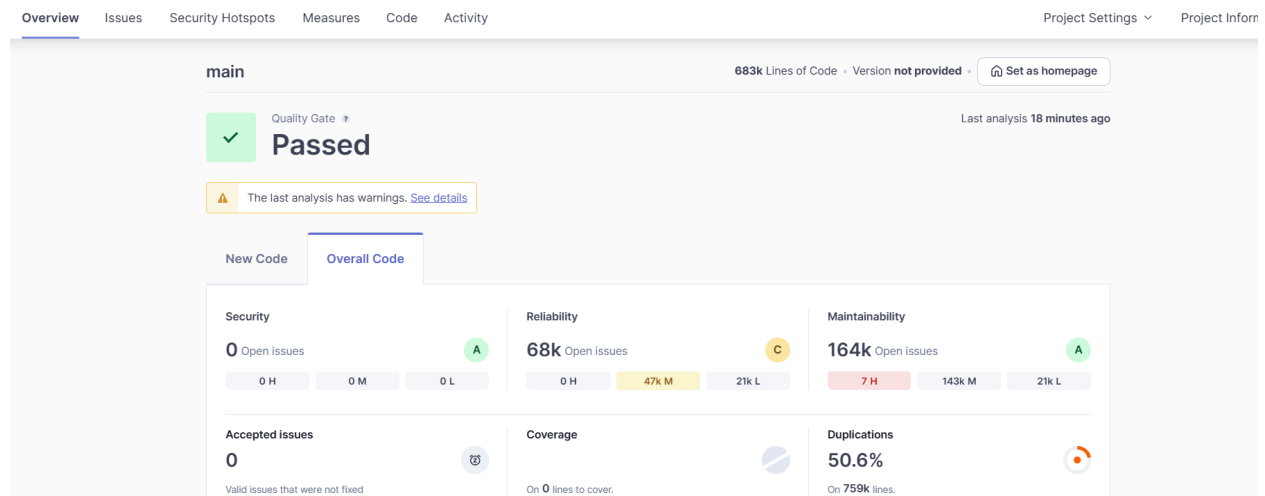
try sample Pipeline...

☒ Use Groovy Sandbox ?

Save Apply

## 11. Build project





## 14. Code Problems

- Consistency

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

☐ [Remove this deprecated "width" attribute.](#) Consistency  
Maintainability html5 obsolete +

- Intentionality



Bulk Change

Select issues  Navigate to issue  210,549 issues 3135d 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

## Bugs

Bulk Change

Select issues  Navigate to issue  67,624 issues 1646d effort

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

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

Intentionality

Reliability

accessibility wcag2-a 

+

Open

Not assigned

L9 • 2min effort • 4 years ago •  Bug •  Major

## Code Smells

Filters 

Clear All Filters

Issues in new code

Clean Code Attribute

Consistency 164k

Intentionality 15

Adaptability 0

Responsibility 0

Software Quality 1 

x

Security 0

Reliability 68k

Maintainability 164k

Add to selection 

Ctrl

 + click

Bulk Change

Select issues  Navigate to issue  163,781 issues 1705d 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 

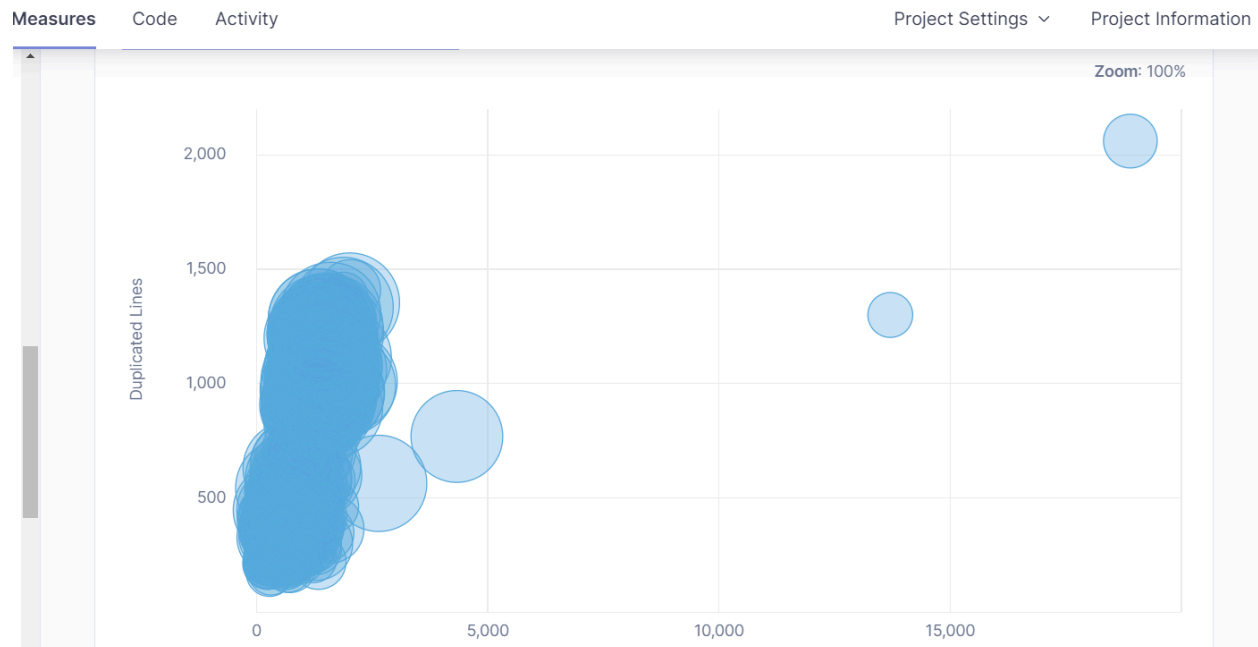
+

Open

Not assigned

L12 • 5min effort • 4 years ago •  Code Smell •  Major

- Duplications



- Cyclomatic Complexities

Cyclomatic Complexity 1,112 <a href="#">See history</a>		New Code: Since September 23, 2024
gameoflife-acceptance-tests	—	
gameoflife-build	—	
gameoflife-core	18	
gameoflife-deploy	—	
gameoflife-web	1,094	

In this way, we have integrated Jenkins with SonarQube for SAST.

### Conclusion:

In this experiment, we integrated Jenkins with SonarQube to implement automated code quality checks within our CI/CD pipeline. The process began with deploying SonarQube using Docker, followed by setting up a dedicated project within SonarQube to perform detailed code quality analysis. We then configured Jenkins by installing the SonarQube Scanner plugin, which allowed Jenkins to communicate with SonarQube. This included adding the necessary SonarQube server details and configuring the scanner tool to analyze code efficiently.

A Jenkins pipeline was developed to automate key tasks, such as cloning the code from a GitHub repository and triggering the SonarQube analysis on the codebase. Through this

integration, the pipeline continuously monitors code quality, providing detailed reports on issues such as bugs, code smells, and security vulnerabilities. This setup not only ensures early detection of potential problems but also promotes consistent improvements in code quality throughout the development lifecycle.