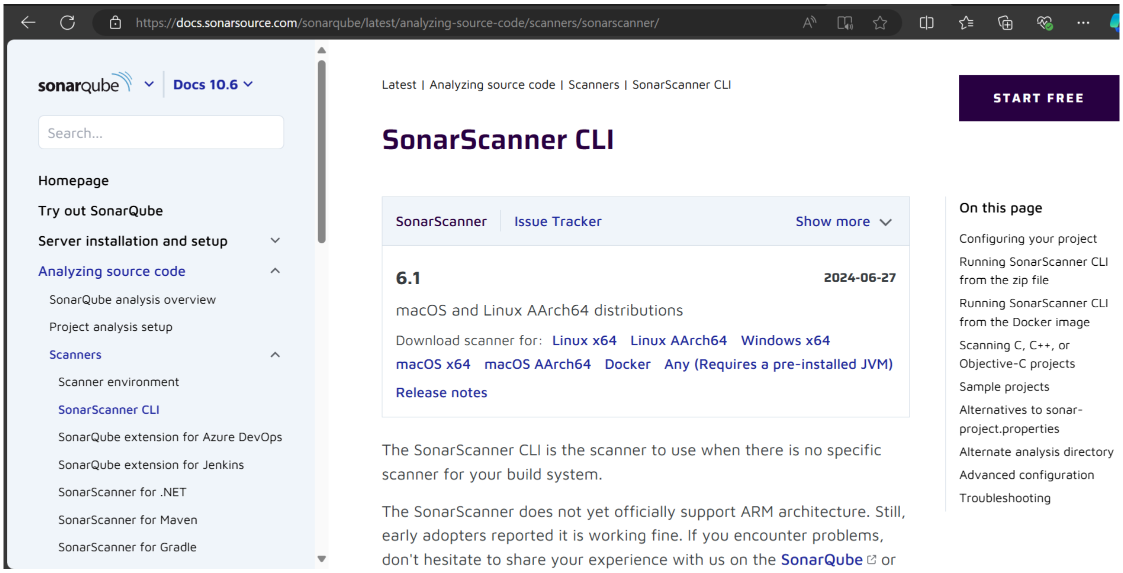
Name: Anish Kulkarni Roll No.: 29 Class: D15C AY: 2024-25

Experiment 8

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.

Steps:

Step 1: Install the SonarScanner CLI from the following link:
https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/


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

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)

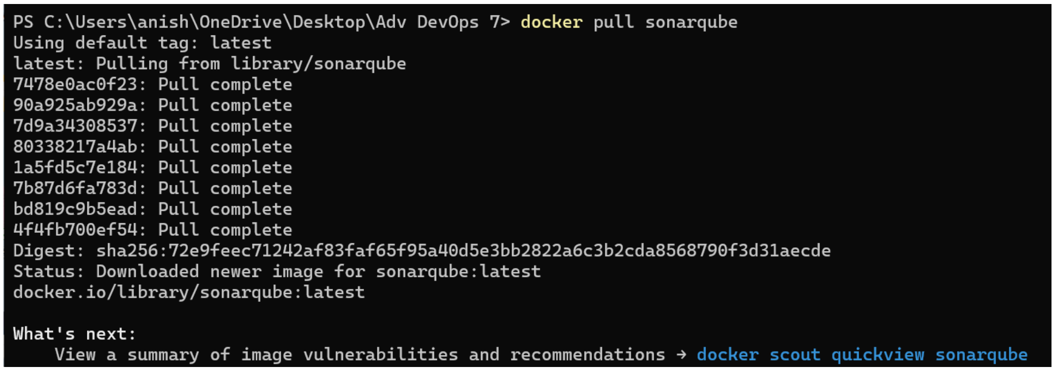
Release notes

The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system.

The SonarScanner does not yet officially support ARM architecture. Still, early adopters reported it is working fine. If you encounter problems, don't hesitate to share your experience with us on the SonarQube or

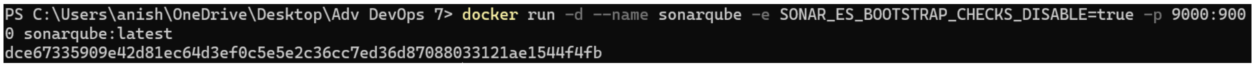
Once download is complete, extract the downloaded files into a folder.

Step 2: Install a SonarQube image by running the 'docker pull sonarqube' command on your terminal. This allows for a Sonarqube image to be used on a local machine without having to install the SonarQube application.



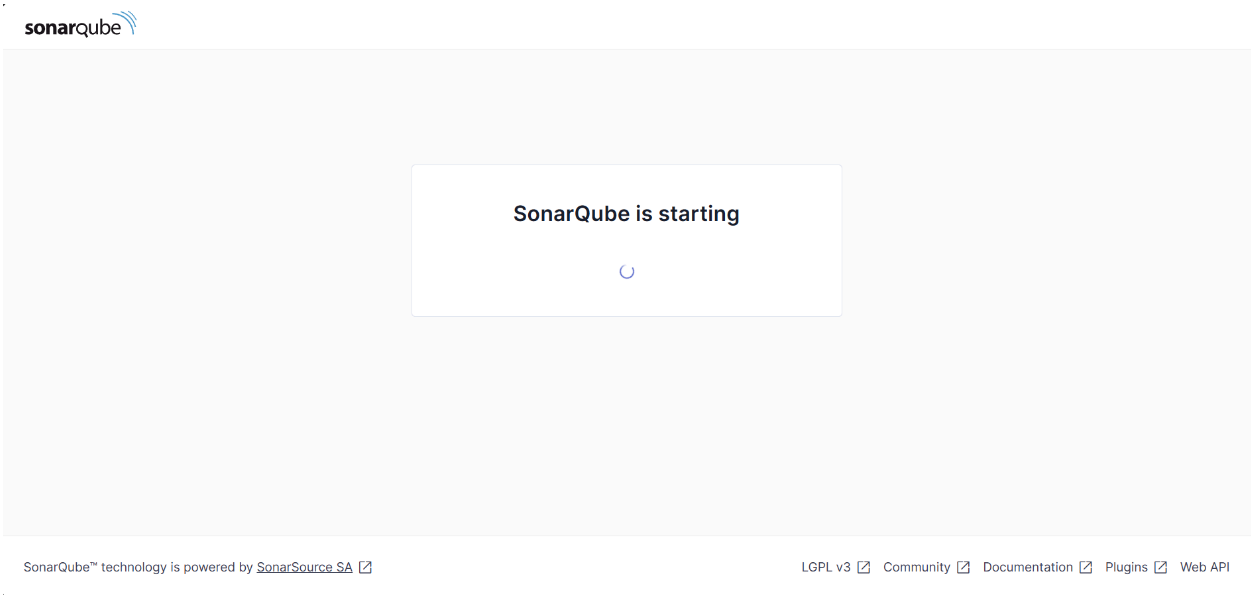
```
PS C:\Users\anish\OneDrive\Desktop\Adv DevOps 7> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
```

Step 3: Execute the following command:
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
This command will run the SonarQube image that was just installed using docker.



```
PS C:\Users\anish\OneDrive\Desktop\Adv DevOps 7> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:900
0 sonarqube:latest
dce67335909e42d81ec64d3ef0c5e5e2c36cc7ed36d87088033121ae1544f4fb
```

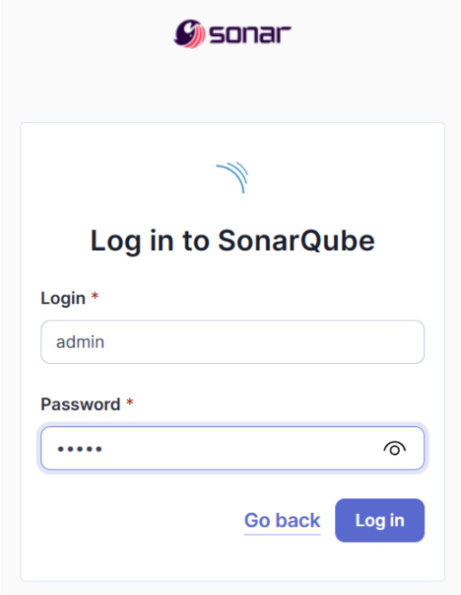
Step 4: Go to http://localhost:9000 on your browser and check if SonarQube is starting or not.



SonarQube is starting

SonarQube™ technology is powered by SonarSource SA

Step 5: On the login page, enter ‘Login’ as admin and ‘Password’ as admin to log in initially. It then asks you to change the password to a password of your choice. Do the same and proceed to the next step.

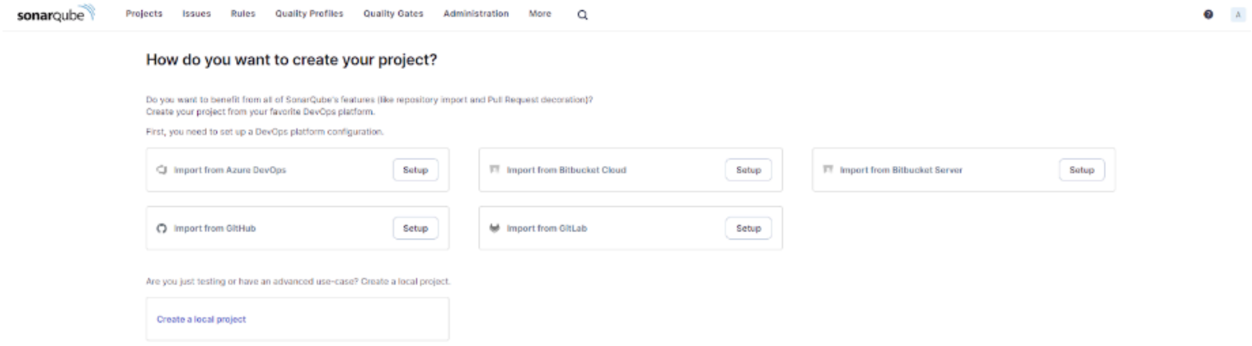


Log in to SonarQube

Login *
admin

Password *
•••••

Step 6: On the SonarQube dashboard, click on 'Create a local project'.



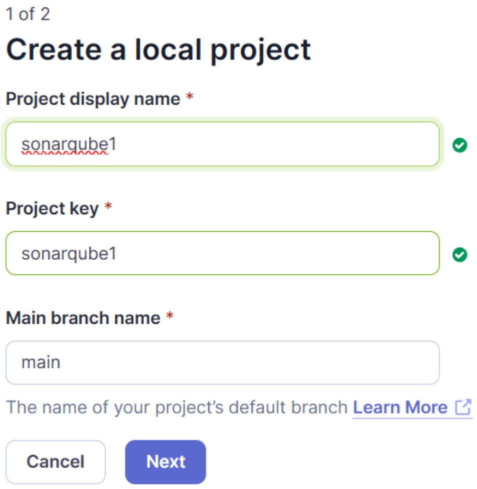
How do you want to create your project?

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.

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

Step 7: Create a local project by entering the project name and key and click on 'Next'.



1 of 2

Create a local project

Project display name *

sonarqube1

Project key *

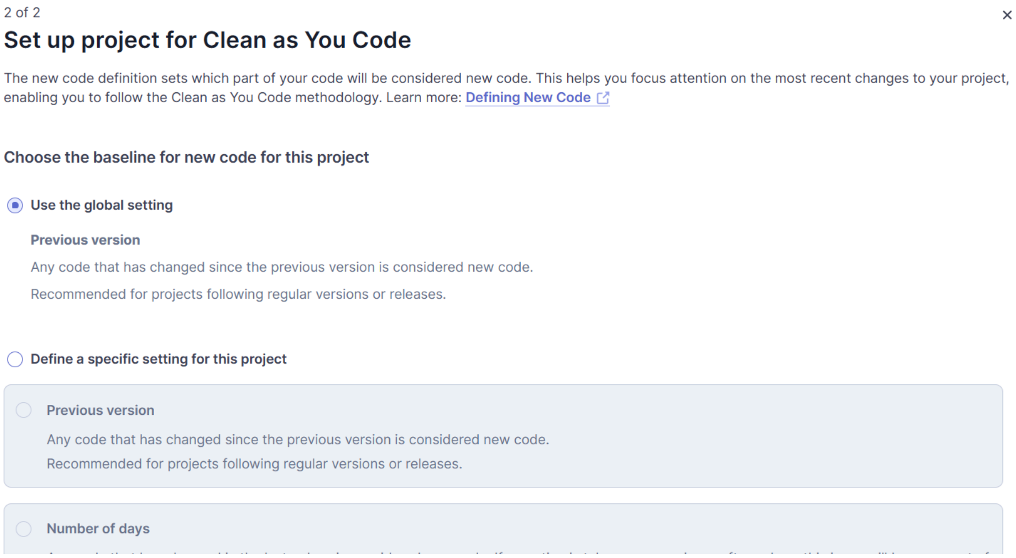
sonarqube1

Main branch name *

main

The name of your project's default branch Learn More

Step 8: Set up your project and click on 'Create project'.



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

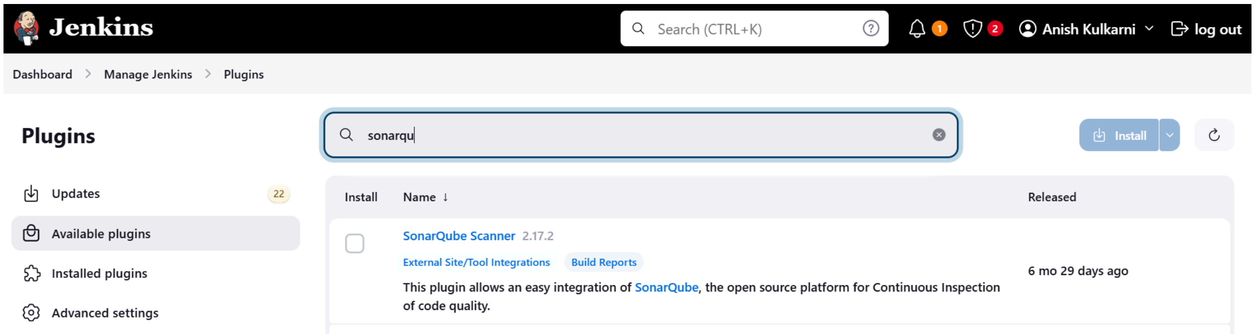
○ Previous version

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

Recommended for projects following regular versions or releases.

○ Number of days

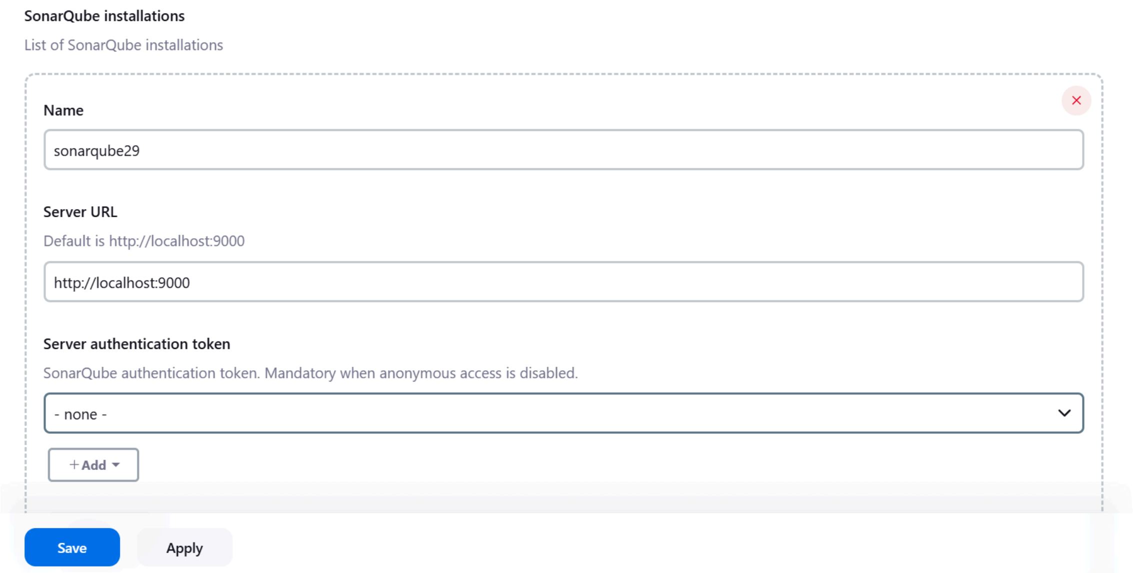
Step 9: Navigate to your Jenkins server (on whichever port it has been installed), click on ‘Manage Jenkins’, click on ‘Plugins’ and search for the ‘SonarQube Scanner’ plugin and install it.



sonarqu

Install	Name ↓	Released
□	SonarQube Scanner 2.17.2 External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	6 mo 29 days ago

Step 10: Under ‘Manage Jenkins’, click on System. Under the ‘Sonarqube installations’ section, add a server and add a server authentication token if needed.



SonarQube installations

List of SonarQube installations

Name

sonarqube29

Server URL

Default is http://localhost:9000

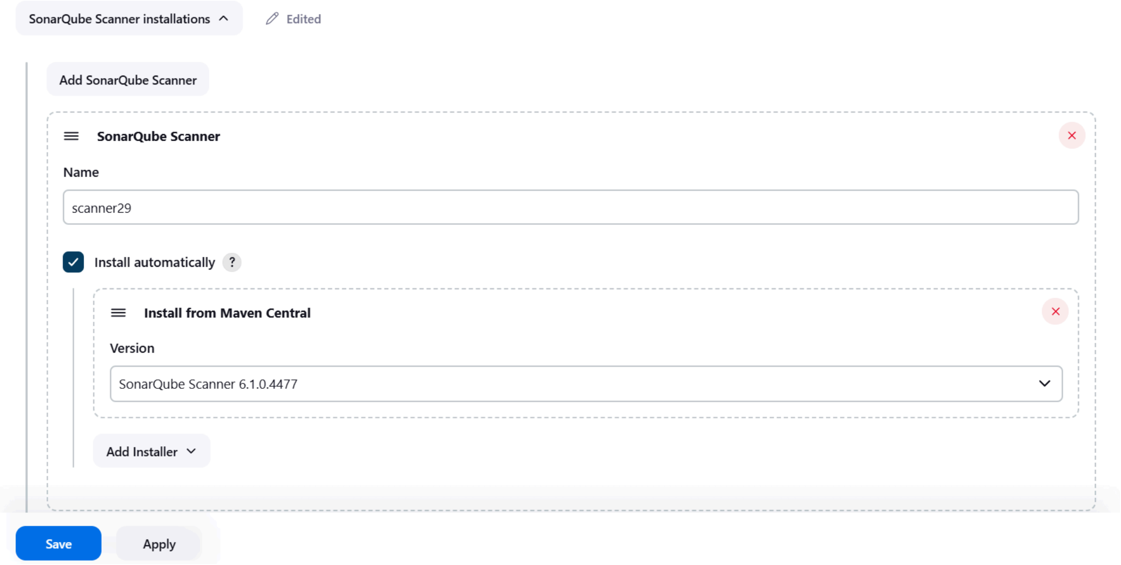
http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

Step 11: Under ‘Manage Jenkins’, click on ‘Tools’. Under the ‘SonarQube Scanner installations’ section, give your scanner a name, choose the latest version and click on ‘Install automatically’.



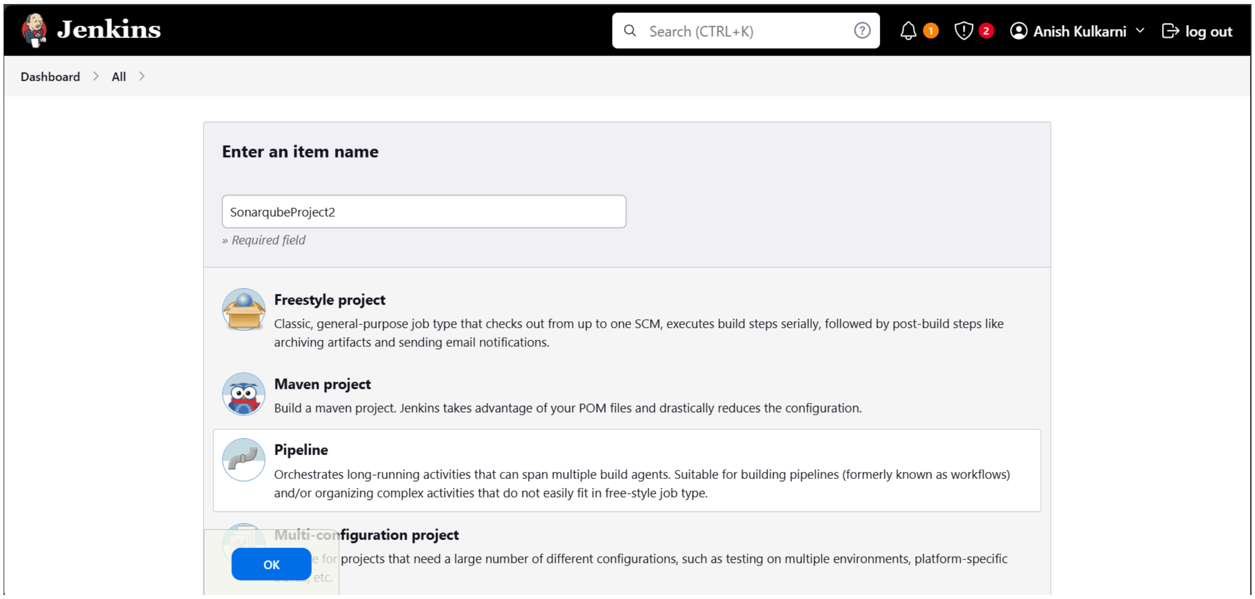
Name: scanner29

☑ Install automatically

Install from Maven Central

Version: SonarQube Scanner 6.1.0.4477

Step 12: Create a new Jenkins project by giving it a name and ensure that it is a pipeline project.



Enter an item name

SonarqubeProject2

» 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.

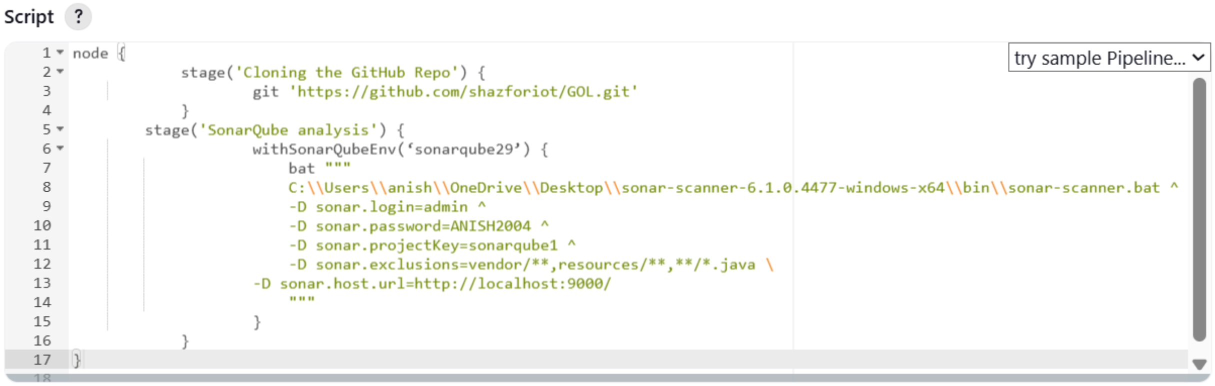
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

Step 13: Under the ‘Pipeline Script’ section, enter the following:-

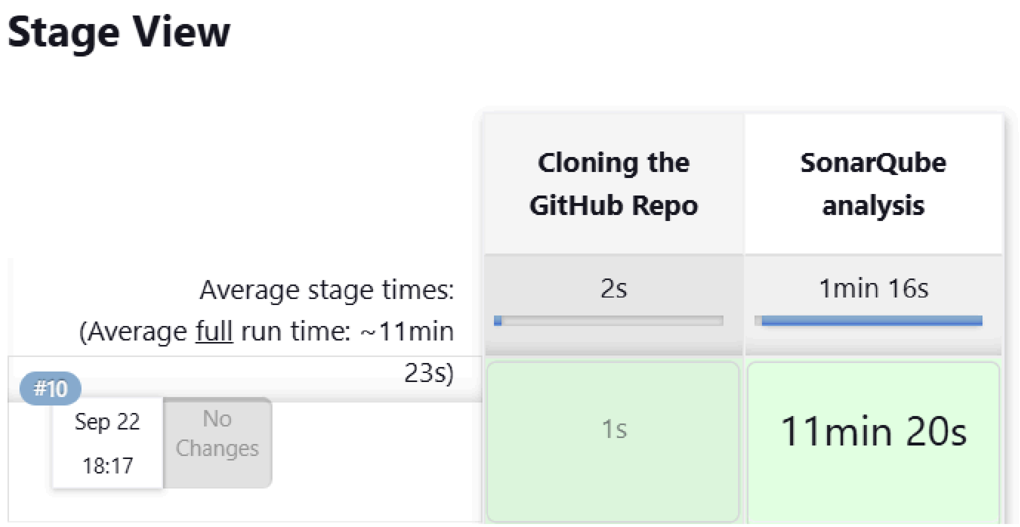
```
node {
                stage('Cloning the GitHub Repo') {
                git 'https://github.com/shazforiot/GOL.git'
                }

                 stage('SonarQube analysis') {
                        withSonarQubeEnv('sonarqube29') {
                                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/
                                 """

                }
        }
}
```



```
node {
        stage('Cloning the GitHub Repo') {
                git 'https://github.com/shazforiot/GOL.git'
        }
    stage('SonarQube analysis') {
                withSonarQubeEnv(‘sonarqube29’) {
                    bat """
                    C:\\Users\\anish\\OneDrive\\Desktop\\sonar-scanner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.bat ^
                    -D sonar.login=admin ^
                    -D sonar.password=ANISH2004 ^
                    -D sonar.projectKey=sonarqube1 ^
                    -D sonar.exclusions=vendor/**,resources/**,**/*.java \
                -D sonar.host.url=http://localhost:9000/
                    """
                }
        }
}
```

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

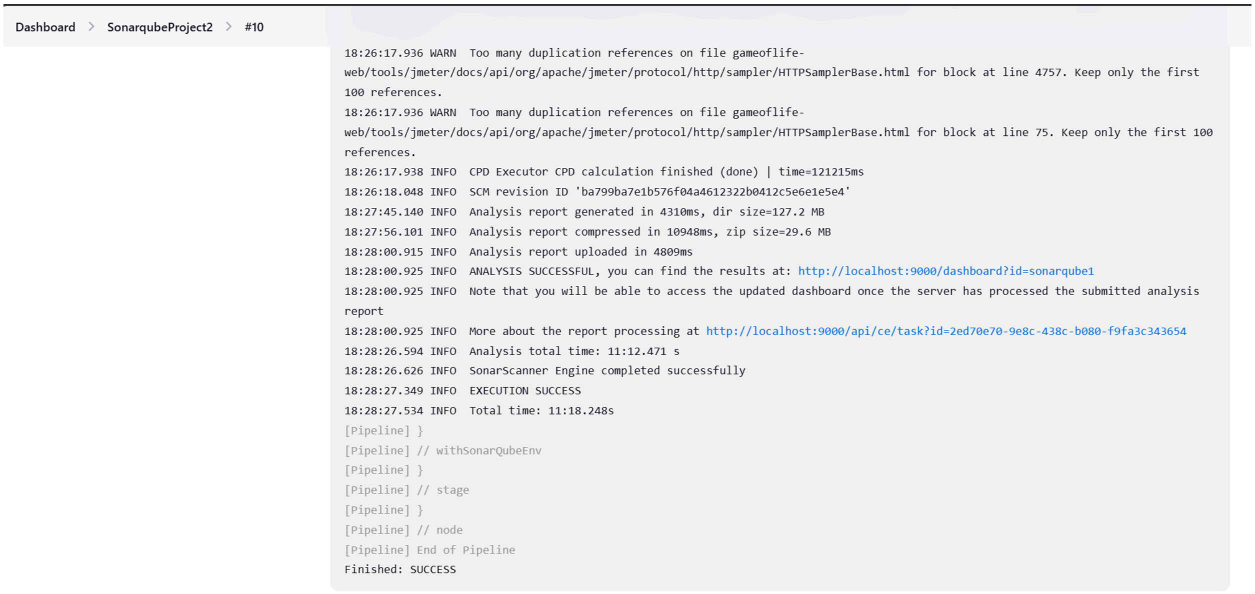
Step 14: Go back to Jenkins, navigate to your Jenkins project and click on ‘Build Now’.



Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~11min 23s)	2s	1min 16s
#10 Sep 22 18:17 No Changes	1s	11min 20s

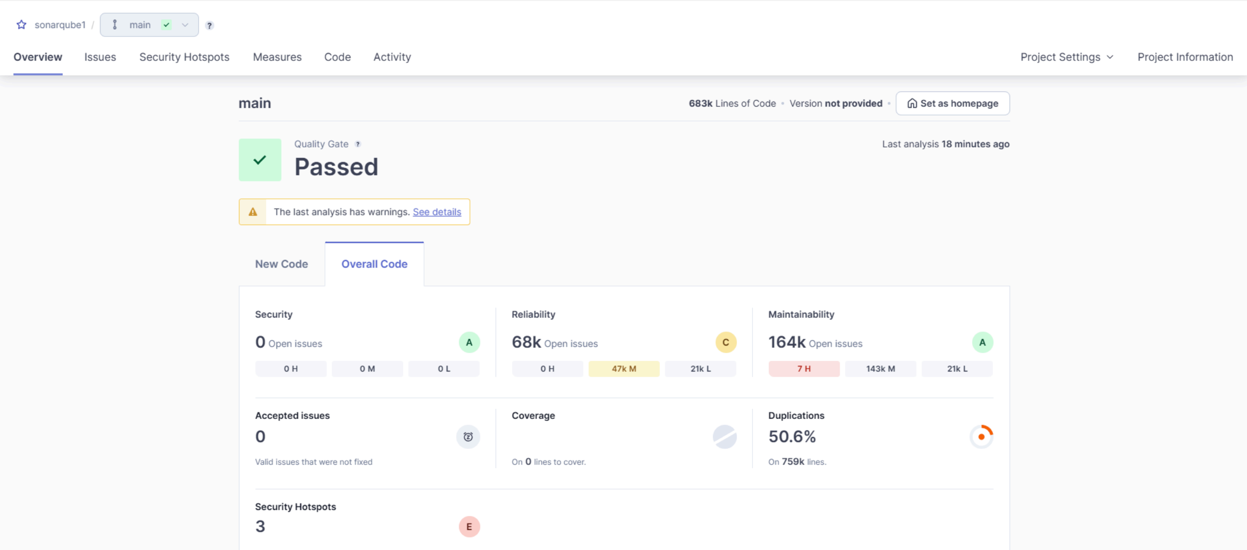
Step 15: Once build is successfully completed, check the console output.



Dashboard > SonarqubeProject2 > #10

```
18:26:17.936 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/sampler/HTTPSamplerBase.html for block at line 4757. Keep only the first
100 references.
18:26:17.936 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/sampler/HTTPSamplerBase.html for block at line 75. Keep only the first 100
references.
18:26:17.938 INFO  CPD Executor CPD calculation finished (done) | time=121215ms
18:26:18.048 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
18:27:45.140 INFO  Analysis report generated in 4310ms, dir size=127.2 MB
18:27:56.101 INFO  Analysis report compressed in 10948ms, zip size=29.6 MB
18:28:00.915 INFO  Analysis report uploaded in 4809ms
18:28:00.925 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube1
18:28:00.925 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis
report
18:28:00.925 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=2ed70e70-9e8c-438c-b080-f9fa3c343654
18:28:26.594 INFO  Analysis total time: 11:12.471 s
18:28:26.626 INFO  SonarScanner Engine completed successfully
18:28:27.349 INFO  EXECUTION SUCCESS
18:28:27.534 INFO  Total time: 11:18.248s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Step 16: Go back to SonarQube and check your project.



sonarqube1 / main ?

Overview · Issues · Security Hotspots · Measures · Code · Activity · Project Settings · Project Information

main

683k Lines of Code • Version not provided • Set as homepage

Quality Gate ?

Passed

Last analysis 18 minutes ago

The last analysis has warnings. See details

New Code · Overall Code

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%

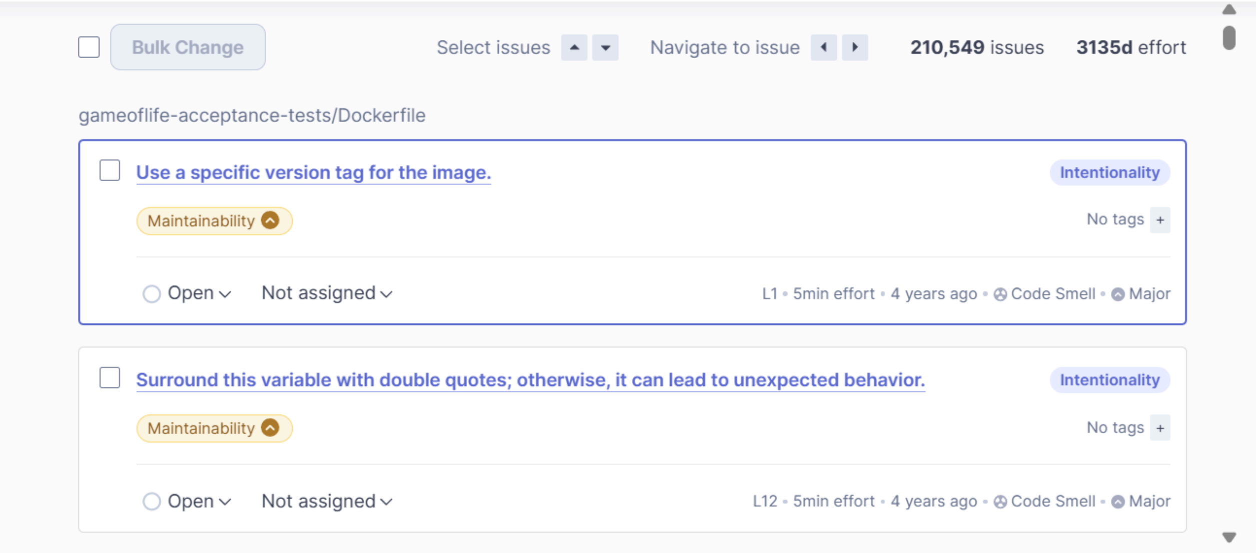
On 759k lines.

Security Hotspots

3 E

Step 17: Check the different types of issues with the code:-

- Code problems:-
    - 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 +

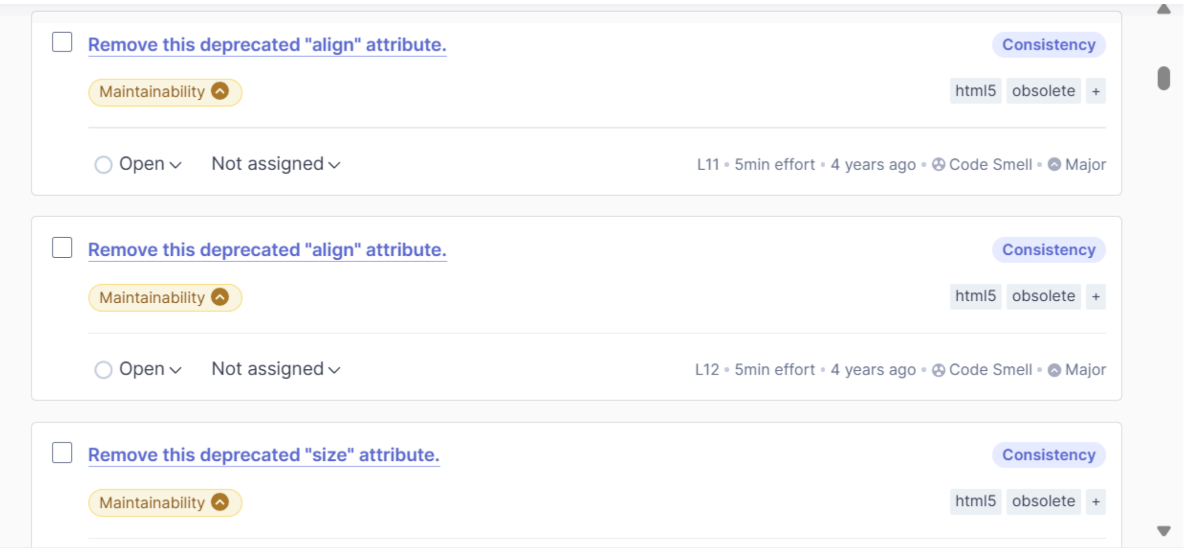
Open · Not assigned · L1 • 5min effort • 4 years ago • Code Smell • Major

□ 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

- Consistency:-

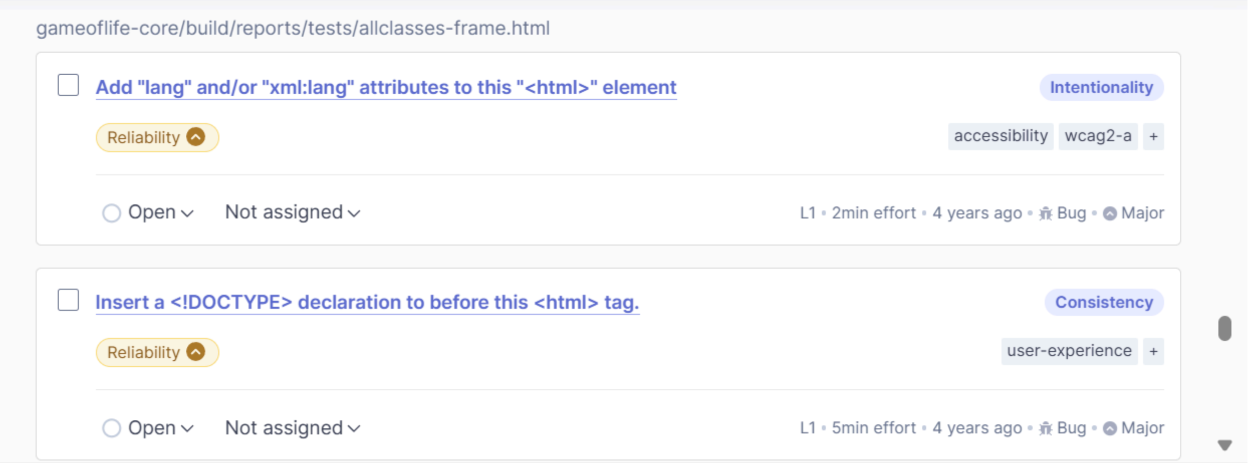


□ Remove this deprecated "align" attribute. Consistency
Maintainability html5 obsolete +
○ Open Not assigned L11 • 5min effort • 4 years ago • Code Smell • Major

□ Remove this deprecated "align" attribute. Consistency
Maintainability html5 obsolete +
○ Open Not assigned L12 • 5min effort • 4 years ago • Code Smell • Major

□ Remove this deprecated "size" attribute. Consistency
Maintainability html5 obsolete +

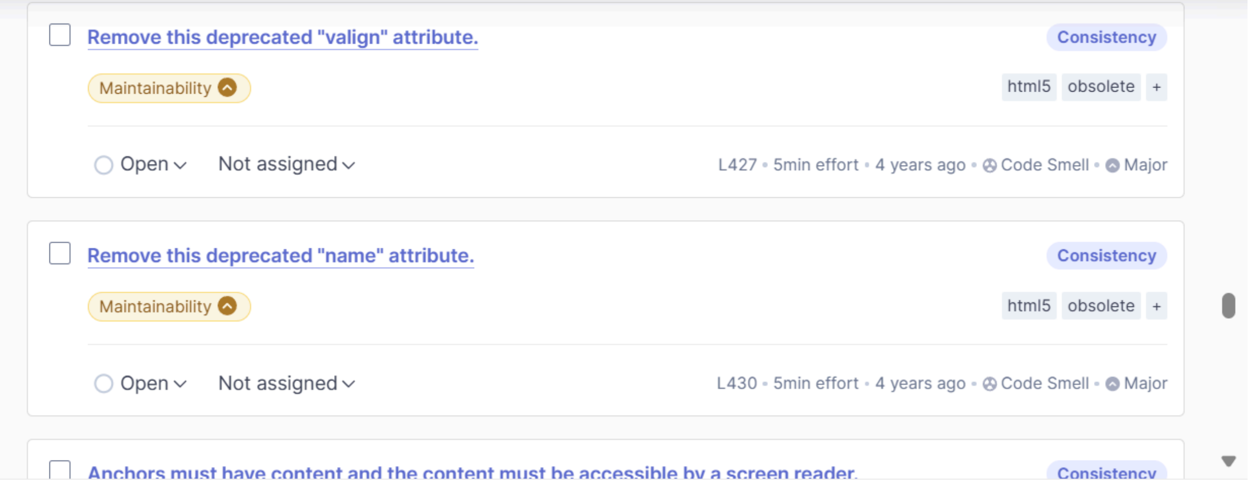
- Bugs and Code Smells:-



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

□ Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency
Reliability user-experience +
○ Open Not assigned L1 • 5min effort • 4 years ago • Bug • Major

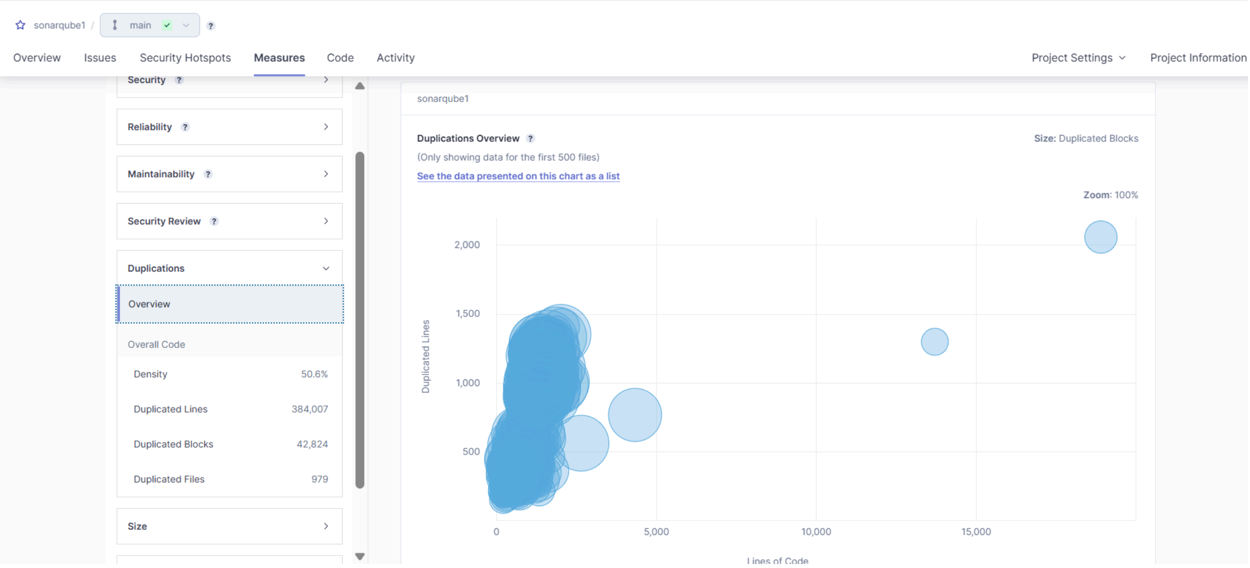


□ Remove this deprecated "valign" attribute. Consistency
Maintainability html5 obsolete +
○ Open Not assigned L427 • 5min effort • 4 years ago • Code Smell • Major

□ Remove this deprecated "name" attribute. Consistency
Maintainability html5 obsolete +
○ Open Not assigned L430 • 5min effort • 4 years ago • Code Smell • Major

□ Anchors must have content and the content must be accessible by a screen reader. Consistency

- Duplications:-



Overall Code

Density 50.6%

Duplicated Lines 384,007

Duplicated Blocks 42,824

Duplicated Files 979

Duplications Overview ?

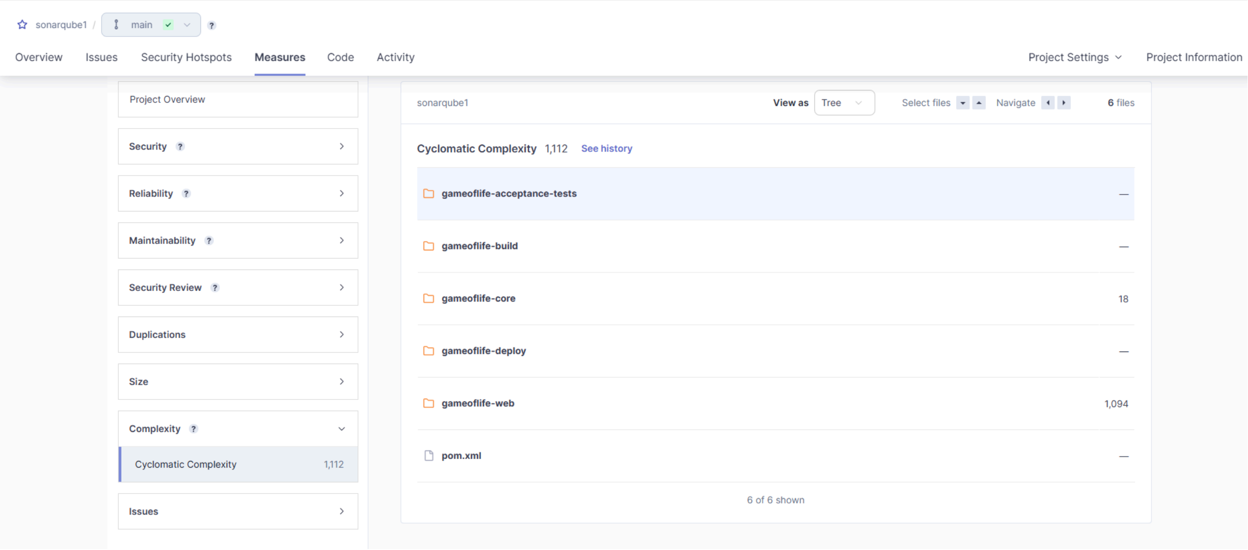
(Only showing data for the first 500 files)

See the data presented on this chart as a list

Size: Duplicated Blocks

Zoom: 100%

- Cyclomatic complexities:-



Cyclomatic Complexity 1,112

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

6 of 6 shown

Conclusion: In this experiment, we learned how to create a Jenkins CI/CD Pipeline with SonarQube integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Java application. A pipeline project is created in Jenkins and a pipeline script contains the link to the Java application on which the SonarQube analysis is to be done. Then the pipeline project is configured as per needs and built. The SonarQube project linked to the pipeline project then successfully does the SonarQube analysis and points out all the issues, bugs, duplications etc in the pipeline project.