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.

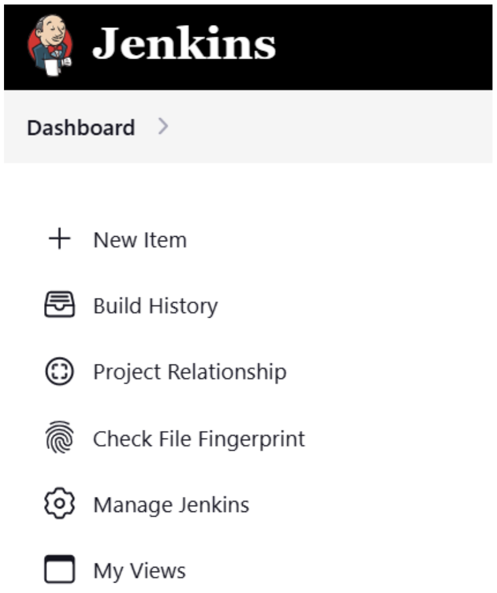
Integrating Jenkins with SonarQube:

Prerequisites:

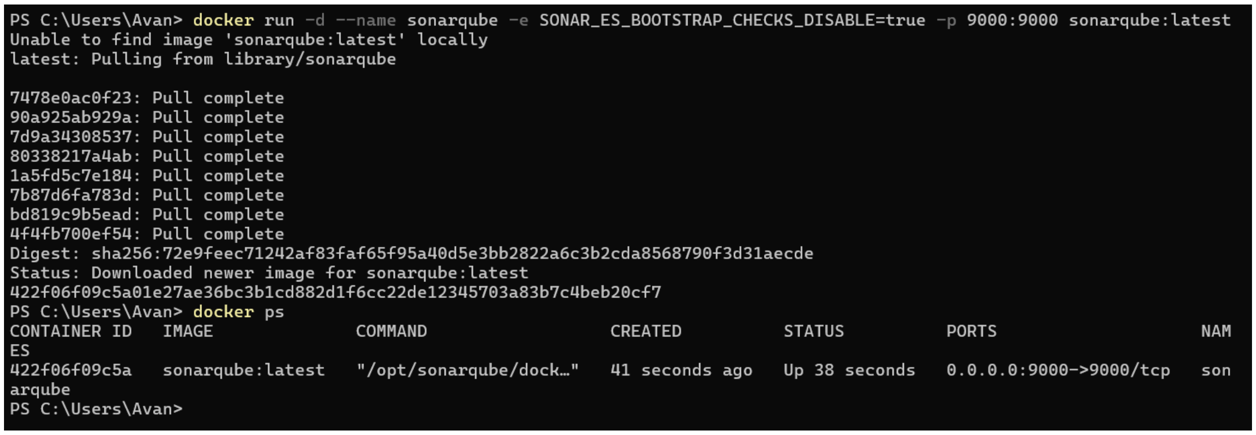
● Jenkins installed
● Docker Installed (for SonarQube)
● SonarQube Docker Image

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.

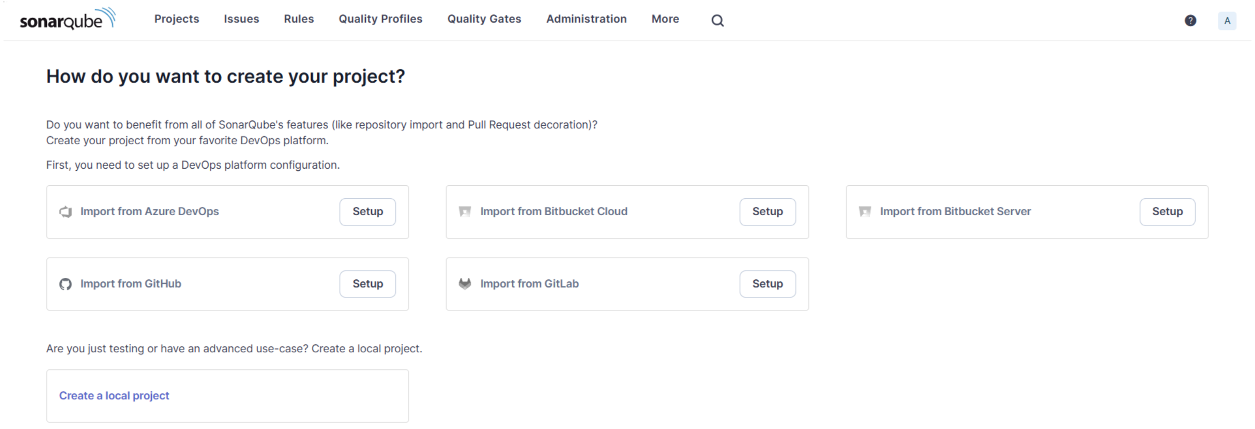


2. Run SonarQube in a Docker container using this command



```
PS C:\Users\Avan> 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
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
422f06f09c5a01e27ae36bc3b1cd882d1f6cc22de12345703a83b7c4beb20cf7
PS C:\Users\Avan> docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                    NAM
ES
422f06f09c5a   sonarqube:latest   "/opt/sonarqube/dock…"   41 seconds ago   Up 38 seconds   0.0.0.0:9000->9000/tcp   son
arqube
PS C:\Users\Avan>
```

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



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.

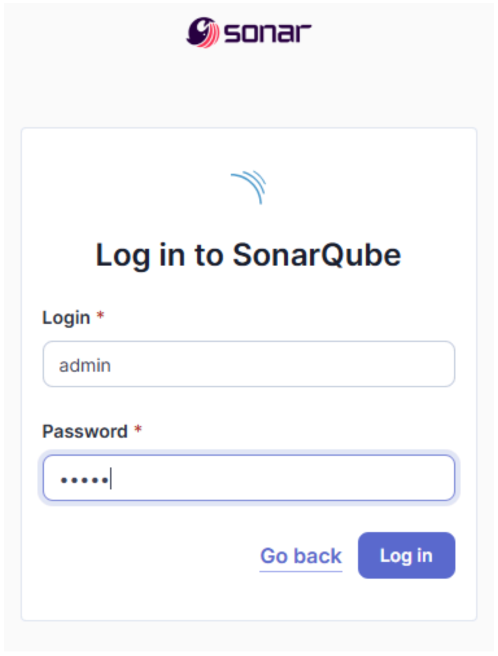
First, you need to set up a DevOps platform configuration.

Import from Azure DevOps — Setup
Import from Bitbucket Cloud — Setup
Import from Bitbucket Server — Setup
Import from GitHub — Setup
Import from GitLab — Setup

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

Create a local project

4. Login to SonarQube using your username and password

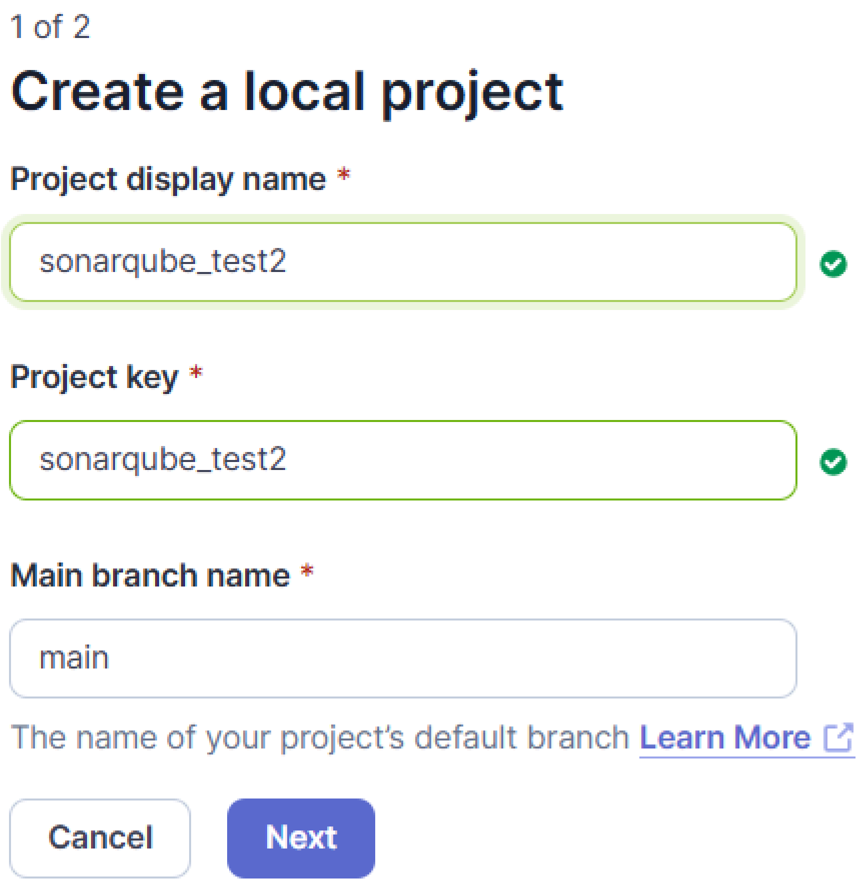


Log in to SonarQube

Login *
admin

Password *
•••••

5. Create a manual project in SonarQube with the name sonarqube-test2



1 of 2

Create a local project

Project display name *

sonarqube_test2

Project key *

sonarqube_test2

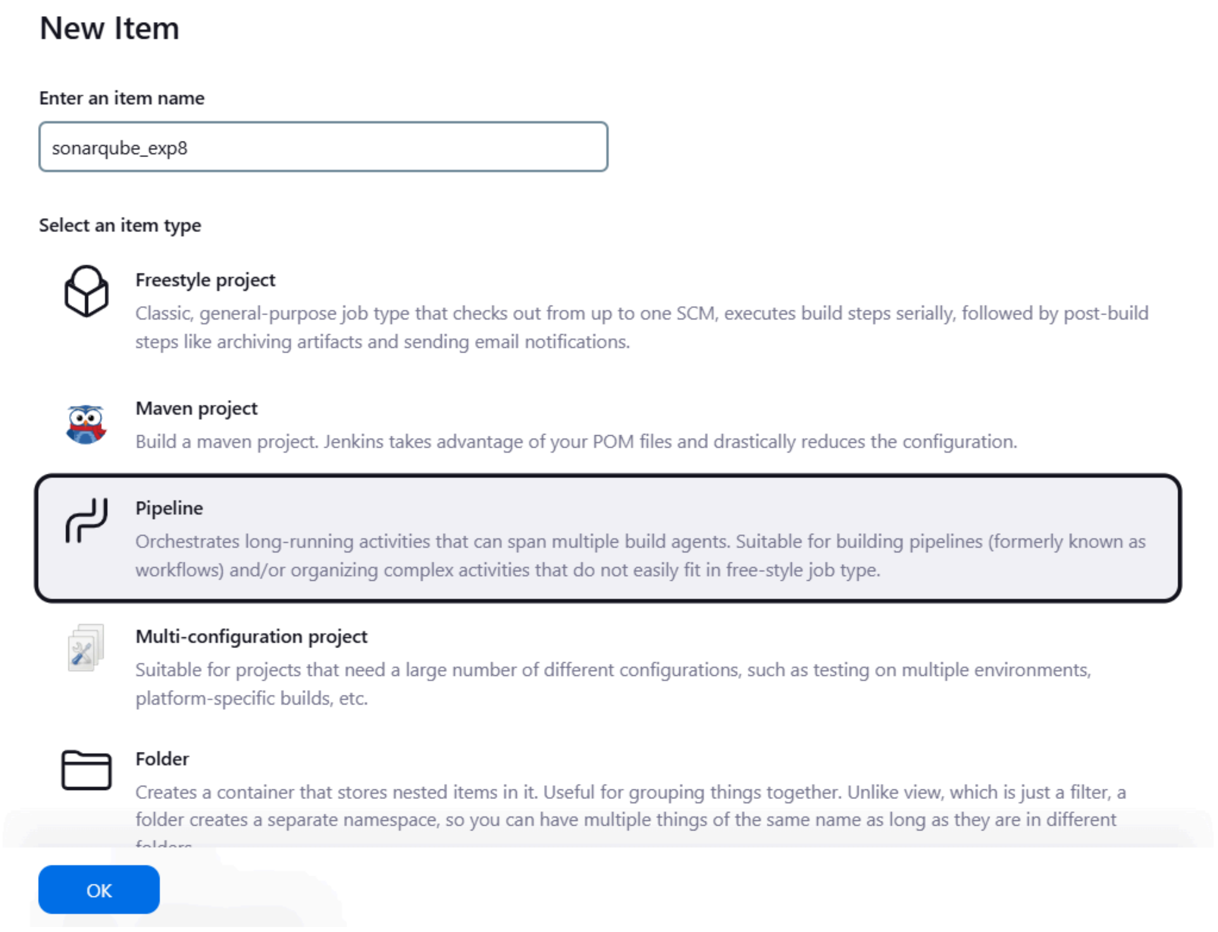
Main branch name *

main

The name of your project's default branch Learn More

Setup the project and come back to Jenkins Dashboard.

6. Create a New Item in Jenkins, choose **Pipeline**.


New Item

Enter an item name

sonarqube_exp8

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

7. Under Pipeline Script, enter the following -

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
sh "<PATH_TO_SONARQUBE_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=http://127.0.0.1:9000/"
}
}
}
```

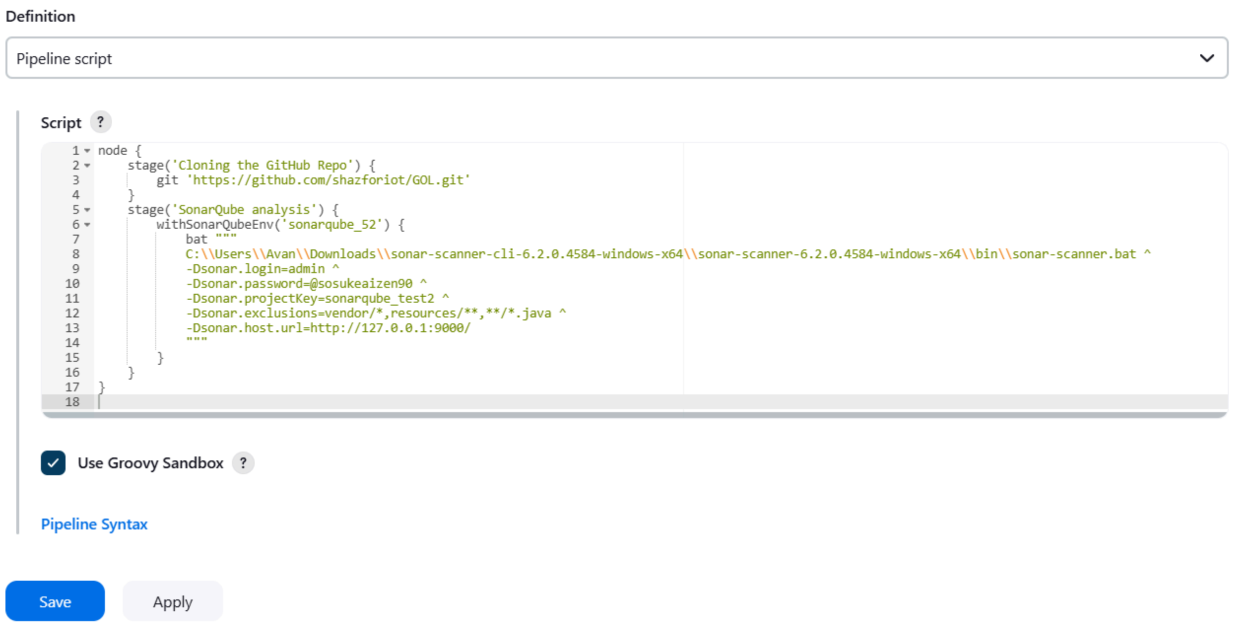
It is a java sample project which has a lot of repetitions and issues that will be detected by

SonarQube.

8. Install sonar-scanner

Now we need to install **sonar-scanner** binary to perform code analysis
To do so, go to
(https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0.4584-windows-x64.zip for windows OS)
For windows path would be
C:\\Users\<USER_NAME>\Downloads\sonar-scanner-cli-6.2.0.4584-windows-x64\bin\sonar-scanner
Now we need to update the required details in the pipeline script as :

```
node {
   stage('Cloning the GitHub Repo') {
      git 'https://github.com/shazforiot/GOL.git'
   }
   stage('SonarQube analysis') {
      withSonarQubeEnv('sonarqube_52') {
         bat """

C:\\Users\\Avan\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
         -Dsonar.login=admin ^
         -Dsonar.password=@sosukeaizen90 ^
         -Dsonar.projectKey=sonarqube_test2 ^
         -Dsonar.exclusions=vendor/*,resources/**,**/*.java ^
         -Dsonar.host.url=http://127.0.0.1:9000/
         """
      }
   }
}
```



Definition

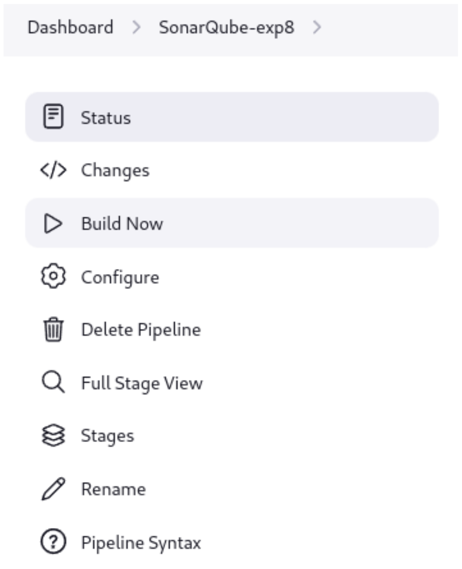
Pipeline script

Script

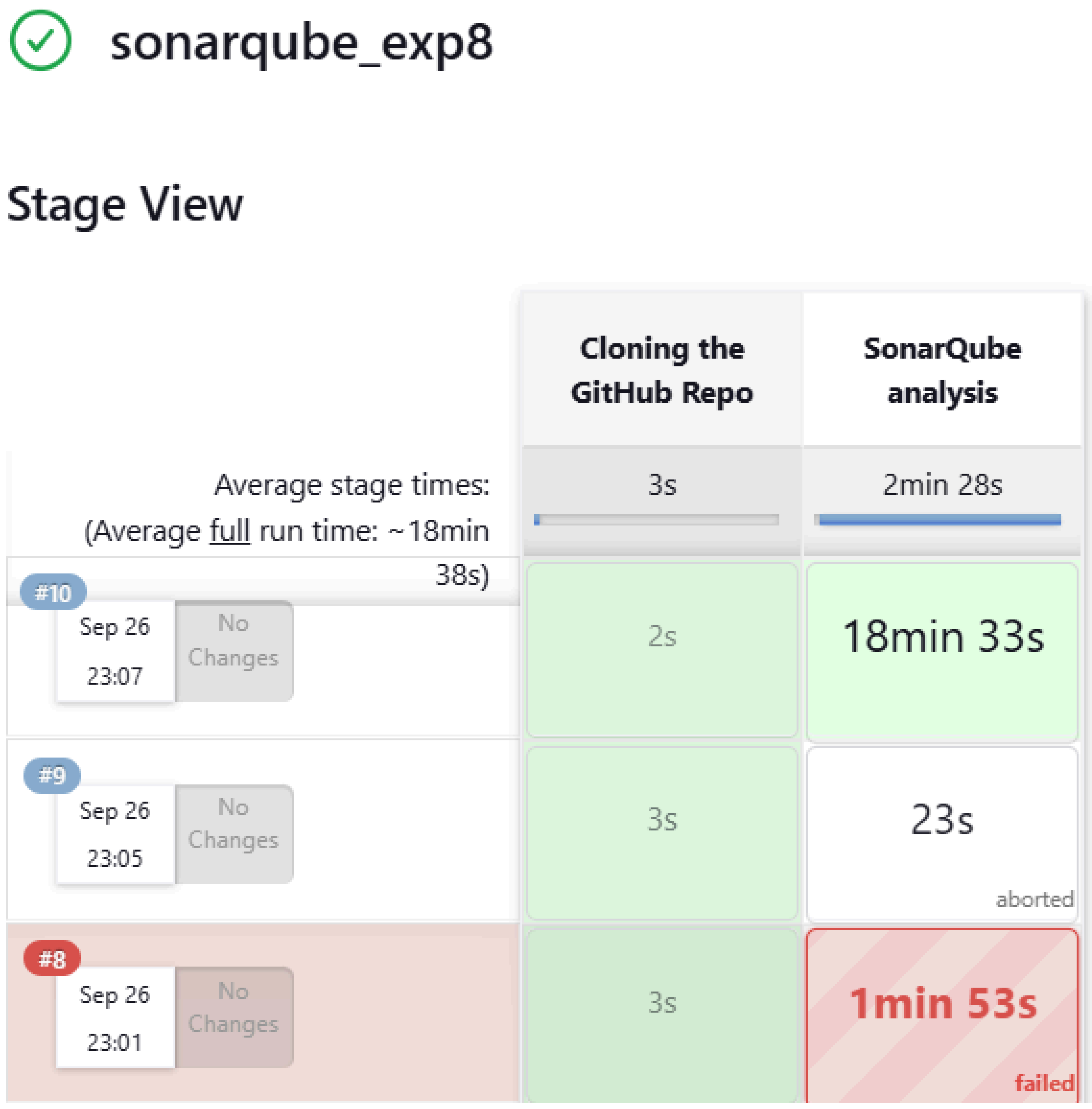
```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube_52') {
            bat """
            C:\\Users\\Avan\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
            -Dsonar.login=admin ^
            -Dsonar.password=@sosukeaizen90 ^
            -Dsonar.projectKey=sonarqube_test2 ^
            -Dsonar.exclusions=vendor/*,resources/**,**/*.java ^
            -Dsonar.host.url=http://127.0.0.1:9000/
            """
        }
    }
}

```

Use Groovy Sandbox [✓]

9. Run the build



Check the status of Build



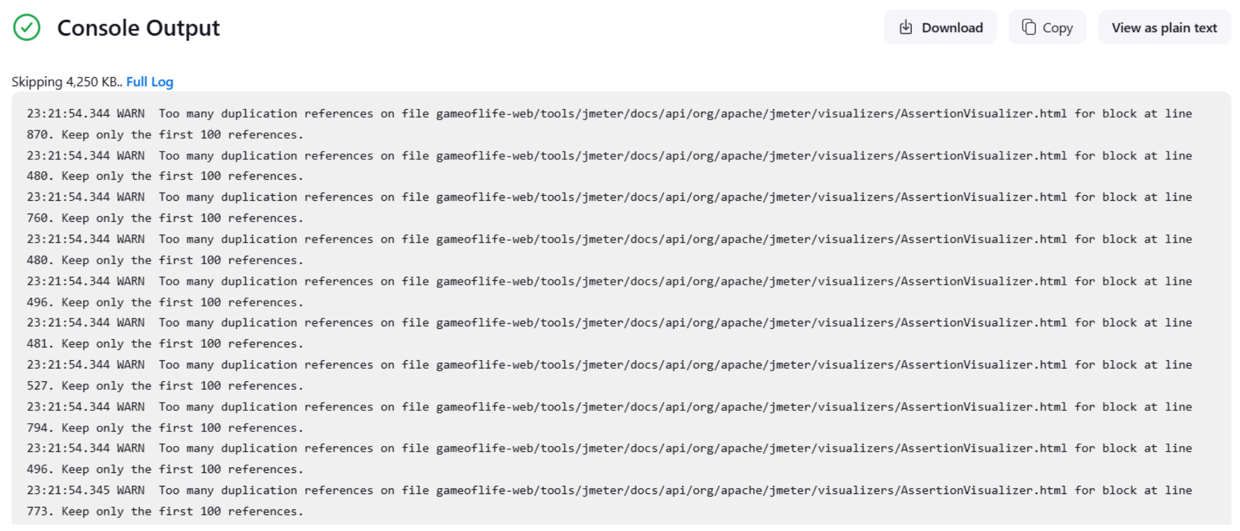
sonarqube_exp8

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~18min 38s)	3s	2min 28s
#10 Sep 26 23:07 No Changes	2s	18min 33s
#9 Sep 26 23:05 No Changes	3s	23s aborted
#8 Sep 26 23:01 No Changes	3s	1min 53s failed

As we can see the SonarQube analysis is completed

10. Check the console output once the build is complete.

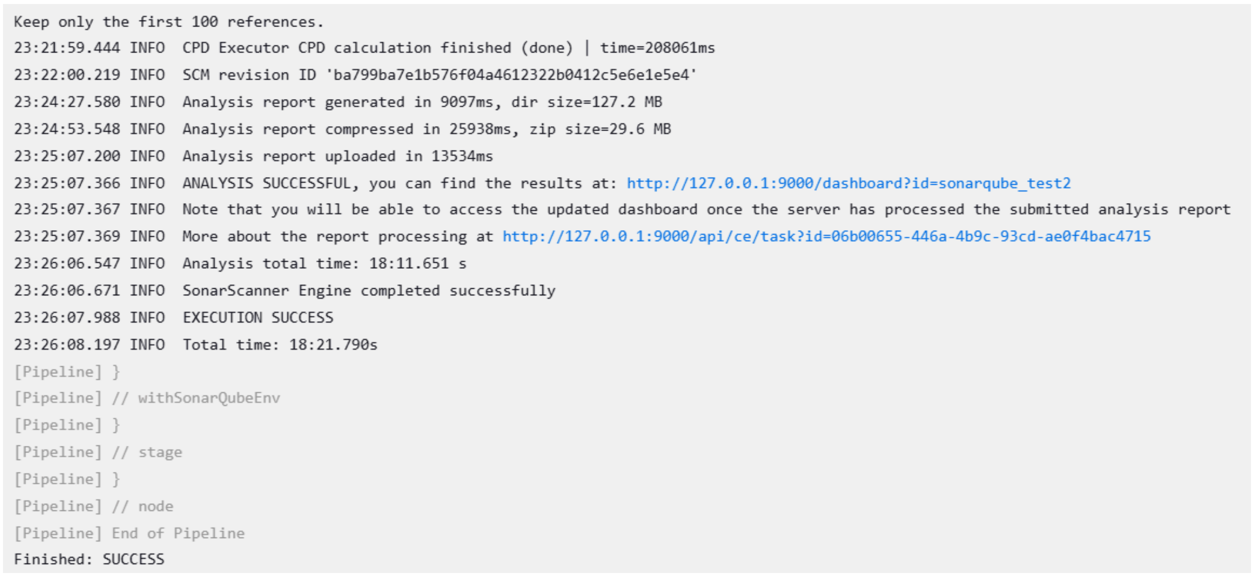


Console Output

Download Copy View as plain text

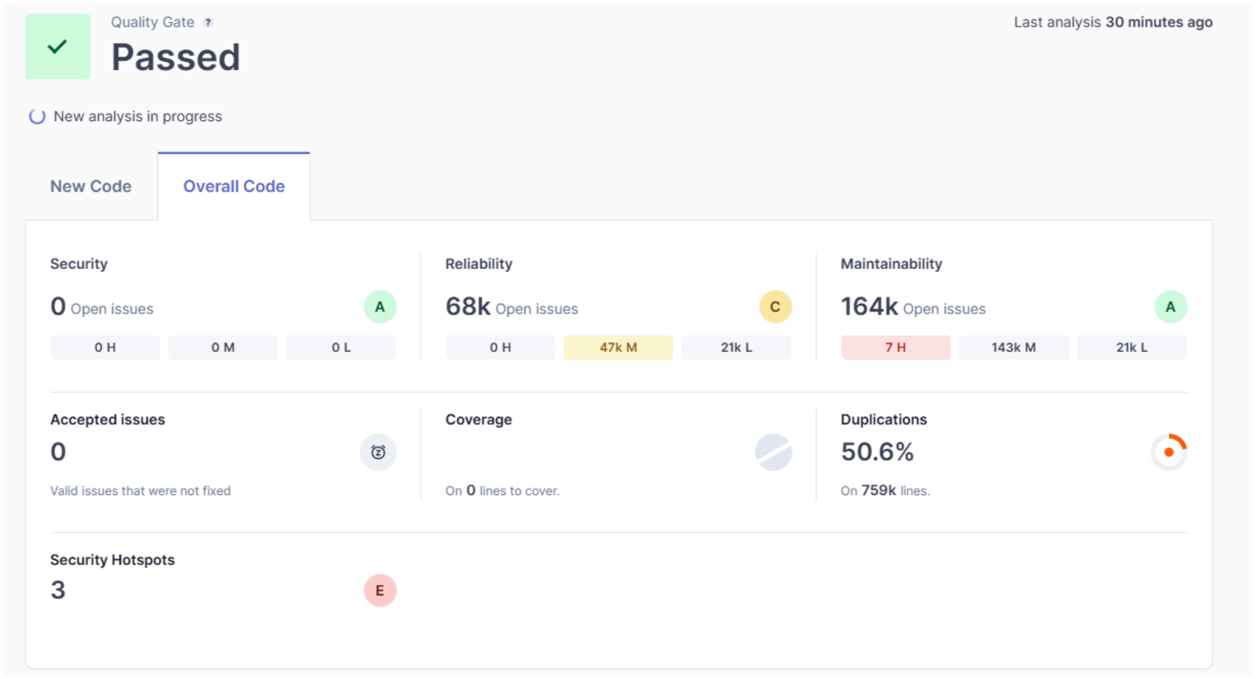
Skipping 4,250 KB.. Full Log

```
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
870. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
480. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
760. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
480. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
496. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
481. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
527. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
794. Keep only the first 100 references.
23:21:54.344 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
496. Keep only the first 100 references.
23:21:54.345 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/AssertionVisualizer.html for block at line
773. Keep only the first 100 references.
```



```
Keep only the first 100 references.
23:21:59.444 INFO  CPD Executor CPD calculation finished (done) | time=208061ms
23:22:00.219 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:24:27.580 INFO  Analysis report generated in 9097ms, dir size=127.2 MB
23:24:53.548 INFO  Analysis report compressed in 25938ms, zip size=29.6 MB
23:25:07.200 INFO  Analysis report uploaded in 13534ms
23:25:07.366 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube_test2
23:25:07.367 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:25:07.369 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=06b00655-446a-4b9c-93cd-ae0f4bac4715
23:26:06.547 INFO  Analysis total time: 18:11.651 s
23:26:06.671 INFO  SonarScanner Engine completed successfully
23:26:07.988 INFO  EXECUTION SUCCESS
23:26:08.197 INFO  Total time: 18:21.790s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

11. After that, check the project in SonarQube.



Quality Gate ?

Passed

Last analysis **30 minutes ago**

New analysis in progress

New Code | Overall Code

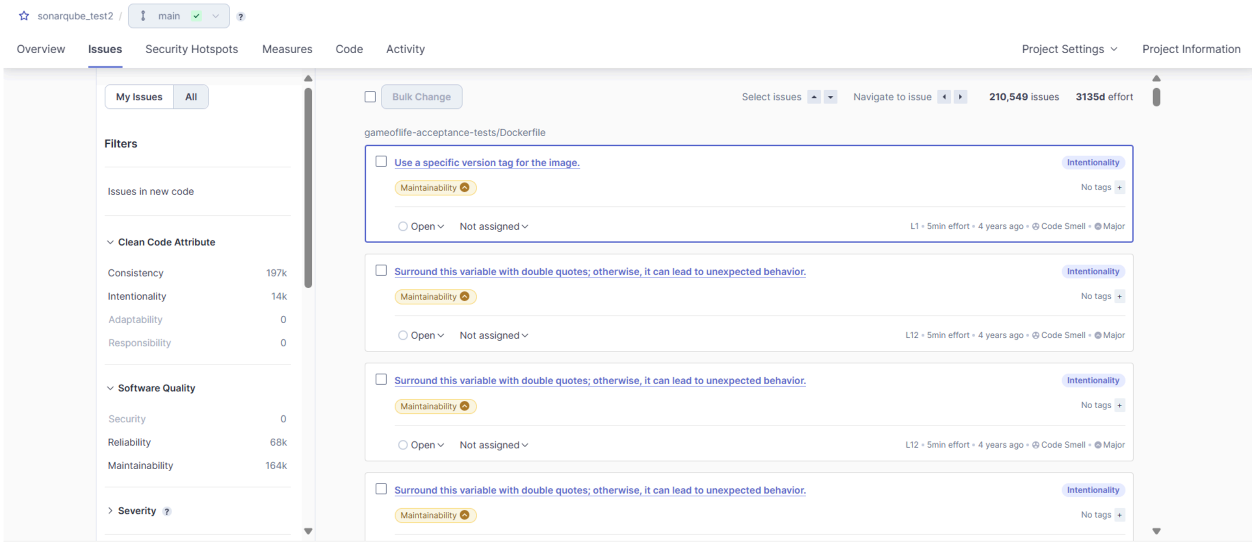
Security	Reliability	Maintainability
0 Open issues (A)	68k Open issues (C)	164k Open issues (A)
0 H · 0 M · 0 L	0 H · 47k M · 21k L	7 H · 143k M · 21k L

Accepted issues	Coverage	Duplications
0		50.6%
Valid issues that were not fixed	On **0** lines to cover.	On **759k** lines.

Security Hotspots
3 (E)

Under different tabs, check all different issues with the code

12. Code Problems-



sonarqube_test2 / main ?

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

My Issues | All

Filters

Issues in new code

Clean Code Attribute

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Software Quality

Security	0
Reliability	68k
Maintainability	164k

Severity ?

Bulk Change

Select issues · Navigate to issue · **210,549** issues · **3135d** effort

gameoflife-acceptance-tests/Dockerfile

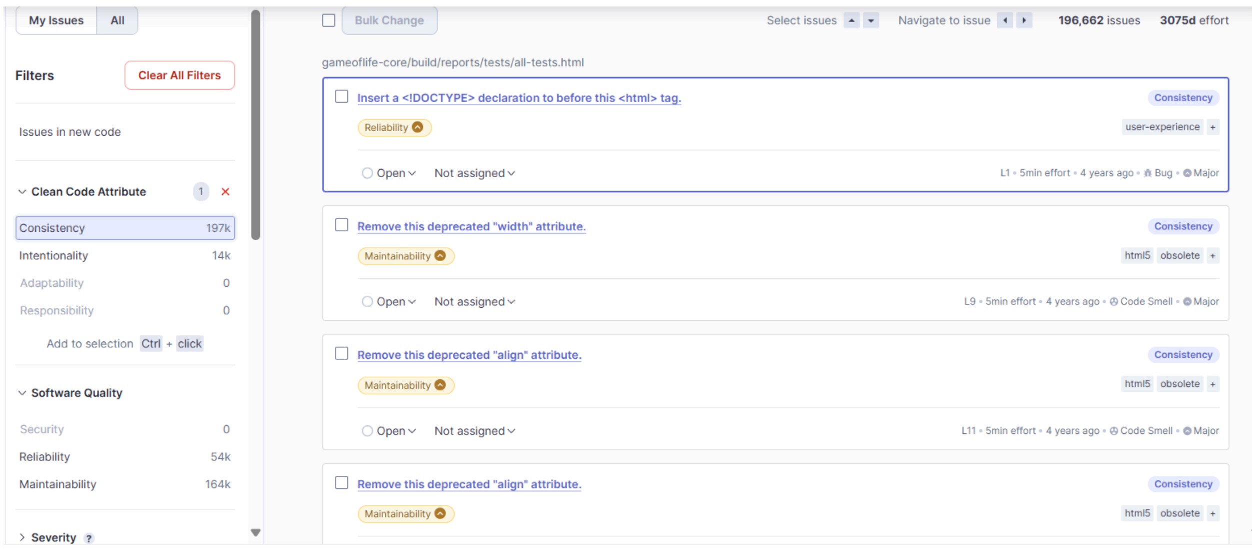
Use a specific version tag for the image.
Maintainability · Intentionality · No tags +
Open · Not assigned · L1 • 5min effort • 4 years ago • Code Smell • Major

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

Code Smells



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
- Maintainability 164k

Severity ?

Bulk Change — Select issues — Navigate to issue — 196,662 issues 3075d effort

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

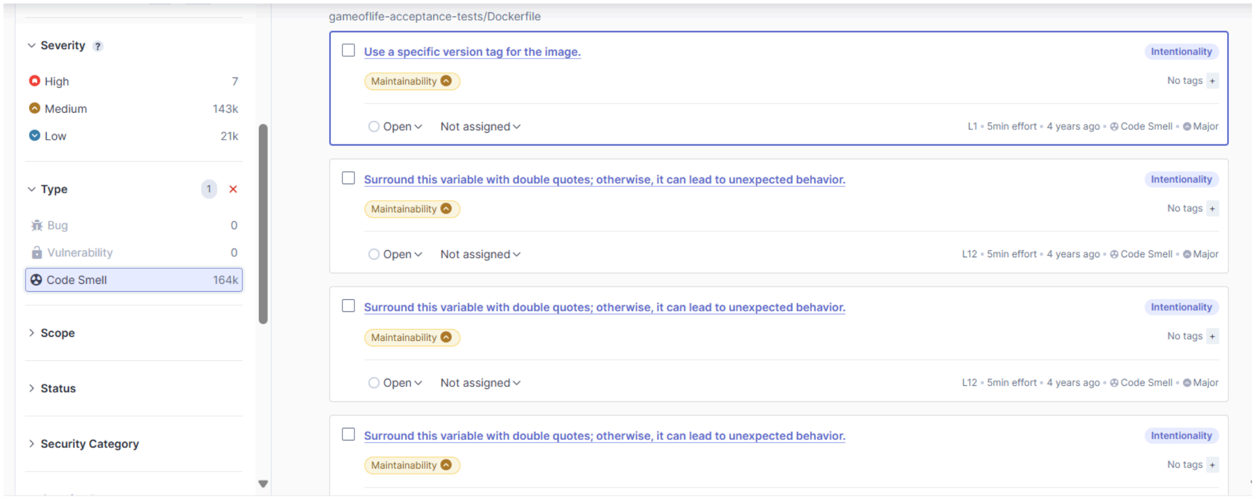
Insert a <!DOCTYPE> declaration to before this <html> tag. — Consistency — Reliability — user-experience
Open · Not assigned — L1 · 5min effort · 4 years ago · Bug · Major

Remove this deprecated "width" attribute. — Consistency — Maintainability — html5 obsolete
Open · Not assigned — L9 · 5min effort · 4 years ago · Code Smell · Major

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

Bugs



Severity ?

- High 7
- Medium 143k
- Low 21k

Type 1 ×

- Bug 0
- Vulnerability 0
- Code Smell 164k

Scope

Status

Security Category

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

Conclusion: We began the experiment by creating a new project in SonarQube and setting up a new pipeline in Jenkins with the proper configuration of the pipeline script. Then we installed the Sonar Scanner CLI, enabling Jenkins to perform code analysis on the Git repository. The pipeline can also be configured to use the installed Sonar Scanner plugin instead of the locally installed binary. Eventually, the pipeline ran successfully, with all tests passing in SonarQube.However, it took me 10 attempts in the Jenkins console to get the build right. The script had to be written in a particular format, ensuring that the credentials were added correctly without any errors. Additionally, SonarQube takes a little time to pass the tests and complete the analysis.