

## EXPERIMENT NO. 8

**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.

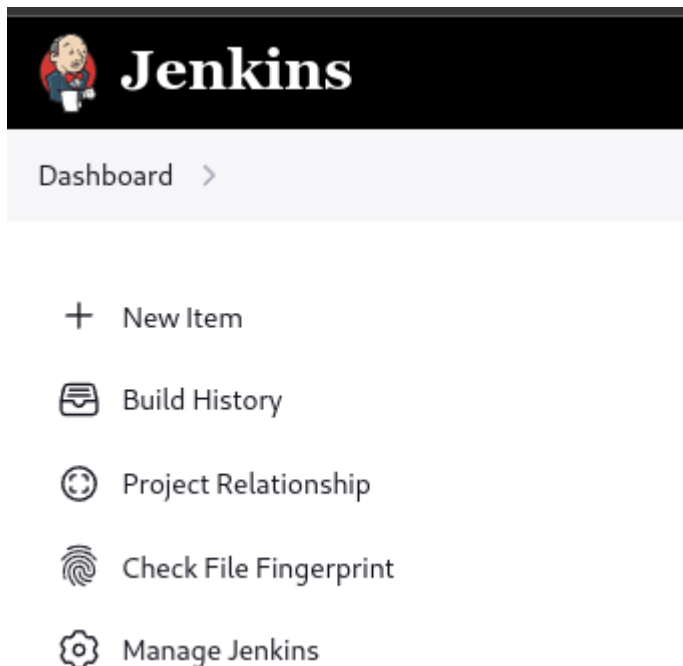
### Integrating Jenkins with SonarQube:

#### Prerequisites:

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

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

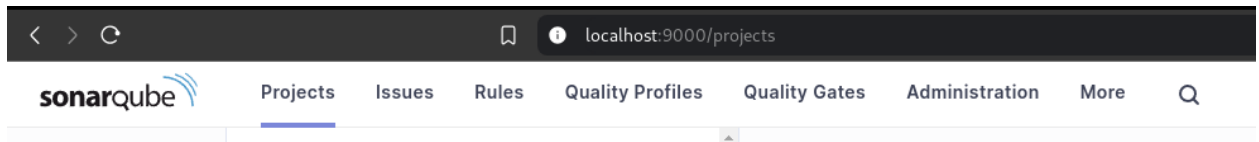
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



2. Run SonarQube in a Docker container using this command

```
quantum@machine ~$ sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

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



- Login to SonarQube using your username and password
- Create a manual project in SonarQube with the name sonarqube-test2

1 of 2

## Create a local project

Project display name \*

 ✓

Project key \*

 ✓

Main branch name \*

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

Cancel

Next

**database should be used for evaluation purposes only**

ed database will not scale, it will not support upgrading to newer versions of Sonar

Setup the project and come back to Jenkins Dashboard.

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

## New Item

Enter an item name

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.



### Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

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-linux-x64.zip> for linux OS

(<https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.2.0.4584-windows-x64.zip> for windows OS)

You will be prompted to download the zip file, download it and extract it and copy the path (absolute path) to <ROOT\_DIRECTORY>/bin/sonar-scanner

For windows path would be

C:\\Users\\<USER\_NAME>\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-cli-6.2.0.4584-windows-x64\\bin\\sonar-scanner in my case its

“C:\\Users\\ADITYA DUBEY\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-cli-6.2.0.4584-windows-x64\\bin\\sonar-scanner” //” ” for paths with space

Now we need to update the required details in the pipeline script as :

```

pipeline {
  agent any
  stages {
    stage('Cloning the GitHub Repo') {
      steps {
        git 'https://github.com/shazforiot/GOL.git'
      }
    }
    stage('SonarQube analysis') {
      steps {
        withSonarQubeEnv('sonarqube') {configuration
          sh """

```

```

"C:\\Users\\ADITYA DUBEY\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\
  sonar-scanner-cli-6.2.0.4584-windows-x64\\bin\\sonar-scanner" ^
  -D sonar.projectKey=sonarqube-test2 ^
  -D sonar.sources=. ^
  -D sonar.exclusions=vendor/**,resources/**,**/*.java ^

```

```
-D sonar.host.url=http://127.0.0.1:9000 ^  
-D sonar.login=admin ^  
-D sonar.password=my pass
```

```
""
```


```
}  
}  
}  
}
```

Script ?

```
1 pipeline {  
2   agent any  
3   stages {  
4     stage('Cloning the GitHub Repo') {  
5       steps {  
6         git 'https://github.com/shazforiot/GOL.git'  
7       }  
8     }  
9     stage('SonarQube Analysis') {  
10      steps {  
11        withSonarQubeEnv('sonarqube') {  
12          bat """  
13            "C:/Users/ADITYA DUBEY/Downloads/sonar-scanner-cli-6.2.0.4584-windows-x64/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar.bat"  
14            -D sonar.projectKey=SONARQUBE_PROJECT2 ^  
15            -D sonar.sources=. ^  
16            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^  
17            -D sonar.host.url=http://127.0.0.1:9000 ^  
18            -D sonar.login=admin ^  
19            -D sonar.password=a1b2d4e5i9j10  
20          """  
21        }  
22      }  
23    }  
24  }  
25 }  
26 }
```

## 9. Run the build

Dashboard &gt; SonarQube-exp8 &gt;

 Status Changes Build Now Configure Delete Pipeline Full Stage View Stages Rename Pipeline Syntax

Check the status of Build



SonarQube analysis for EXP 8

## Stage View



As we can see the SonarQube analysis is completed

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

### Console Output

Skipping 4,226 KB.. [Full Log](#)

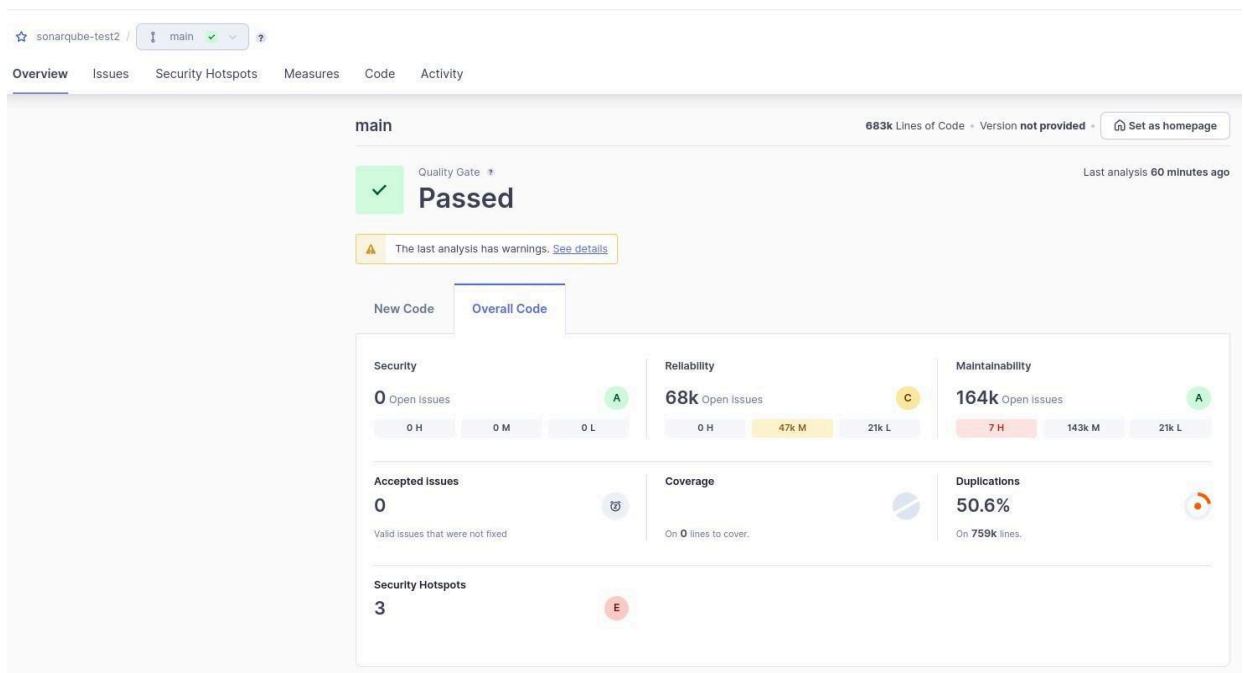
```
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 296. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 17. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 298. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 300. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep only the first 100 references.
20:11:28.619 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 225. Keep only the first 100 references.
```

```

20:11:31.504 INFO   CPD Executor CPD calculation finished (done) | time=104775ms
20:11:31.748 INFO   SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:11:33.306 INFO   Analysis report generated in 1507ms, dir size=127.2 MB
20:11:40.819 INFO   Analysis report compressed in 7511ms, zip size=29.6 MB
20:11:42.147 INFO   Analysis report uploaded in 1322ms
20:11:42.154 INFO   ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test2
20:11:42.154 INFO   Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:11:42.154 INFO   More about the report processing at http://127.0.0.1:9000/api/ce/task?id=670aec3c-6c20-460b-ad52-ec6ce041fb1f
20:11:46.802 INFO   Analysis total time: 3:42.216 s
20:11:46.835 INFO   SonarScanner Engine completed successfully
20:11:48.173 INFO   EXECUTION SUCCESS
20:11:48.296 INFO   Total time: 3:47.113s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

11. After that, check the project in SonarQube.



Under different tabs, check all different issues with the code

## 12. Code Problems-

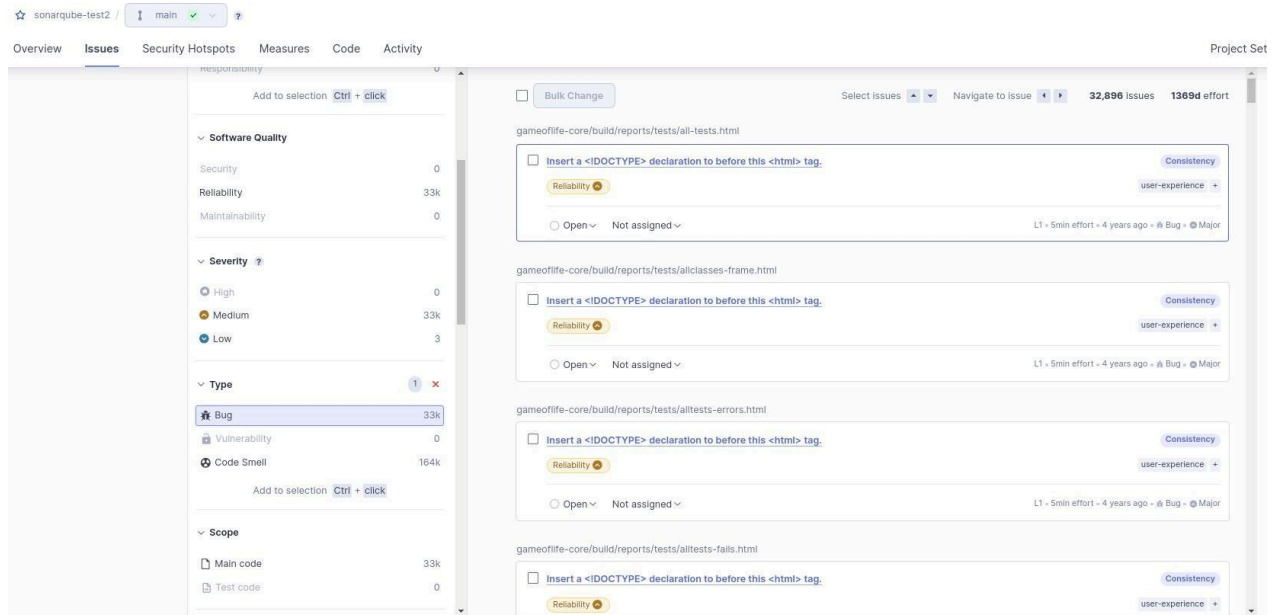
The screenshot shows the SonarQube interface for the 'sonarqube-test2' project, specifically the 'Issues' tab. The left sidebar contains filters for 'My Issues', 'All', and 'Issues in new code'. Under 'Clean Code Attribute', 'Consistency' is selected with 197k issues. Under 'Software Quality', 'Security' has 0, 'Reliability' has 54k, and 'Maintainability' has 164k. Under 'Severity', 'High' has 0, 'Medium' has 176k, and 'Low' has 21k. The main panel shows a list of issues for the file 'gameoflife-core/build/reports/tests/all-tests.html'. The first issue is 'Insert a <DOCTYPE> declaration to before this <html> tag.' with a 'Consistency' attribute and 'user-experience' tag. The second issue is 'Remove this deprecated "width" attribute.' with a 'Consistency' attribute and 'html5 obsolete' tag. The third issue is 'Remove this deprecated "align" attribute.' with a 'Consistency' attribute and 'html5 obsolete' tag. The fourth issue is 'Remove this deprecated "align" attribute.' with a 'Consistency' attribute and 'html5 obsolete' tag. The fifth issue is 'Remove this deprecated "size" attribute.' with a 'Consistency' attribute. The top right shows '190,662 Issues' and '3075d effort'.

## Code Smells

The screenshot shows the SonarQube interface for the 'sonarqube-test2' project, specifically the 'Issues' tab. The left sidebar contains filters for 'My Issues', 'All', and 'Issues in new code'. Under 'Clean Code Attribute', 'Consistency' is selected with 197k issues. Under 'Software Quality', 'Security' has 0, 'Reliability' has 54k, and 'Maintainability' has 164k. Under 'Severity', 'High' has 0, 'Medium' has 176k, and 'Low' has 21k. The main panel shows a list of issues for the file 'gameoflife-acceptance-tests/Dockerfile'. The first issue is 'Use a specific version tag for the image.' with an 'Intentionality' attribute and 'No tags' tag. The second issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with an 'Intentionality' attribute and 'No tags' tag. The third issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with an 'Intentionality' attribute and 'No tags' tag. The fourth issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with an 'Intentionality' attribute and 'No tags' tag. The top right shows '164,034 Issues' and '1708d effort'.

## Bugs





**Conclusion:** We began the experiment with creating a new project in SonarQube and setting up a new Pipeline in Jenkins with proper configuration of pipeline script. Then we installed Sonar Scanner CLI so that jenkins can do code analysis of Git Repository. We can also configure the pipeline to use the installed Sonar Scanner plugin instead of locally installed Binary of Sonar Scanner. The pipeline ran successfully with all tests passed in SonarQube