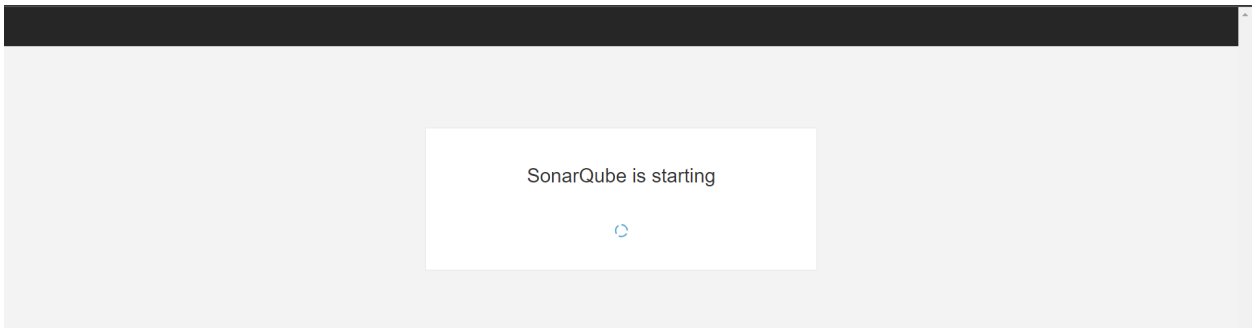
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST:

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

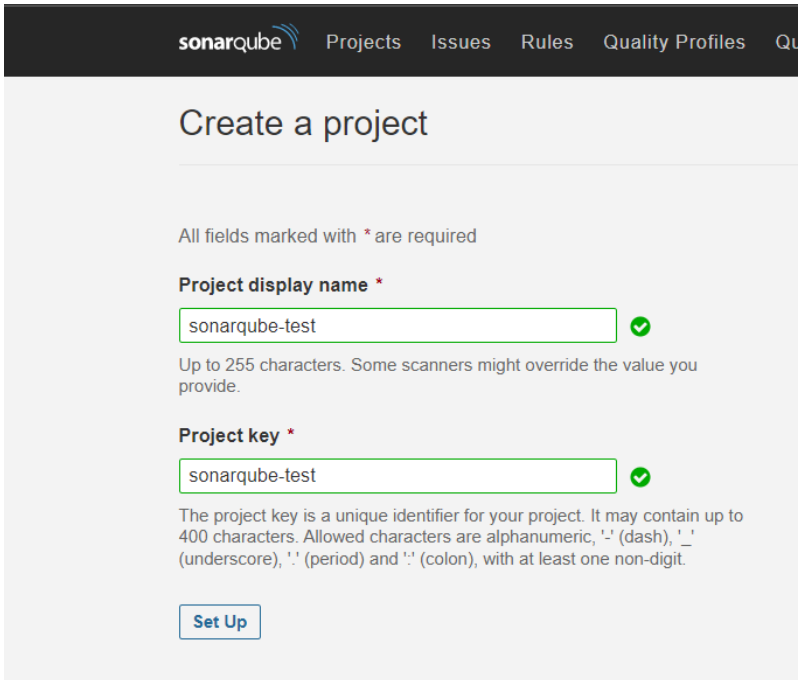
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Mikil> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:lates
t
5d39cfd39cc2dcc0bdcad5f0111585930bda84d835a8c79f0ea0377c50a1de70
PS C:\Users\Mikil>
```

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



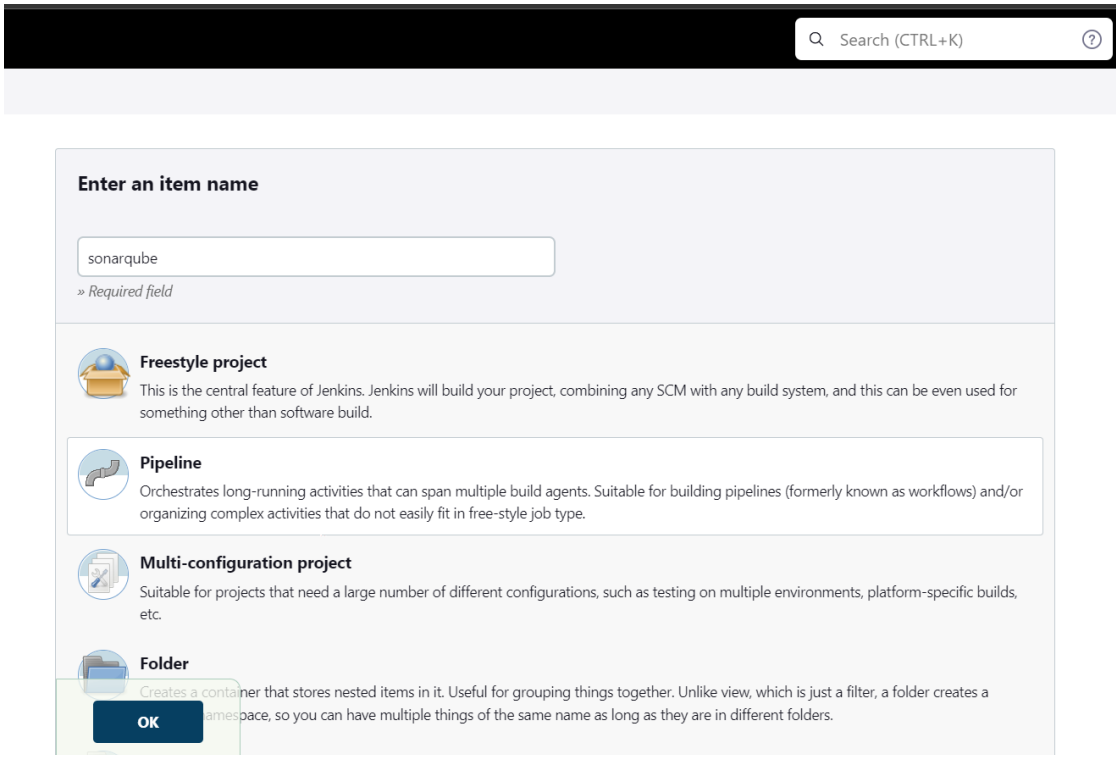

4. Login to SonarQube using username admin and password admin.

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




Set up the project and come back to Jenkins Dashboard.

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




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
"/c/ProgramData//Jenkins/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqub
e/bin//sonar-scanner \
      -D sonar.login=admin \
      -D sonar.password=mikami \
      -D sonar.projectKey=demoapp-project \
      -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. Run The Build.

# Pipeline sonarqube

## Stage View

| | Cloning the GitHub Repo | SonarQube analysis |
|---|---|---|
| Average stage times: (Average full run time: ~4min 16s) | 11s | 4min 4s |
| #1 Oct 07 18:53 No Changes | 11s | 4min 4s |

## Permalinks

- Last build (#1), 13 min ago
- Last stable build (#1), 13 min ago
- Last successful build (#1), 13 min ago
- Last completed build (#1), 13 min ago

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

Search (CTRL+K) admin log out

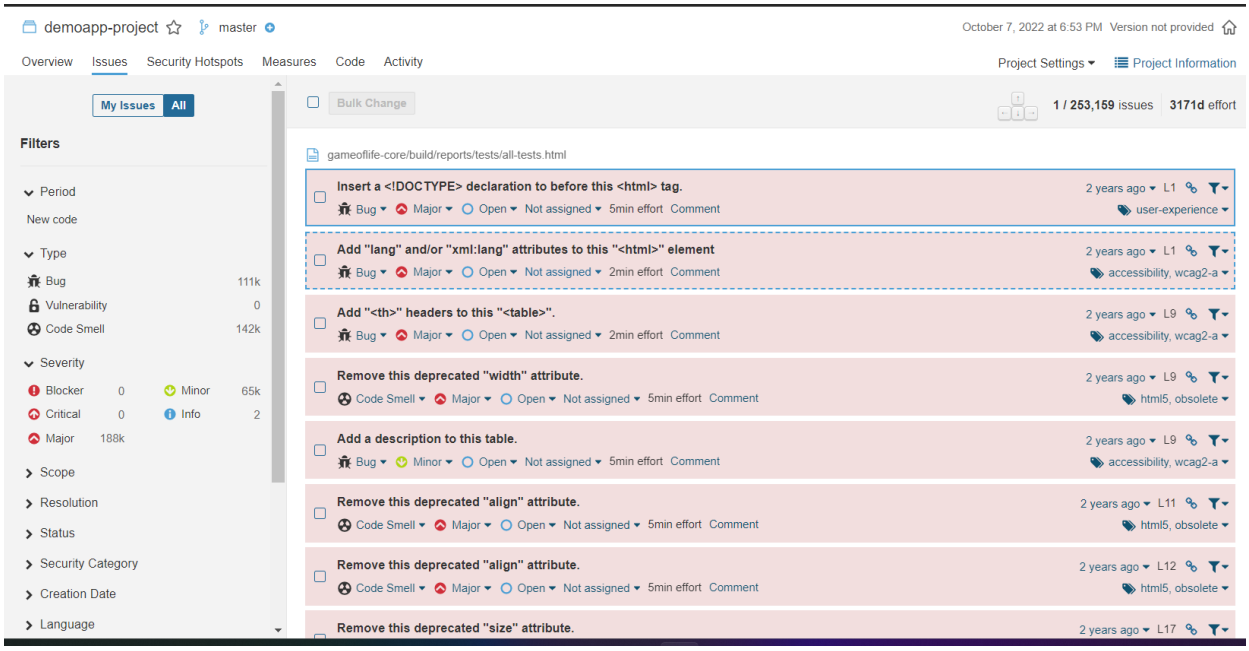
## Console Output

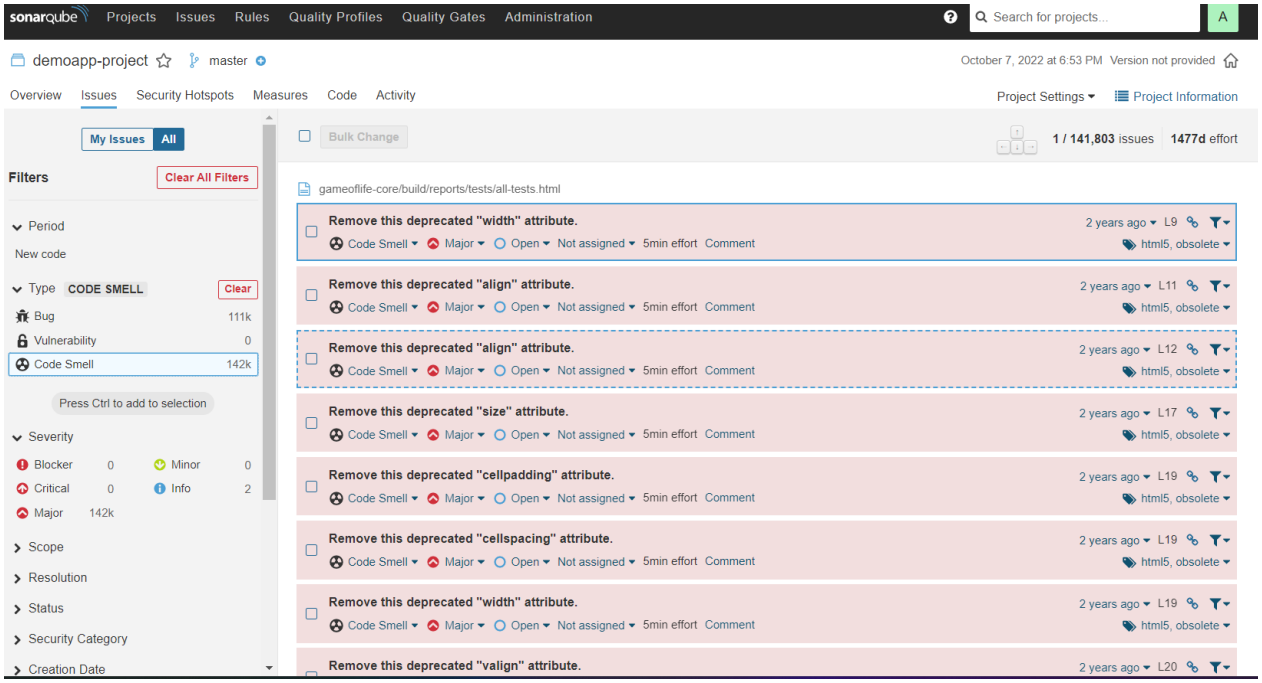
```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/GOL.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.37.3.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
```

```
WARN: Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/engine/util/CompoundVariable.html for block at line
17. Keep only the first 100 references.
WARN: Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/engine/util/CompoundVariable.html for block at line
151. Keep only the first 100 references.
WARN: Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/engine/util/CompoundVariable.html for block at line
610. Keep only the first 100 references.
WARN: Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/engine/util/CompoundVariable.html for block at line
74. Keep only the first 100 references.
INFO: CPD Executor CPD calculation finished (done) | time=82320ms
INFO: Analysis report generated in 1968ms, dir size=129.8 MB
INFO: Analysis report compressed in 8183ms, zip size=29.8 MB
INFO: Analysis report uploaded in 3365ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=demoapp-project
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://127.0.0.1:9000/api/ce/task?id=AYOyoTWoSTNws3yuswAt
INFO: Analysis total time: 3:57.572 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 3:59.581s
INFO: Final Memory: 16M/88M
INFO: ------------------------------------------------------------------------
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

10. After that, check the project in SonarQube.
Under different tabs, check all the different issues with the code.

11. Code Problems -




In this way, we have created a CI/CD Pipeline with Jenkins and integrated it with SonarQube to find issues in the code like bugs, code smells, duplicates, cyclomatic complexities, etc.