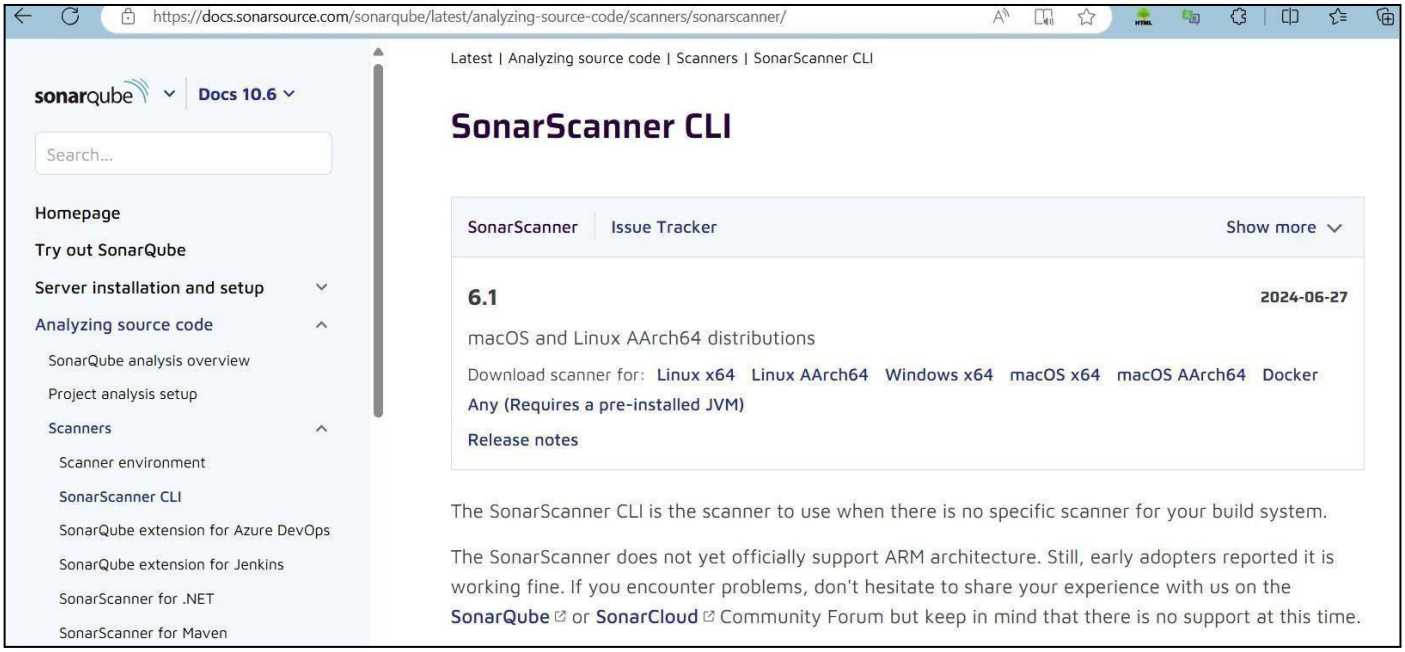
ADVANCE DEVOPS EXP 8

NAME: PRAJYOT SHINDE 57 / D15A

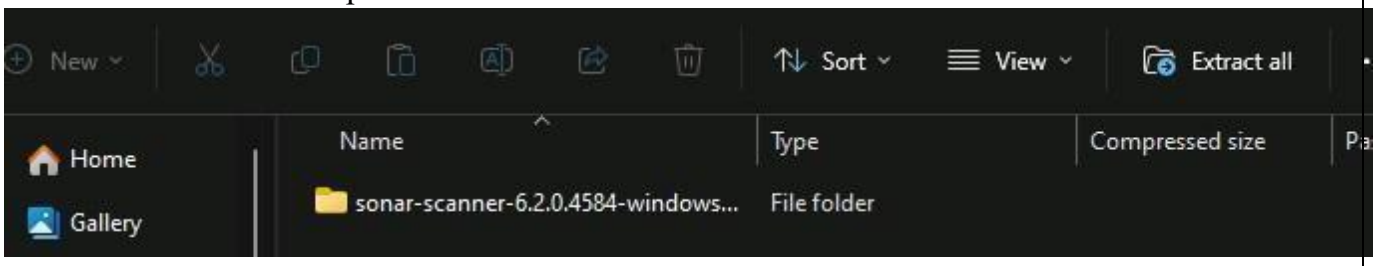
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner https://docs.sonarsource.com/sonarqube/latest/analyzingsource-code/scanners/sonarscan



ner/ Visit this link and download the sonarqube scanner CLI.

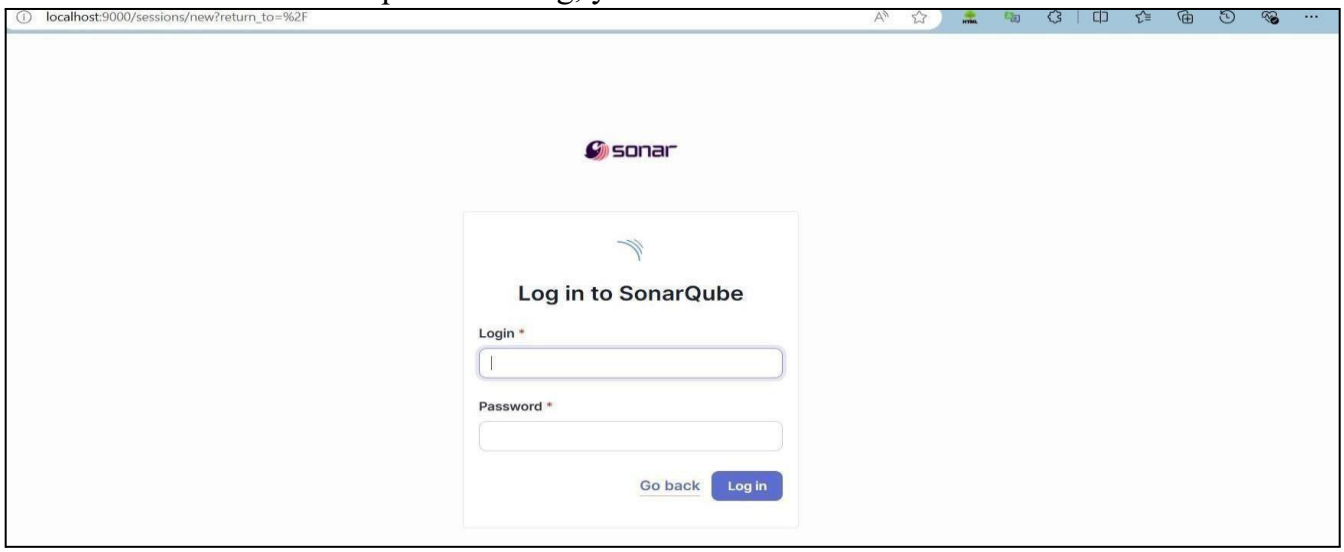
Extract the downloaded zip file in a folder.



1. Install sonarqube image Command: **docker pull sonarqube**

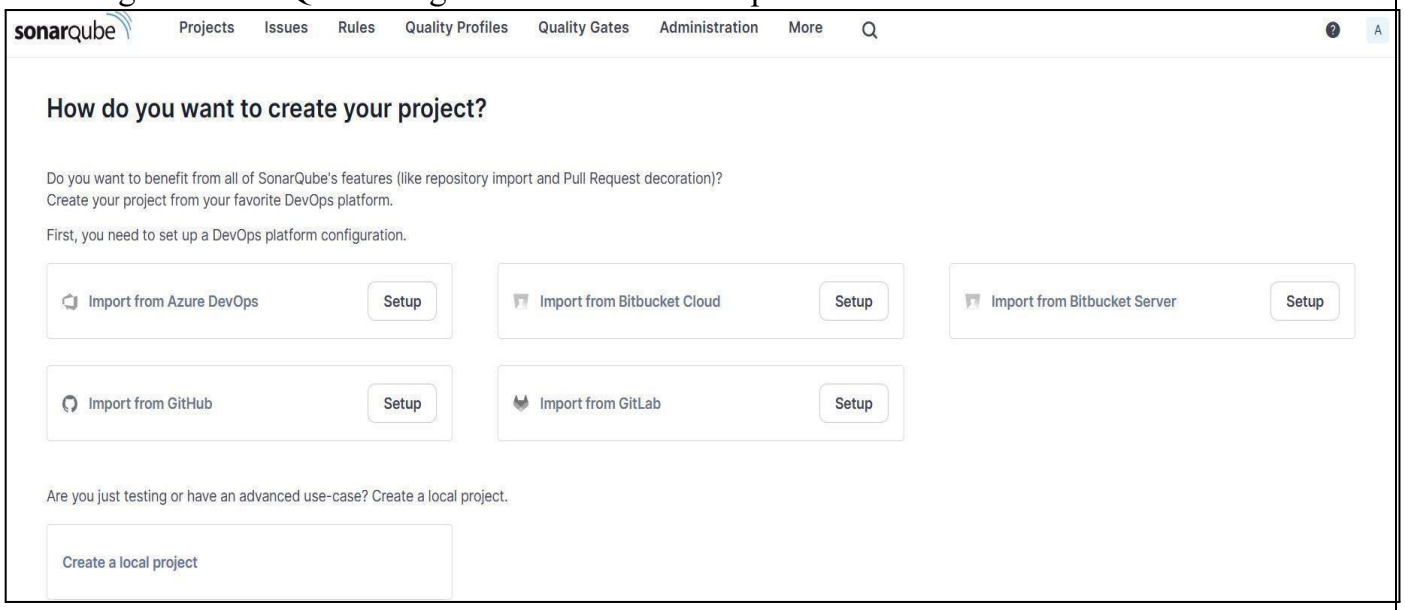
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindo

PS C:\Users\Soham Satpute> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

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

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

1 of 2

Create a local project

Project display name *

Sonarqube-test

Project key *

Sonarqube-test

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 you focus att
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.

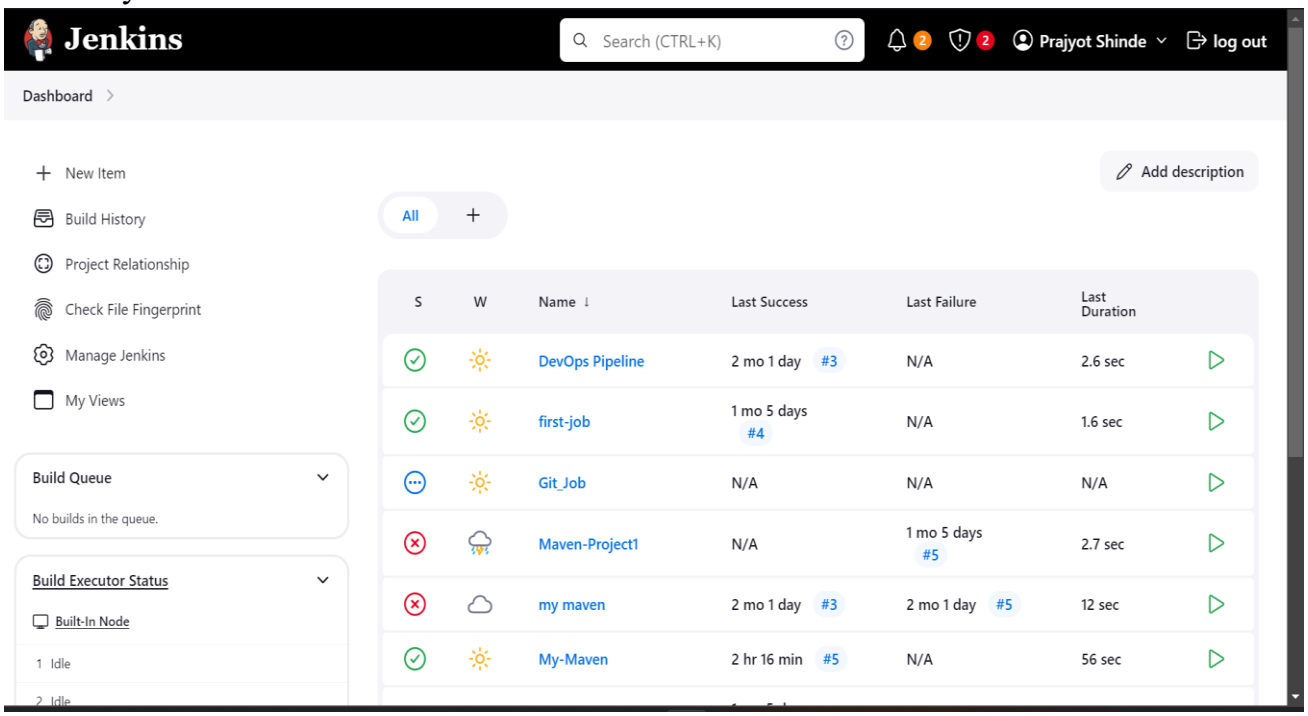
Recommended for projects following regular versions or releases.

Define a specific setting for this project

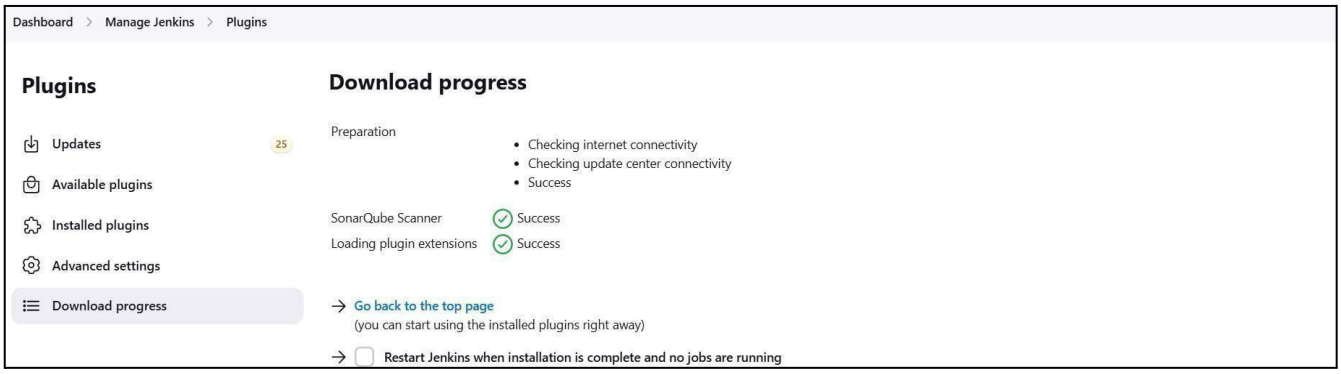
Previous version

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

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



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



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**

8. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

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

Dashboard > Manage Jenkins > Tools

SonarQube Scanner installations Edited

Add SonarQube Scanner

SonarQube Scanner

Name

SonarQube

Install automatically ?

Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer

Add SonarQube Scanner

Save Apply

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

Dashboard > All > New Item

New Item

Enter an item name

AdDevops-8

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.

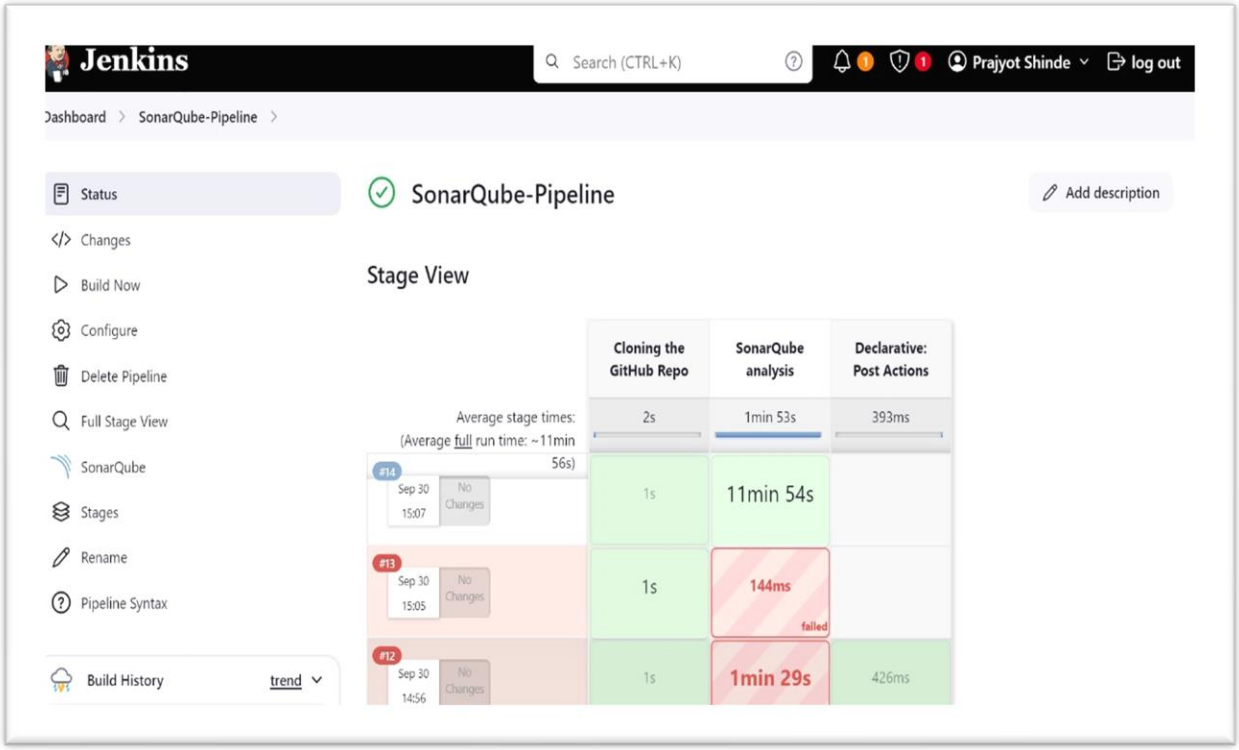
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

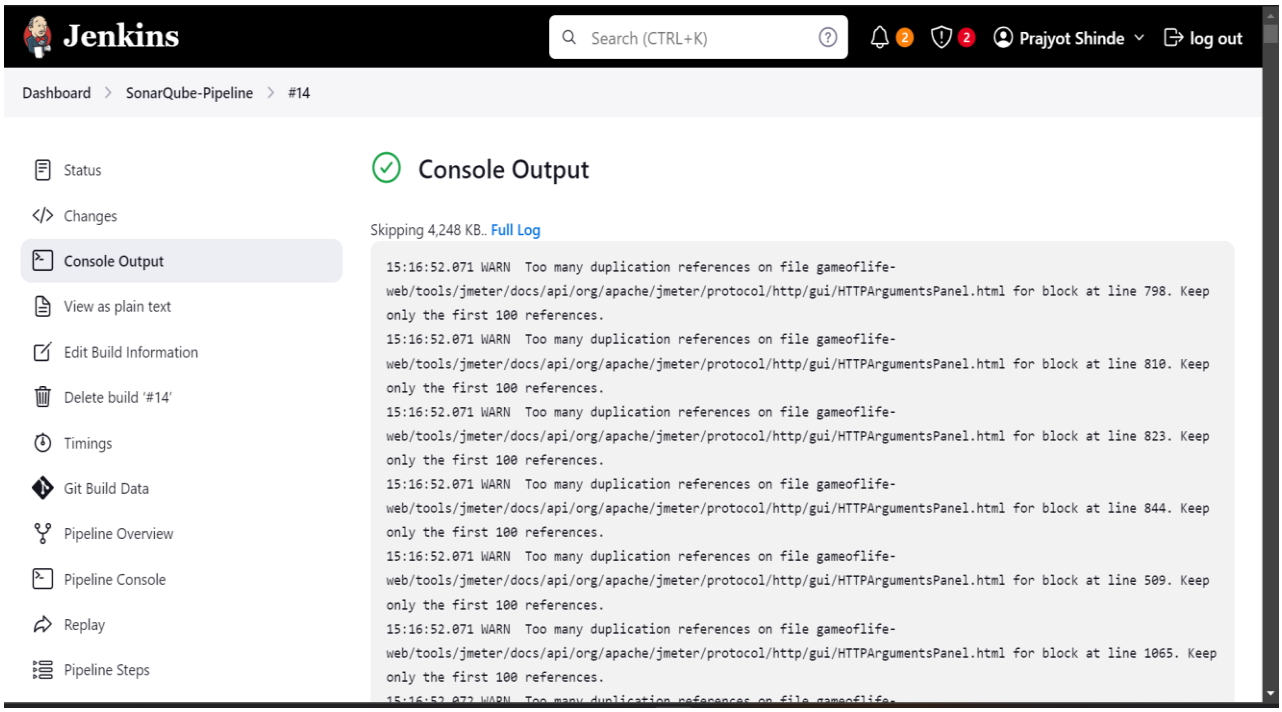
10. Under Pipeline script, enter the following:

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

  analysis') {

    withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenk
    i ns>') { 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/)
      """
    }
  }
}
```

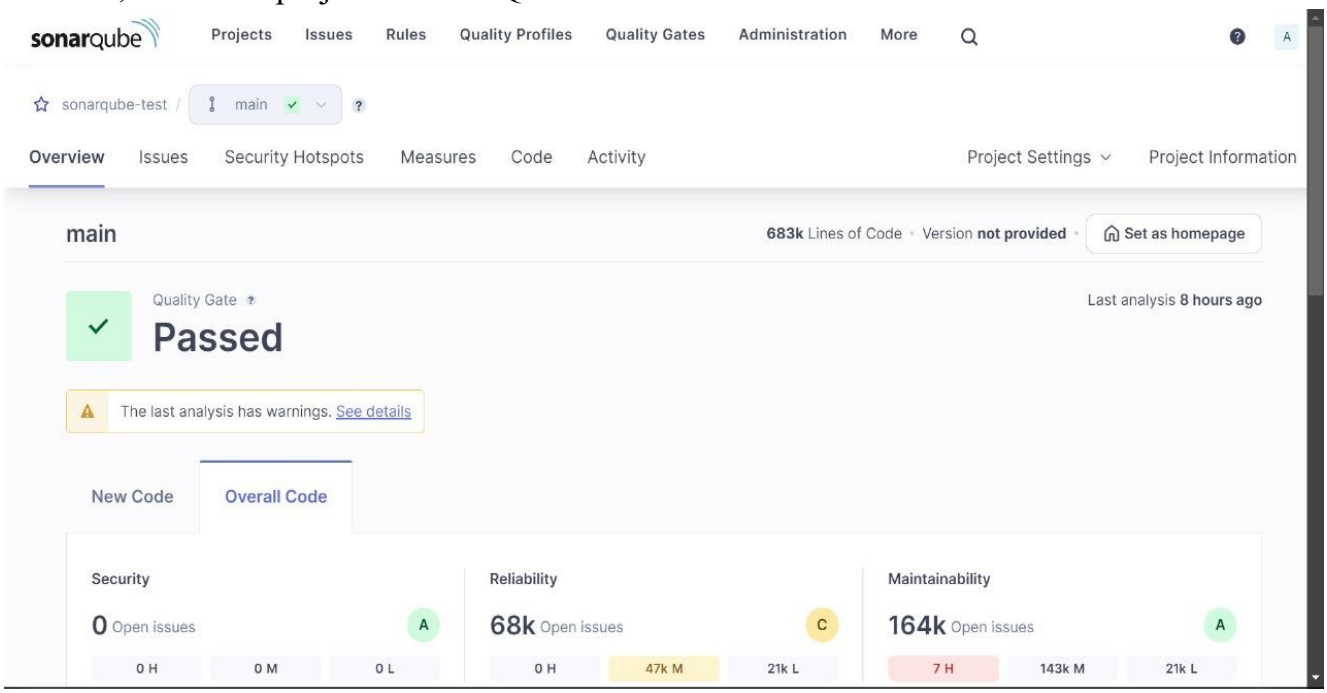
It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.



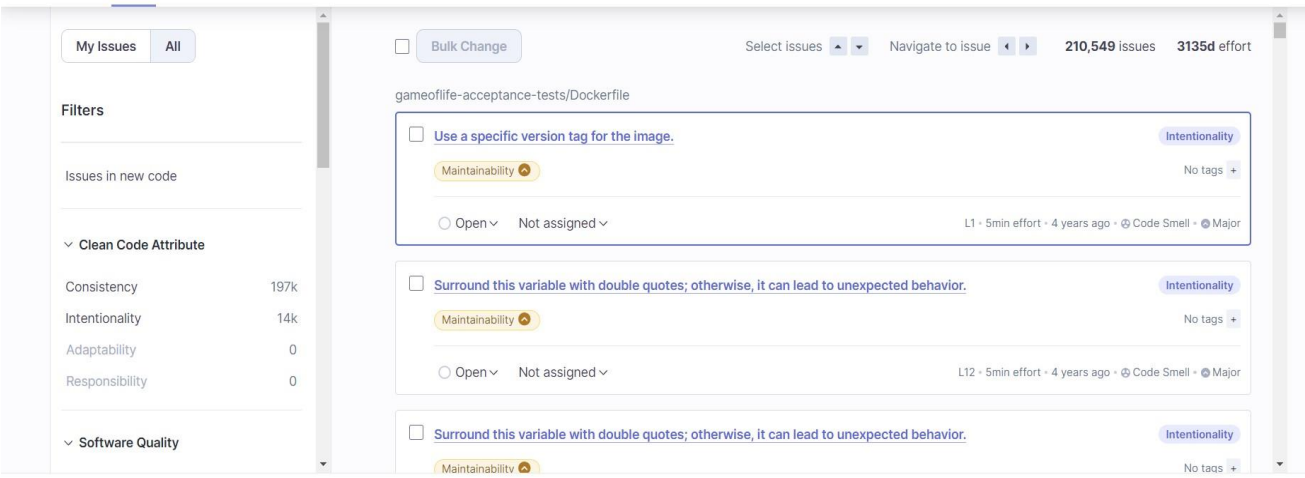
11.Check console



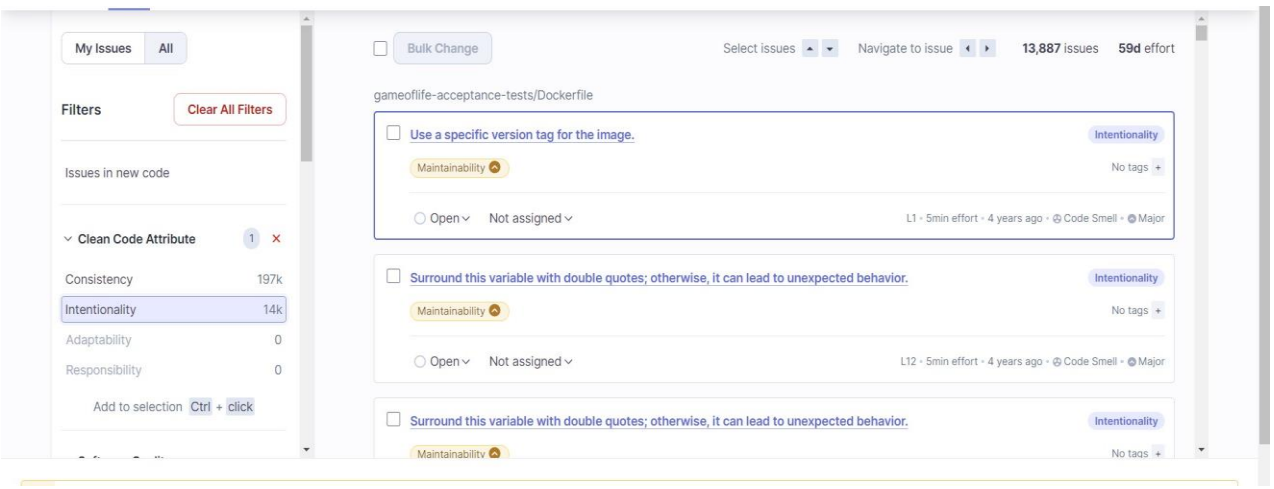
12.Now, check the project in SonarQube:



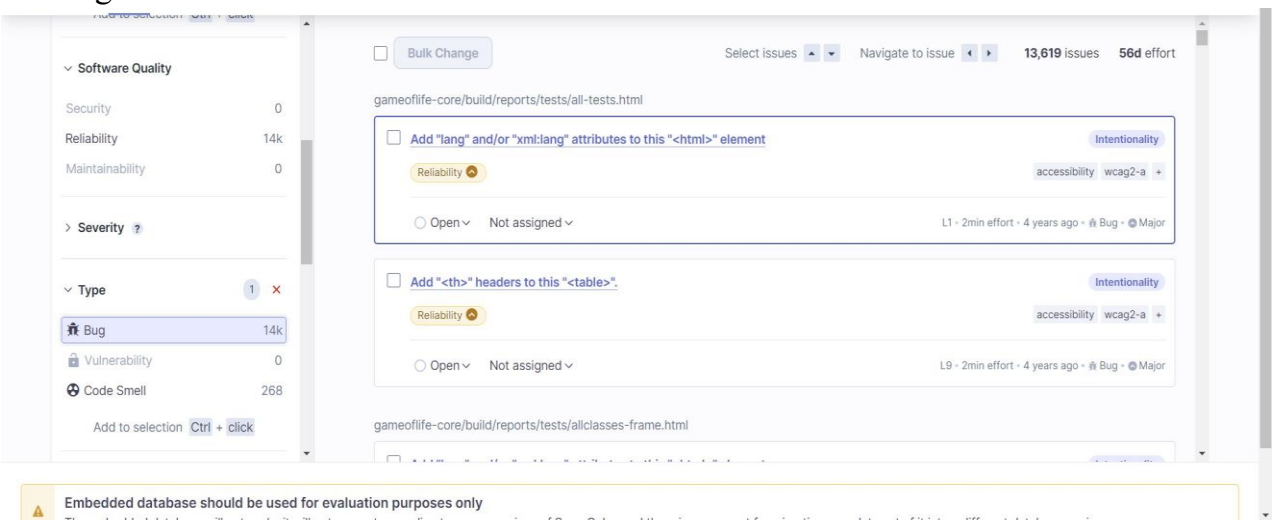
13.code problems consistency:



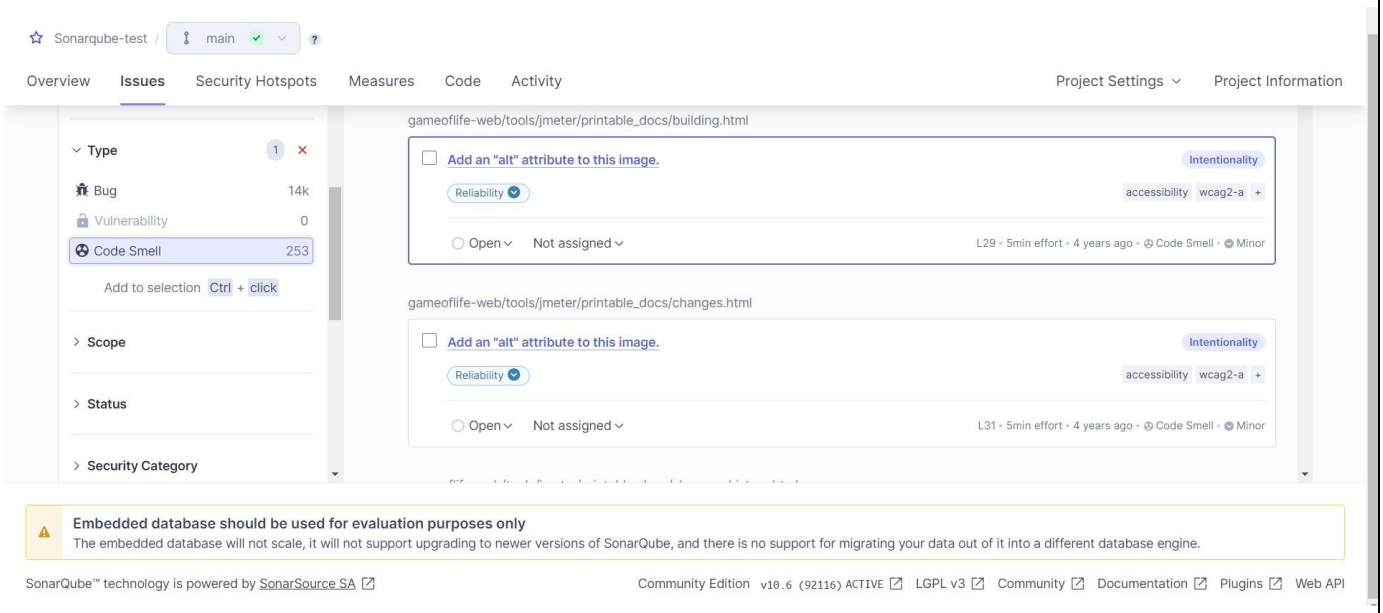
14.Intentionality:



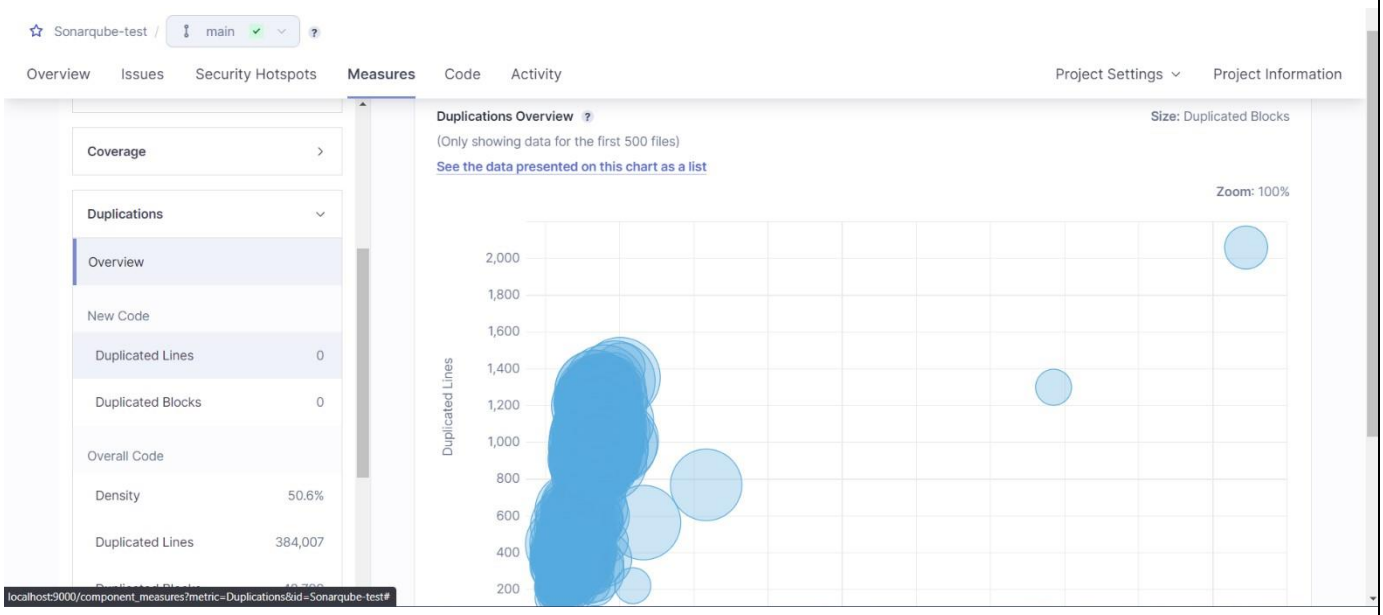
15.Bugs



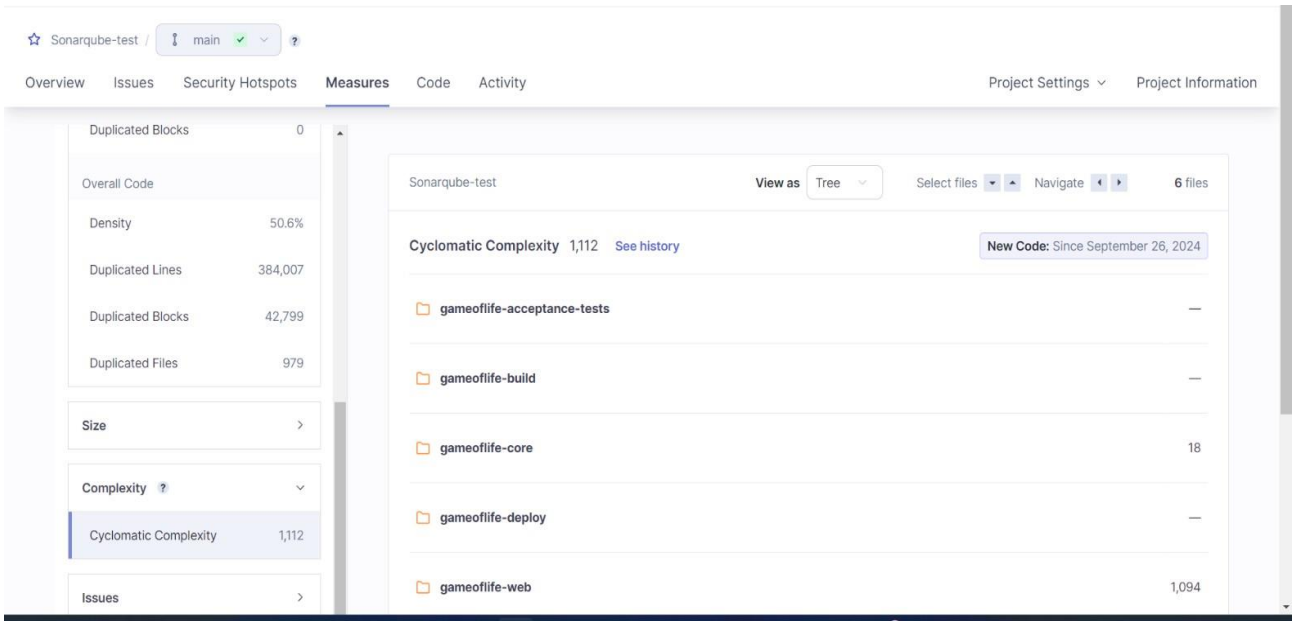
Code smells:



Duplications:



Cyclomatic Complexities:



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