

## ADVANCE DEVOPS EXP 8

**Name:**Manav punjabi

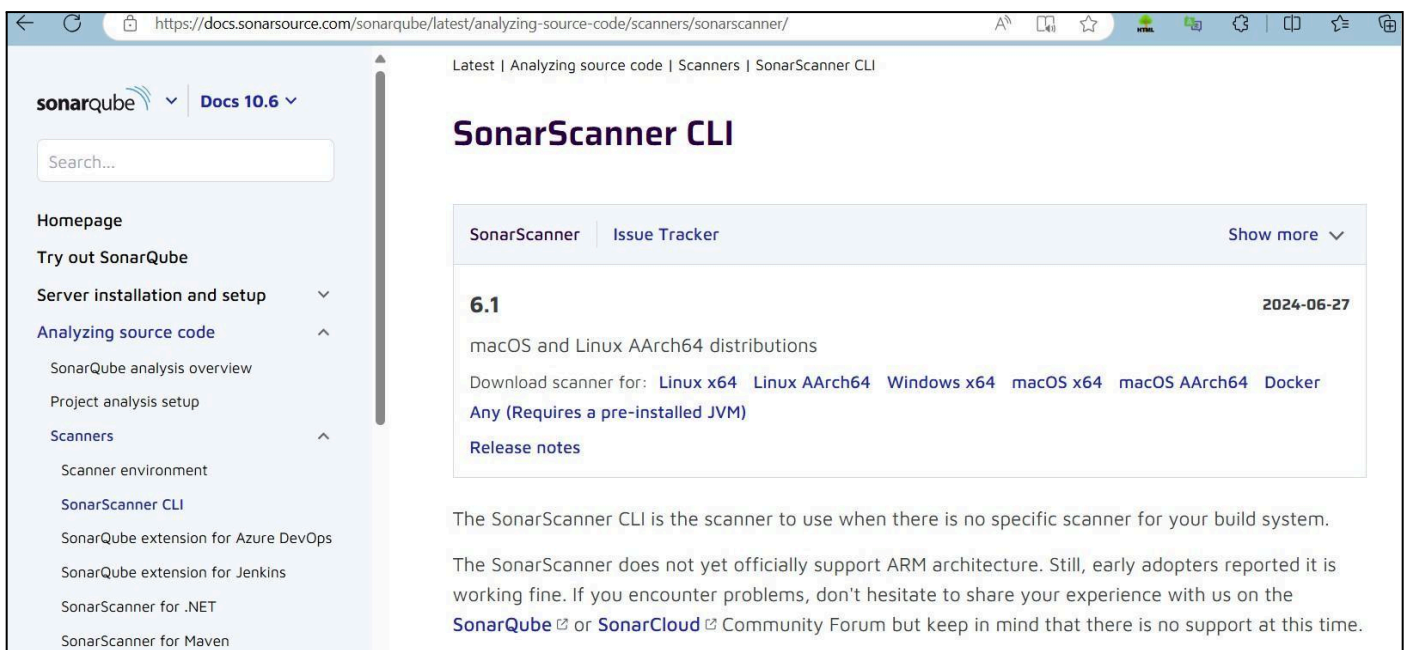
**Class:**D15A

**Roll No:**45

**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

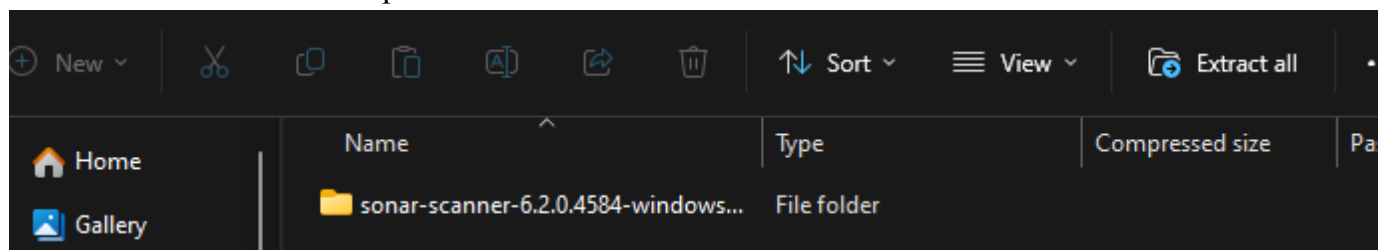
<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>



The screenshot shows the SonarScanner CLI documentation page. The left sidebar contains a navigation menu with links to SonarQube, Docs 10.6, and various installation and setup guides. The main content area is titled "SonarScanner CLI" and includes a version number "6.1" and a date "2024-06-27". It lists supported operating systems: macOS and Linux AArch64 distributions. Below this, it provides download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, and Docker. A "Release notes" link is also present. The text explains that the SonarScanner CLI is used when there is no specific scanner for the build system and mentions that it does not yet officially support ARM architecture.

[ner/](#) Visit this link and download the sonarqube scanner CLI.

Extract the downloaded zip file in a folder.

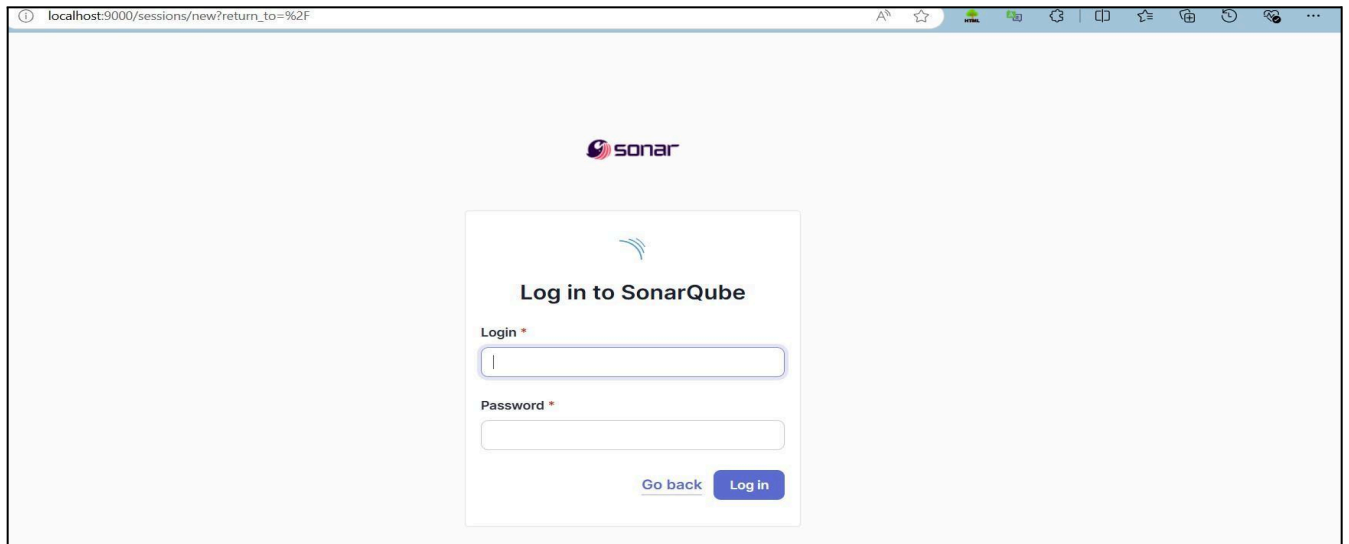


1. Install sonarqube image

Command: **docker pull**  
**sonarqube**

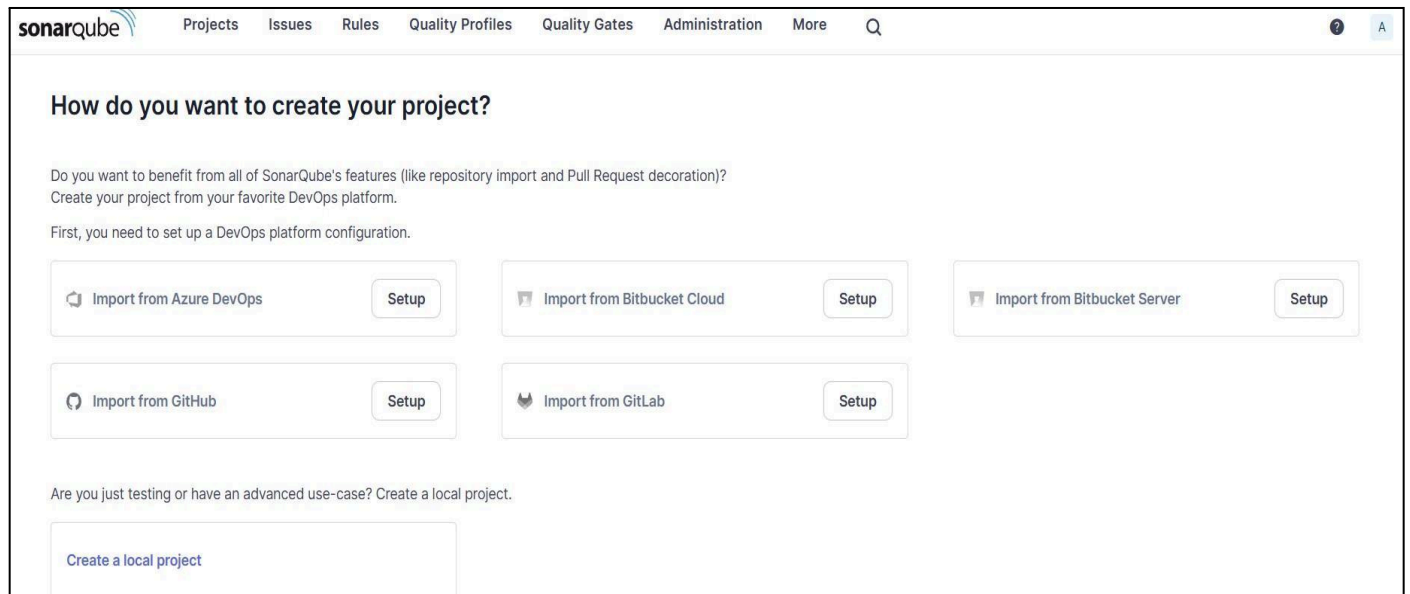
```
C:\Windows\System32>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

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

1 of 2

## Create a local project

**Project display name \***


sonarqube ✓

**Project key \***

sonarqube ✓

**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 attention on the code that has changed since the previous version. 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.

The screenshot shows the Jenkins Dashboard at localhost:8080. The left sidebar contains links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. The main area displays a table of build history with columns for status (S), weather icon (W), name, last success, last failure, and last duration. The table lists five builds: 'Devops Pipeline', 'devops\_exp6\_pipeline', 'maven\_exp\_6', 'maven\_project', and 'myNewJob'. Below the table, there are filters for 'Icon', 'S', 'M', and 'L'. On the left, there are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle', '2 Idle', and 'Node\_1 (offline)').

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Devops Pipeline	1 mo 13 days #4	N/A	0.61 sec
✓	☀	devops_exp6_pipeline	24 days #1	N/A	2.2 sec
✓	☁	maven_exp_6	17 days #13	17 days #12	9.2 sec
✗	☁	maven_project	1 mo 13 days #3	1 mo 7 days #10	12 sec
✓	☀	myNewJob	24 days #1	N/A	0.49 sec

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

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. The 'Plugins' section has a search bar with 'sonarq' entered. Below the search bar, there is a table of available plugins. The table has columns for 'Install', 'Name', and 'Released'. The first entry is 'SonarQube Scanner 2.17.2', which is currently not installed. The description for this plugin states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' The 'Released' column shows '6 mo 29 days ago'.

Install	Name	Released
<input type="checkbox"/>	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

The screenshot shows the 'Download progress' page in Jenkins. The left sidebar contains links for 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area displays the progress of the installation. It shows 'Preparation' with steps: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this, it shows 'SonarQube Scanner' with a 'Success' status and 'Loading plugin extensions' with a 'Success' status. At the bottom, there are two links: 'Go back to the top page (you can start using the installed plugins right away)' and 'Restart Jenkins when installation is complete and no jobs are running'.

**Download progress**

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner **Success**

Loading plugin extensions **Success**

→ [Go back to the top page](#)  
(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

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>



The screenshot shows the 'Manage Jenkins' system configuration page for SonarQube Servers. It features a form with three main sections: 'Name', 'Server URL', and 'Server authentication token'. The 'Name' field contains 'sonarqube'. The 'Server URL' field contains 'http://localhost:9000'. The 'Server authentication token' dropdown menu is set to '- none -'. There is a '+ Add' button and an 'Advanced' dropdown menu at the bottom.

Name ✕

sonarqube

Server URL

Default is http://localhost:9000

http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

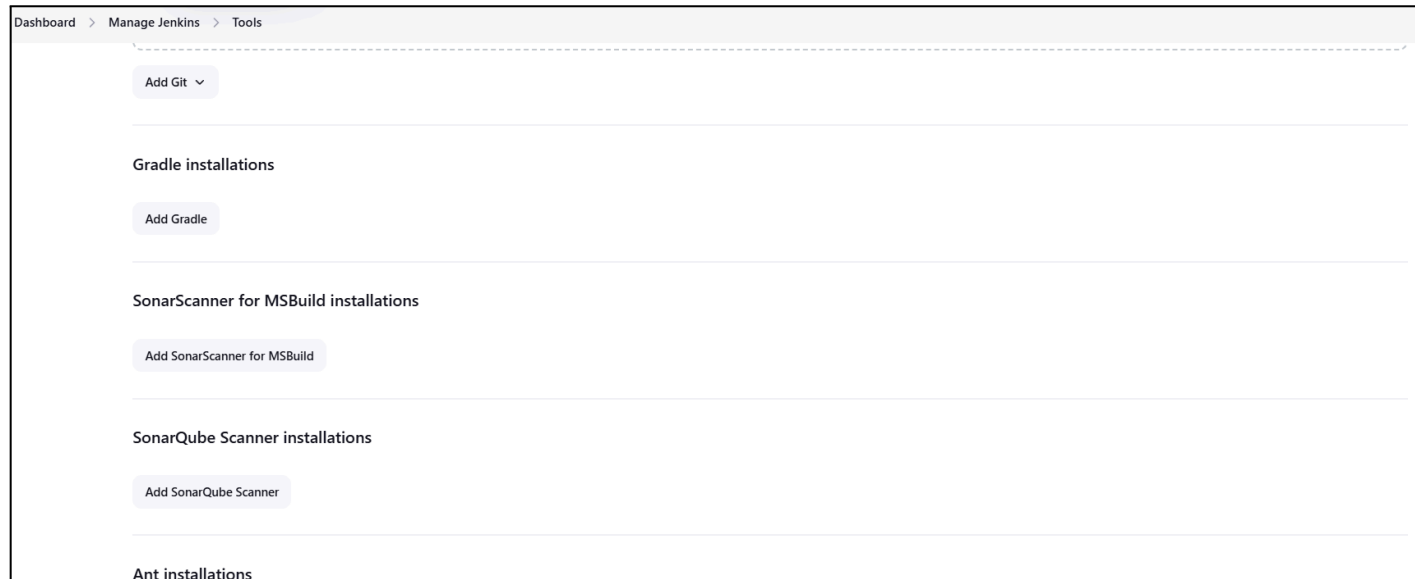
- none - ▼

+ Add ▼

Advanced ▼

8. Search for SonarQube Scanner under Global Tool Configuration.  
Choose the latest configuration and choose Install automatically.

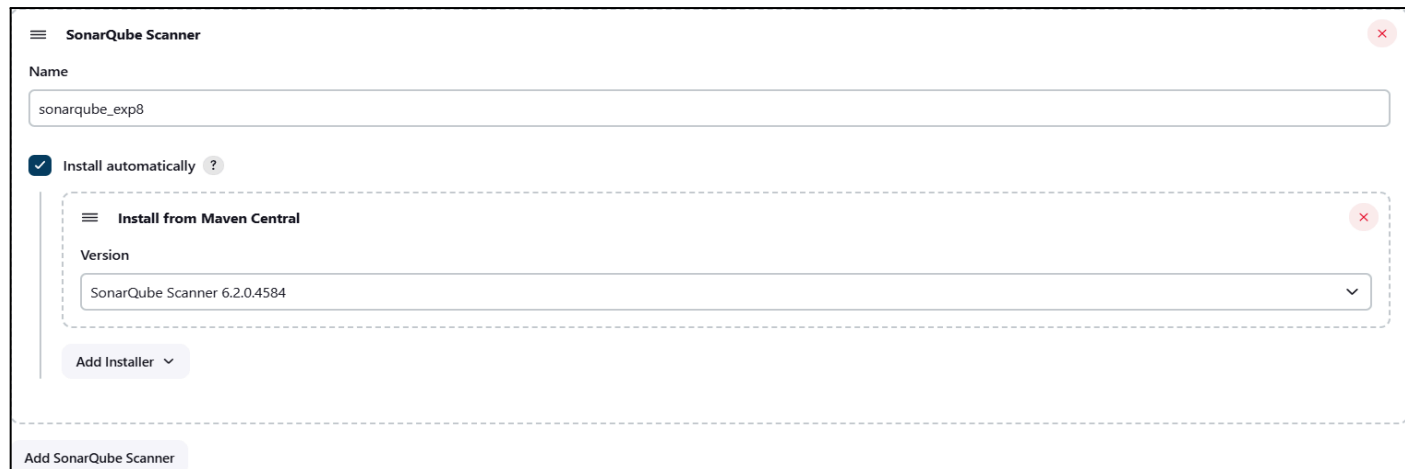
**Dashboard > Manage Jenkins > Tools**



The screenshot shows the Jenkins 'Tools' configuration page. At the top, there is a breadcrumb trail: 'Dashboard > Manage Jenkins > Tools'. Below this, there are several sections, each with an 'Add' button and a dropdown arrow:

- Add Git** (dropdown arrow)
- Gradle installations**
  - Add Gradle**
- SonarScanner for MSBuild installations**
  - Add SonarScanner for MSBuild**
- SonarQube Scanner installations**
  - Add SonarQube Scanner**
- Ant installations**

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



The screenshot shows the configuration form for the 'SonarQube Scanner' tool. The form has a title bar with a hamburger menu icon, the text 'SonarQube Scanner', and a close button (X). The form contains the following fields and options:

- Name:** A text input field containing 'sonarqube\_exp8'.
- Install automatically:** A checkbox that is checked, followed by a help icon (?).
- Install from Maven Central:** A dashed box containing:
  - Version:** A dropdown menu showing 'SonarQube Scanner 6.2.0.4584'.
- Add Installer:** A button with a dropdown arrow.
- Add SonarQube Scanner:** A button at the bottom of the form.


the “Install automatically” option.

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


### 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.

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_Jenki
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.

Definition

Pipeline script

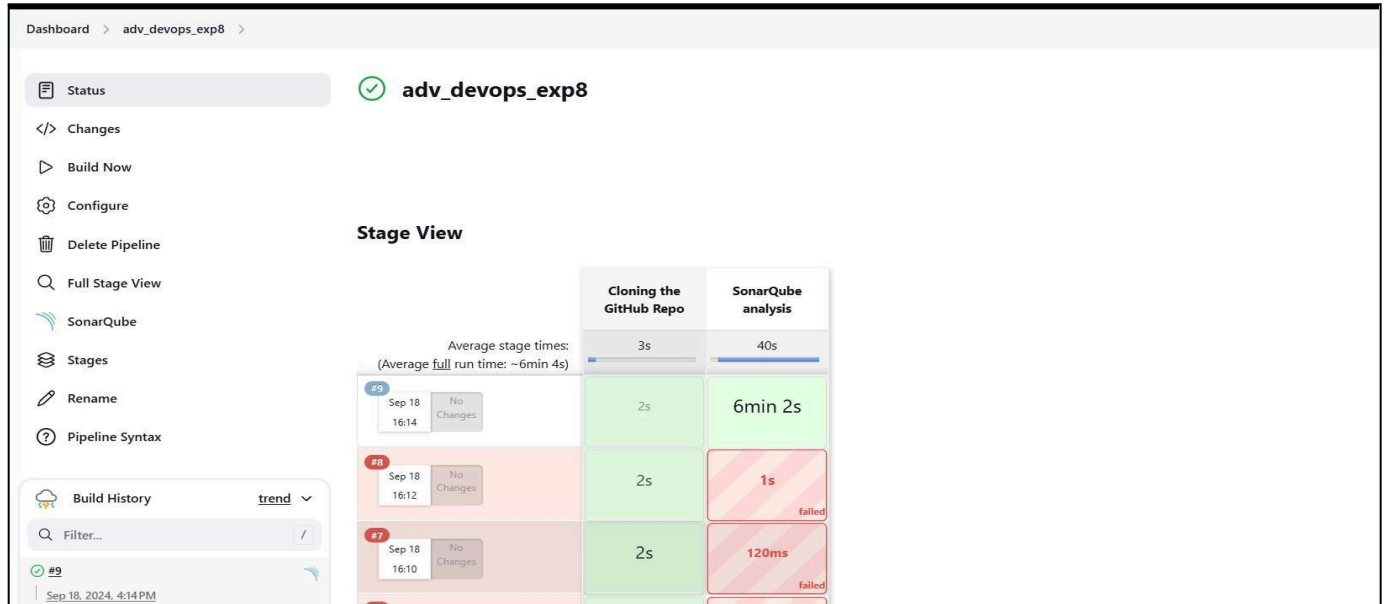
Script ?

```
1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5
6   stage('SonarQube analysis') { withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
7     sh """
8       <PATH_TO_SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
9       -D sonar.login=admin \
10      -D sonar.password=admin> \
11      -D sonar.projectKey=sonarqube \
12      -D sonar.exclusions=vendor/**,resources/**,/**/*.java \
13      -D sonar.host.url=http://localhost:9000
14      """
15    }
16  }
17 }
18
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

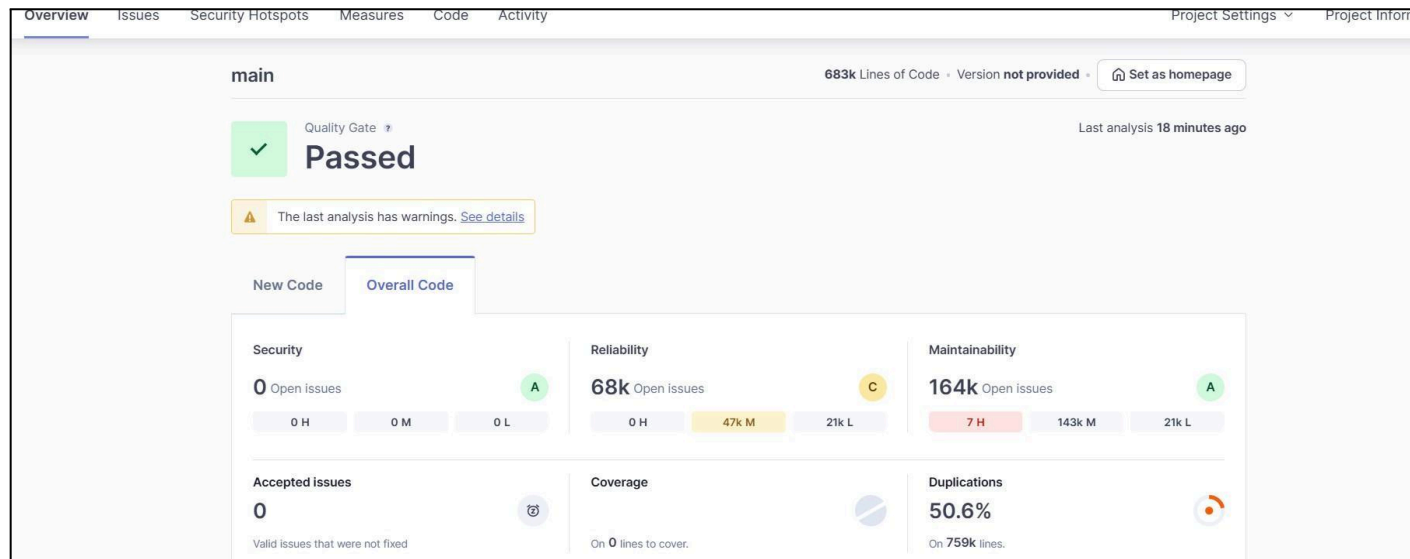
## 11. Build project



## 12. Check console

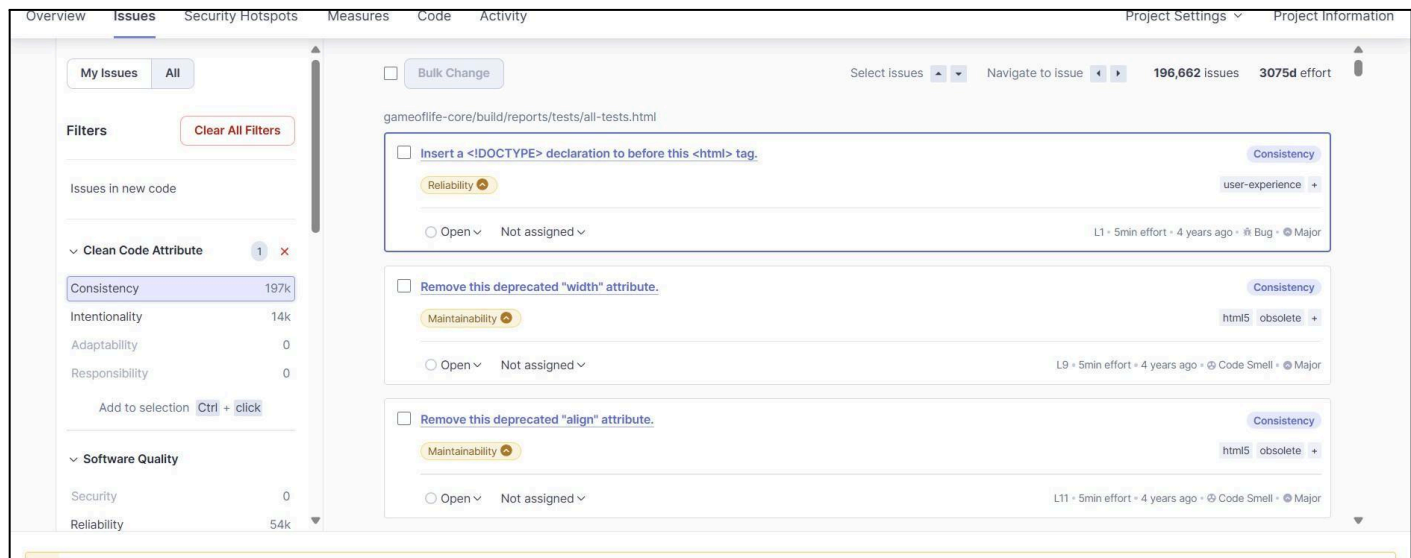


### 13. Now, check the project in SonarQube



### 14. Code Problems

- Consistency



● Intentionality

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project Settings ▾Project Information

My IssuesAll

Filters

Clear All Filters

Issues in new code

Clean Code Attribute1 ✕

Consistency197k

Intentionality14k

Adaptability0

Responsibility0

Add to selectionCtrl + click

Software Quality

Security0

Reliability14k

☐Bulk Change

Select Issues ▴ ▾

Navigate to issue ◀ ▶

13,887 issues

59d 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

☐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

- Bugs

The screenshot shows a bug tracking interface with the following details:

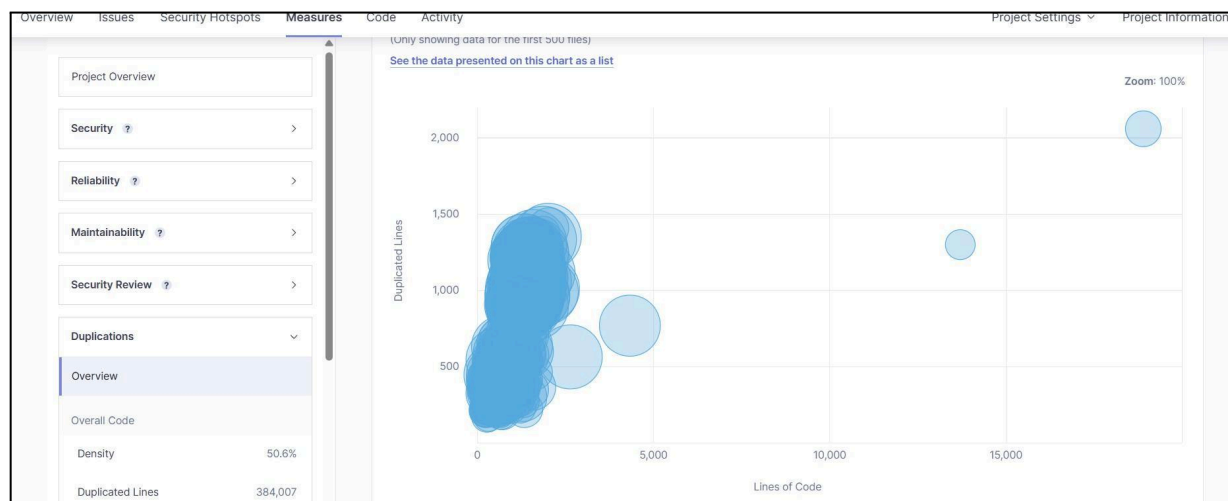
- Top Bar:** "Bulk Change" button, "Select issues" dropdown, "Navigate to issue" dropdown, "67,624 issues", "1646d effort".
- Path:** gameoflife-core/build/reports/tests/all-tests.html
- Bug 1:**
  - ☐ Add "lang" and/or "xml:lang" attributes to this "<html>" element
  - Reliability (tag)
  - Intentionality (tag)
  - accessibility wcag2-a (tag)
  - Open (radio), Not assigned (radio)
  - L1 • 2min effort • 4 years ago • Bug • Major
- Bug 2:**
  - ☐ Insert a <!DOCTYPE> declaration to before this <html> tag.
  - Reliability (tag)
  - Consistency (tag)
  - user-experience (tag)
  - Open (radio), Not assigned (radio)
  - L1 • 5min effort • 4 years ago • Bug • Major
- Bug 3:**
  - ☐ Add "<th>" headers to this "<table>".
  - Reliability (tag)
  - Intentionality (tag)
  - accessibility wcag2-a (tag)
  - Open (radio), Not assigned (radio)
  - L9 • 2min effort • 4 years ago • Bug • Major

- Code Smells

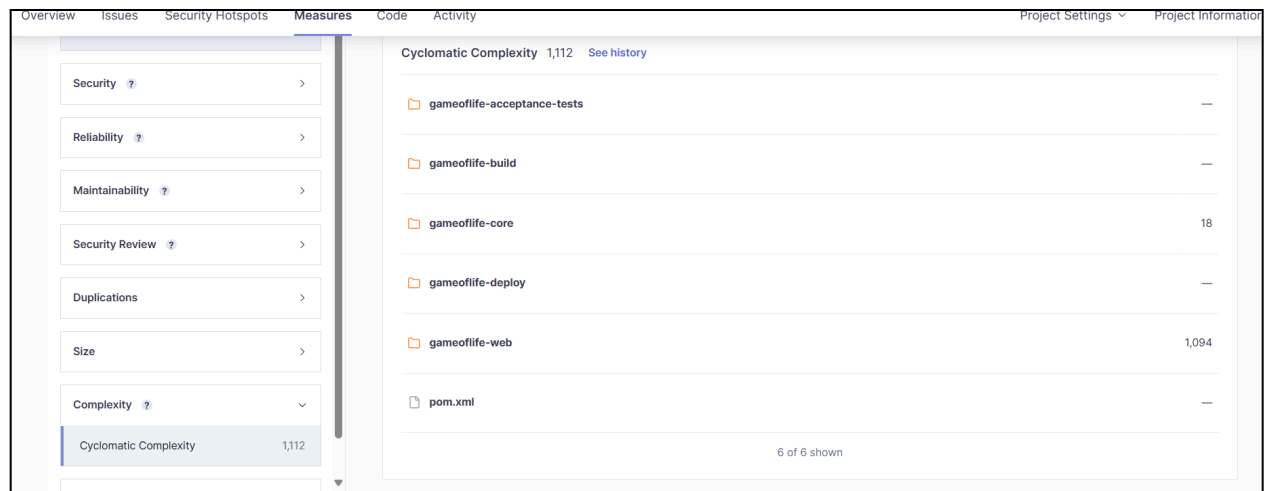
The screenshot shows a code smell tracking interface with the following details:

- Filters:** "Clear All Filters" button. "Issues in new code" section. "Clean Code Attribute" section with "Consistency" (164k), "Intentionality" (15), "Adaptability" (0), "Responsibility" (0). "Software Quality" section with "Security" (0), "Reliability" (68k), "Maintainability" (164k).
- Top Bar:** "Bulk Change" button, "Select issues" dropdown, "Navigate to issue" dropdown, "163,781 issues", "1705d effort".
- Path:** gameoflife-acceptance-tests/Dockerfile
- Issue 1:**
  - ☐ Use a specific version tag for the image.
  - Maintainability (tag)
  - Intentionality (tag)
  - No tags (tag)
  - Open (radio), Not assigned (radio)
  - L1 • 5min effort • 4 years ago • Code Smell • Major
- Issue 2:**
  - ☐ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
  - Maintainability (tag)
  - Intentionality (tag)
  - No tags (tag)
  - Open (radio), Not assigned (radio)
  - L12 • 5min effort • 4 years ago • Code Smell • Major
- Issue 3:**
  - ☐ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
  - Maintainability (tag)
  - Intentionality (tag)
  - No tags (tag)
  - Open (radio), Not assigned (radio)
  - L12 • 5min effort • 4 years ago • Code Smell • Major

- Duplications



## ● Cyclomatic Complexities



The screenshot shows the SonarQube interface with the 'Measures' tab selected. On the left, a sidebar lists various quality measures, with 'Cyclomatic Complexity' selected and showing a value of 1,112. The main area displays a table of Cyclomatic Complexity for different modules. The table has two columns: the module name and its complexity value. The modules listed are gameoflife-acceptance-tests, gameoflife-build, gameoflife-core, gameoflife-deploy, gameoflife-web, and pom.xml. The complexity values are: gameoflife-acceptance-tests (—), gameoflife-build (—), gameoflife-core (18), gameoflife-deploy (—), gameoflife-web (1,094), and pom.xml (—). At the bottom of the table, it says '6 of 6 shown'.

Cyclomatic Complexity 1,112 <a href="#">See history</a>	
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

6 of 6 shown

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