

## 08 Advanced DevOps Lab

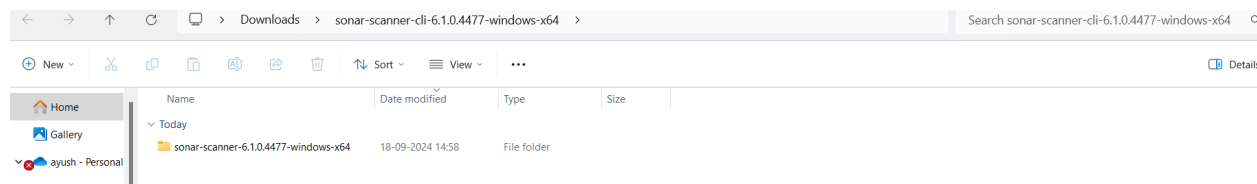
Aim: Create a Jenkins CI/CD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

### Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>  
Visit this link and download the sonarqube scanner CLI.

The screenshot shows the SonarScanner CLI documentation page. The left sidebar contains a navigation menu with links like 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code', 'Scanners', and 'SonarScanner CLI'. The main content area is titled 'SonarScanner CLI' and features a table with columns 'SonarScanner' and 'Issue Tracker'. The 'SonarScanner' column shows version '6.1' and a date '2024-06-27'. Below the table, there is a section for 'macOS and Linux AArch64 distributions' with download links for 'Linux x64', 'Linux AArch64', 'Windows x64', 'macOS x64', 'macOS AArch64', and 'Docker'. A 'Release notes' link is also present. The right sidebar contains a 'Show more' link.

Extract the downloaded zip file in a folder.



### 1. Install sonarqube image

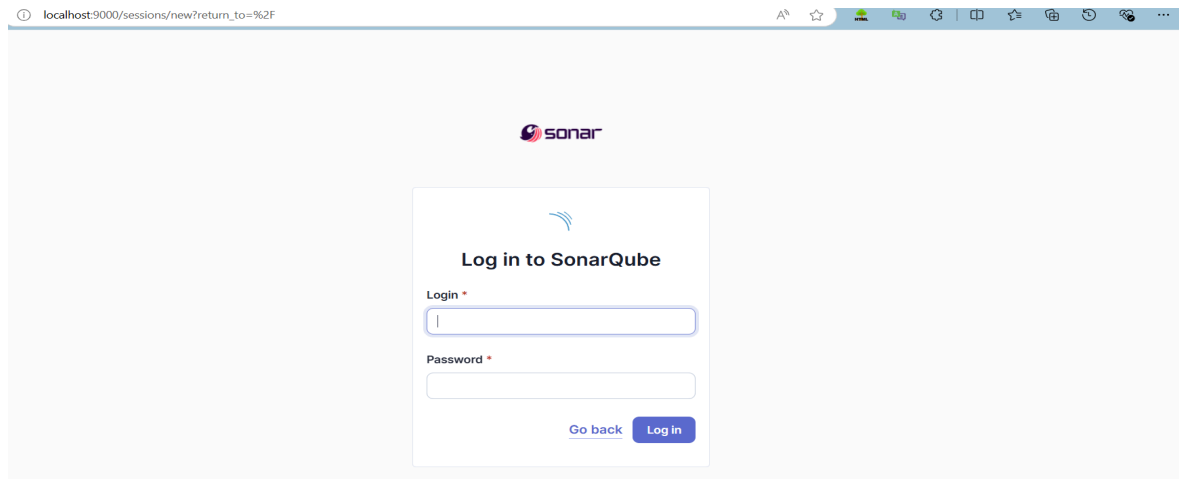
Command: **docker pull sonarqube**

```
C:\Users\Ayush Maurya>docker pull sonarqube
Using default tag: latest
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: sha256:72e9fec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecd
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

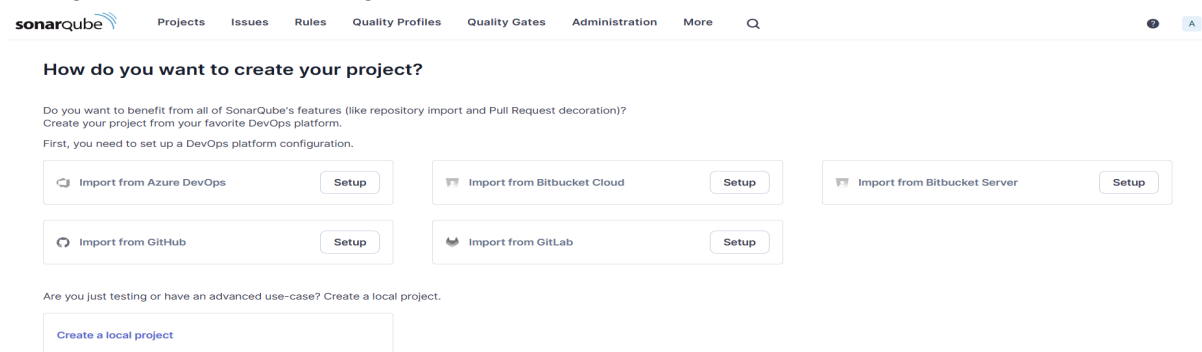
What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube

C:\Users\Ayush Maurya>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
dfe47cea99897c7099833dc0d7fa99279ef09d4f3038622910a74df2630afed5
```

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 at 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.

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.

**Plugins**

Search: sonarq

Install	Name	Released
<input type="checkbox"/>	SonarQube Scanner 2.17.2	6 mo 29 days ago

**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**

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**

Dashboard > Manage Jenkins > Tools

Add Git

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 “Install automatically” option.

SonarQube Scanner

Name

sonarqube\_exp8

☒ Install automatically

Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584


Add Installer

Add SonarQube Scanner


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

Dashboard > All >


**Enter an item name**  
  
= Required field




**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.

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_Jenkins>') {
        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.

Pipeline

Definition

Pipeline script

Script ?

```
1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') { // Ensure this matches the SonarQube environment name in Jenkins
7       bat """
8         "C:\\Users\\Ayush Maurya\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-scanner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.
9         -D sonar.login=admin ^
10        -D sonar.password=Ayush3114 ^
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 }
```

11. Build project

Dashboard > adv\_devops\_exp8 >

Status

adv\_devops\_exp8

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

SonarQube

Stages

Rename

Pipeline Syntax

Build History

Filter...

#9

Sep 18, 2024, 4:14 PM

Stage View

Average stage times:  
(Average full run time: ~6min 4s)

	Cloning the GitHub Repo	SonarQube analysis
#5 Sep 18 16:14 No Changes	2s	6min 2s
#6 Sep 18 16:12 No Changes	2s	1s
#7 Sep 18 16:10 No Changes	2s	120ms

12. Check console

Status

Console Output

View as plain text

Edit Build Information

Delete build '#9'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Replay

Pipeline Steps

Workspaces

Previous Build

Console Output

Skipping 4,246 KB. [Full Log](#)

```
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 512. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 248. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 886. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 249. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 662. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 615. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 664. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 913. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 810. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 668. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 548. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 543. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 152. Keep only the first 100 references.
16:19:49.753 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line
```

## 13. Now, check the project in SonarQube

The screenshot shows the SonarQube project overview for the 'main' branch. The Quality Gate is 'Passed' (green checkmark). The last analysis was 18 minutes ago. The project has 683k Lines of Code and Version is not provided. A warning indicates the last analysis has warnings. The 'Overall Code' tab is selected, showing metrics for Security (0 Open Issues), Reliability (68k Open Issues), Maintainability (164k Open Issues), Accepted issues (0), Coverage (50.6%), and Duplications (50.6%).

Metric	Value	Quality
Security	0 Open Issues	A
Reliability	68k Open Issues	C
Maintainability	164k Open Issues	A
Accepted issues	0	
Coverage	50.6%	
Duplications	50.6%	

## 14. Code Problems

### Consistency

The screenshot shows the SonarQube Issues page for 'Consistency' problems. The left sidebar shows filters for 'Clean Code Attribute' (1) and 'Software Quality' (0). The main area displays three issues related to HTML attributes: 'Insert a <DOCTYPE> declaration to before this <html> tag.', 'Remove this deprecated "width" attribute.', and 'Remove this deprecated "align" attribute.' Each issue is marked as 'Open' and 'Not assigned'.

Issue	Category	Severity	Effort	Age	Tags
Insert a <DOCTYPE> declaration to before this <html> tag.	Consistency	Reliability	L1 - 5min effort	4 years ago	user-experience
Remove this deprecated "width" attribute.	Consistency	Maintainability	L9 - 5min effort	4 years ago	html5, obsolete
Remove this deprecated "align" attribute.	Consistency	Maintainability	L11 - 5min effort	4 years ago	html5, obsolete

### Intentionality

The screenshot shows the SonarQube Issues page for 'Intentionality' problems. The left sidebar shows filters for 'Clean Code Attribute' (1) and 'Software Quality' (0). The main area displays three issues related to Dockerfile tags: 'Use a specific version tag for the image.', 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.', and 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' Each issue is marked as 'Open' and 'Not assigned'.

Issue	Category	Severity	Effort	Age	Tags
Use a specific version tag for the image.	Intentionality	Maintainability	L1 - 5min effort	4 years ago	No tags
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality	Maintainability	L12 - 5min effort	4 years ago	No tags
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality	Maintainability	L12 - 5min effort	4 years ago	No tags

• Bugs

Bulk Change

Select Issues

Navigate to Issue

67,624 Issues

1646d effort

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element

Intentionality

accessibilitywcag2-a

Reliability

Open

Not assigned

L1

2min effort

4 years ago

Bug

Major

Insert a <!DOCTYPE> declaration to before this <html> tag.

Consistency

user-experience

Reliability

Open

Not assigned

L1

5min effort

4 years ago

Bug

Major

Add "<th>" headers to this "<table>".

Intentionality

accessibilitywcag2-a

Reliability

Open

Not assigned

L9

2min effort

4 years ago

Bug

Major

• Code Smells

Bulk Change

Select Issues

Navigate to Issue

163,781 Issues

1705d effort

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image.

Intentionality

No tags

Maintainability

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

No tags

Maintainability

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

No tags

Maintainability

Open

Not assigned

L12

5min effort

4 years ago

Code Smell

Major

Filters

Clear All Filters

Issues in new code

Clean Code Attribute

Consistency

164k

Intentionality

15

Adaptability

0

Responsibility

0

Software Quality

1

Security

0

Reliability

68k

Maintainability

164k

Add to selectionCtrl + click

• Duplications

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

(Only showing data for the first 300 files)  
[See the data presented on this chart as a list](#)

Project Overview

Security

Reliability

Maintainability

Security Review

Duplications

Overview

Overall Code

Density50.6%

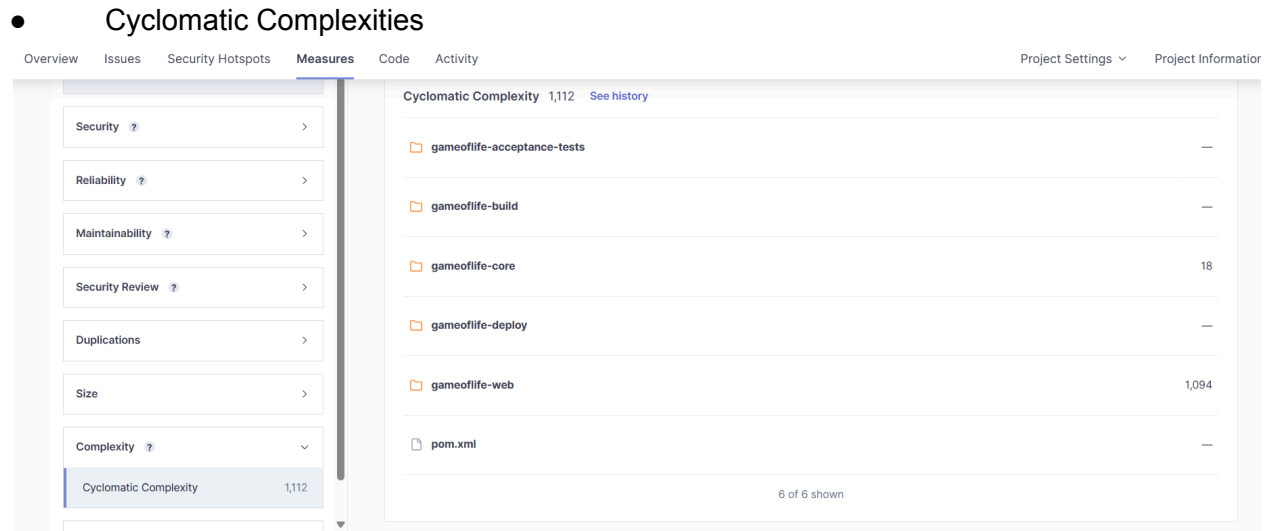
Duplicated Lines384,007

Duplicated Lines

Lines of Code

Zoom: 100%





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

### Conclusion:

This experiment established a seamless integration of Jenkins with SonarQube to automate code quality assessments within the CI/CD pipeline. SonarQube was deployed using Docker, and after setting up a project, it was configured to analyze the codebase for potential quality issues. Jenkins was configured with the necessary SonarQube plugins, allowing automated code checks through a pipeline that cloned a GitHub repository and performed a SonarQube scan. This integration ensures continuous monitoring throughout the development cycle, effectively identifying and addressing bugs, code smells, and security vulnerabilities, thereby enhancing code quality and security.