

# ADVANCE DEVOPS EXP-7

RIYA VARYANI

D15A/64

**Aim:** To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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



The screenshot shows the Jenkins Dashboard interface. At the top, there's a search bar and user information for 'Riya Varyani'. The left sidebar contains navigation links like 'New Item', 'Build History', 'Project Relationship', etc. The main area displays a table of build history with columns for status, name, last success, last failure, and last duration. Below the table, there are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing the built-in node).

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	first-job	18 hr #1	N/A	0.62 sec
✓	☀	MavenBuild	18 hr #1	N/A	34 sec
✓	☀	pipeline	18 hr #2	N/A	4.1 sec

**Step-2:** Run SonarQube in a Docker container using this command :- a]docker -v  
b] docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\varya>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
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: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecd
Status: Downloaded newer image for sonarqube:latest
b186ed1336af1631917ede88ae9f1d4d688ccd35cc77eb5bfd47b1764967a2c6
C:\Users\varya>
```

**Step-3:** Once the container is up and running, you can check the status of SonarQube at localhost port 9000. The login id is “admin” and the password is also “admin”.

## Log in to SonarQube

Login \*

Password \*

[Go back](#) [Log in](#)

**Step-4:** Create a local project in SonarQube with the name sonarqube

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](#)

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 most recent changes to your project, enabling you to follow the Clean as 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.  
 Recommended for projects following regular versions or releases.

☐ Number of days  
 Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.  
 Recommended for projects following continuous delivery.

☐ Reference branch  
 Choose a branch as the baseline for the new code.  
 Recommended for projects using feature branches.

**Step-5:** Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins → Plugins and search for SonarQube Scanner in Available Plugins and install it.

Search (CTRL+K)

Riya Varyani

log out

Dashboard > Manage Jenkins > Plugins

Plugins
 

sona

Updates 33

Available plugins

Installed plugins

Advanced settings

Name	Enabled
<b>SonarQube Scanner for Jenkins</b> 2.17.2 This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality. <a href="#">Report an issue with this plugin</a>	<div> </div>

**Step-6:** Under 'Manage Jenkins → System', look for SonarQube Servers and enter these details. Name : sonarqube, Server URL : http://localhost:9000

### SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

☒ Environment variables

### SonarQube installations

List of SonarQube installations

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

**Step-7:** Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically. Manage Jeknins → Tools → SonarQube Scanner Installation.

### SonarQube Scanner installations

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


**Step-8:** After the configuration, create a New Item in Jenkins, choose a freestyle project named sonarqube.


## 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.

OK

**Step-9:** Choose this GitHub repository in Source Code Management.

[https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git). It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

**Configure**

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

**Source Code Management**

☐ None

☒ Git ?

**Repositories** ?

Repository URL ?

https://github.com/shazforiot/MSBuild\_firstproject.git

Credentials ?

- none -

+ Add

Advanced

Add Repository

**Branches to build** ?

Branch Specifier (blank for 'any') ?

\*/master

Add Branch

Repository browser ?

(Auto)

Save Apply

**Step-10:** Under Build-> Execute SonarQube Scanner, enter these Analysis Properties.  
Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

sonar.projectKey=sonarqube

sonar.login=admin

sonar.password=riya 123

sonar.sources=.

sonar.host.url=http://localhost:9000

## Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

### Build Steps

Execute SonarQube Scanner

JDK ?

JDK to be used for this SonarQube analysis

(inherit From Job)

Path to project properties ?

Analysis properties ?

sonar.projectKey=sonarqube  
sonar.login=admin  
sonar.password=ansh  
sonar.sources=.  
sonar.host.url=http://localhost:9000

Additional arguments ?

JVM Options ?

**Step-11:** Go to <http://localhost:9000/admin/permissions> and allow Execute Permissions to the Admin user.

Global Permissions

Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

AllUsersGroups

Q Search for users or groups...


	Administer System ?	Administer ?	Execute Analysis ?	Create ?
<div><div>sonar-administrators</div><div>System administrators</div></div>	<input checked="" type="checkbox"/>	<div><input checked="" type="checkbox"/> Quality Gates</div> <div><input checked="" type="checkbox"/> Quality Profiles</div>	<input type="checkbox"/>	<div><input checked="" type="checkbox"/> Projects</div>
<div><div>sonar-users</div><div>Every authenticated user automatically belongs to this group</div></div>	<input type="checkbox"/>	<div><input type="checkbox"/> Quality Gates</div> <div><input type="checkbox"/> Quality Profiles</div>	<input checked="" type="checkbox"/>	<div><input checked="" type="checkbox"/> Projects</div>
<div><div>Anyone <div>DEPRECATED</div></div><div>Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.</div></div>	<input type="checkbox"/>	<div><input type="checkbox"/> Quality Gates</div> <div><input type="checkbox"/> Quality Profiles</div>	<input type="checkbox"/>	<div><input type="checkbox"/> Projects</div>
<div><div>Administrator admin</div></div>	<input checked="" type="checkbox"/>	<div><input type="checkbox"/> Quality Gates</div> <div><input type="checkbox"/> Quality Profiles</div>	<input checked="" type="checkbox"/>	<div><input checked="" type="checkbox"/> Projects</div>


4 of 4 shown

**Step-12:** Run The Build and check the console output.

Dashboard > SonarQube >

Status

 **SonarQube**

 Add description

 Changes

 Workspace

 Build Now

 Configure

 Delete Project

 SonarQube

 Rename



### Permalinks

- [Last build \(#2\), 1 hr 14 min ago](#)
- [Last stable build \(#2\), 1 hr 14 min ago](#)
- [Last successful build \(#2\), 1 hr 14 min ago](#)
- [Last failed build \(#1\), 1 hr 19 min ago](#)
- [Last unsuccessful build \(#1\), 1 hr 19 min ago](#)
- [Last completed build \(#2\), 1 hr 14 min ago](#)

Build History

trend

Filter...


 #2

Sep 26, 2024, 10:51 PM


Dashboard > SonarQube > #2 > Console Output

Status


 **Console Output**

 Download

 Copy

 View as plain text


 Changes

 Console Output

 Edit Build Information

 Delete build '#2'

 Timings

 Git Build Data

 Previous Build

```
Started by user Riya Varyani
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
```




```
22:52:22.244 INFO Sensor Zero Coverage Sensor
22:52:22.262 INFO Sensor Zero Coverage Sensor (done) | time=18ms
22:52:22.264 INFO SCM Publisher SCM provider for this project is: git
22:52:22.266 INFO SCM Publisher 4 source files to be analyzed
22:52:22.679 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=413ms
22:52:22.683 INFO CPD Executor Calculating CPD for 0 files
22:52:22.684 INFO CPD Executor CPD calculation finished (done) | time=0ms
22:52:22.691 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
22:52:22.947 INFO Analysis report generated in 128ms, dir size=201.0 kB
22:52:23.019 INFO Analysis report compressed in 58ms, zip size=22.5 kB
22:52:23.156 INFO Analysis report uploaded in 135ms
22:52:23.156 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
22:52:23.158 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
22:52:23.158 INFO More about the report processing at http://localhost:9000/api/ce/task?id=7cb01bac-c199-40ab-8c99-3c35629e6a80
22:52:23.165 INFO Analysis total time: 17.327 s
22:52:23.165 INFO SonarScanner Engine completed successfully
22:52:23.216 INFO EXECUTION SUCCESS
22:52:23.227 INFO Total time: 24.808s
Finished: SUCCESS
```

REST API Jenkins 2.462.2

**Step-13:** Once the build is complete, check the project in SonarQube.

The screenshot displays the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The 'Projects' tab is active. On the left, there are tabs for 'My Favorites' and 'All'. Below these are filters for 'Quality Gate' (1 Passed, 0 Failed) and 'Reliability'. The main content area shows a search bar, a 'Create Project' button, and a list of projects. The first project, 'sonarqube PUBLIC', is highlighted with a green checkmark and the status 'Passed'. Below the project name, it says 'Last analysis: 18 minutes ago' and 'The main branch of this project is empty.'

## main

Version **not provided** \* Set as homepageQuality Gate **Passed**

Last analysis 17 minutes ago

The last analysis has warnings. [See details](#)

New Code

Overall Code

## Security

0 Open issues

0 H

0 M

0 L

## Reliability

0 Open issues

0 H

0 M

0 L

## Maintainability

0 Open issues

0 H

0 M

0 L

## Accepted issues

0

Valid issues that were not fixed

## Coverage

On 0 lines to cover.

## Duplications

0.0%

On 86 lines.

## Security Hotspots

0