

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

### **Theory:**

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

### **What problems does SAST solve?**

**SAST** takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

**SAST** tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise. It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

### **Why is SAST important?**

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the

codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence. Thus, integrating static analysis into the SDLC can yield dramatic results in the overall quality of the code developed.

## **What are the key steps to run SAST effectively?**

There are six simple steps needed to perform SAST efficiently in organizations that have a very large number of applications built with different languages, frameworks, and platforms.

1. Finalize the tool. Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.

2. Create the scanning infrastructure, and deploy the tool. This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.

3. Customize the tool. Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security

vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.

4. Prioritize and onboard applications. Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk

applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.

5. Analyze scan results. This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be

tracked and provided to the deployment teams for proper and timely remediation.

6. Provide governance and training. Proper governance ensures that your development teams are employing the scanning tools properly. The software

security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

## **Integrating Jenkins with SonarQube: Windows installation**

Step 1 Install JDK 1.8

Step 2 download and install

jenkins

**installing-the-default-jre-jdk**

Step 1 Install JDK 1.8

sudo apt-get install

openjdk-8-jre sudo apt

install default-jre /

## **how-to-install-jenkins-on-ubuntu-20-04**

### **Open SSH**

#### **Prerequisites:**

- Jenkins installed
- Docker Installed (for SonarQube)

(sudo apt-get install docker-ce=5:20.10.15~3-0~ubuntu-jammy

docker-ce-cli=5:20.10.15~3-0~ubuntu-jammy containerd.io docker-compose-plugin)

- SonarQube Docker Image

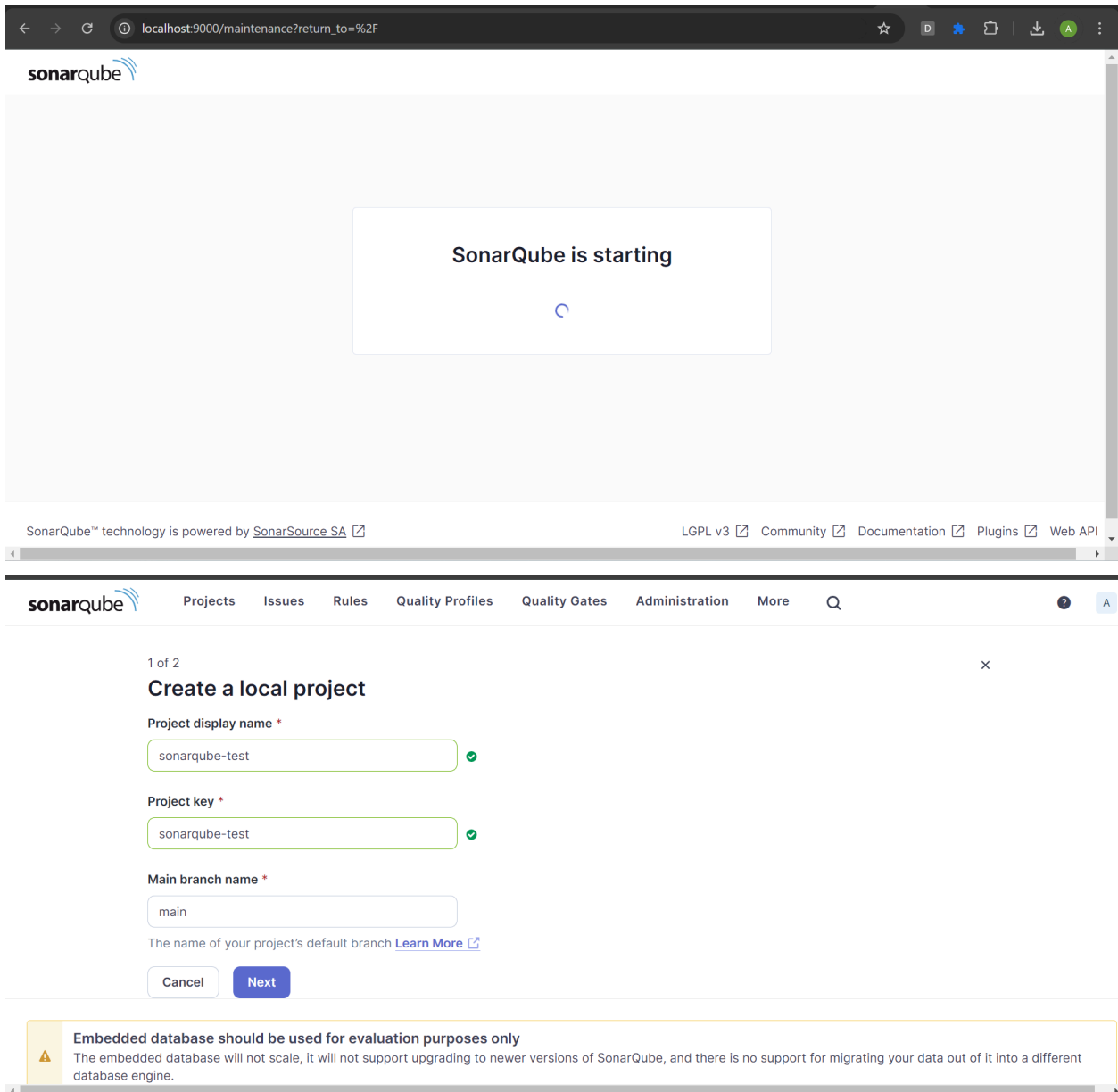
```

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\Santosh Sawant> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
docker: error during connect: Head "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktopLinuxEngine:
The system cannot find the file specified.
See 'docker run --help'.
PS C:\Users\Santosh Sawant> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
docker: Error response from daemon: Conflict. The container name "/sonarqube" is already in use by container "4f18dd2a380b6123c23b375
b26ff242670188ab50f190e8f86ac0889441a113a". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
PS C:\Users\Santosh Sawant> docker rm 4f18dd2a380b6123c23b375b26ff242670188ab50f190e8f86ac0889441a113a
4f18dd2a380b6123c23b375b26ff242670188ab50f190e8f86ac0889441a113a
PS C:\Users\Santosh Sawant> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
c03b891be990f1eb8b41085ce2b42c481c2751f4e9bd6b14a147d49bfe67fc6a

```



localhost:9000/maintenance?return\_to=%2F

sonarqube

SonarQube is starting

SonarQube™ technology is powered by [SonarSource SA](#)

LGPL v3 [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

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

**Embedded database should be used for evaluation purposes only**

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

Ok so go to Jenkins dashboard and then go to Manage jenkins then plugins , install plugins and then install “sonarqube scanner”

Dashboard > Manage Jenkins > Plugins

Plugins

Updates29

Available plugins

Installed plugins

Advanced settings

Search installed plugins

Script Security1341.va\_2819d\_414686

Allows Jenkins administrators to control what in-process scripts can be run by less-privileged users.[Report an issue with this plugin](#)

SNAKEYAML API2.2-111.vc6598e30cc65

This plugin provides SNAKEYAML for other plugins.[Report an issue with this plugin](#)

SonarQube Scanner for Jenkins2.17.2

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.[Report an issue with this plugin](#)

SSH Build Agents plugin2.973.v0fa\_8c0dea\_f9f

Allows to launch agents over SSH, using a Java implementation of the SSH protocol.[Report an issue with this plugin](#)

SSH Credentials Plugin343.v884f71d78167

Allows storage of SSH credentials in Jenkins[Report an issue with this plugin](#)

Jenkins

Search (CTRL+K)

1

2

ARNAV SANTOSH SAWANT

log out

Dashboard > Manage Jenkins > Plugins

Plugins

Updates28

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

Installing

Loading plugin extensions

Pending

→ [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

REST API

Jenkins 2.452.3

Dashboard > Manage Jenkins > System >

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

Save

Apply

Jenkins

Search (CTRL+K)

1

2

ARNAV SANTOSH SAWANT

log out

Dashboard > All >

Enter an item name

sonarQube

» 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) organizing complex activities that do not easily fit in free-style job type.

OK

## 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

Save

Apply

## Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

#### Path to project properties ?

#### Analysis properties ?

sonar.projectKey=sonarqube  
sonar.login=sqp\_018cb0bf9d002dc66c6c3f278248d119e5c2e0ab  
sonar.sources=HelloWorldCore  
sonar.host.url=http://localhost:9000

#### Additional arguments ?

#### JVM Options ?

Save

Apply

sonarqube

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

More

Q

?

A

Administration

Configuration

Security

Projects

System

Marketplace

	Administer System ?	Administer ?	Execute Analysis ?	Create ?
<div>sonar-administrators</div> <div>System administrators</div>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
<div>sonar-users</div> <div>Every authenticated user automatically belongs to this group</div>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
<div>Anyone <b>DEPRECATED</b></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>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
<div>A Administrator admin</div>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects

After 7 failures finally there was 8th success which is shown down below in image





## Build History

trend ▾



Filter...



#8

26 Sept 2024, 13:12



#7

26 Sept 2024, 13:09



#6

26 Sept 2024, 13:05



#5

26 Sept 2024, 13:03



#4

26 Sept 2024, 13:03



#3

26 Sept 2024, 12:59



#2

26 Sept 2024, 12:47



#1

26 Sept 2024, 12:45




Atom feed for all





Atom feed for failures

 Status

 Changes

 Console Output

 View as plain text

 Edit Build Information

 Timings

 Previous Build

## Console Output

```
Started by user ARNAV SANTOSH SAWANT
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sonarQube2
[sonarQube2] $
C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarQube\bin\sonar-scanner.bat
-Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test2 -
Dsonar.login=sqp_018cb0bf9d002dc66c6c3f278248d119e5c2e0ab -Dsonar.host.url=http://localhost:9000 -
Dsonar.sources=HelloWorldCore -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarQube2
13:12:17.541 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value
'http://localhost:9000'
13:12:17.641 INFO Scanner configuration file:
C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarQube\bin\..\conf\sonar-
scanner.properties
13:12:17.648 INFO Project root configuration file: NONE
13:12:17.763 INFO SonarScanner CLI 6.2.0.4584
13:12:17.770 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
13:12:17.795 INFO Windows 11 10.0 amd64
```