Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra

Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology A.Y. 2024-25

Advance DevOps Lab
Experiment 07

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

Roll No.	43
Name	Harsh Pramod Padyal
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO4: To identify and remediate application vulnerabilities earlier and help integrate security in the development process using SAST Techniques.
Grade:	

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

THEORY :

What is SAST?

Static Application Security Testing (SAST) is a method that analyzes source code to find security weaknesses before the code is compiled. This is often called white box testing.

What Problems Does SAST Solve?

SAST occurs early in the Software Development Life Cycle (SDLC) and does not require a working application. It helps developers find and fix vulnerabilities during development, preventing issues from reaching the final product. SAST tools provide real-time feedback, allowing developers to address problems before moving on. They also give visual representations of issues and highlight the exact locations of vulnerabilities, guiding developers on how to fix them without needing extensive security knowledge. It's crucial to run SAST tools regularly, such as during daily builds or code releases.

Why is SAST Important?

There are more developers than security staff, making it hard to review all code manually. SAST tools can analyze 100% of the codebase quickly, scanning millions of lines in minutes. They automatically find critical vulnerabilities, such as buffer overflows and SQL injection, which improves the overall quality of the code developed.

Key Steps to Run SAST Effectively:

1. **Choose the Right Tool**: Select a SAST tool that supports the programming languages and frameworks you use.

2. **Set Up the Infrastructure**: Handle licensing, access control, and resources needed to deploy the tool.

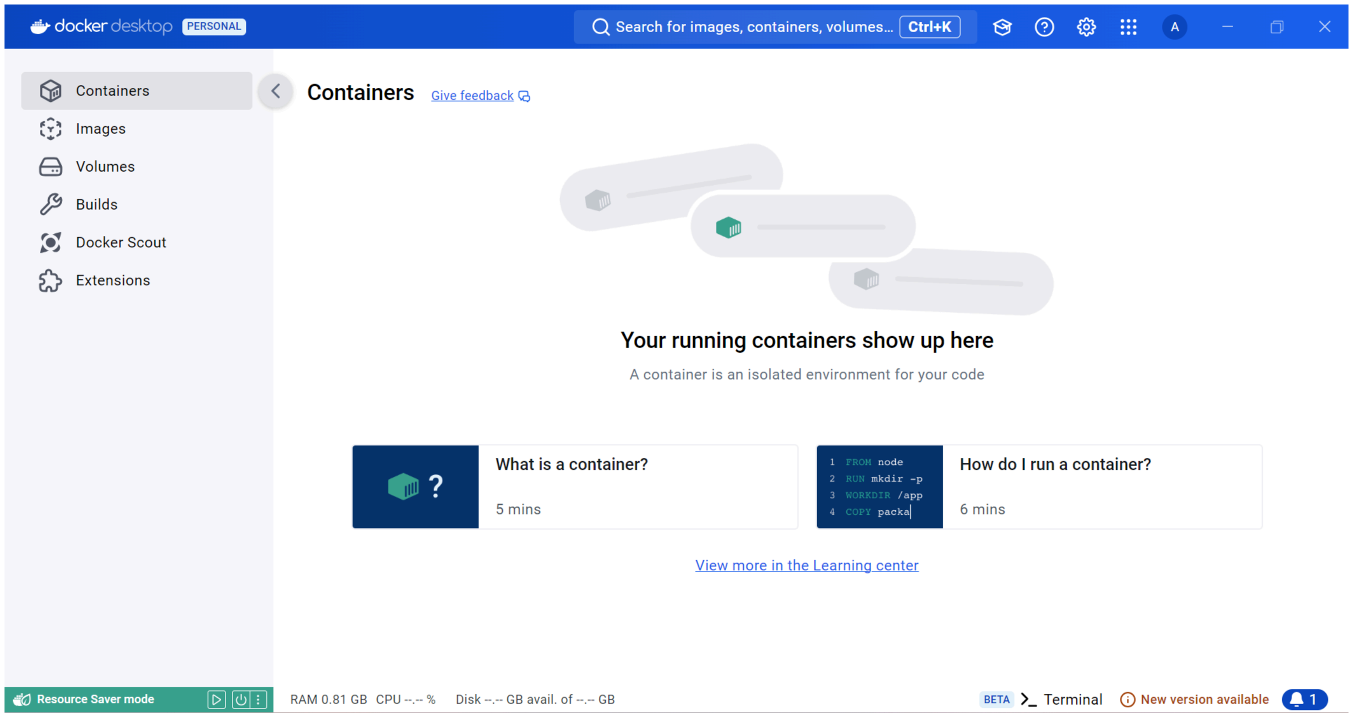
3. **Customize the Tool**: Fine-tune the tool to meet your organization's needs, reducing false positives and adding rules for better vulnerability detection.

4. **Onboard Applications**: Prioritize scanning high-risk applications first, then gradually onboard all applications for regular scans.

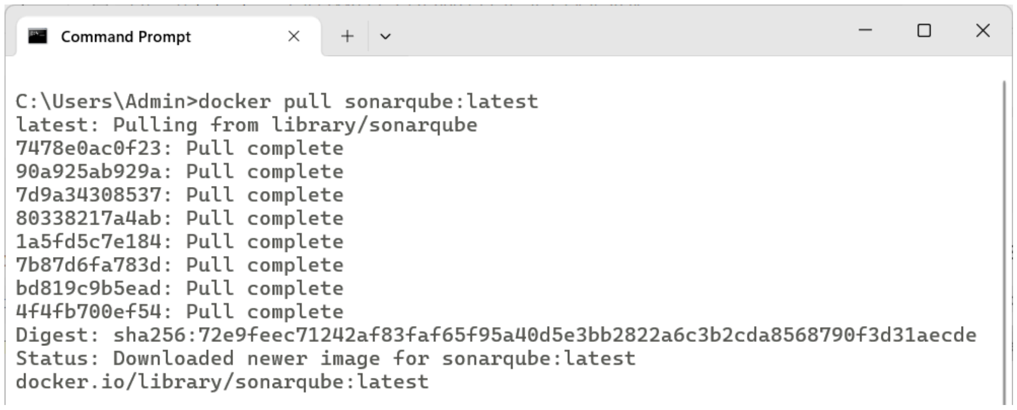
5. **Analyze Results**: Review the scan results, remove false positives, and ensure issues are tracked for timely fixing.

6. **Provide Governance and Training**: Ensure development teams use the tools properly and integrate SAST into the application development process.

Ensure Docker is running.

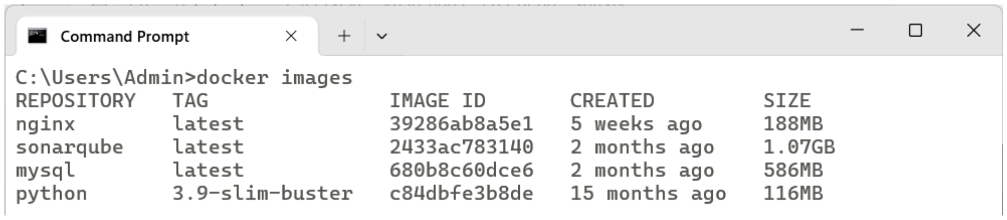

docker desktop PERSONAL
Search for images, containers, volumes... Ctrl+K
Containers
Images
Volumes
Builds
Docker Scout
Extensions
Containers Give feedback
Your running containers show up here
A container is an isolated environment for your code
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Resource Saver mode
RAM 0.81 GB CPU --.-- % Disk --.-- GB avail. of --.-- GB
BETA Terminal New version available 1

Open Command Prompt or PowerShell and execute the following command to pull the SonarQube image from Docker Hub.
docker pull sonarqube:latest



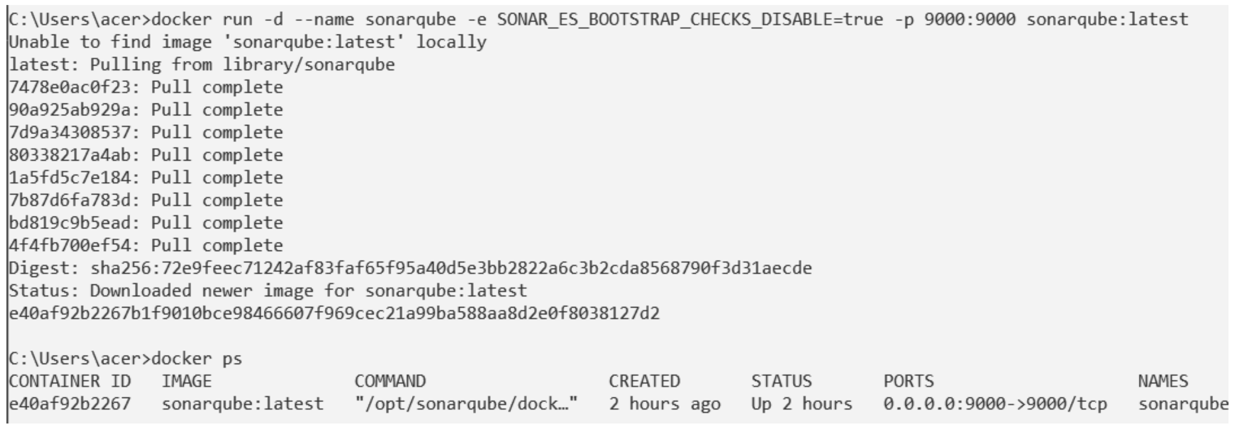
```
C:\Users\Admin>docker pull sonarqube:latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest
```

Verify that the image has been downloaded successfully by running.
docker images



```
C:\Users\Admin>docker images
REPOSITORY   TAG                IMAGE ID       CREATED         SIZE
nginx        latest             39286ab8a5e1   5 weeks ago     188MB
sonarqube    latest             2433ac783140   2 months ago    1.07GB
mysql        latest             680b8c60dce6   2 months ago    586MB
python       3.9-slim-buster    c84dbfe3b8de   15 months ago   116MB
```

Execute the following command in your terminal to run the SonarQube Container.
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

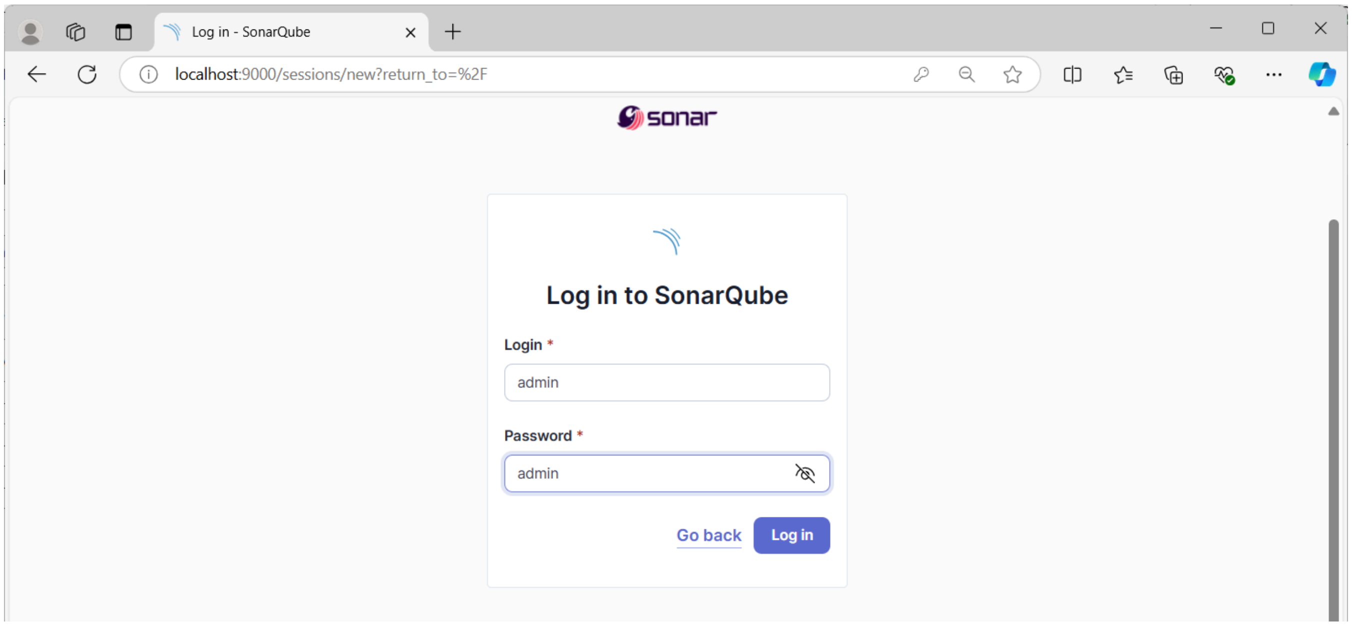


```
C:\Users\acer>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
e40af92b2267b1f9010bce98466607f969cec21a99ba588aa8d2e0f8038127d2

C:\Users\acer>docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS        PORTS                    NAMES
e40af92b2267   sonarqube:latest   "/opt/sonarqube/dock…"   2 hours ago    Up 2 hours    0.0.0.0:9000->9000/tcp   sonarqube
```

Login to SonarQube: Use the default credentials:

- Username: admin
- Password: admin



Log in - SonarQube

localhost:9000/sessions/new?return_to=%2F

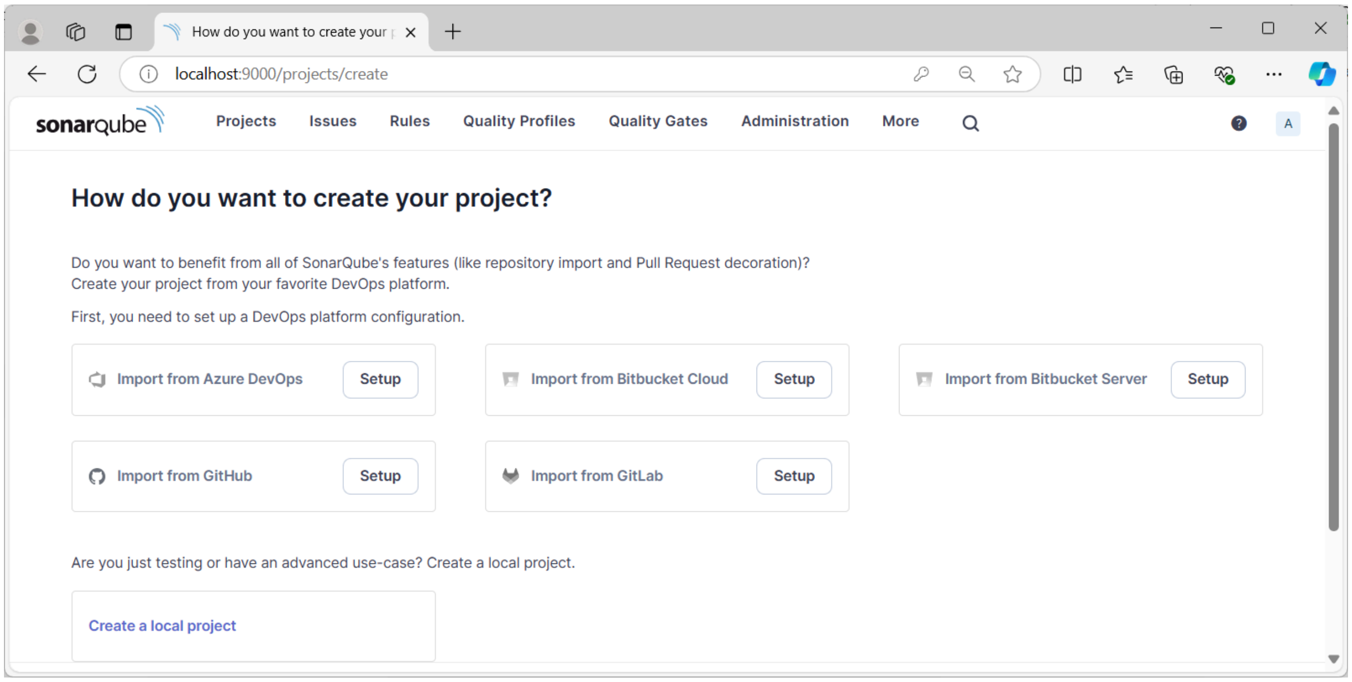
Log in to SonarQube

Login *

admin

Password *

admin



How do you want to create your project?

localhost:9000/projects/create

How do you want to create your project?

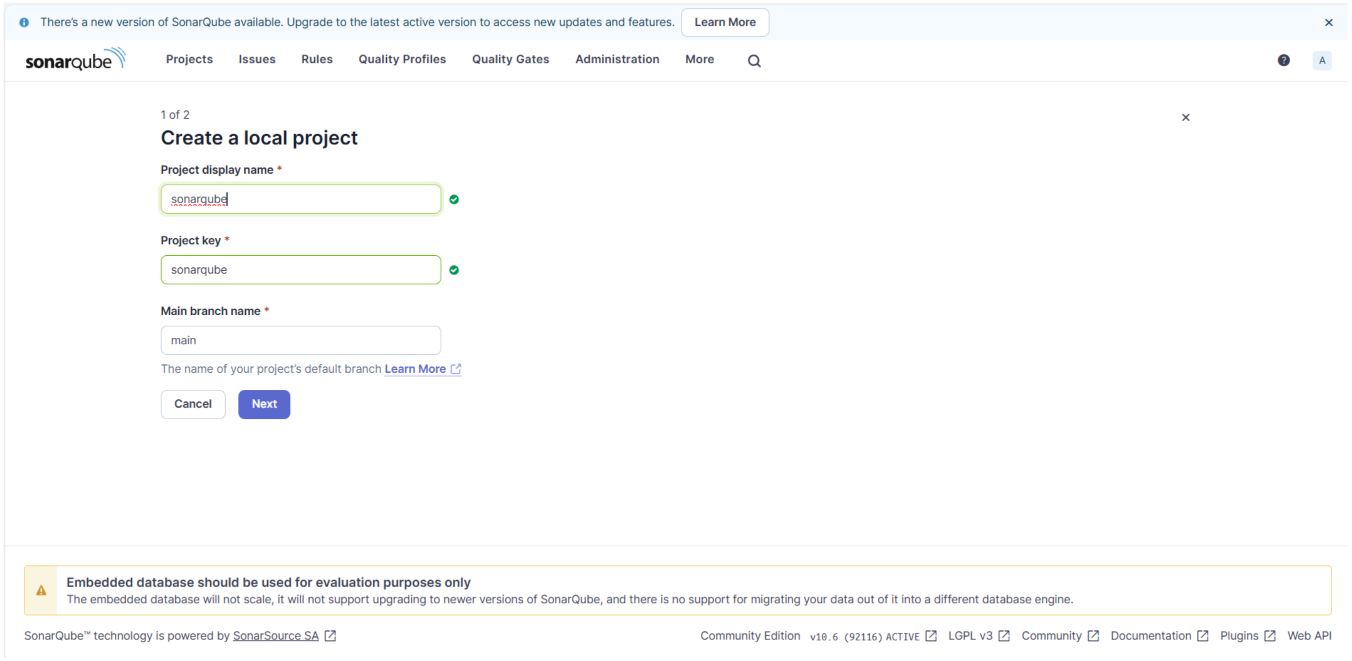
Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

Import from Azure DevOps Setup
Import from Bitbucket Cloud Setup
Import from Bitbucket Server Setup
Import from GitHub Setup
Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a local project.

Click on Create Project. Name the project and follow the prompts to set it up.



There's a new version of SonarQube available. Upgrade to the latest active version to access new updates and features. Learn More

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

1 of 2

Create a local project

Project display name *: sonarqube

Project key *: sonarqube

Main branch name *: main

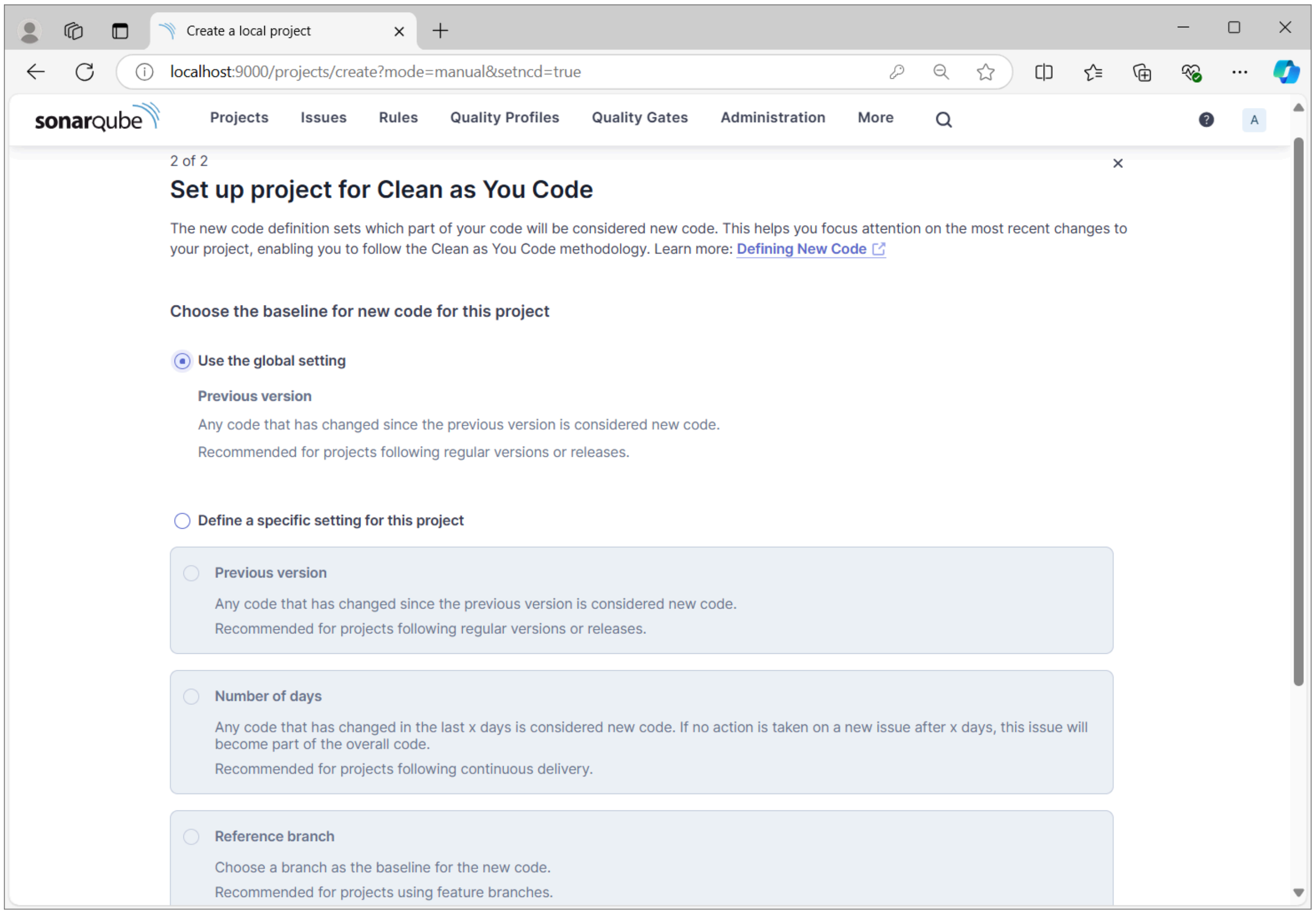
The name of your project's default branch Learn More

Cancel Next

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.

SonarQube™ technology is powered by SonarSource SA Community Edition v10.6 (92116) ACTIVE LGPL v3 Community Documentation Plugins Web API



Create a local project

localhost:9000/projects/create?mode=manual&setncd=true

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

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.

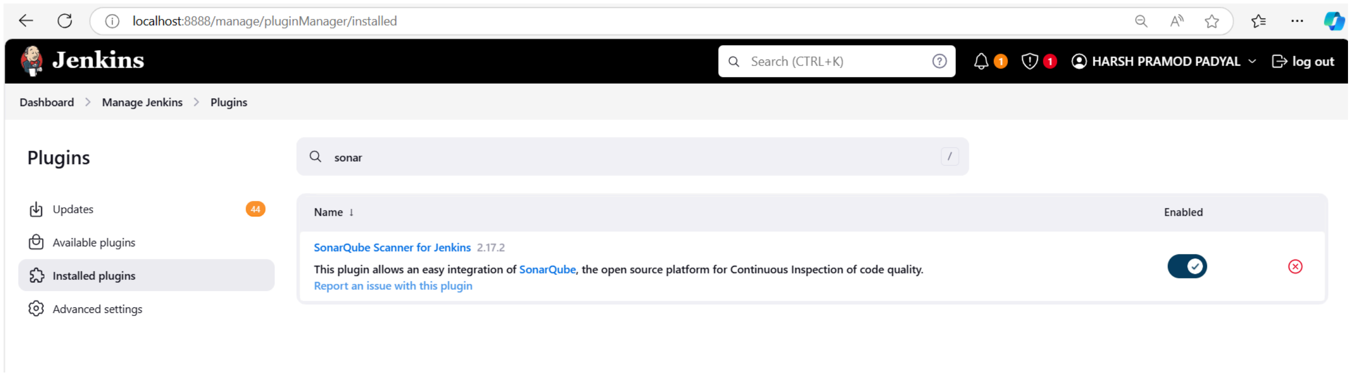
Reference branch

Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

Open Jenkins Dashboard: Go to http://localhost:8080 in your web browser (or the port where Jenkins is running).

Manage Jenkins → Manage Plugins → Available tab, search for SonarQube Scanner. Check the box next to it and click Install without Restart.



localhost:8888/manage/pluginManager/installed

Jenkins

Dashboard > Manage Jenkins > Plugins

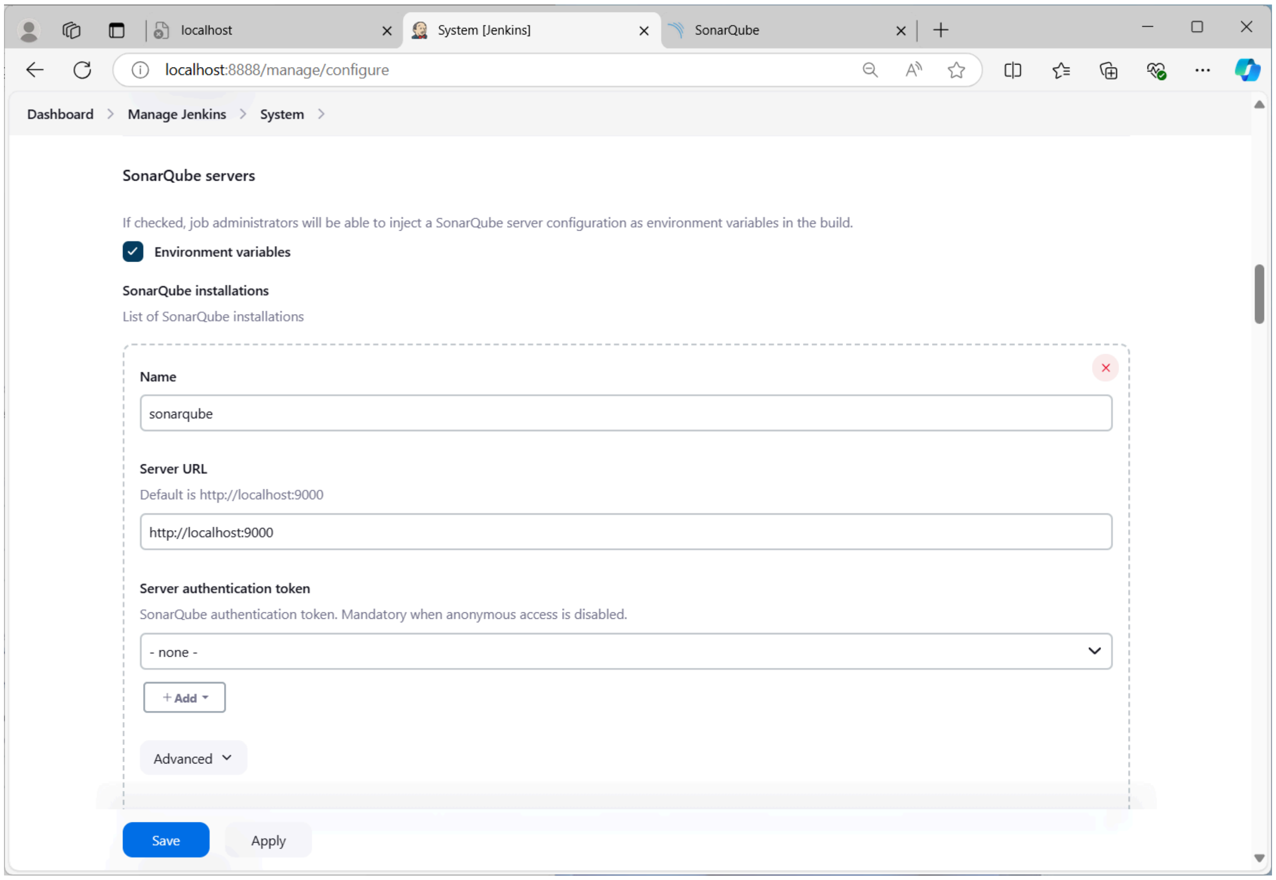
Plugins

sonar

Name ↓	Enabled
SonarQube Scanner for Jenkins 2.17.2 This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. Report an issue with this plugin	✓

Manage Jenkins → Configure System. Scroll down to the SonarQube Servers section and add a new SonarQube server.

- Name: Give it a name (e.g., SonarQube).
- Server URL: Enter http://localhost:9000.



localhost:8888/manage/configure

Dashboard > Manage Jenkins > System >

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

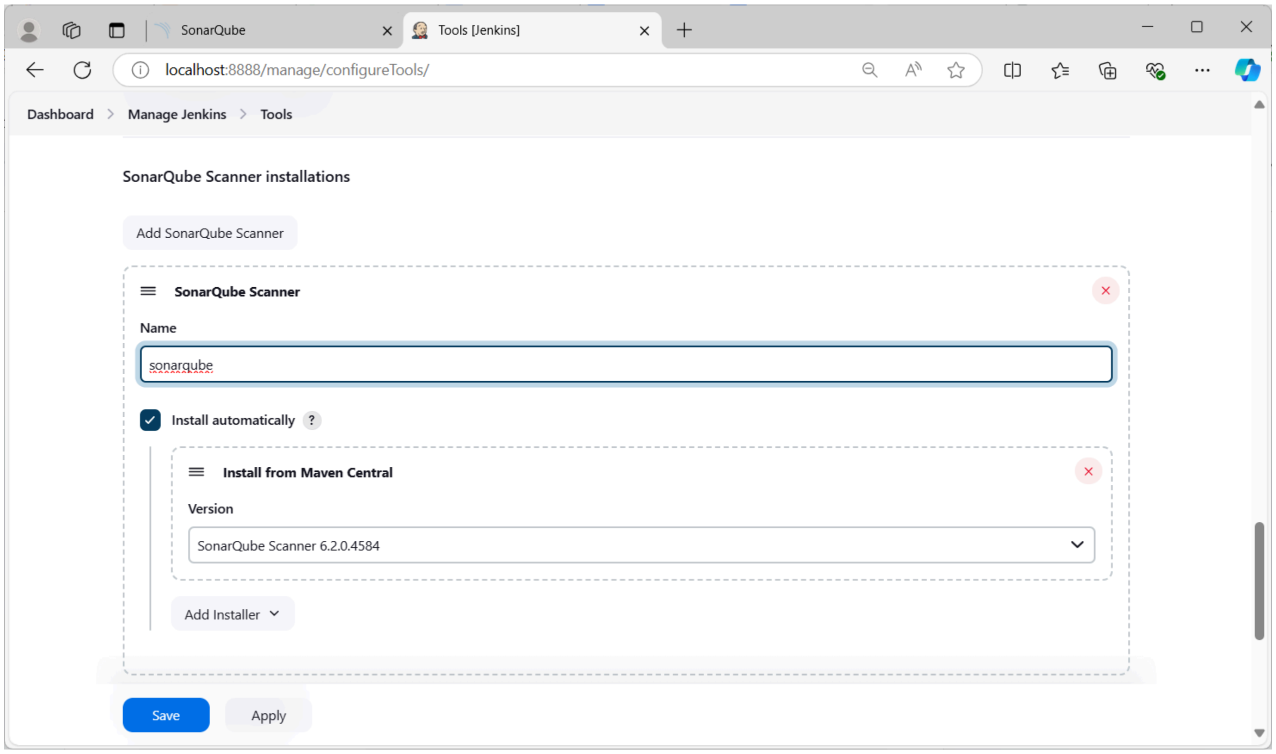
http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

Manage Jenkins → Global Tool Configuration. Find SonarQube Scanner and choose the latest version. Check the Install automatically option.



SonarQube
Tools [Jenkins]
localhost:8888/manage/configureTools/
Dashboard > Manage Jenkins > Tools
Add SonarQube Scanner
Add Installer
Save
Apply

SonarQube Scanner installations

SonarQube Scanner

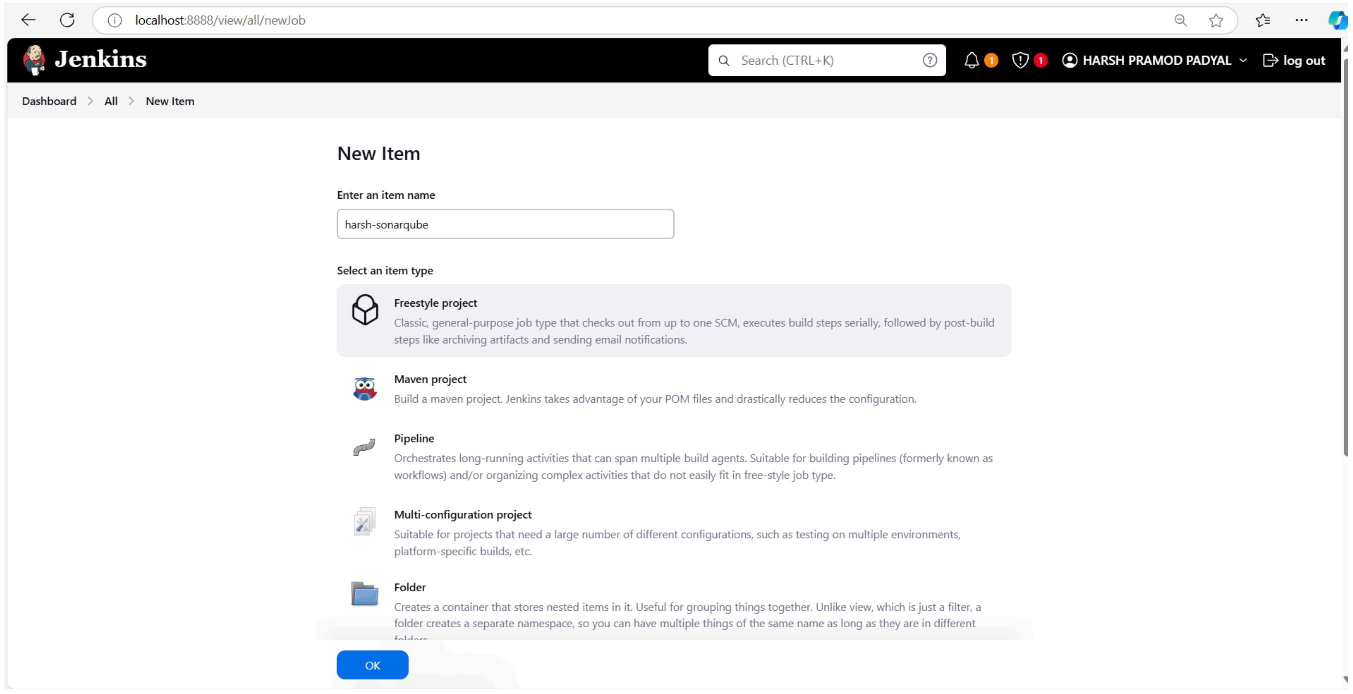
Name: sonarqube

Install automatically ✓

Install from Maven Central

Version: SonarQube Scanner 6.2.0.4584

On the Jenkins dashboard, click on New Item. Enter a name for your project. Choose the Freestyle project and click OK.



New Item

Enter an item name: harsh-sonarqube

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.

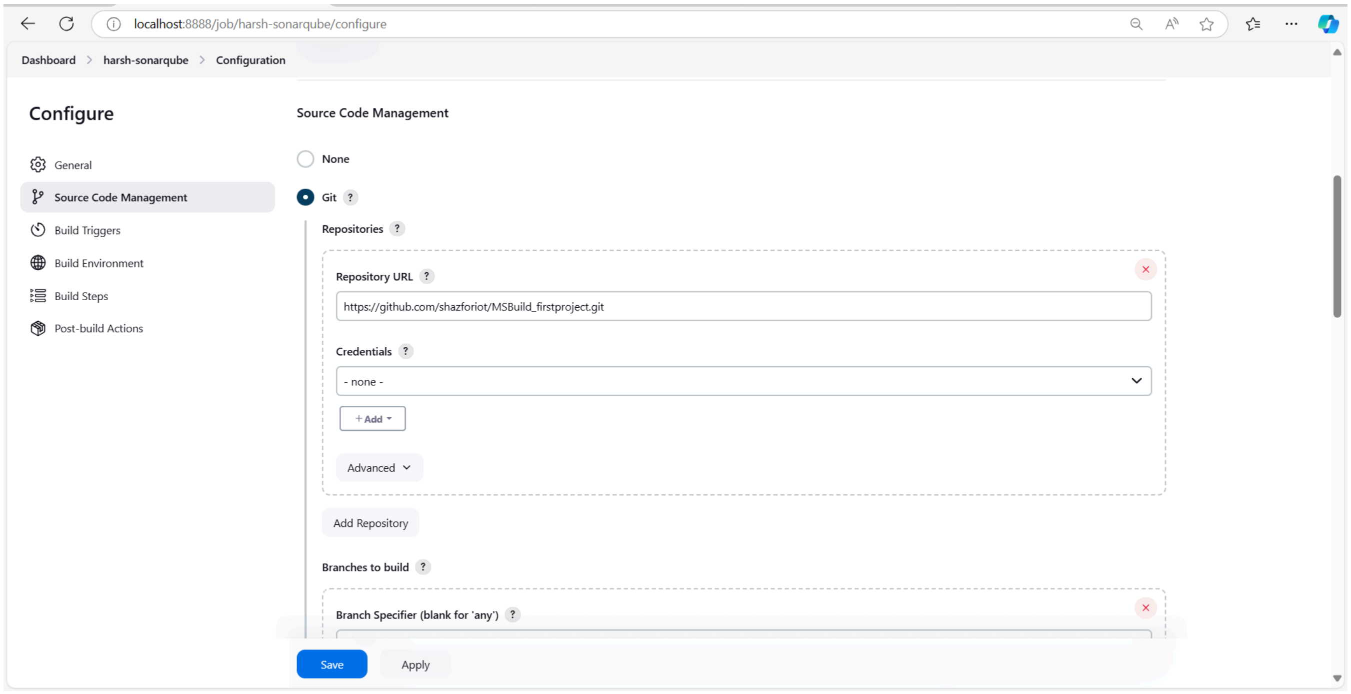
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

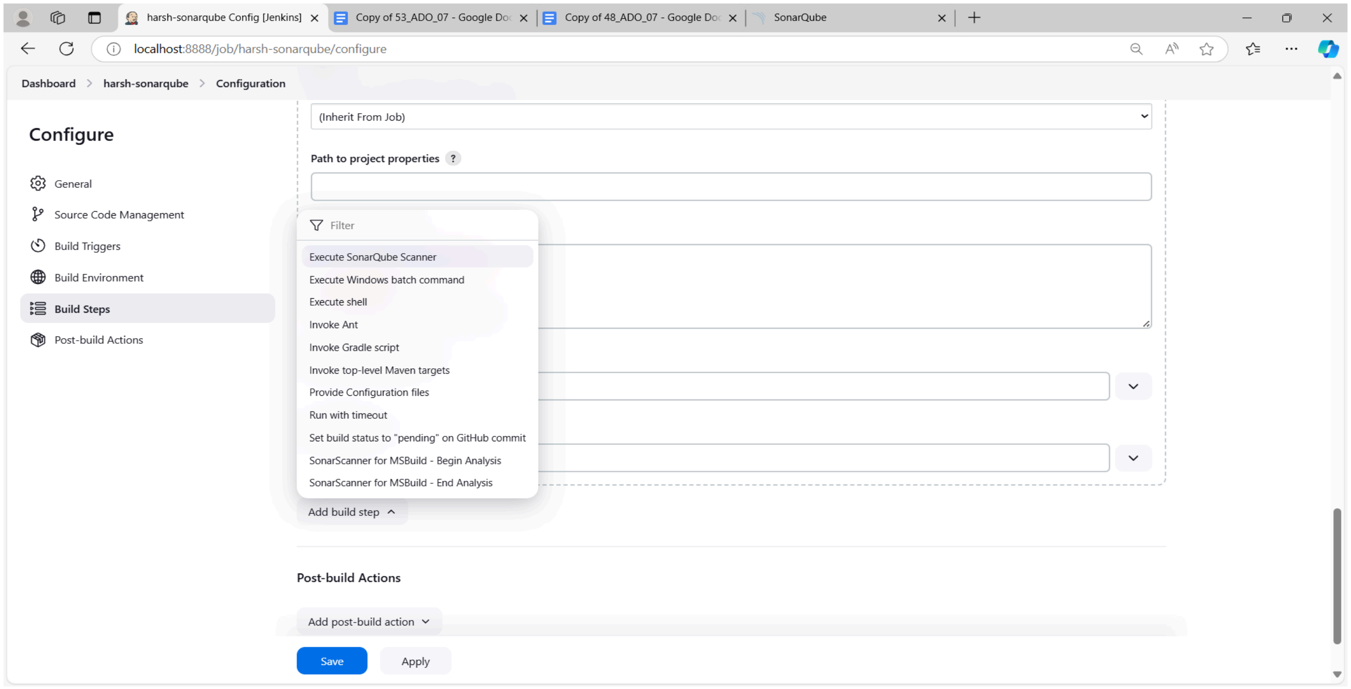
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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.

In the project configuration, look for the Source Code Management section. Choose Git and enter the repository URL.
https://github.com/shazforiot/MSBuild_firstproject.git

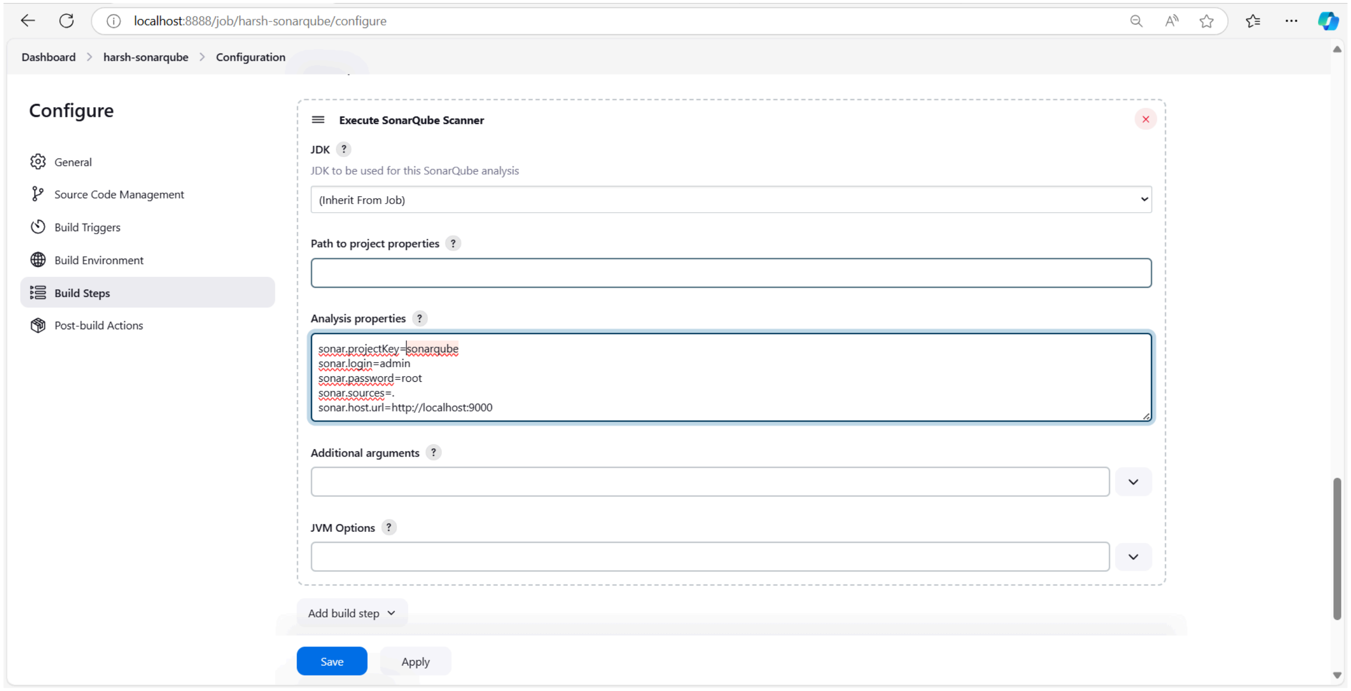


Scroll down to the Build section. Click on Add build step and select Execute SonarQube Scanner.


harsh-sonarqube Config [Jenkins]
Copy of 53_ADO_07 - Google Do
Copy of 48_ADO_07 - Google Do
SonarQube
localhost:8888/job/harsh-sonarqube/configure
Dashboard > harsh-sonarqube > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
(Inherit From Job)
Path to project properties
Filter
Execute SonarQube Scanner
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Provide Configuration files
Run with timeout
Set build status to "pending" on GitHub commit
SonarScanner for MSBuild - Begin Analysis
SonarScanner for MSBuild - End Analysis
Add build step
Post-build Actions
Add post-build action
Save
Apply

Enter Analysis Properties

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



Execute SonarQube Scanner

JDK: (Inherit From Job)
JDK to be used for this SonarQube analysis

Path to project properties:

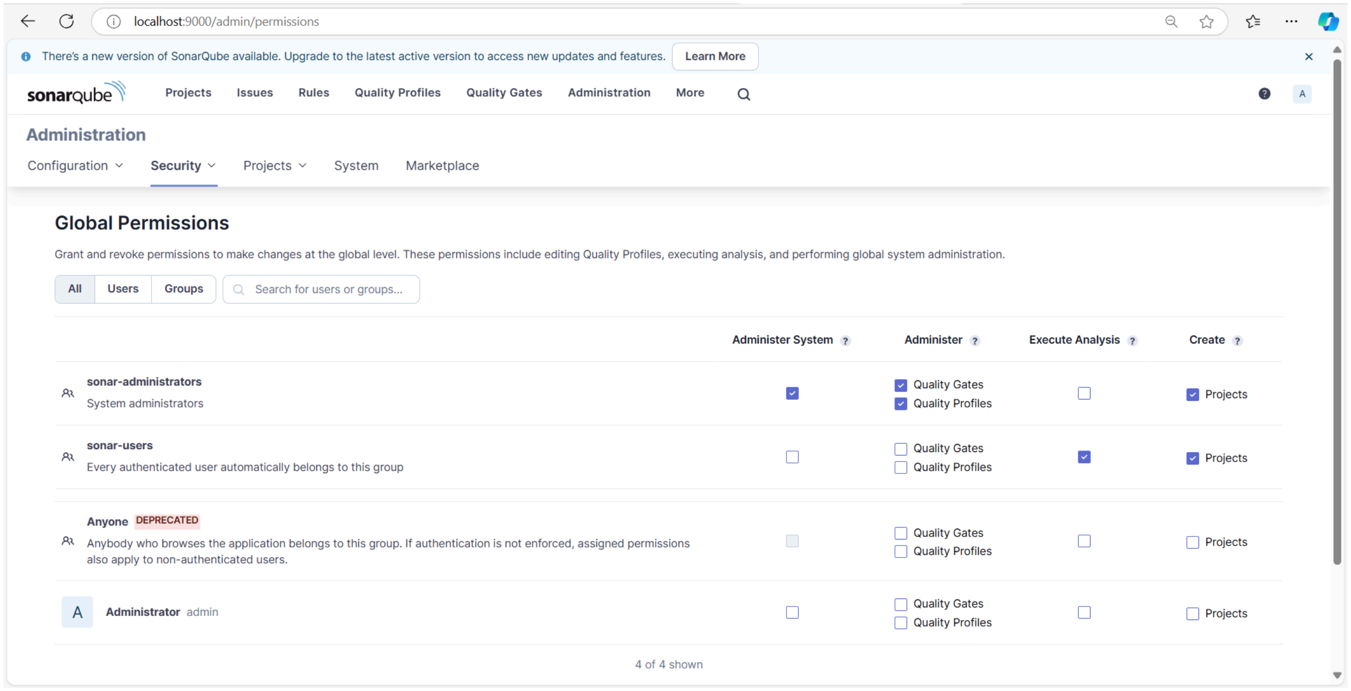
Analysis properties:

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

Additional arguments:

JVM Options:

Go to http://localhost:9000/admin/permissions and give the Admin user execute permissions.


localhost:9000/admin/permissions
There's a new version of SonarQube available. Upgrade to the latest active version to access new updates and features.
Learn More
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Administration
Configuration
Security
Projects
System
Marketplace
All
Users
Groups
Search for users or groups...

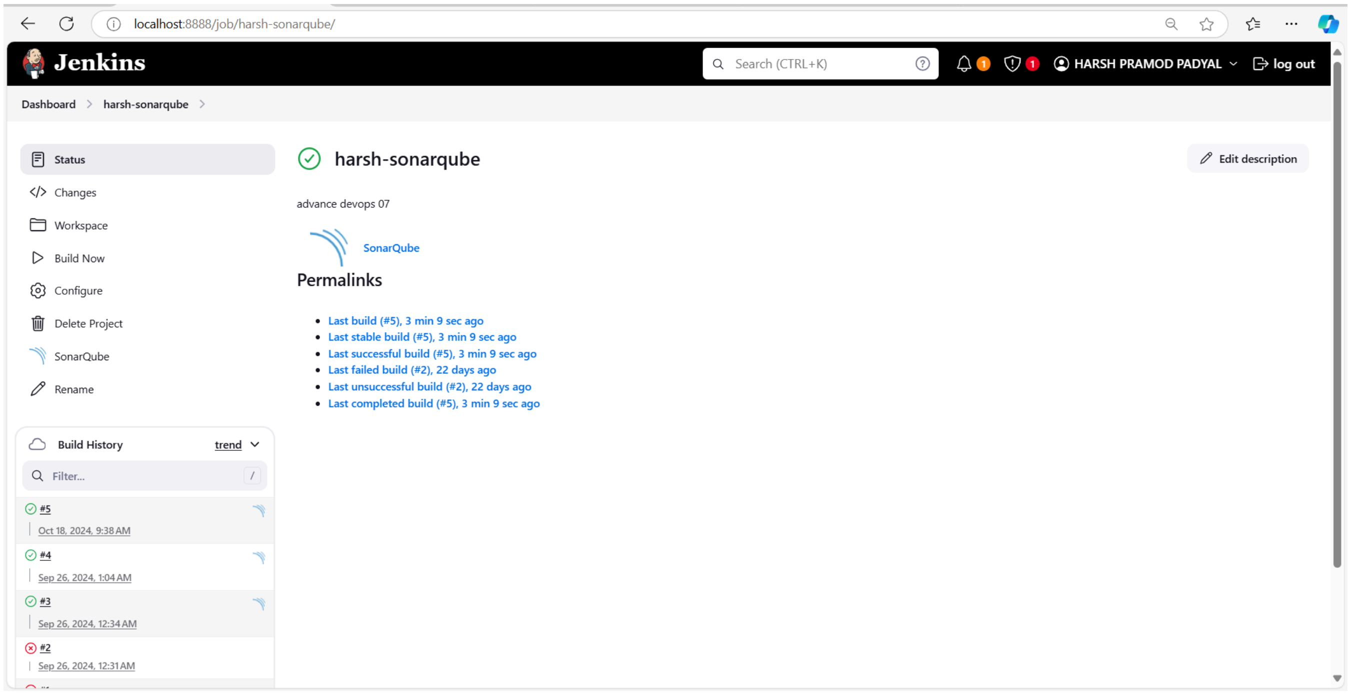
Global Permissions

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

	Administer System	Administer	Execute Analysis	Create
sonar-administrators System administrators	✓	✓ Quality Gates ✓ Quality Profiles	☐	✓ Projects
sonar-users Every authenticated user automatically belongs to this group	☐	☐ Quality Gates ☐ Quality Profiles	✓	✓ Projects
Anyone DEPRECATED Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.	☐	☐ Quality Gates ☐ Quality Profiles	☐	☐ Projects
Administrator admin	☐	☐ Quality Gates ☐ Quality Profiles	☐	☐ Projects

4 of 4 shown

Save the project configuration and click Build Now.



harsh-sonarqube

advance devops 07

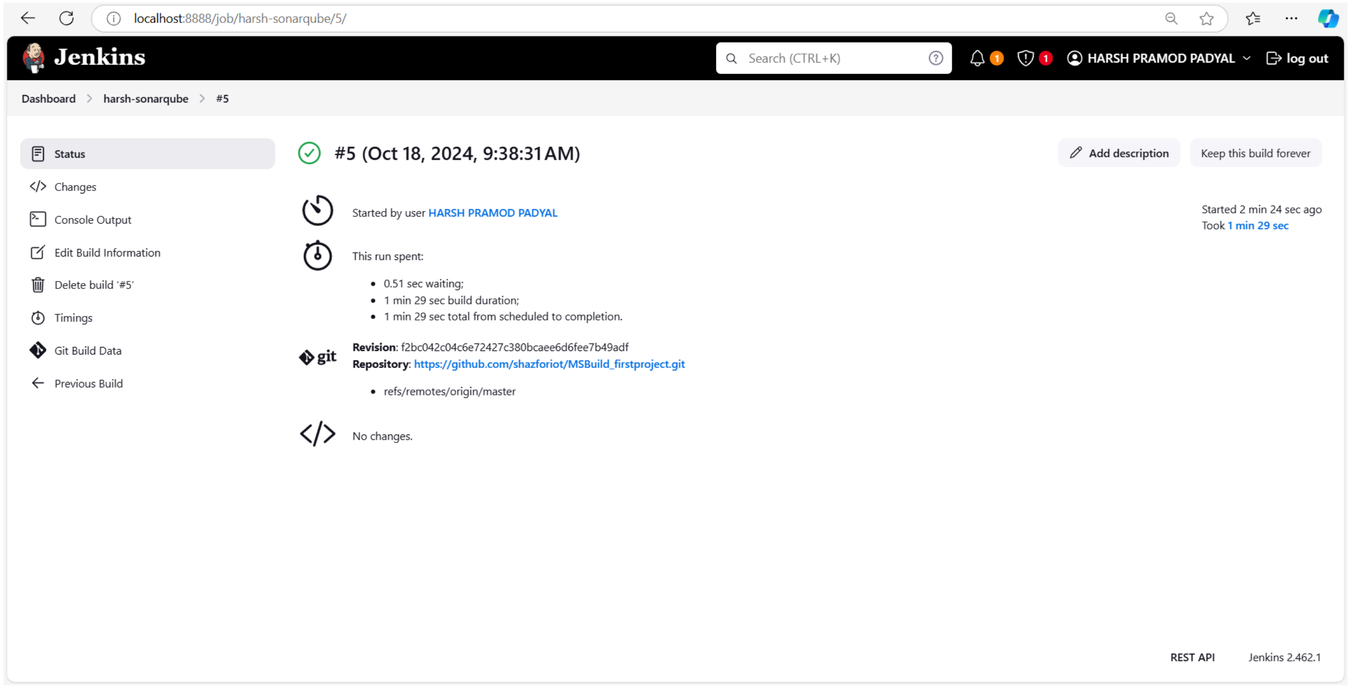
SonarQube

Permalinks

- Last build (#5), 3 min 9 sec ago
- Last stable build (#5), 3 min 9 sec ago
- Last successful build (#5), 3 min 9 sec ago
- Last failed build (#2), 22 days ago
- Last unsuccessful build (#2), 22 days ago
- Last completed build (#5), 3 min 9 sec ago

#5 Oct 18, 2024, 9:38 AM
#4 Sep 26, 2024, 1:04 AM
#3 Sep 26, 2024, 12:34 AM
#2 Sep 26, 2024, 12:31 AM

Click on the build number in the build history.



#5 (Oct 18, 2024, 9:38:31 AM)

Started by user HARSH PRAMOD PADYAL

Started 2 min 24 sec ago
Took 1 min 29 sec

This run spent:

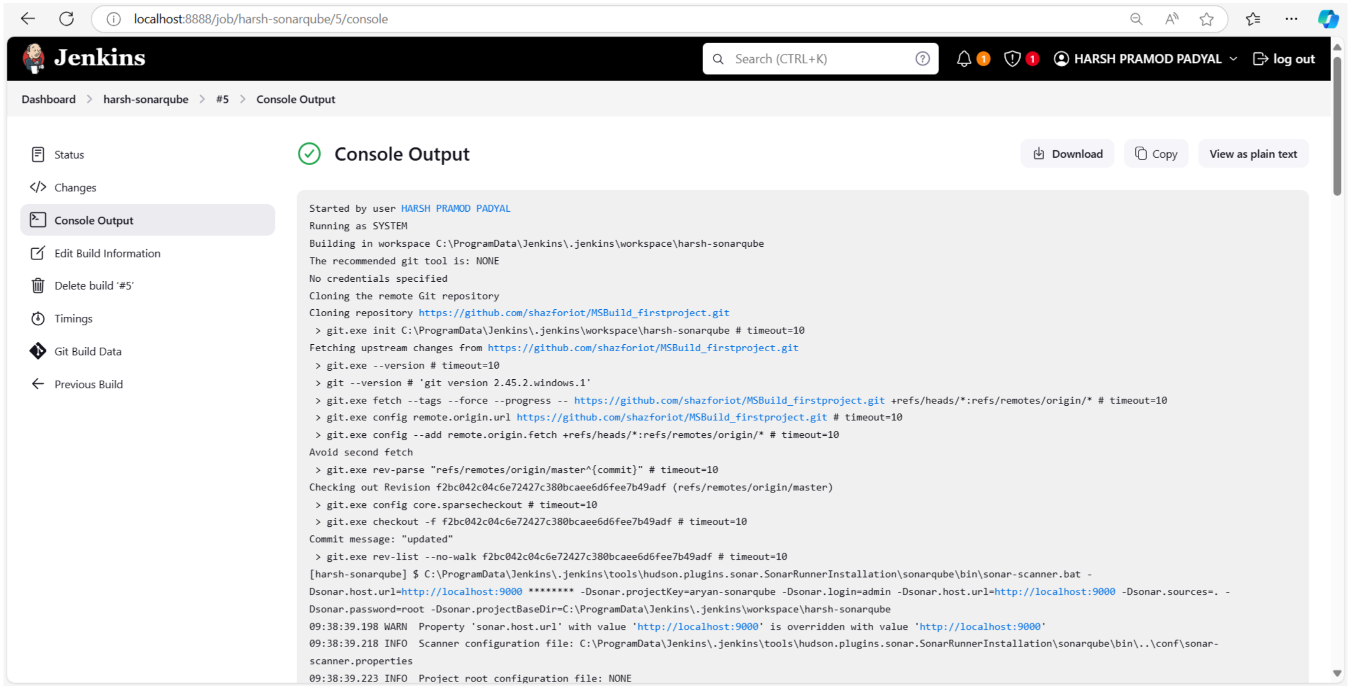
- 0.51 sec waiting;
- 1 min 29 sec build duration;
- 1 min 29 sec total from scheduled to completion.

Revision: f2bc042c04c6e72427c380bcaee6d6fee7b49adf
Repository: https://github.com/shazforiot/MSBuild_firstproject.git

- refs/remotes/origin/master

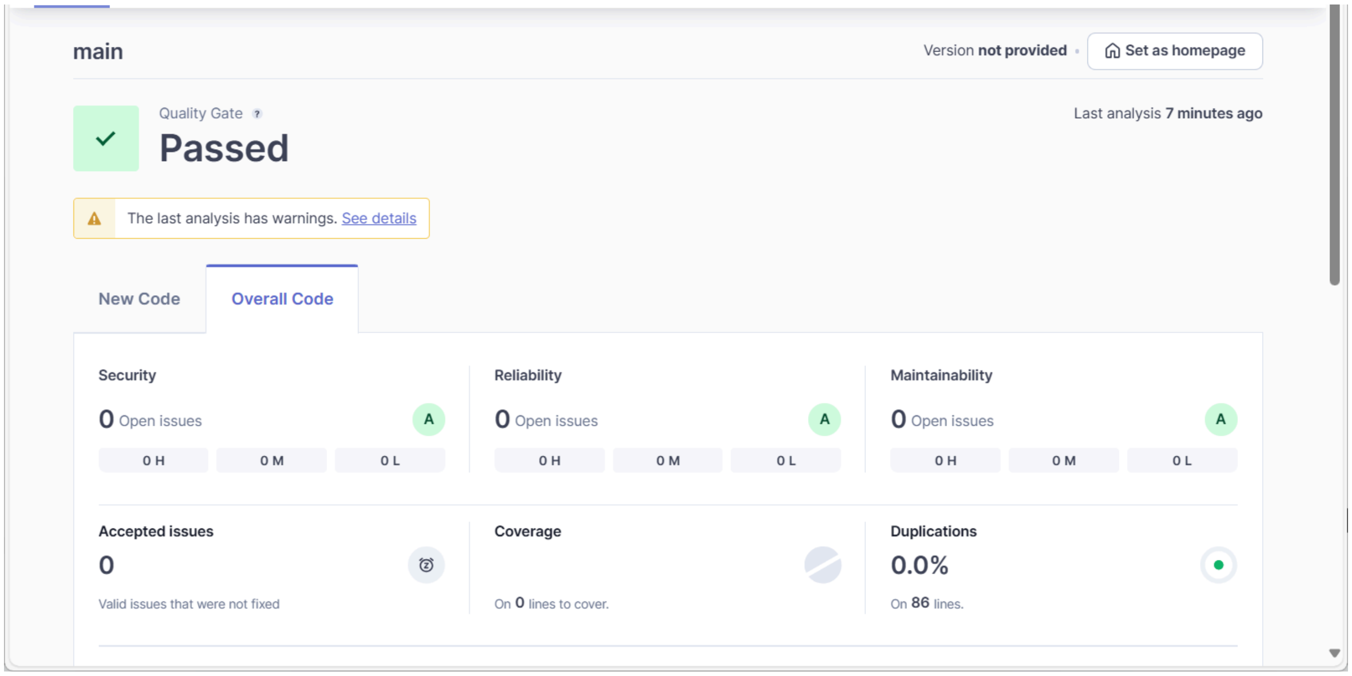
No changes.

Click on Console Output to see the build logs.



```
Started by user HARSH PRAMOD PADYAL
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\harsh-sonarqube
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\harsh-sonarqube # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > 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
[harsh-sonarqube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 ******** -Dsonar.projectKey=aryan-sonarqube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -
Dsonar.password=root -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\harsh-sonarqube
09:38:39.198 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
09:38:39.218 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-
scanner.properties
09:38:39.223 INFO  Project root configuration file: NONE
```

Go back to http://localhost:9000 and navigate to your project. Review the results of the analysis.



Security	Reliability	Maintainability
0 Open issues A	0 Open issues A	0 Open issues A
0 H 0 M 0 L	0 H 0 M 0 L	0 H 0 M 0 L

Accepted issues	Coverage	Duplications
0		0.0%
Valid issues that were not fixed	On 0 lines to cover.	On 86 lines.

CONCLUSION :

Integrating Static Application Security Testing (SAST) into the software development process helps identify and fix security vulnerabilities early, improving the overall quality of applications. By regularly running SAST tools, organizations can ensure safer code and reduce the risk of security issues in their final products.