

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

Docker Run docker -v command.g Use this command to check if docker is installed and running on your system.

```
C:\Users\2022k>docker -v
Docker version 27.1.1, build 6312585
```

Install SonarQube image Command: docker pull sonarqube This command helps you to install an image of SonarQube that can be used on the local system without actually installing the SonarQube installer.

```
C:\Users\2022k>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

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
C:\Users\2022k>
```

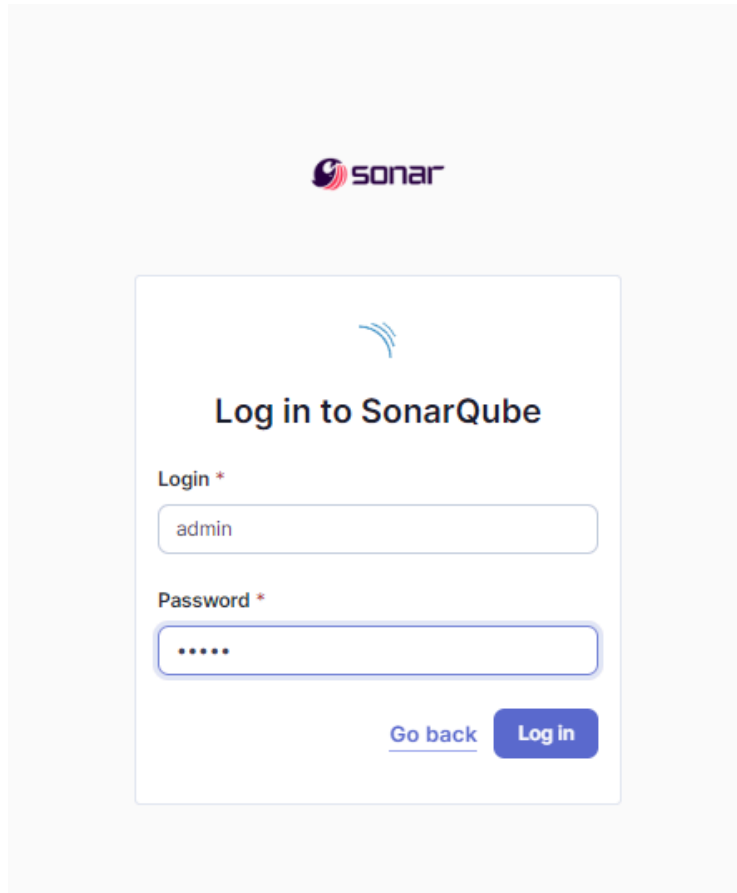
3) Keep jenkins installed on your system. Experiment Steps: Step 1: Run SonarQube image docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest This command will run the SonarQube image that was just installed using docker

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

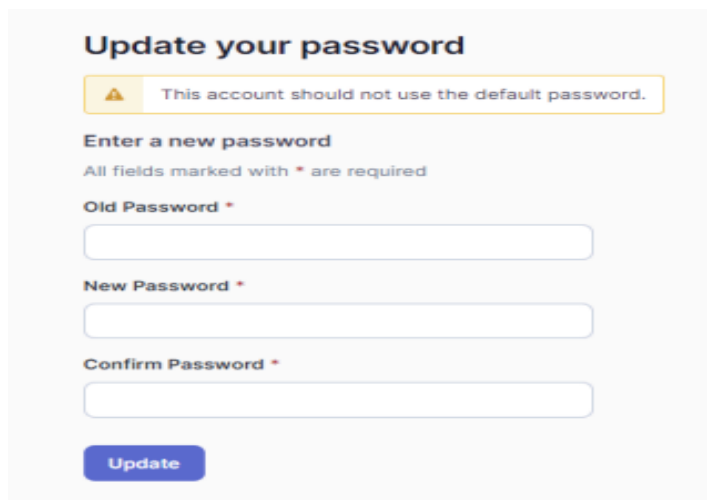
PS C:\Users\2022k> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Already exists
90a925ab929a: Already exists
7d9a34308537: Already exists
80338217a4ab: Already exists
1a5fd5c7e184: Already exists
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
6d247c43e5980146f4e62fe0c2a278fa537f94ff932607c5fa7975aa5009e472
PS C:\Users\2022k> |
```

Once the SonarQube image is started, you can go to <http://localhost:9000> to find the SonarQube that has started. Update this password later on.



The image shows the SonarQube login page. At the top is the Sonar logo. Below it is a white box with a blue Sonar icon and the text "Log in to SonarQube". There are two input fields: "Login *" with the text "admin" and "Password *" with six dots. At the bottom right of the box are two buttons: "Go back" (a link) and "Log in" (a blue button).

On this interface, login with username = 'admin' and password = 'admin'. Once logged in successfully, SonarQube will ask you to reset this password. Reset it and remember this password.



The image shows the "Update your password" page. At the top is the title "Update your password". Below it is a yellow warning box with a triangle icon and the text "This account should not use the default password.". Underneath is the heading "Enter a new password" and a note "All fields marked with * are required". There are three input fields: "Old Password *" (empty), "New Password *" (empty), and "Confirm Password *" (empty). At the bottom is a blue "Update" button.


Once logged in successfully, SonarQube will ask you to reset this password. Reset it and remember this password then create a Project .
Give the project a display name and project key. Setup the project and create.

[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

Set up the project as required and click on create.

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

☐ 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.

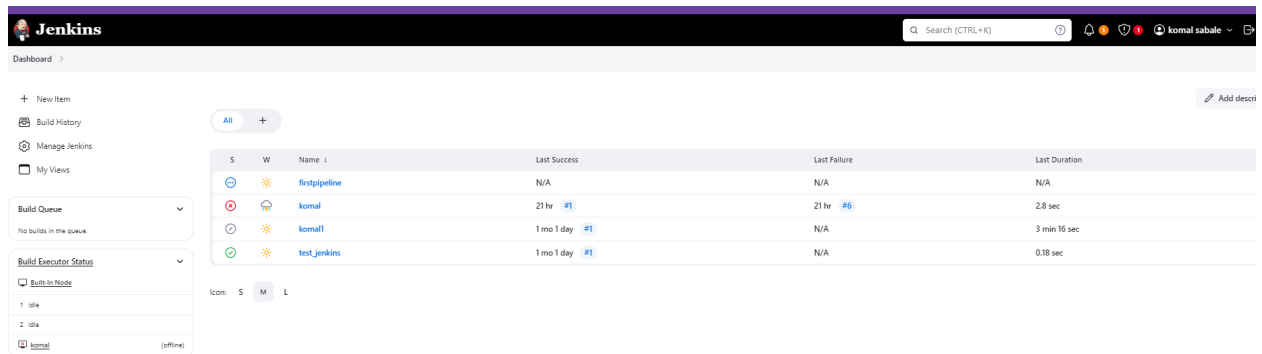
☐ Reference branch

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

Back

Create project

Step 5: Open Jenkins on whichever port it is installed. (http://localhost:).



The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with links: New Item, Build History, Manage Jenkins, and My Views. Below these are sections for Build Queue (showing 'No builds in the queue') and Build Executor Status (showing two idle executors named '1' and '2', and one named 'komal' which is offline). The main area displays a table of build jobs. The table has columns for status, name, last success, last failure, and last duration. The jobs listed are 'firstpipeline', 'komal', 'komal!', and 'test_jenkins'. The 'komal' job is highlighted with a red status icon and shows a last failure 21 hours ago.

Status	Name	Last Success	Last Failure	Last Duration
Success	firstpipeline	N/A	N/A	N/A
Failure	komal	21 hr #1	21 hr #6	2.8 sec
Success	komal!	1 mo 1 day #1	N/A	3 min 16 sec
Success	test_jenkins	1 mo 1 day #1	N/A	0.18 sec

Go to manage jenkins → Search for Sonarqube Scanner for Jenkins and install it.



The screenshot shows the 'Manage Jenkins' page, specifically the 'Plugins' section. A search bar at the top contains the text 'sonarqube'. Below the search bar, there's a table of installed and available plugins. The table has columns for 'Install', 'Name', and 'Released'. The 'SonarQube Scanner' plugin is listed with version 2.7.7.2 and a release date of 7 mo 9 days ago. It has a description: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' There are also links for 'External Site/Tool Integrations' and 'Build Reports'.

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.7.7.2 External Site/Tool Integrations Build Reports	7 mo 9 days ago
<input type="checkbox"/>	SonarQube Scanner 3.0.0.0 External Site/Tool Integrations	3 mo 23 days ago

Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.



The screenshot shows the 'System Configuration' page in Jenkins, specifically the 'SonarQube Servers' section. At the top, there's a checkbox for 'Environment variables' which is checked. Below this, there's a section for 'SonarQube installations' with a list of installations. A form is shown for adding a new installation. The form has fields for 'Name' (containing 'sonarqube'), 'Server URL' (with a default value of 'http://localhost:9000'), and 'Server authentication token' (with a dropdown menu set to 'none'). There's an 'Add' button and an 'Advanced' dropdown menu.

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


Server authentication token: SonarQube authentication token. Mandatory when anonymous access is disabled. - none -

+ Add +


Advanced

Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose install automatically.


SonarQube Scanner installations


SonarQube Scanner installations ^  Edited

Add SonarQube Scanner

☰ SonarQube Scanner 

Name

☒ Install automatically 

☰ Install from Maven Central 









Version

Add Installer

▼

Add SonarQube Scanner

After configuration, create a New Item → choose a freestyle project.


 **Jenkins**      komal sabale   log out


Dashboard > All > New Item


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.

 **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.

If you want to create a new item from other existing, you can use this option:

Copy from

OK

Use this github repository in Source Code Management.
https://github.com/komalsabale2815/MSBuild_firstproject.
It is a sample hello-world project with no vulnerabilities

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

Advanced ▾

Under Build Steps, enter Sonarqube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

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

Path to project properties ?

Analysis properties ?

Additional arguments ?

JVM Options ?

Add build step ▾

Post-build Actions

Add post-build action ▾

Now, you need to grant the local user (here admin user) permissions to Execute the Analysis stage on SonarQube. For this, go to <http://localhost:admin/permissions> and check the 'Execute Analysis' checkbox under Administrator.


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.

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

4 of 4 shown

Step 13: Go back to Jenkins. Go to the job you had just built and click on Build Now.

**Jenkins**

Search (CTRL+K) ?

log out

Dashboard > sonarqube >

Status

Changes

Workspace

Build Now


Configure

Delete Project

SonarQube

Rename

sonarqube

 SonarQube

Permalinks

- Last build (#7), 33 min ago
- Last stable build (#7), 33 min ago
- Last successful build (#7), 33 min ago
- Last failed build (#6), 34 min ago
- Last unsuccessful build (#6), 34 min ago
- Last completed build (#7), 33 min ago


Build History

trend

Filter...

#7 26 Sept 2024, 18:25

Check the console Output

 **Jenkins**

Search (CTRL+K) ?

🔔 1 🛡️ 1 👤 komal sabale 🚪 log out

Dashboard > sonarqube > #7 > Console Output

📄 Status

</> Changes

📄 Console Output

✏️ Edit Build Information

🗑️ Delete build '#7'

🕒 Timings

📊 Git Build Data

⬅️ Previous Build

✅ Console Output

Download Copy View as plain text

```
Started by user komal sabale
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/komalsabale2815/MSBuild_firstproject #
timeout=10
Fetching upstream changes from https://github.com/komalsabale2815/MSBuild_firstproject
> git.exe --version # timeout=10
> git --version # 'git version 2.42.0.windows.2'
> git.exe fetch --tags --force --progress --
https://github.com/komalsabale2815/MSBuild_firstproject +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
[sonarqube] $
C:\ProgramData\Jenkins\jenkins\tools\udson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\s
onar-scanner.bat -Dsonar.projectKey=sonarqube -
Dsonar.login=squ_8dbedadf1bcd6fa691911f5eef0092088ea13a58 -Dsonar.host.url=http://localhost:9000 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube -Dsonar.sources=. -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
18:25:32.129 INFO Scanner configuration file:
C:\ProgramData\Jenkins\jenkins\tools\udson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\
.\conf\sonar-scanner.properties
18:25:32.138 INFO Project root configuration file: NONE
18:25:32.176 INFO SonarScanner CLI 6.2.0.4584
18:25:32.179 INFO Java 21 Oracle Corporation (64-bit)
18:25:32.188 INFO Windows 11 10.0 amd64
18:25:32.244 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
18:25:35.304 INFO JRE provisioning: os[windows], arch[amd64]
18:25:35.552 INFO Communicating with SonarQube Server 10.6.0.92116
18:25:36.377 INFO Starting SonarScanner Engine...
18:25:36.377 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
18:25:40.113 INFO Load global settings
18:25:40.310 INFO Load global settings (done) | time=195ms
18:25:40.323 INFO Server id: 1478411E-AZIpjKX8vBvle7fi6uzc
```


Once the build is complete, go back to SonarQube and check the project linked.

The screenshot shows the SonarQube interface for a project named 'main'. The top navigation bar includes 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. The 'Overview' tab is selected. The project name 'main' is displayed at the top left, with a dropdown menu showing 'main' and a green checkmark. To the right, it says 'Version not provided' and 'Set'. Below the project name, there is a green checkmark icon and the text 'Quality Gate Passed'. A warning message states 'The last analysis has warnings. See details'. The main content area is divided into two tabs: 'New Code' and 'Overall Code'. The 'Overall Code' tab is active. The 'Overall Code' tab displays several metrics: 'Security' with '0 Open issues' and a green 'A' grade; 'Reliability' with '0 Open issues' and a green 'A' grade; 'Maintainability' with '0 Open issues'; 'Accepted issues' with '0' and a clock icon; 'Coverage' with 'On 0 lines to cover.' and a pie chart; 'Duplications' with '0.0%' and 'On 86 lines.'; and 'Security Hotspots' with '0' and a green 'A' grade. The bottom section is titled 'Activity' and has a dropdown menu set to 'Issues'.

sonarqube / main main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings

main Version not provided Set

Quality Gate ? **Passed** Last analysis

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

New Code Overall Code

Security
0 Open issues A
0 H 0 M 0 L

Reliability
0 Open issues A
0 H 0 M 0 L

Maintainability
0 Open issues
0 H 0 M

Accepted issues
0 🕒
Valid issues that were not fixed

Coverage
📊
On 0 lines to cover.

Duplications
0.0%
On 86 lines.

Security Hotspots
0 A

Activity

Issues ▼