



VASANTDADA PATIL PRATISHTHAN'S COLLEGE OF ENGINEERING AND VISUAL ARTS

ISO 9001:2015 Certified Institute

Department of Information Technology

NBA Accredited Course (Dated 01/07/2024 to 30/06/2027)

EXPERIMENT - 7

Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to Sonar Qube/Git Lab.

Theory:

- SAST:

Static Application Security Testing is an automated process that analyzes source code or compiled versions of code to identify potential security vulnerabilities. It's "static" because it examines the code without executing it.

- Jenkins:

Jenkins is an open-source automation server that enables developers to reliably build, test, and deploy their software. It is a server-based system that runs in servlet containers such as Apache Tomcat. It is an CI/CD tool used in most of the industries.

- SonarQube:

SonarQube is an open-source platform for continuous inspection of code quality, developed by SonarSource. It provides automated analysis and integration with various development tools, enabling developers to write clean, readable, and understandable code.

Integrating SAST into Jenkins allows for automated security checks as part of your build process. Also It can be integrated with SonarQube to provide detailed code analysis reports.

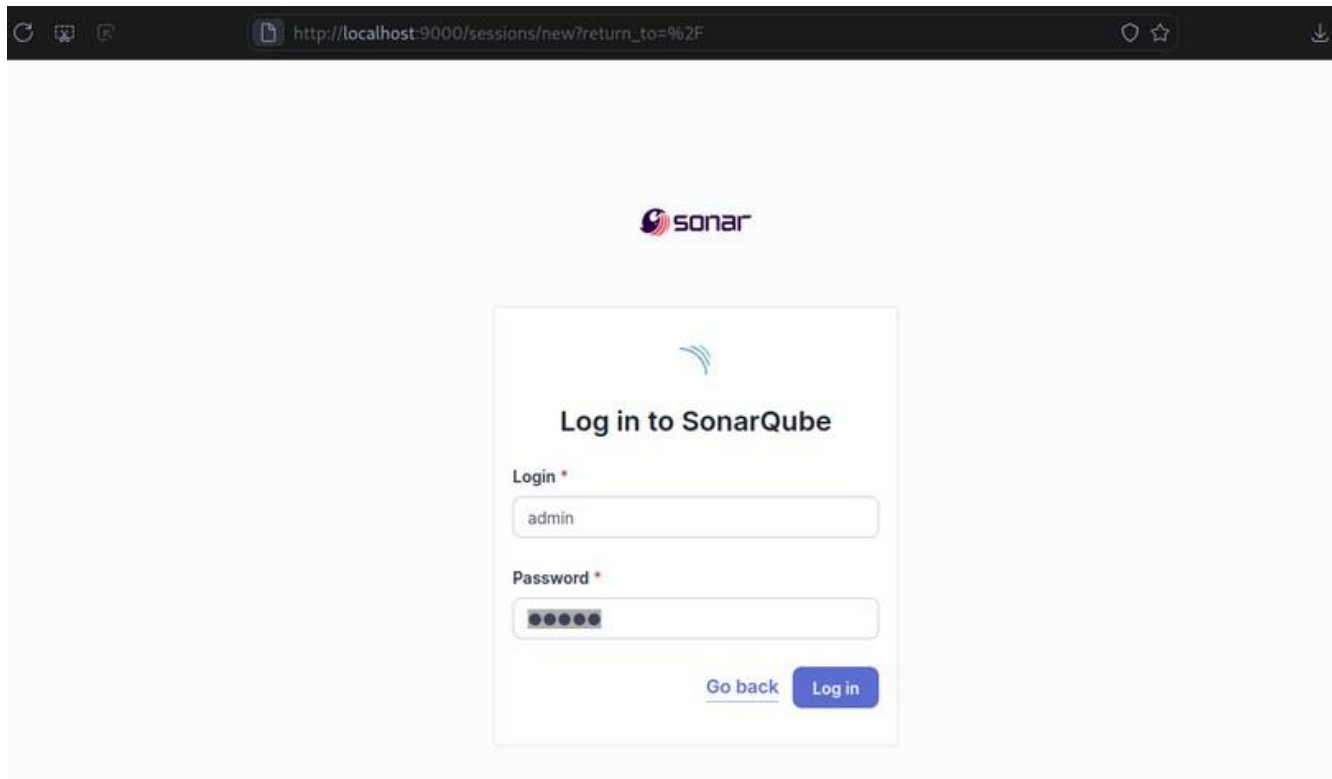
STEPS:

1. First you need to go with the installation part.

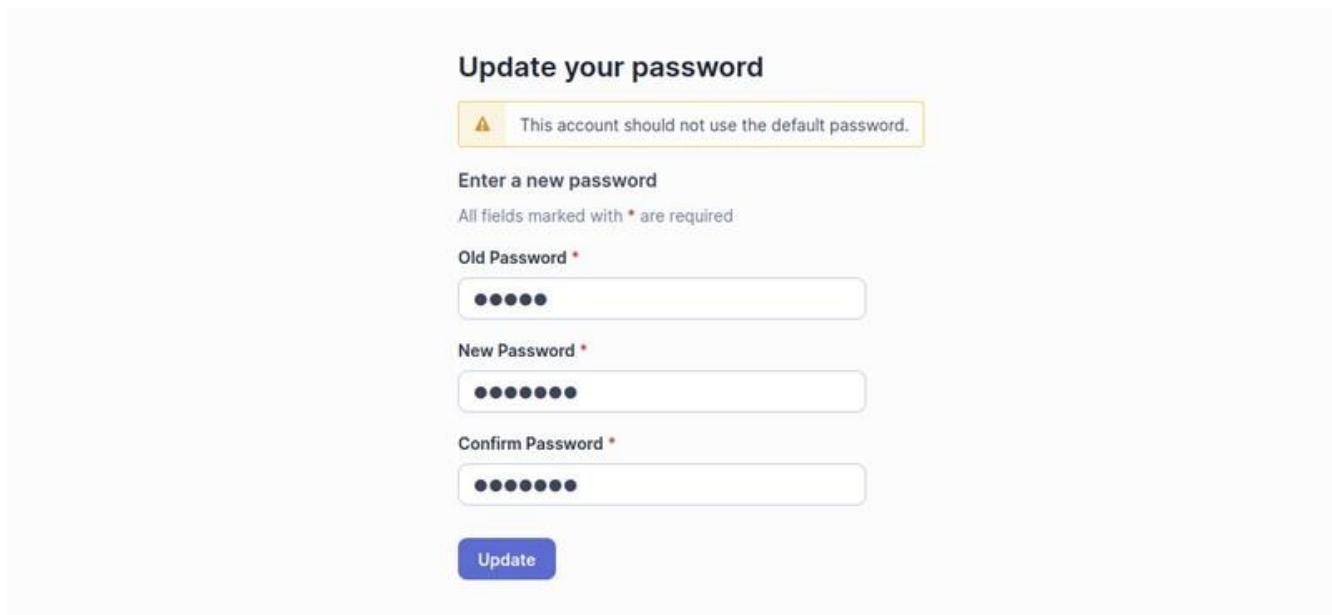
```
> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

I have used docker, zip file also works well.

2. Then access it on localhost:9000



The screenshot shows a web browser window with the address bar displaying `http://localhost:9000/sessions/new?return_to=%2F`. The page features the Sonar logo at the top center. Below it is a white box containing the text "Log in to SonarQube" with a small blue icon above it. There are two input fields: "Login *" with the text "admin" and "Password *" with masked characters. At the bottom right of the box are two buttons: "Go back" (a link) and "Log in" (a blue button).



The screenshot shows the "Update your password" page. At the top, there is a yellow warning box with a triangle icon and the text "This account should not use the default password." Below this is the heading "Enter a new password" followed by the note "All fields marked with * are required". There are three input fields: "Old Password *" with masked characters, "New Password *" with masked characters, and "Confirm Password *" with masked characters. At the bottom is a blue "Update" button.

3. Now login to your jenkins dashboard and install SonarQube Scanner plugin.

The screenshot shows the Jenkins 'Plugins' page. The top navigation bar includes the Jenkins logo and a search bar with the text 'Search (CTRL+K)'. Below the navigation bar, the breadcrumb trail reads 'Dashboard > Manage Jenkins > Plugins'. On the left side, there is a sidebar with links: 'Updates', 'Available plugins' (highlighted), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main content area has a search bar containing 'sonarqube'. Below the search bar, there is a table of plugins with columns 'Install' and 'Name'. The table lists three plugins: 'SonarQube Scanner 2.17.2', 'Sonar Gerrit 388.v9b_f1cb_e42306', and 'SonarQube Generic Coverage 1.0'. Each plugin entry includes a checkbox in the 'Install' column and a brief description in the 'Name' column.

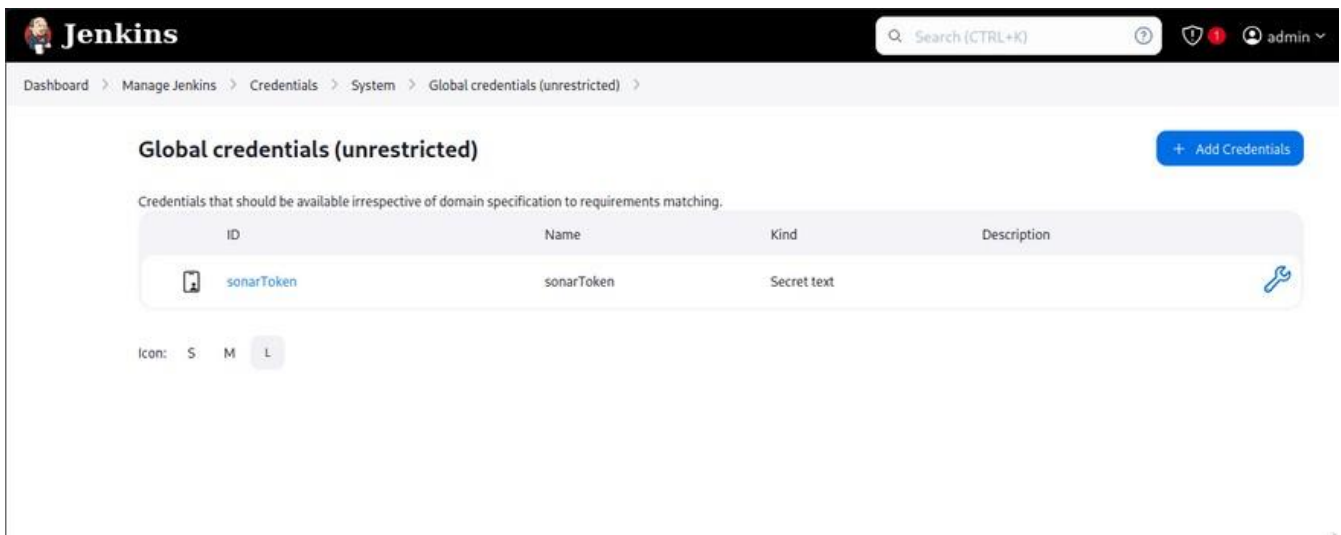
Install	Name ↓
<input type="checkbox"/>	SonarQube Scanner 2.17.2 External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.
<input type="checkbox"/>	Sonar Gerrit 388.v9b_f1cb_e42306 External Site/Tool Integrations This plugin allows to submit issues from SonarQube to Gerrit as comments directly.
<input type="checkbox"/>	SonarQube Generic Coverage 1.0 TODO

4. Then Go to SonarQube dashboard → Account settings → Security → Generate Token. (Make sure you have an project, if not then create)

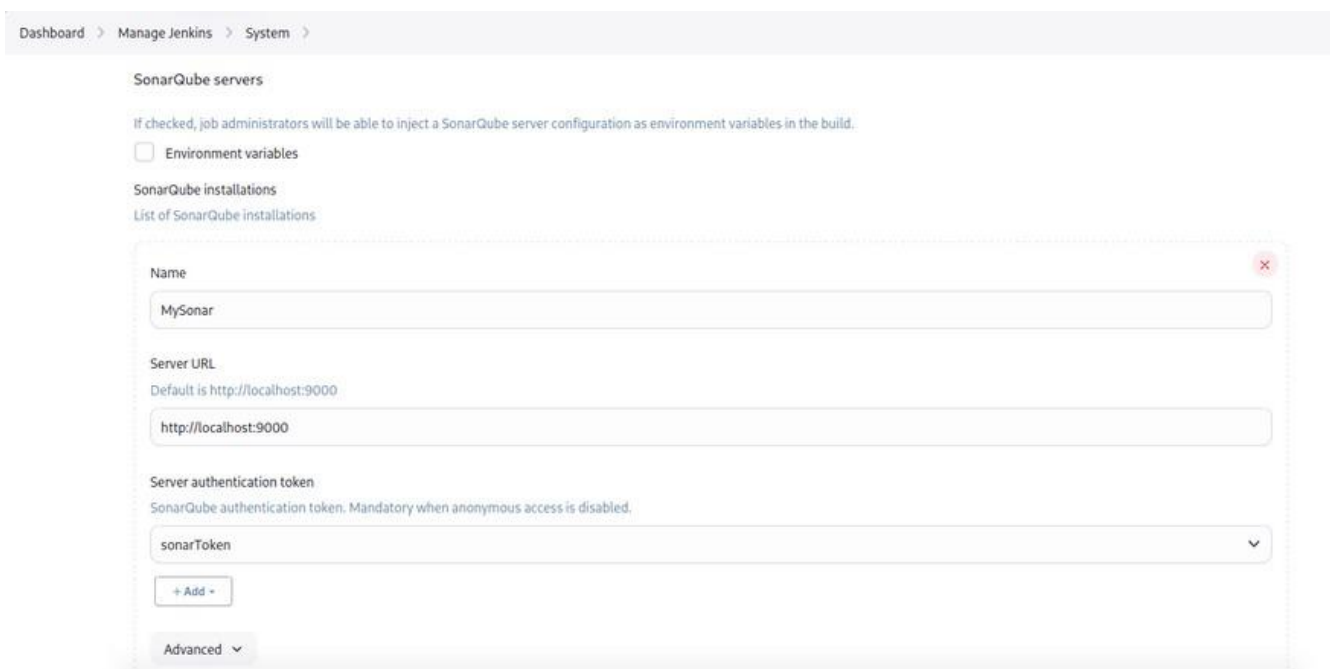
The screenshot shows the SonarQube 'Administrator' page, specifically the 'Security' tab. The page has a header with 'A Administrator' and a sub-header with 'Profile', 'Security' (highlighted), 'Notifications', and 'Projects'. The main content area is titled 'Security' and contains a paragraph explaining the purpose of tokens. Below this, there is a section titled 'Generate Tokens' with a table of input fields and a 'Generate' button. The table has columns: 'Name', 'Type', 'Project', and 'Expires in'. The 'Name' field contains 'sonarToken', the 'Type' field is a dropdown menu with 'Project Analysis Token' selected, the 'Project' field is a dropdown menu with 'new' selected, and the 'Expires in' field is a dropdown menu with '30 days' selected. The 'Generate' button is a blue button with white text. Below the table, there is an 'Advanced' link with a caret icon. At the bottom of the page, there are two buttons: 'Save' (blue) and 'Apply' (grey).

Name	Type	Project	Expires in	
sonarToken	Project Analysis Token	new	30 days	Generate

5. Once token is generated add it to Jenkins Credentials as a secret. (Global)



6. Then, Add your SonarQube server with jenkins, and add that credential here and save.



7. Now, setup the SonarQube scanner installations, If you want you can go with manual configuration or just tick "Install Automatically".

Dashboard > Manage Jenkins > Tools

SonarQube Scanner installations

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

☰ SonarQube Scanner

Name

MySonar

☒ Install automatically ?

☰ Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner

8. Start new item, with type "Freestyle project".

New Item

Enter an item name

testline

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.

9. Select your source code repository if it on any git server.

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/Dark-Kernel/node-api

10. You need to create new file 'sonar-project.properties' with following configuration:

```
sonar.projectKey=new
sonar.sources=.
sonar.exclusions=**/node_modules--*,**/*.spec.js
sonar.tests=.
sonar.test.inclusions=**/*.spec.js
sonar.javascript.lcov.reportPaths=coverage/lcov.info
```

Or just add it in Build step.



Execute SonarQube Scanner

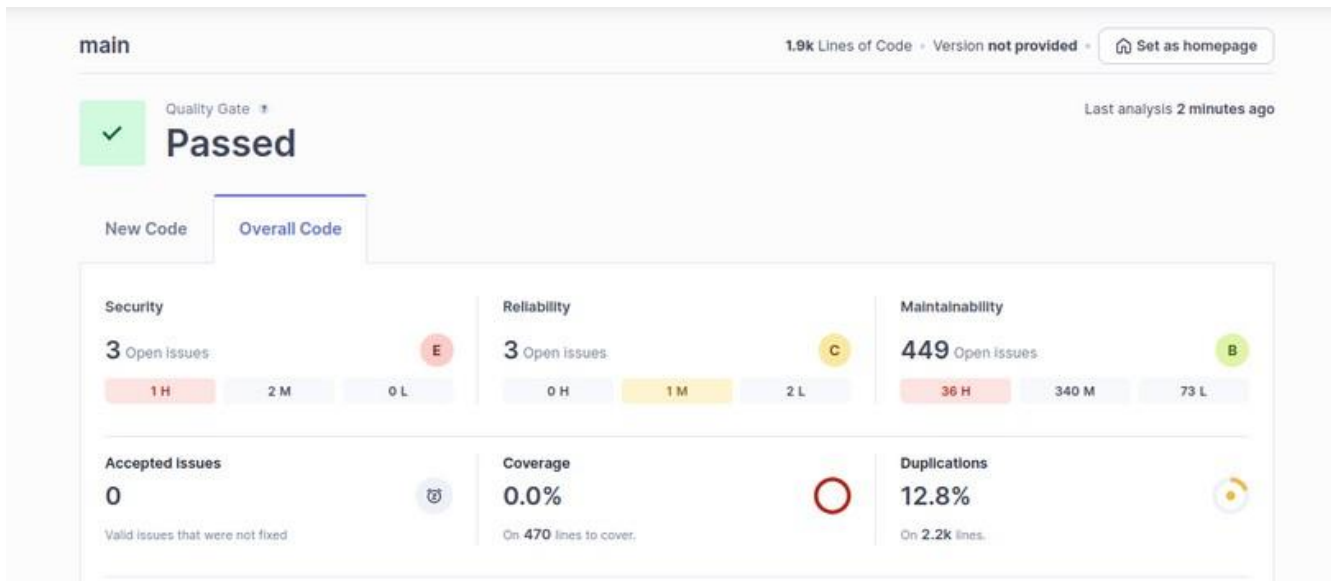
JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=new
sonar.sources=.
sonar.exclusions=**/node_modules--*,**/*.spec.js
sonar.tests=.
sonar.test.inclusions=**/*.spec.js
sonar.javascript.lcov.reportPaths=coverage/lcov.info
```

11. Build the project, then go to sonarQube project.



Conclusion: Thus, we have successfully understood Static Analysis SAST process and learned to integrate Jenkins SAST to SonarQube/GitLab.