Anushka Shahane D15A 55

EXPERIMENT 8 - ADVANCE DEVOPS

Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web /

Theory:
What is SAST?
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.
What is a CI/CD Pipeline?

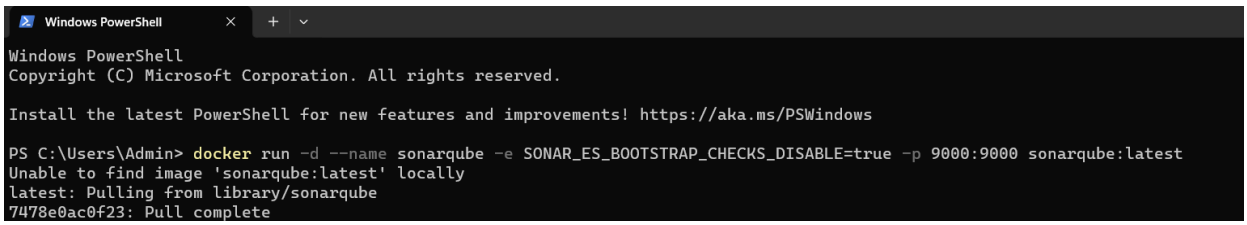
.
What is SonarQube
SonarQube is an open-source platform developed by SonarSource for continuous inspection of
code quality. Sonar does static code analysis, which provides a detailed report of bugs, code
smells, vulnerabilities, code duplications.

It supports 25+ major programming languages through built-in rulesets and can also be extended
with various plugins.

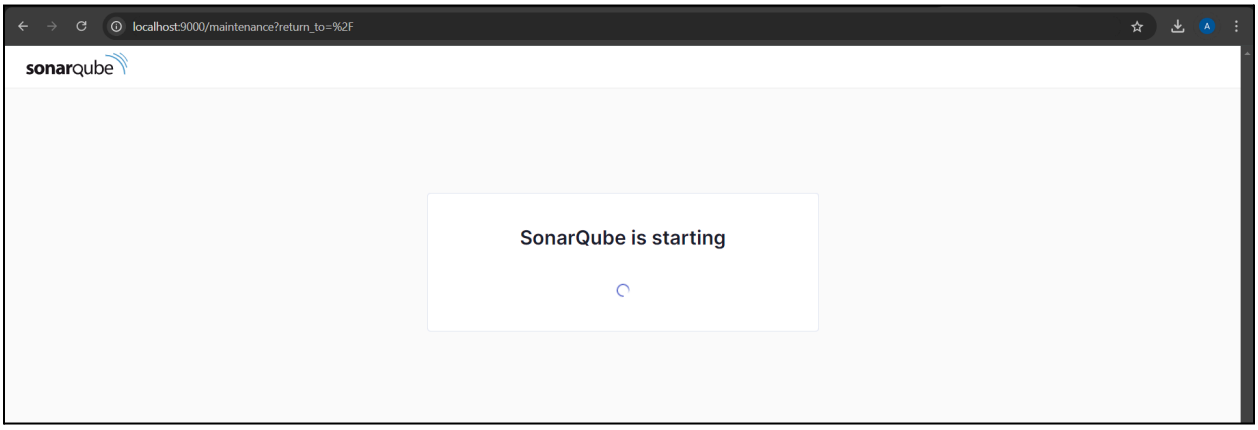
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

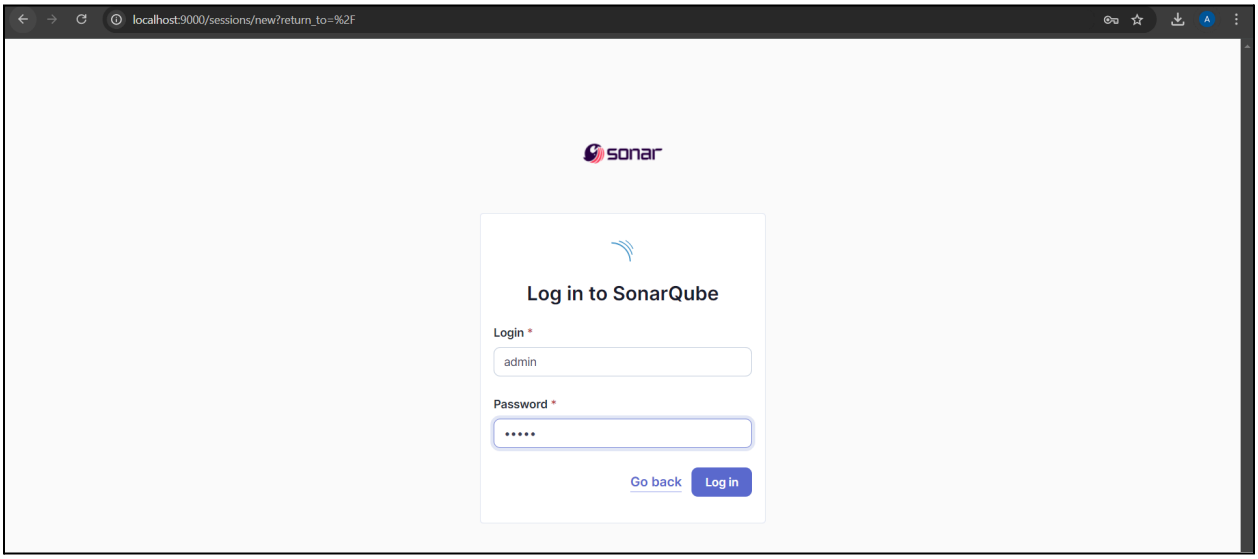
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.

2. Run SonarQube in a Docker container using this command -



3. Once the container is up and running, you can check the status of SonarQube at localhost
port 9000.

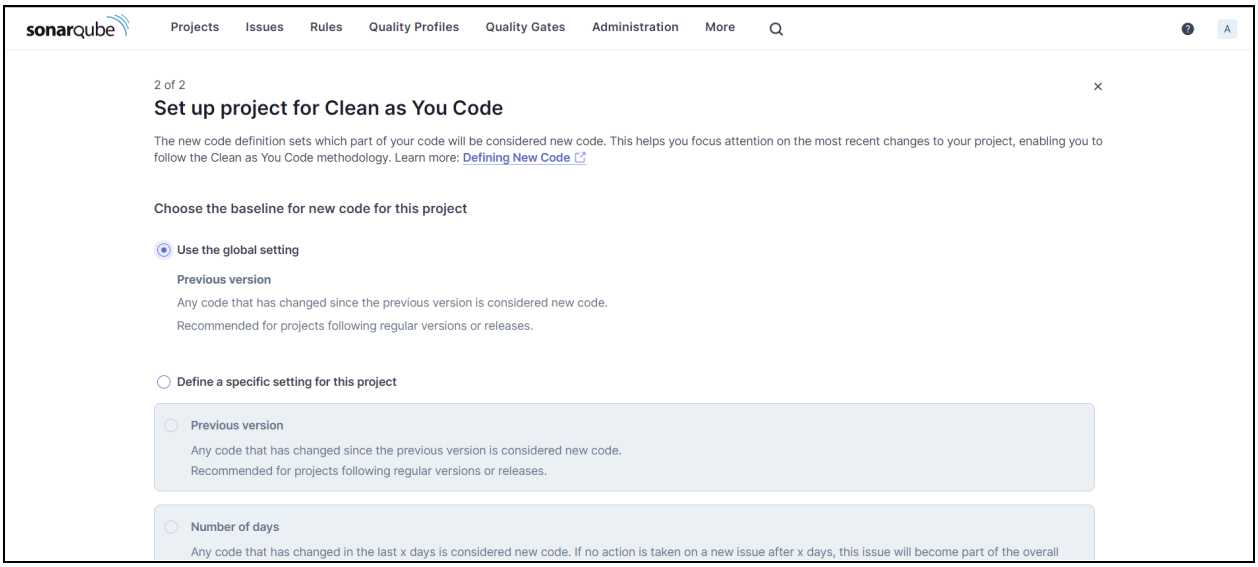


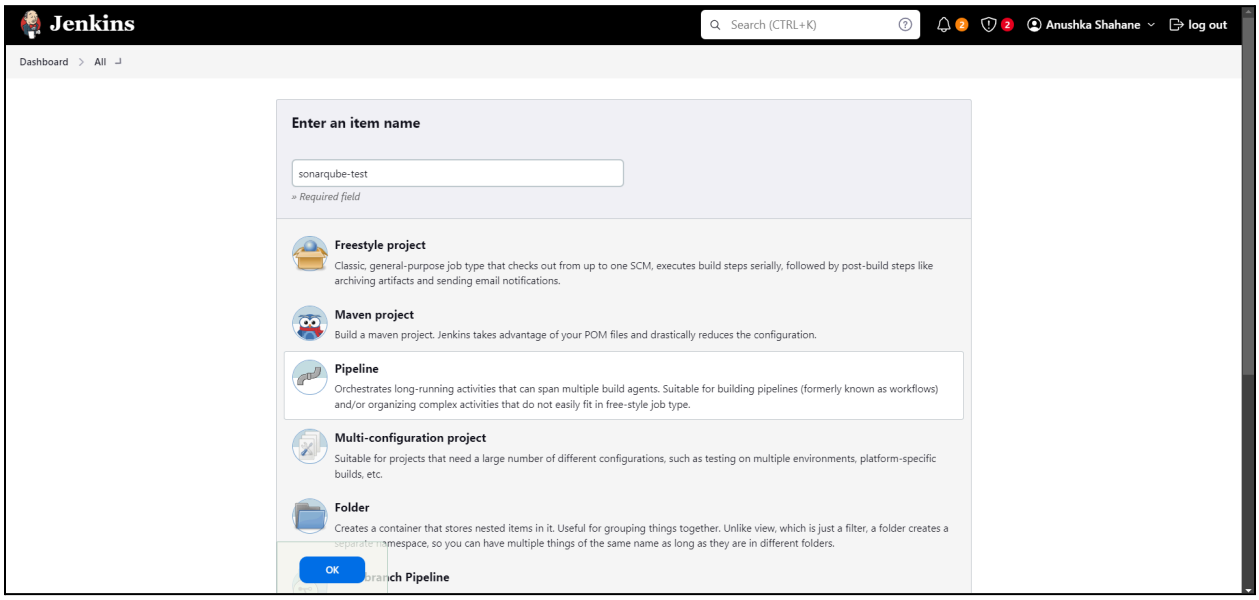


4. Login to SonarQube using username admin and password admin.

5. Create a manual project in SonarQube with the name sonarqube-test

Anushka Shahane D15A 55

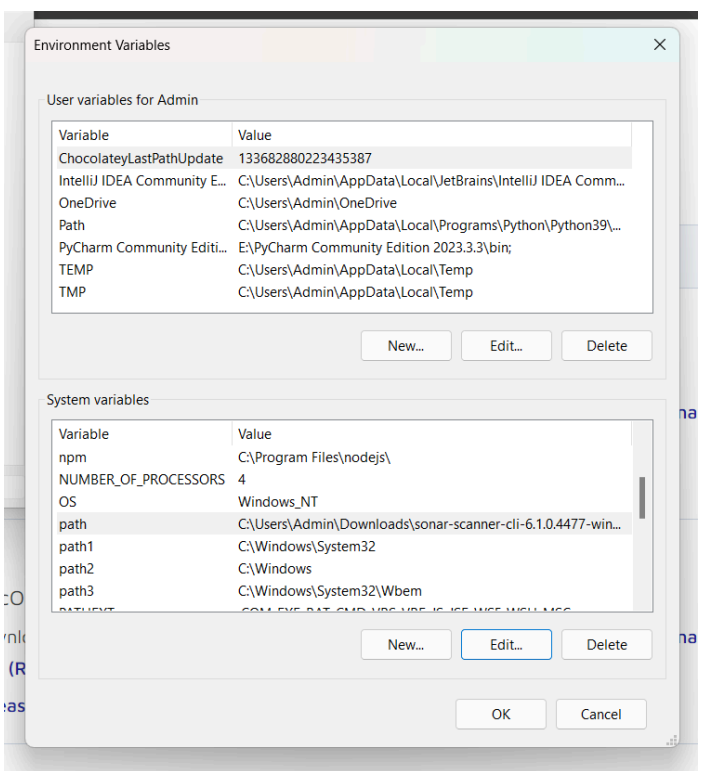




Anushka Shahane D15A 55

Step 6 :Go to download_sonarscanner to download sonar scanner

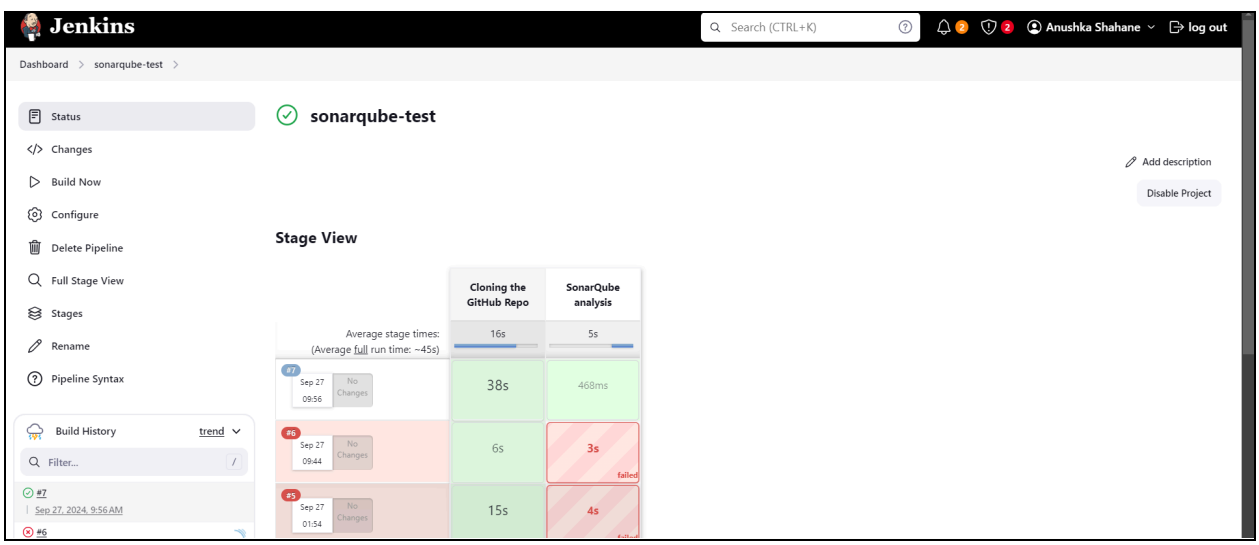
Step 7: After the download is complete, extract the file and copy the path to bin folder
Go to environment variables, system variables and click on path
Add a new path, paste the path copied earlier.



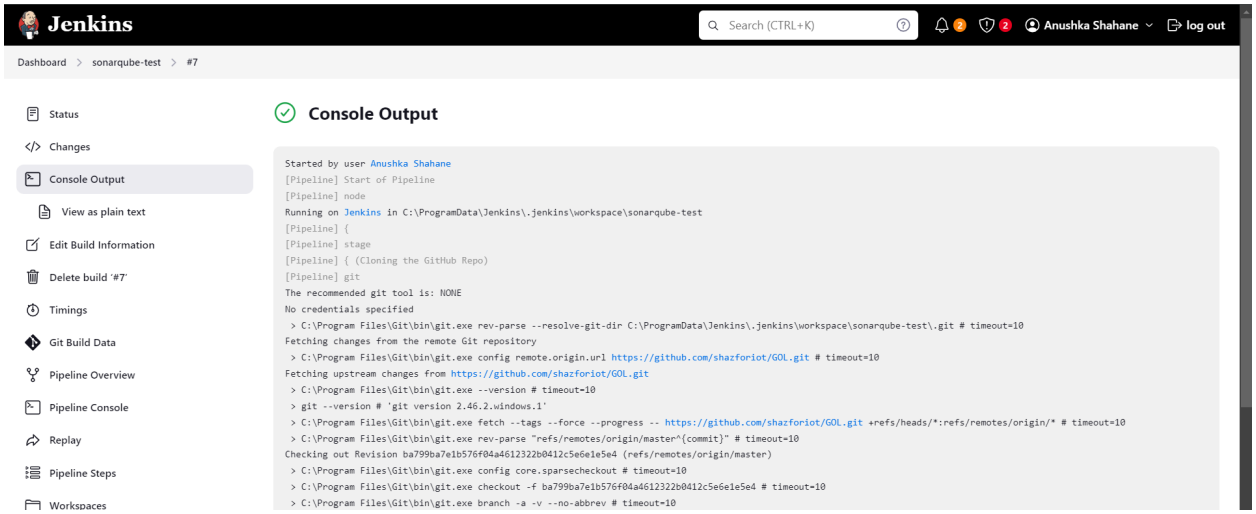
Anushka Shahane D15A 55

Step 8 : Save the pipeline and build it.

Output :



Anushka Shahane D15A 55



Under different tabs, check all different issues with the code.