Veydant Sharma D15C 50 AdvDevops

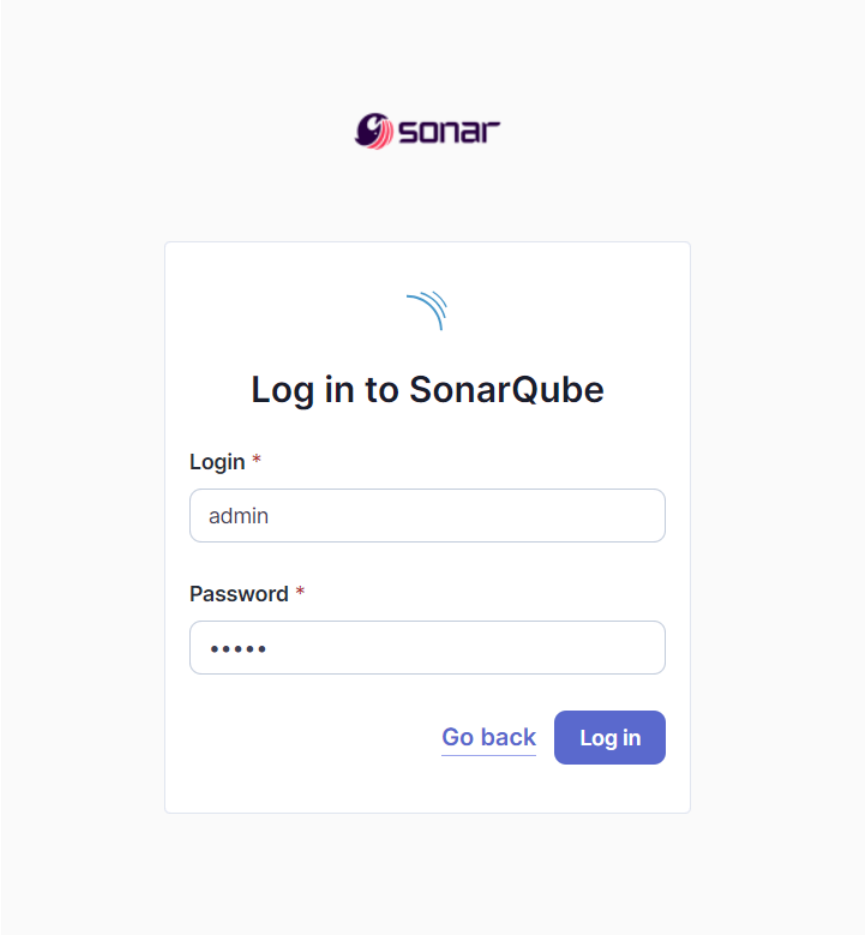
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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

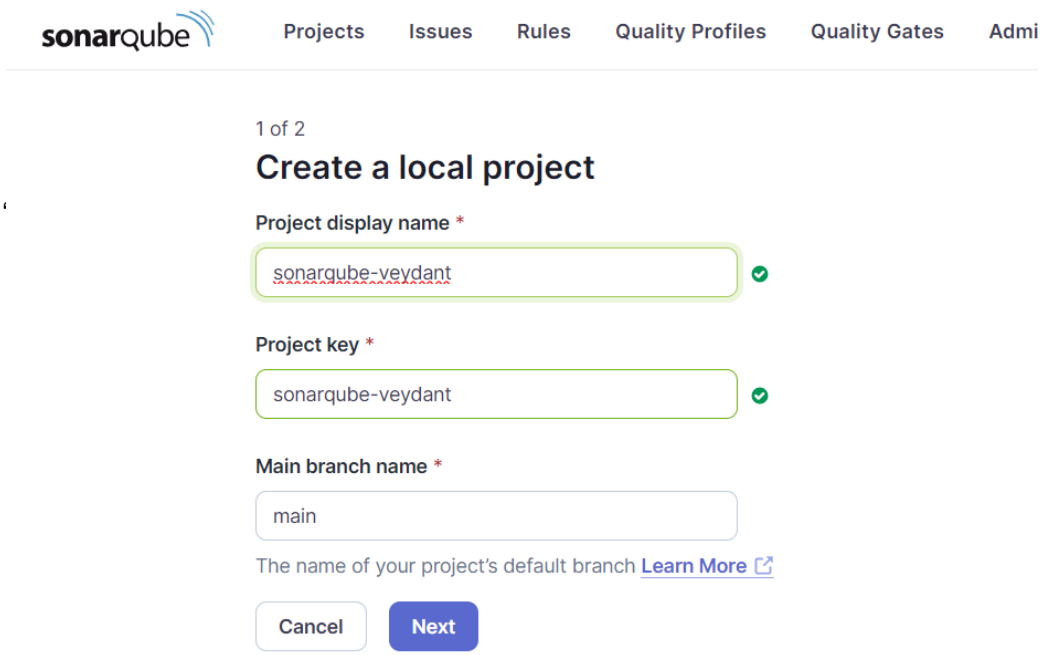
2. Run SonarQube in a Docker container using this command -

```
C:\Windows\system32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
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
b6dd73afc810a20ec3d643e9a148ab9643a3b5beff2766406df21f5f54a090c1
```

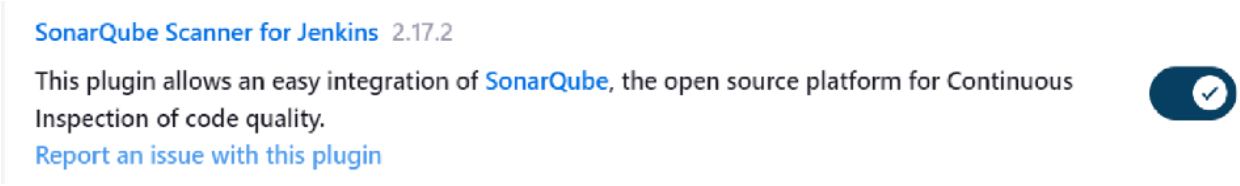
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



4. Login to SonarQube using username admin and password admin.
5. Create a manual project in SonarQube with the name sonarqube



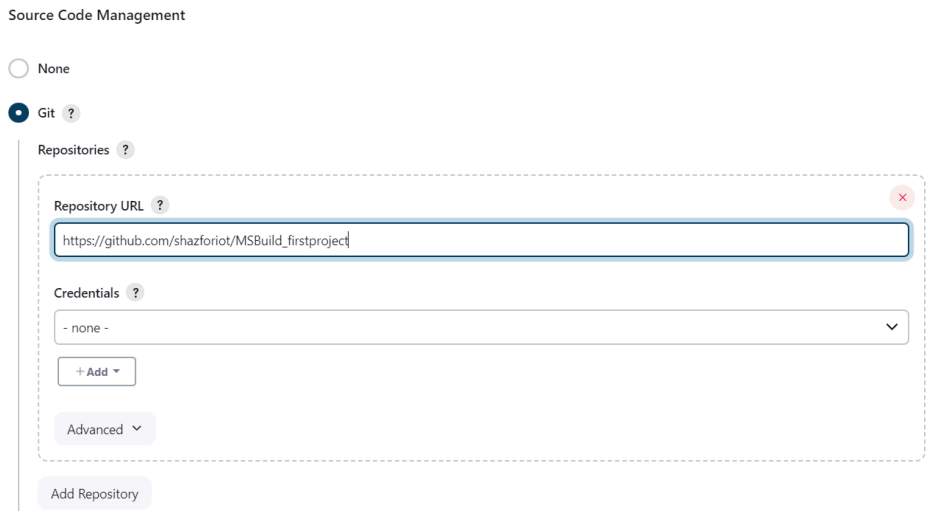
Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



6. Under Jenkins 'Configure System', look for SonarQube Servers and enter the details. Enter the Server Authentication token if needed.

7. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.
8. After the configuration, create a New Item in Jenkins, choose a freestyle project.

9. Choose this GitHub repository in Source Code Management.
https://github.com/shazforiot/MSBuild_firstproject.git



10. Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.
11. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.

Veydant Sharma D15C 50 AdvDevops

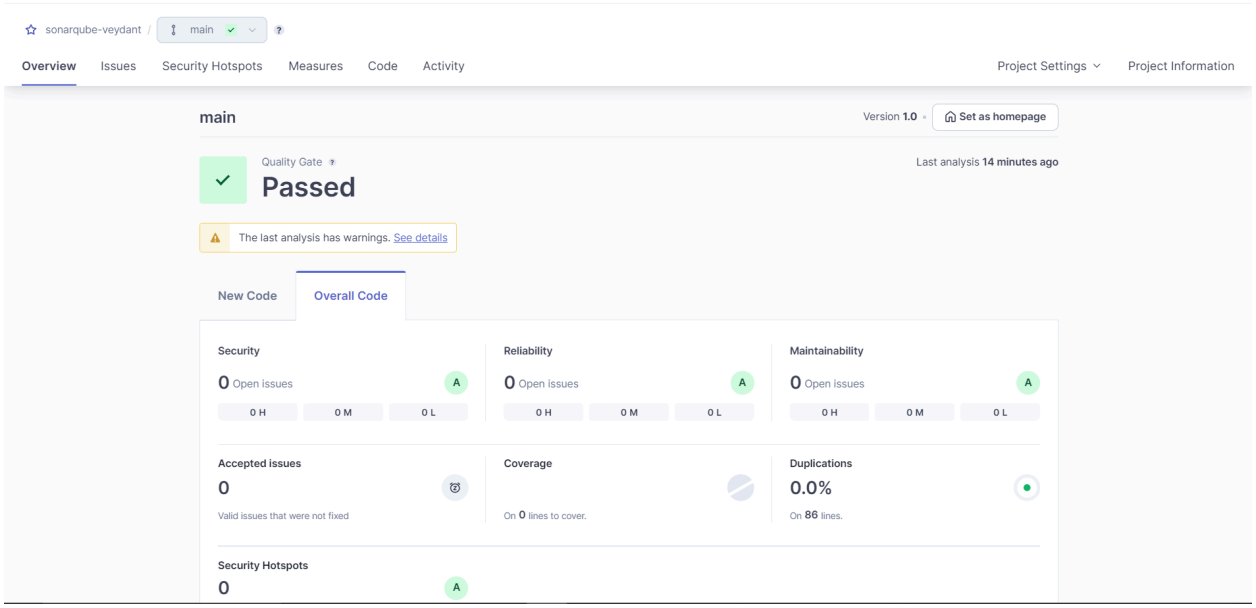
12. Run the build and check the output

Console Output

Download Copy View as plain text

```
Started by user unknown or anonymous
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject
 > 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 +refs/heads/*:refs/remotes/origin/* # timeout=10
 > 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
[sonarqube-test] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000
-Dsonar.projectKey=sonarqube-veydant -Dsonar.projectName=sonarqube-veydant -Dsonar.host.url=http://localhost:9000 -Dsonar.login=squ_3c67bbd5196ad7f467c5a12b65dfb090e8769e1c
-Dsonar.projectVersion=1.0 -Dsonar.sources=. -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test
21:42:11.084 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
21:42:11.111 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\..\conf\sonar-
scanner.properties
21:42:11.114 INFO  Project root configuration file: NONE
21:42:11.182 INFO  SonarScanner CLI 6.2.0.4584
21:42:11.186 INFO  Java 21.0.4 Eclipse Adoptium (64-bit)
21:42:11.195 INFO  Windows 10 10.0 amd64
21:42:11.257 INFO  User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
21:42:12.960 INFO  JRE provisioning: os[windows], arch[amd64]
21:42:25.421 INFO  Communicating with SonarQube Server 10.6.0.92116
21:42:27.194 INFO  Starting SonarScanner Engine...
21:42:27.196 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
```

13. Once the build is complete check on sonarqube



Veydant Sharma D15C 50 AdvDevops

Conclusion

- In this experiment we worked on the sonarqube project along with jenkins.
- Project Build step issues: Issues faced were due to permissions from sonarqube project.
- The steps involved logging into SonarQube, creating a project, and configuring necessary settings within Jenkins to facilitate automated analysis of our sample GitHub repository. This integration not only enhances our ability to identify vulnerabilities early in the development lifecycle but also promotes a culture of security within our development practices.
- By integrating Jenkins with SonarQube, we established an automated framework for continuous static analysis, enhancing our CI/CD pipeline's security posture.