Name: Alok Yadav Div: D15C Roll No.: 59

EXPERIMENT NO. 7

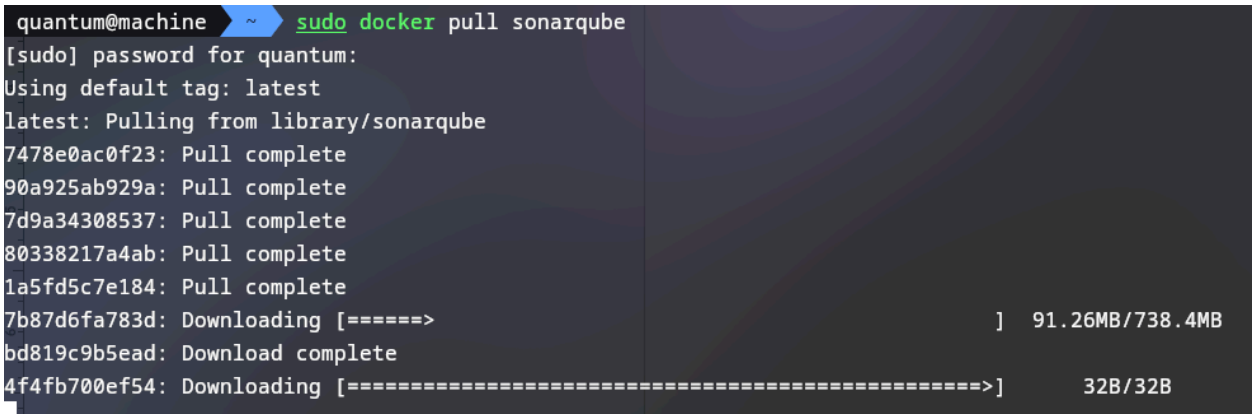
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Prerequisites:
- Jenkins installed (Java JDK required)
- Docker Installed (for SonarQube)

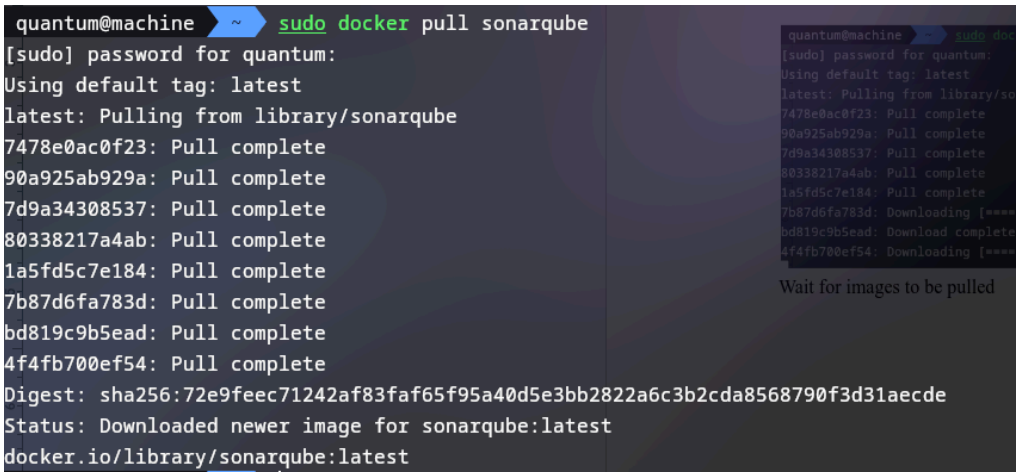
docker engine service should be running (sudo systemctl start docker)

1. Install SonarQube Docker image

   sudo docker pull sonarqube

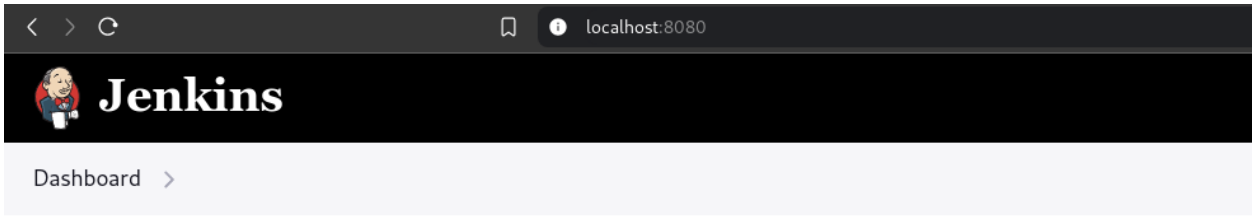


   Wait for images to be pulled



SonarQube installed successfully

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



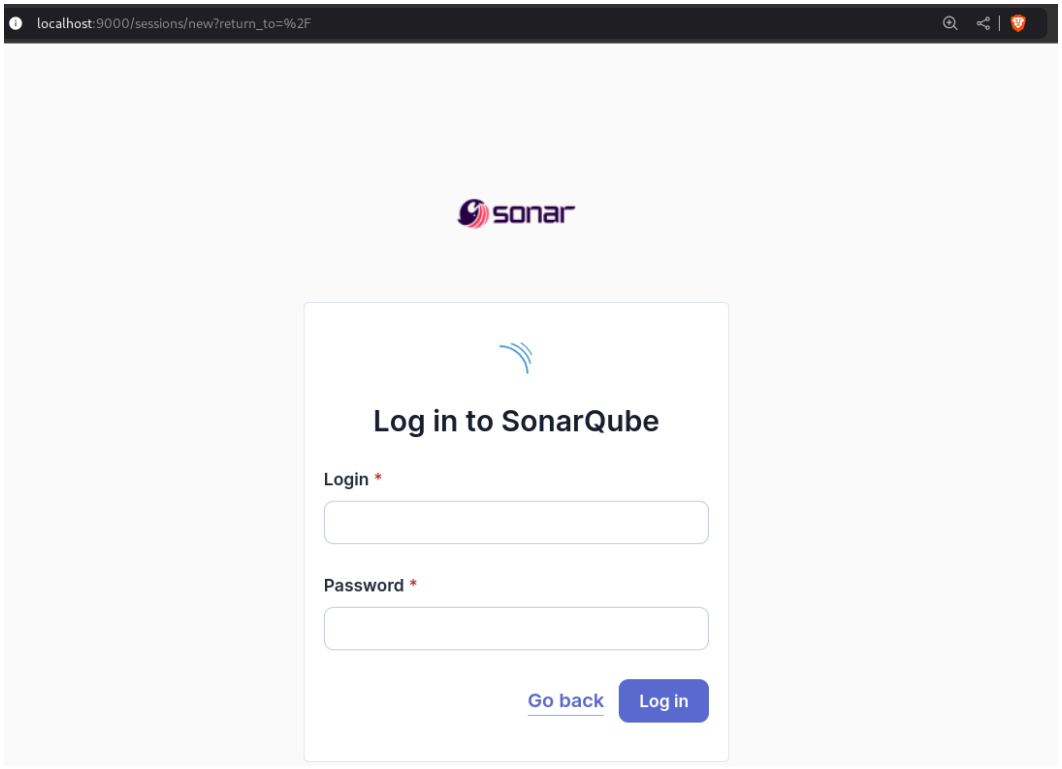
Our jenkins is running on port 8080

3. Run SonarQube in a Docker container using this command
sudo docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

Warning: run below command only once

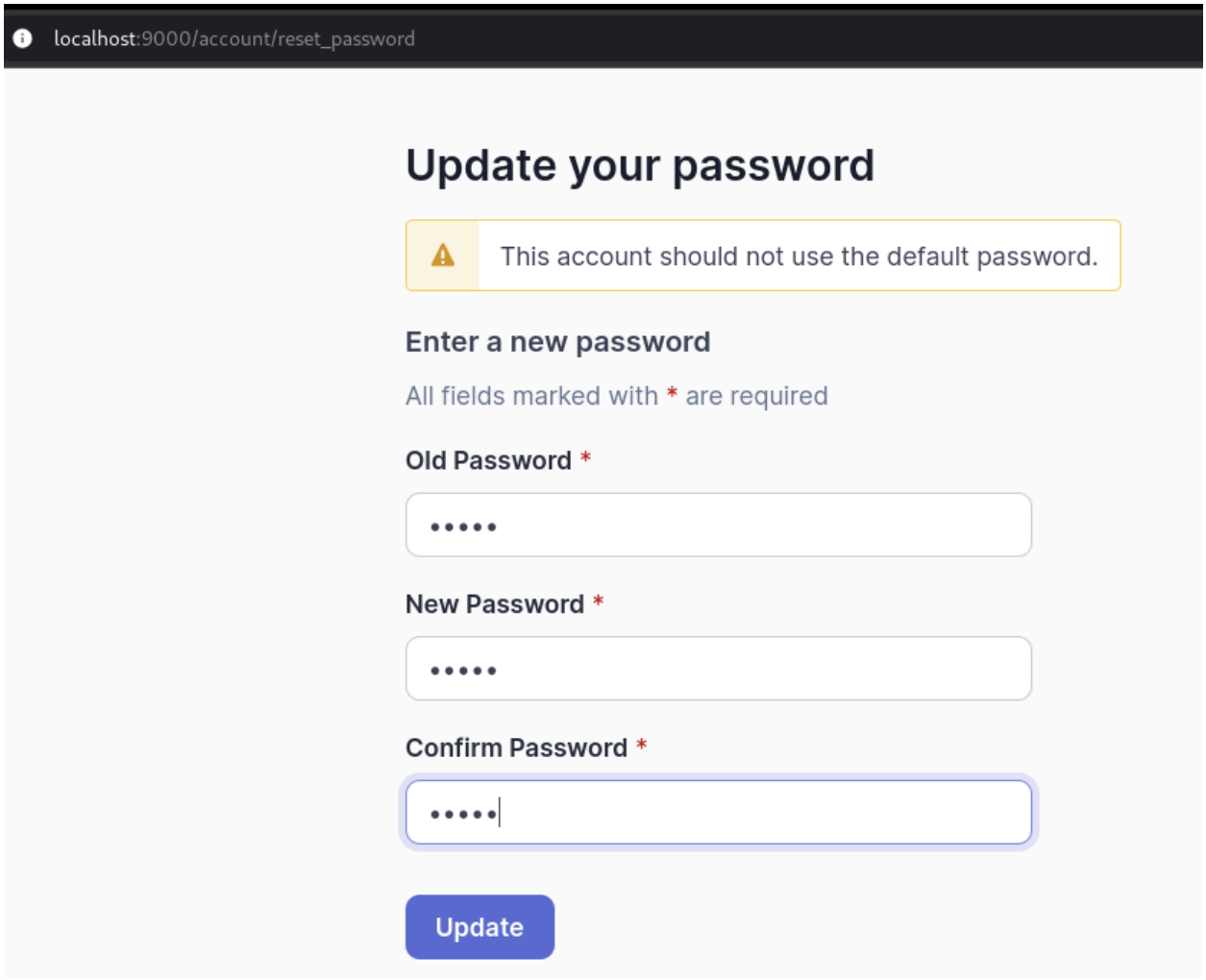
4. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.

Name: Alok Yadav Div: D15C Roll No.: 59



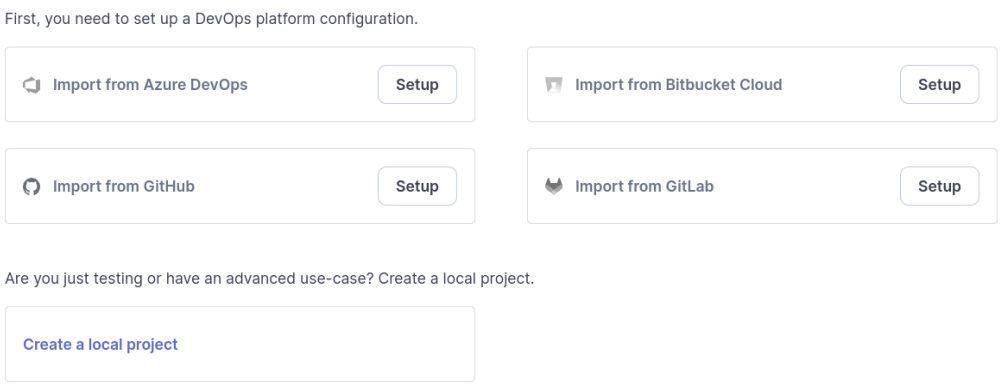
5. Login to SonarQube using username admin and password admin.

Name: Alok Yadav Div: D15C Roll No.: 59

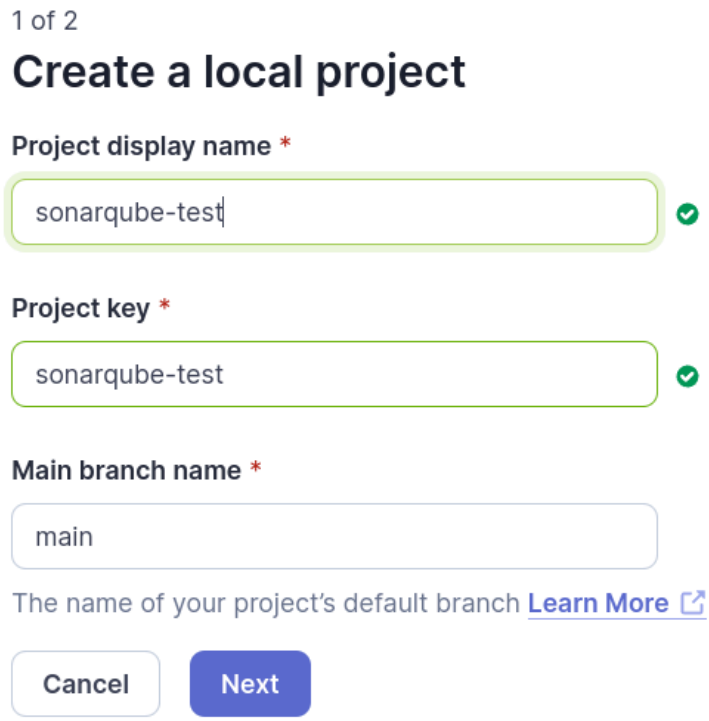


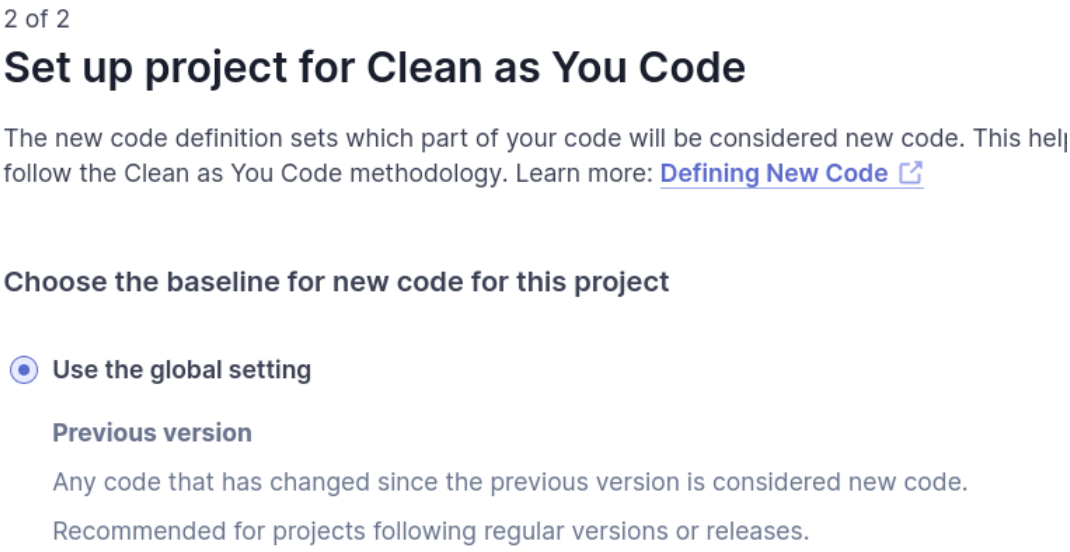
After logging, we have to change default password

6. Create a manual project in SonarQube with the name sonarqube
   Click on **create a local project** on dashboard

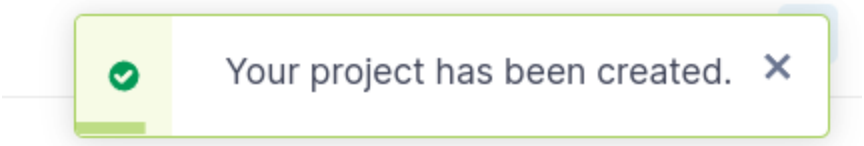


Name: Alok Yadav Div: D15C Roll No.: 59



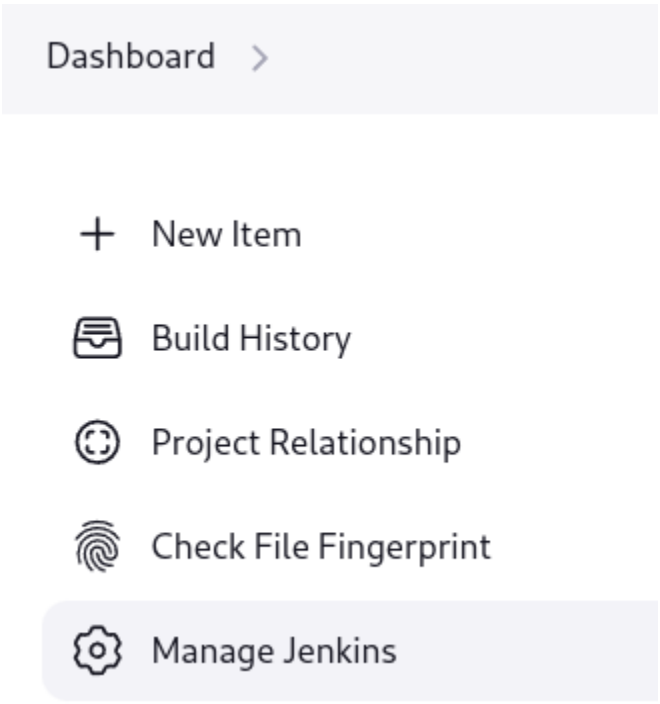


We can either use new custom settings for the project or use global settings
Here I'm using global setting

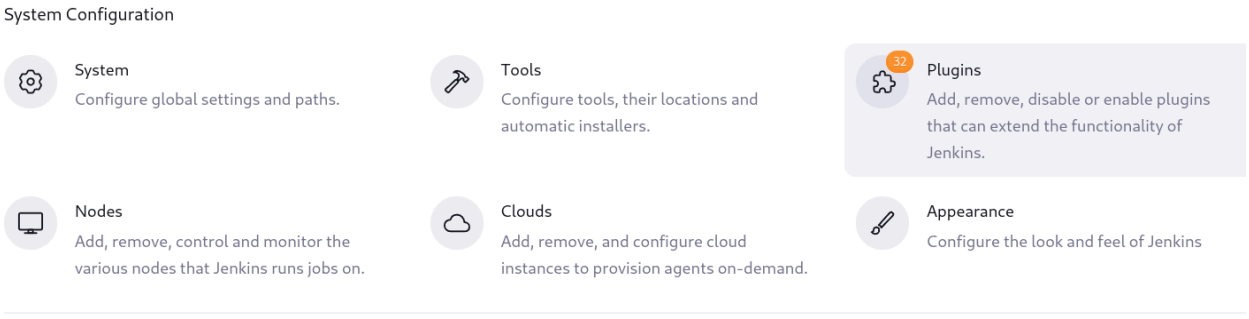


On successful creation of project, we a popup for the same

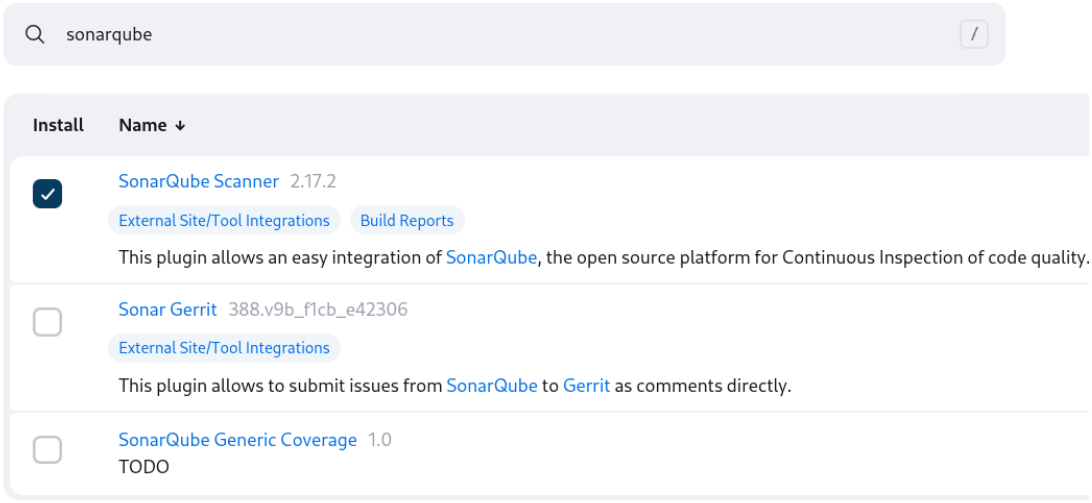
7. After setting project in sonarqube, go to **Jenkins Dashboard**

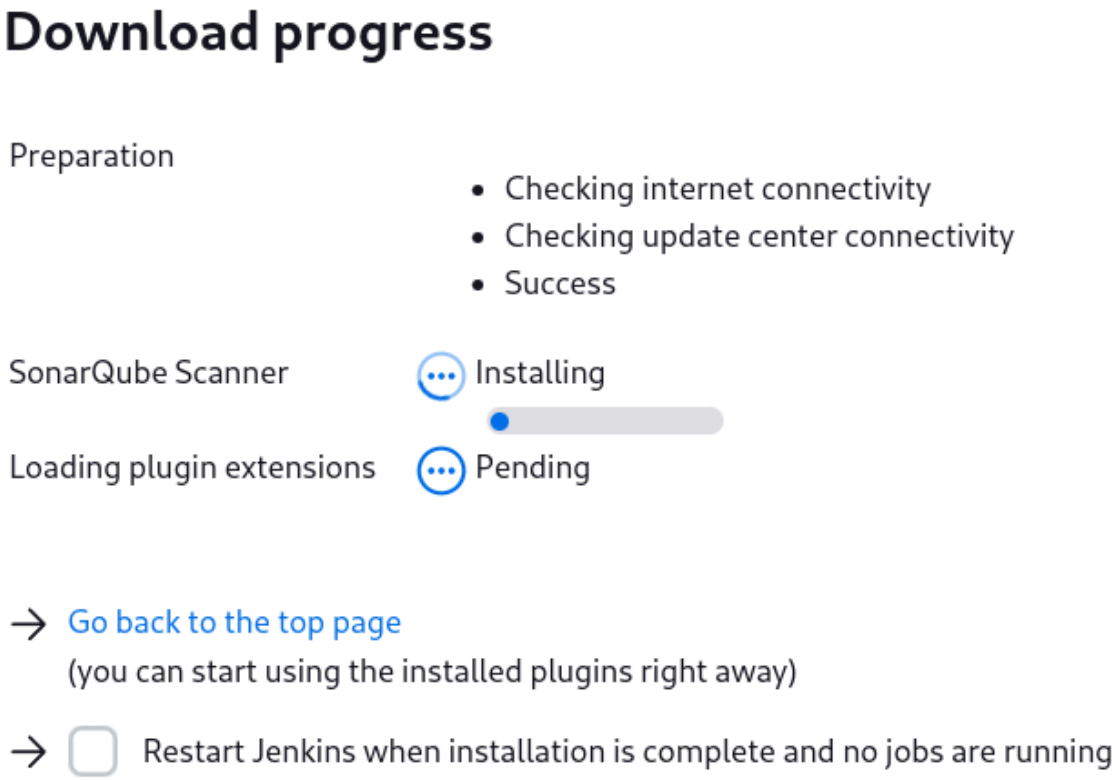


Go to Manage Jenkins and search for SonarQube Scanner in Plugins settings and install it.

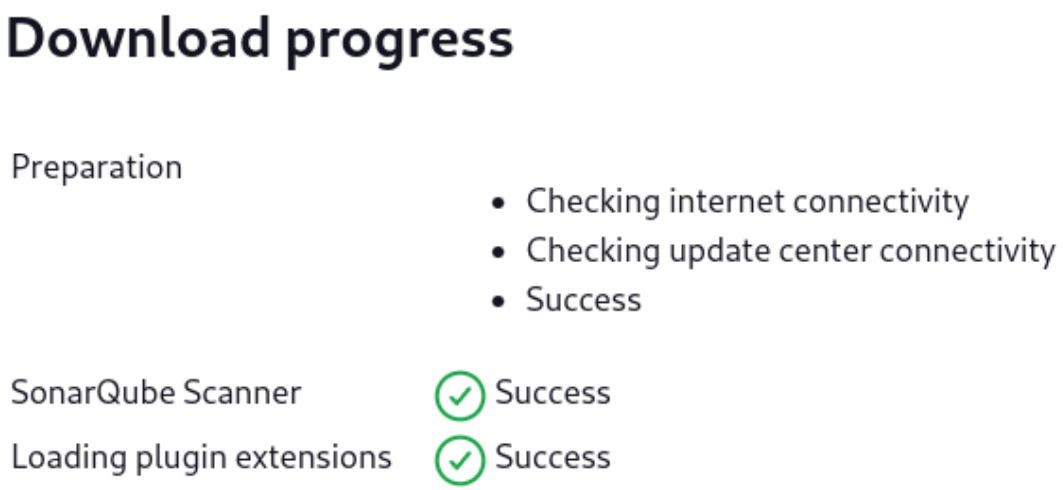


Name: Alok Yadav Div: D15C Roll No.: 59



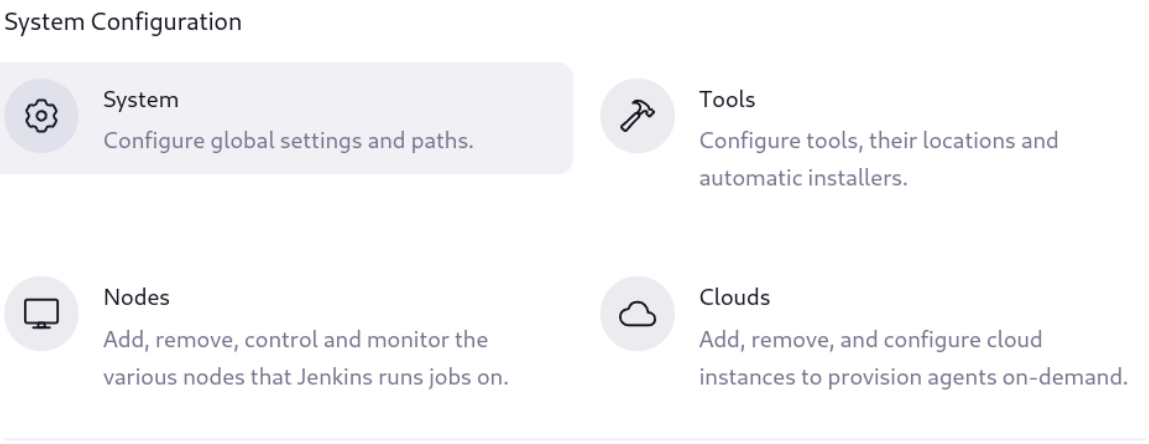


Our installation is in progress wait for it to download and install packages

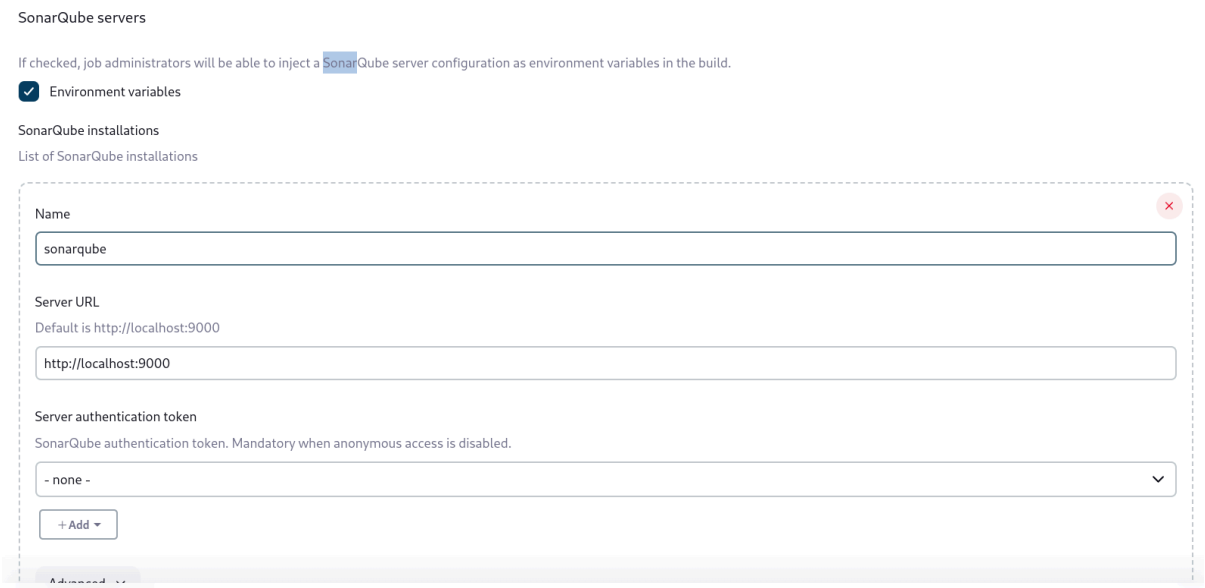


Plugin installed successfully

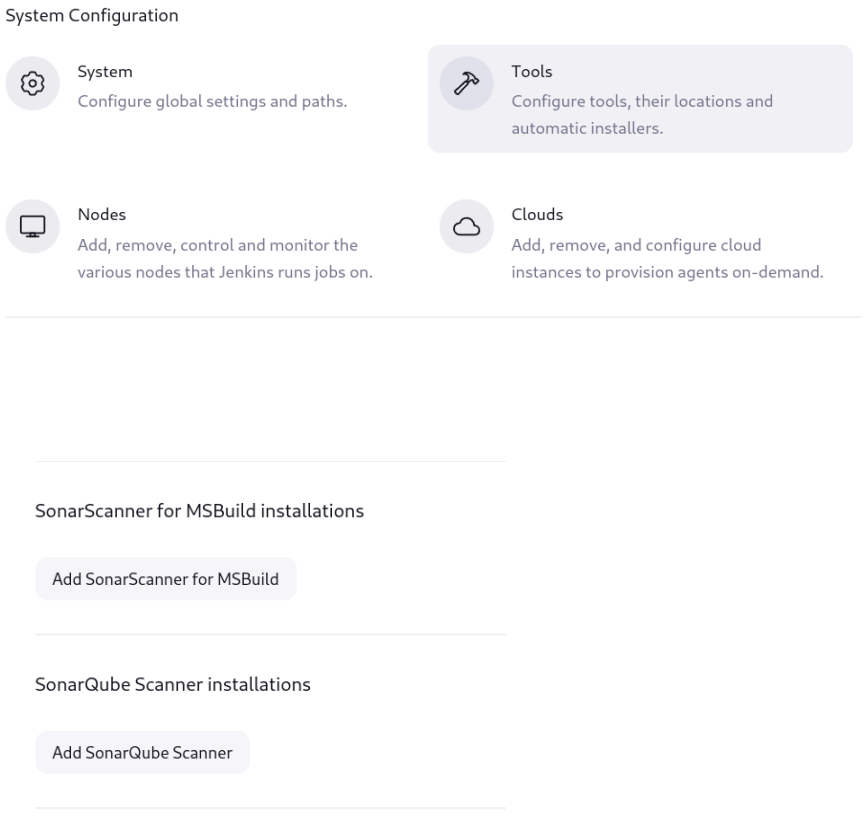
8. Under Jenkins dashboard ‘Configure System’, look for SonarQube Servers and enter the details.



Name: Alok Yadav Div: D15C Roll No.: 59

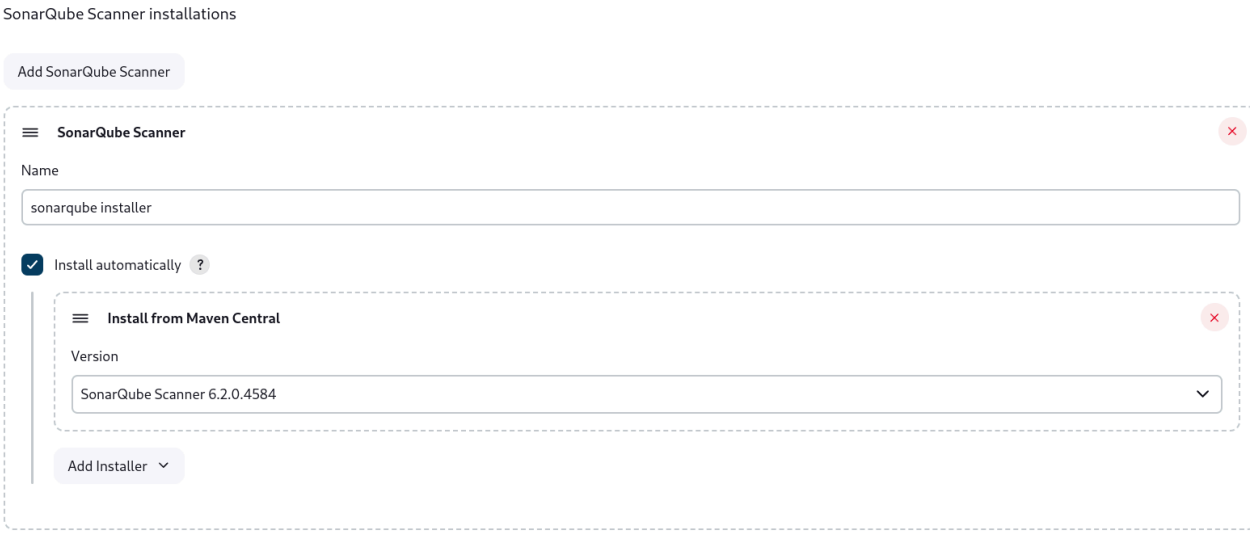


9. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.



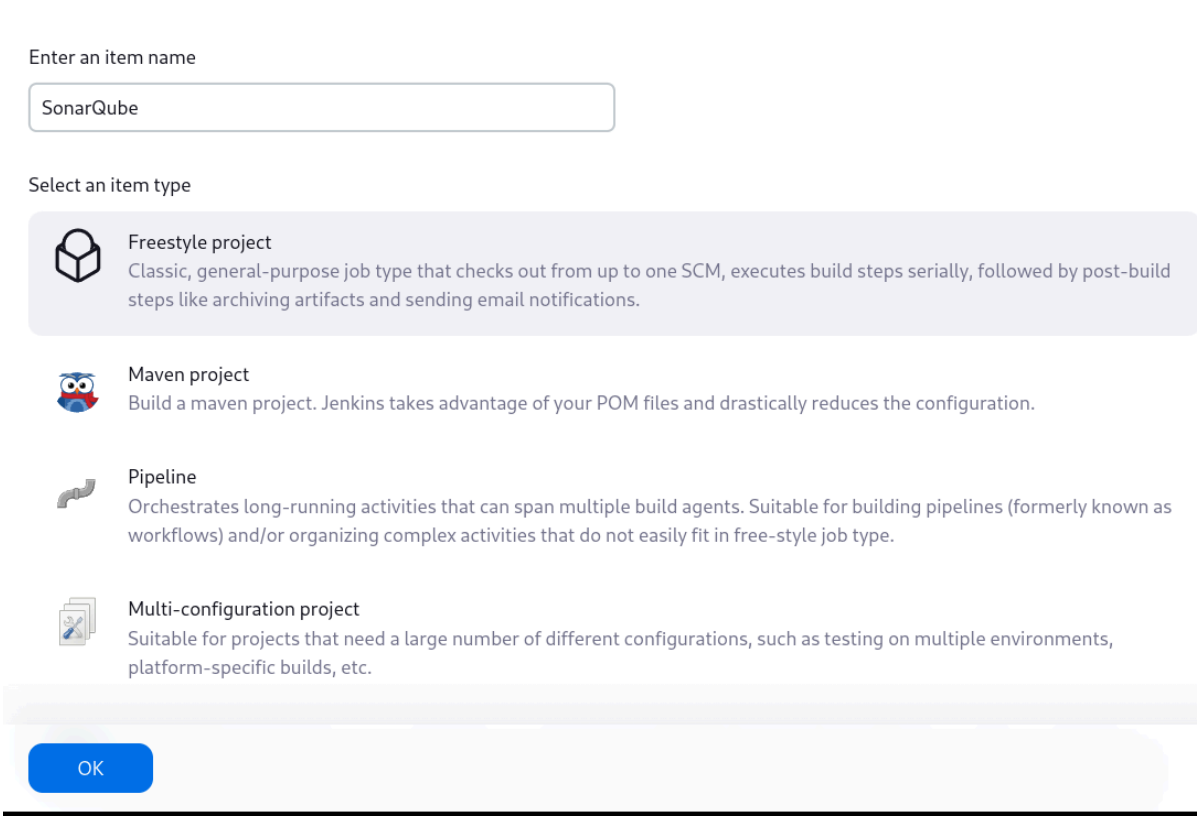
Click on **Add SonarQube Scanner**

Name: Alok Yadav Div: D15C Roll No.: 59



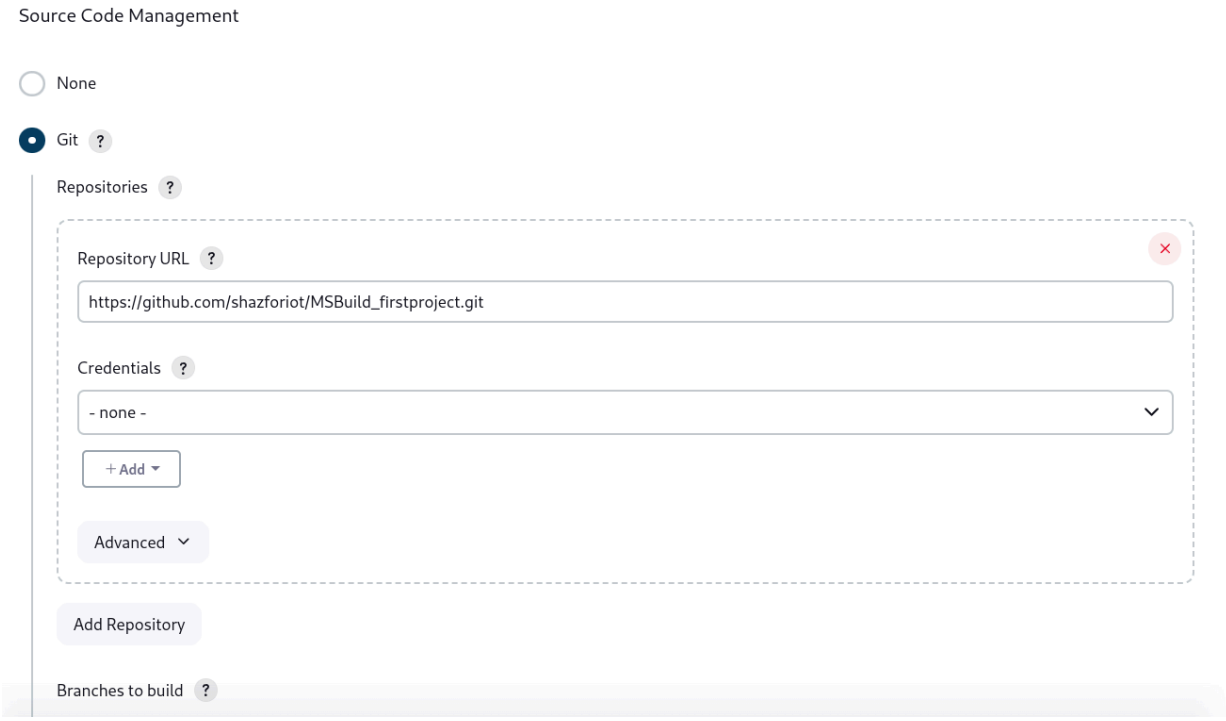
Select Latest version and save configuration

10. After the configuration, create a New Item in Jenkins, choose a freestyle project.

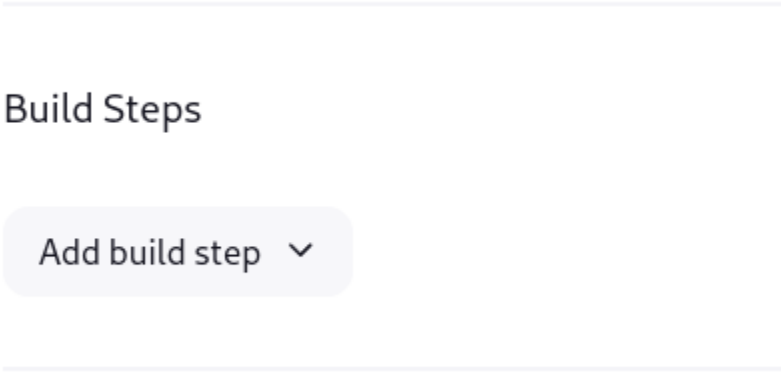


Name: Alok Yadav Div: D15C Roll No.: 59

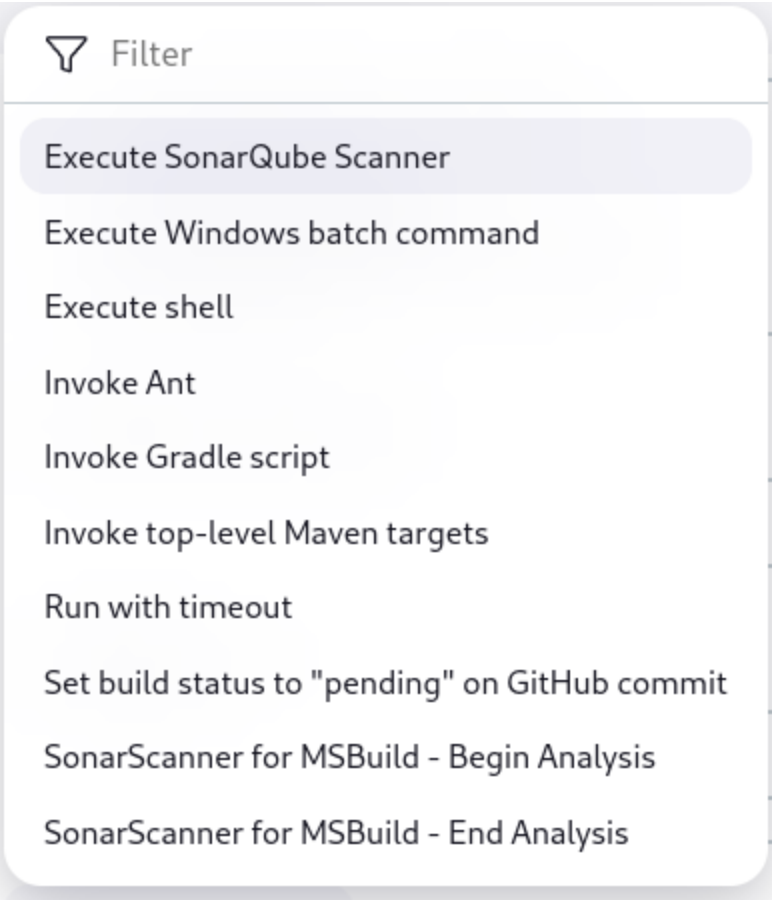
11. Choose this GitHub repository in Source Code Management. https://github.com/shazforiot/MSBuild_firstproject.git It is a sample hello-world project with no vulnerabilities and issues, just to test the integration



12. Under Build-> Execute SonarQube Scanner

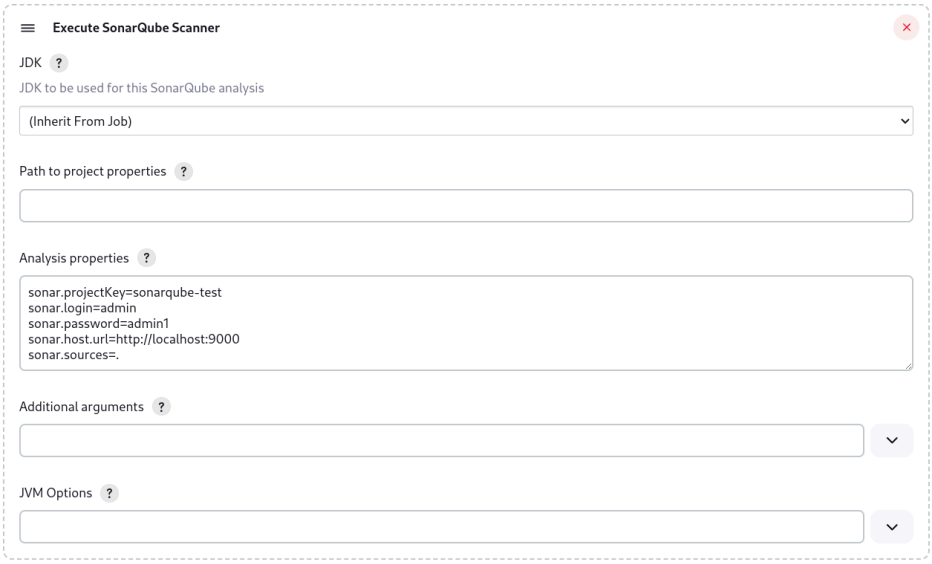


Click on add build steps



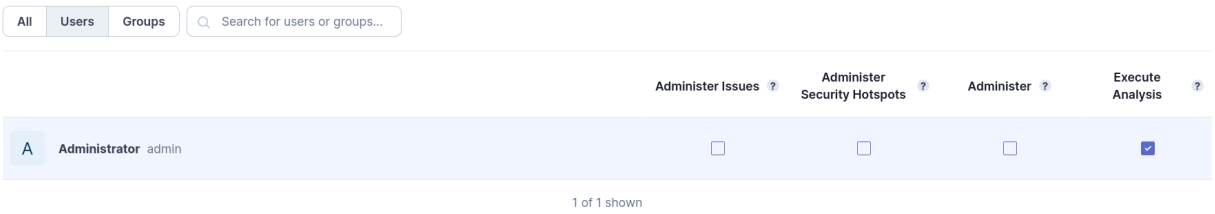
Then click on **Execute SonarQube Scanner**

13. Mention the SonarQube Project Key, Login, Password, Source path and Host URL in Analysis properties

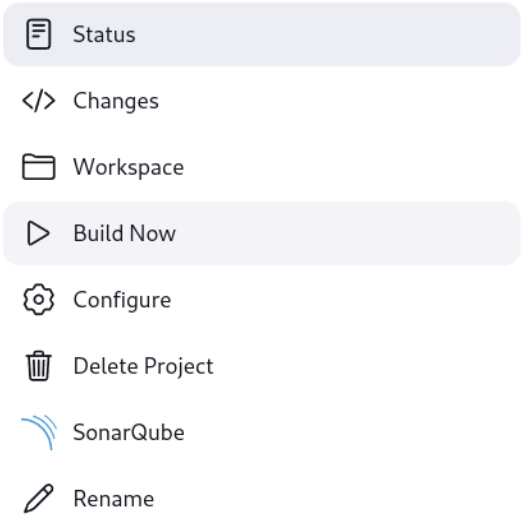


Name: Alok Yadav Div: D15C Roll No.: 59

14. Go to http://localhost:9000/project_roles?id=<project_key> and allow Execute Permissions to the Admin user.



15. Run The Build.

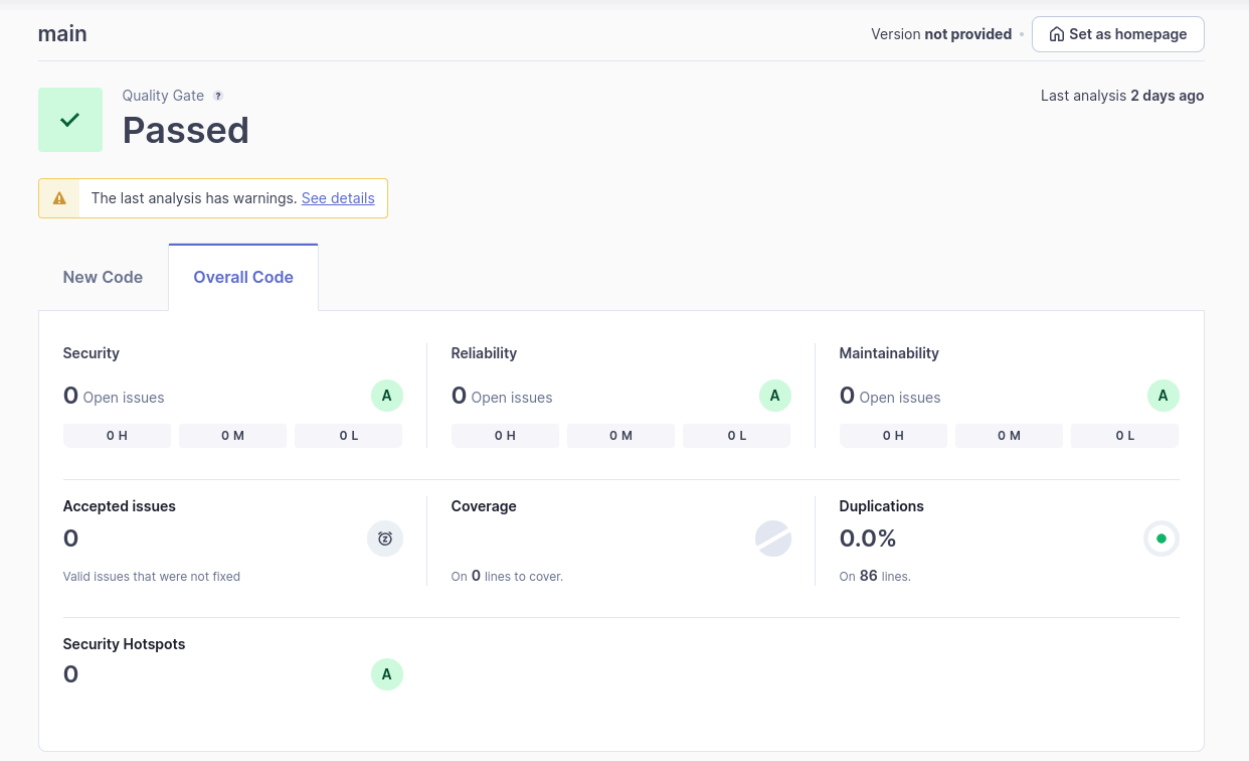


16. Check the console output

Name: Alok Yadav Div: D15C Roll No.: 59

```
10:30:32.234 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:30:32.234 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:30:32.235 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=96795973-9667-4456-a15c-3311dbc9d067
10:30:32.244 INFO  Analysis total time: 9.533 s
10:30:32.245 INFO  SonarScanner Engine completed successfully
10:30:32.300 INFO  EXECUTION SUCCESS
10:30:32.301 INFO  Total time: 14.090s
Finished: SUCCESS
```

17. Once the build is complete, check the project in SonarQube.



In this way, we have integrated Jenkins with SonarQube for SAST.

Conclusion: We began the experiment with installation of SonarQube Docker Image followed by setting up a new project in SonarQube. Then we installed the SonarQube scanner plugin and then created a new freestyle project in Jenkins with a Git repository for code analysis. Then we configured the jenkins with appropriate settings to work with sonarqube. Gave permissions to Jenkins to perform code analysis. It is essential to provide correct properties in **Analysis Properties** for Jenkins to run correctly.The jenkins project ran successfully with all tests passed in SonarQube

Name: Alok Yadav Div: D15C Roll No.: 59