Advance DevOps

Experiment 7

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

Steps:

1. Firstly, we will ensure whether is installed or not by running docker -v in the command prompt.

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

2. Run docker login command and add your username and password for docker.

```
C:\Users\Dell>docker login
Authenticating with existing credentials...
Stored credentials invalid or expired
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head
over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and
is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username (dimple866): dimple866
Password:

Login Succeeded
```

3. Run docker pull SonarQube command to install SonarQube image.

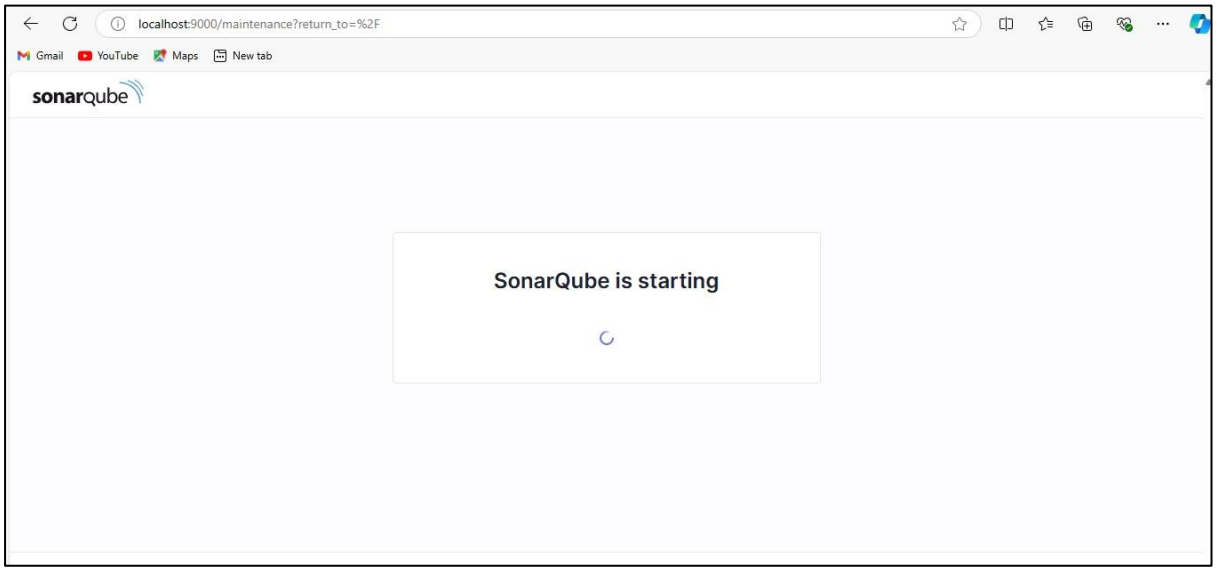
```
C:\Users\Dell>docker pull sonarqube
Using default tag: latest
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
```

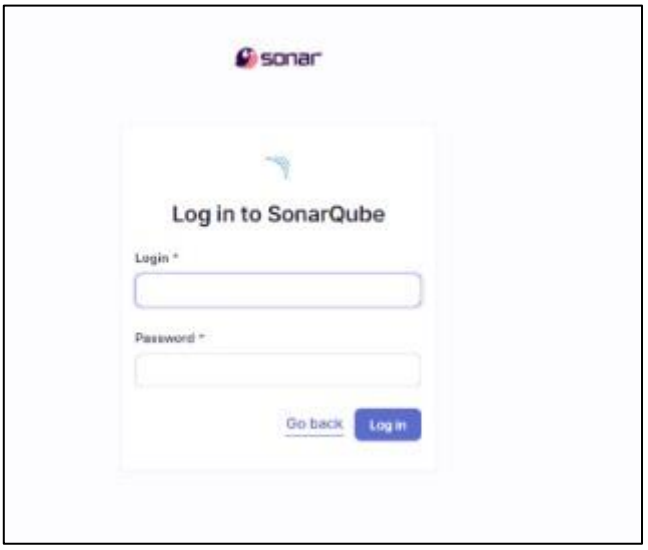
4. Run docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest Command to run the sonarqube.

```
C:\Users\Dell>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
ac1f985dedebc00a642a4c69a502d611389e8f9fa46610febe75aa5021767cab
```

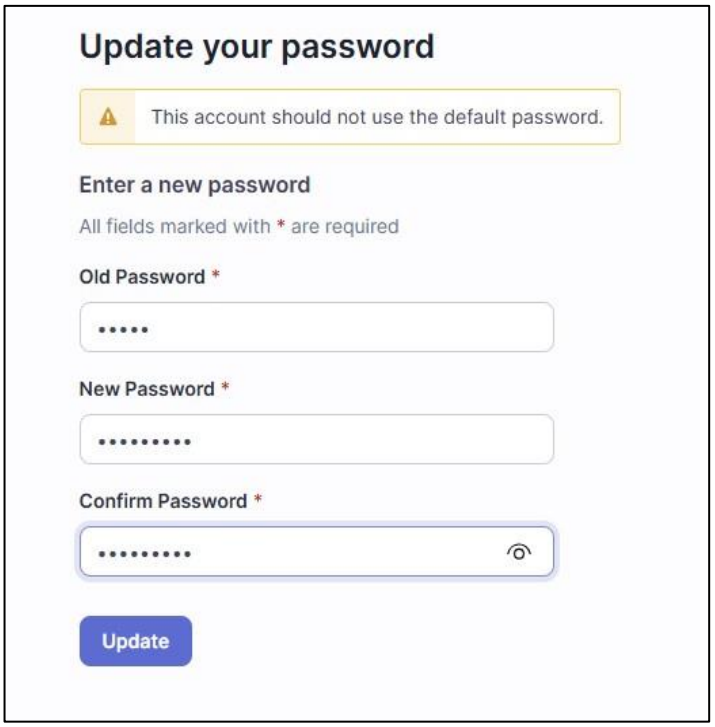
5. Once the container is running go to your web browser and check status of SonarQube at port 9000.



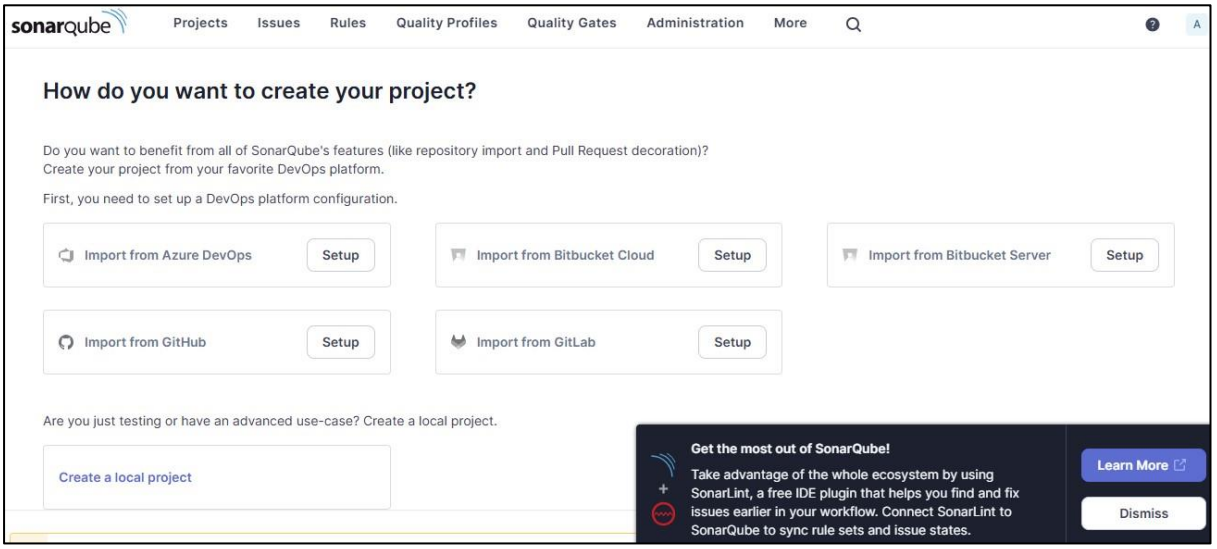
6. Once SonarQube is started it will redirect you to login page. The login and password both for SonarQube is ‘admin’



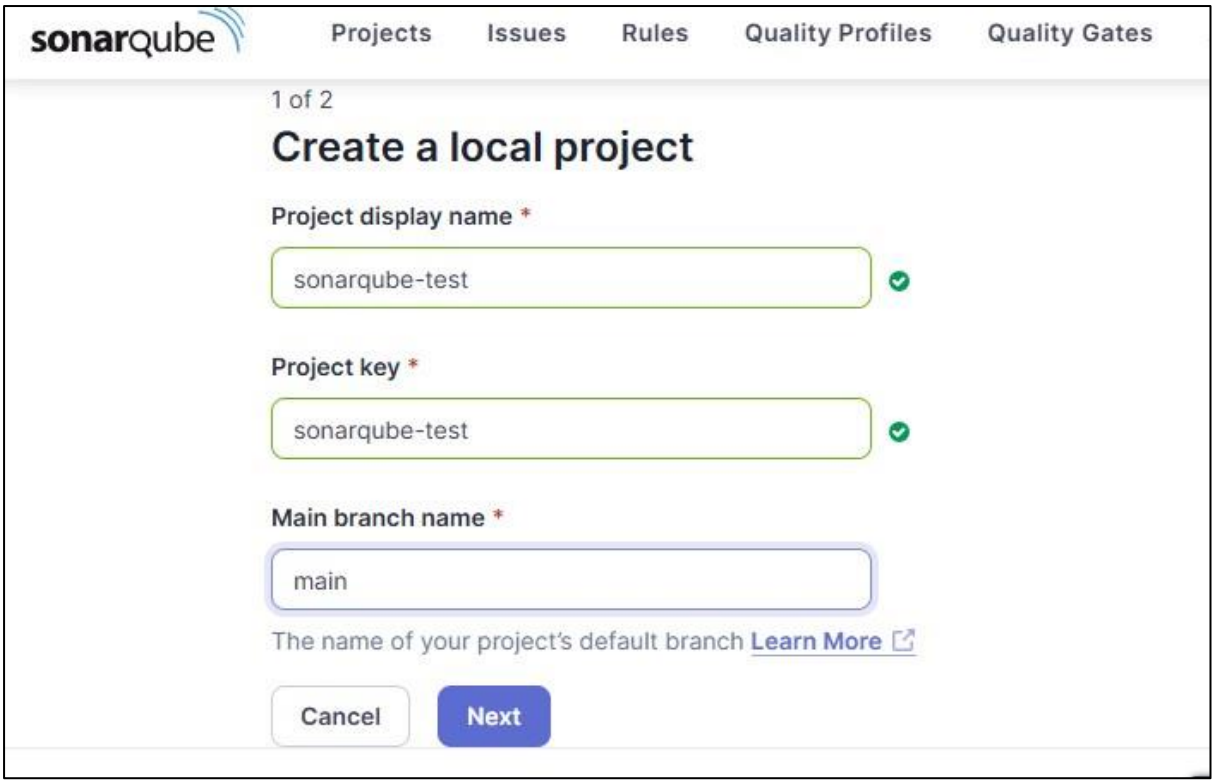
7. Change the password for your SonarQube account.



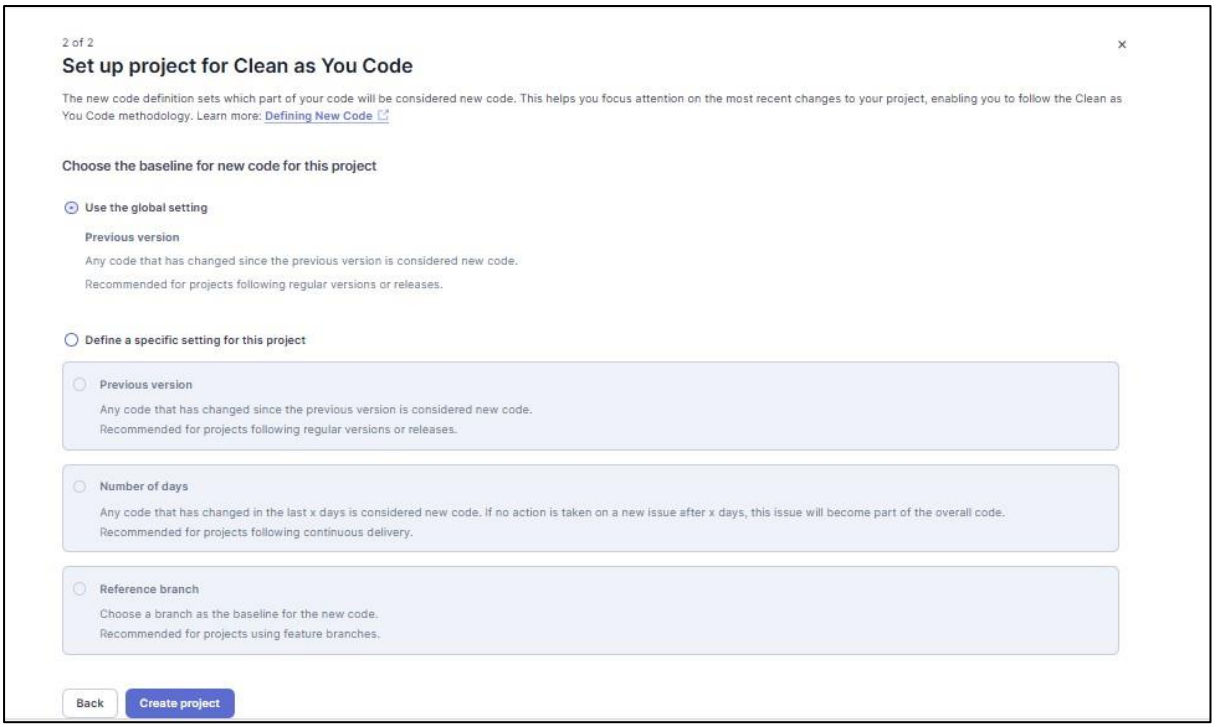
8. After changing the password, you will be directed to this screen. Click on Create a Local Project.



9. Add name of the project and project key and select the main branch name and click on next.

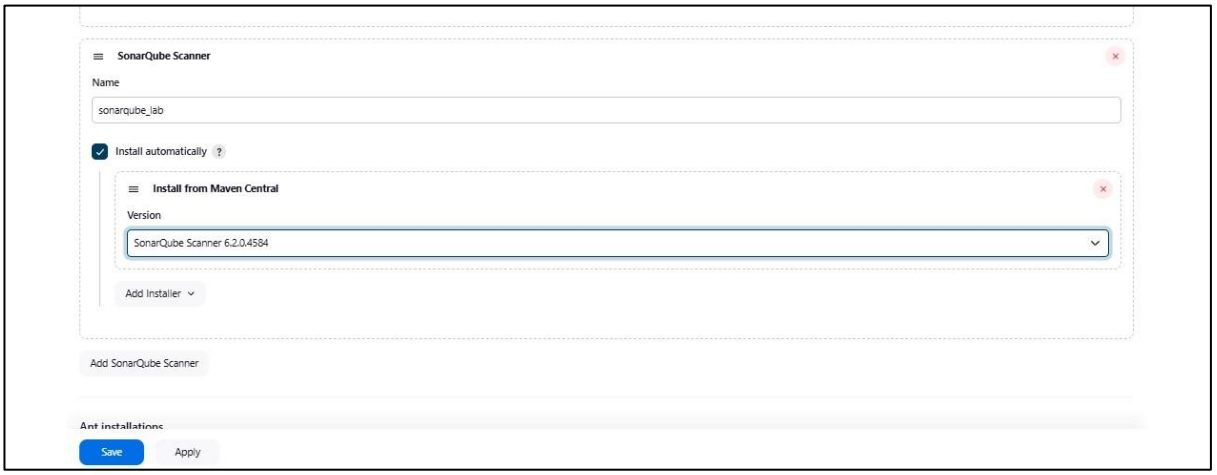


10. Set up the project as required and click on create.

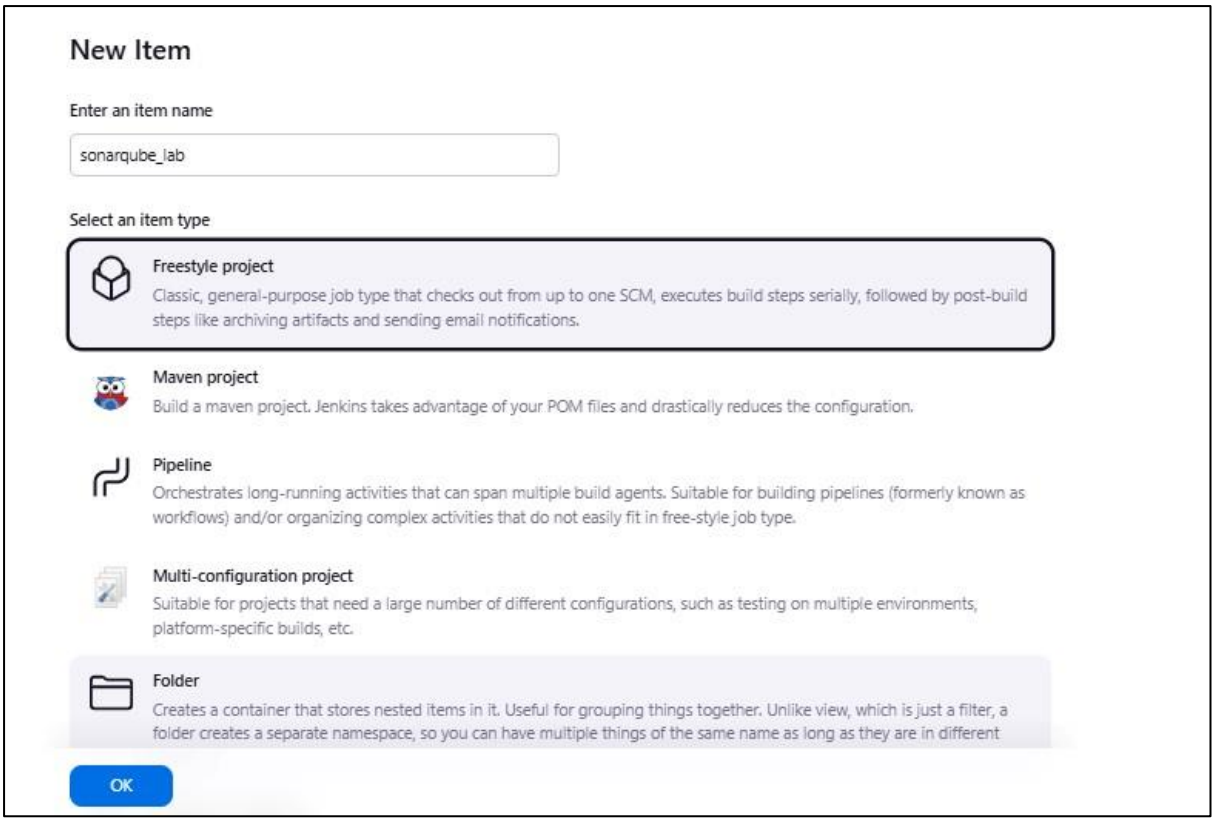


11. Go to Jenkins dashboard->Manage Jenkins->System and scroll down to SonarQube installations. Enter the name and URL in the fields and save the changes.

12. In SonarQube Scanner add the latest version then apply the changes and save it.



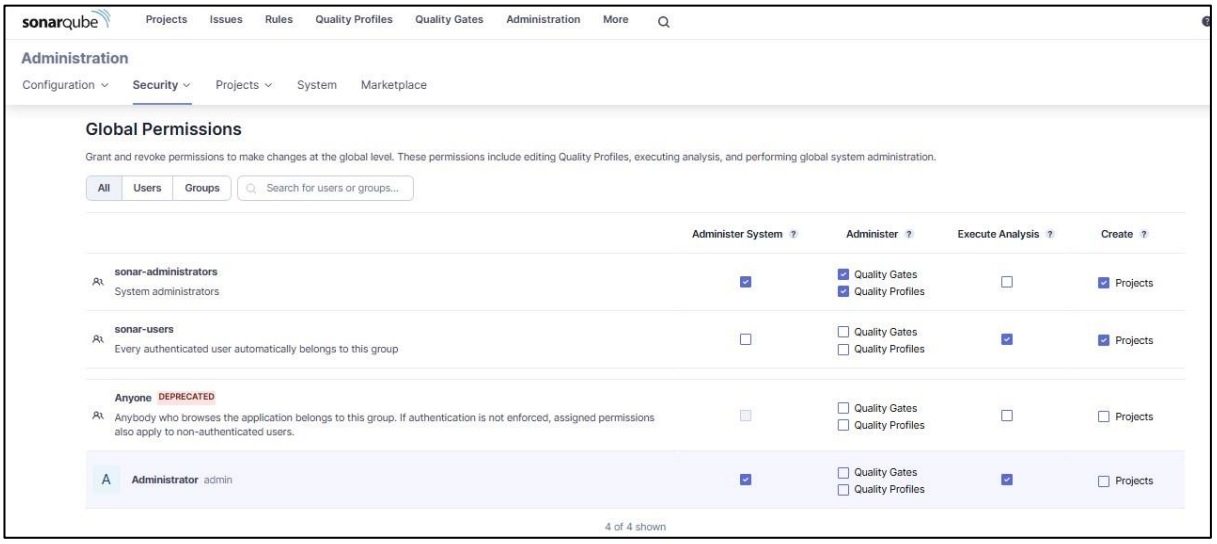
13. Go to Jenkins and then create a new item, enter the item name and select “Freestyle project” and then click on ok.



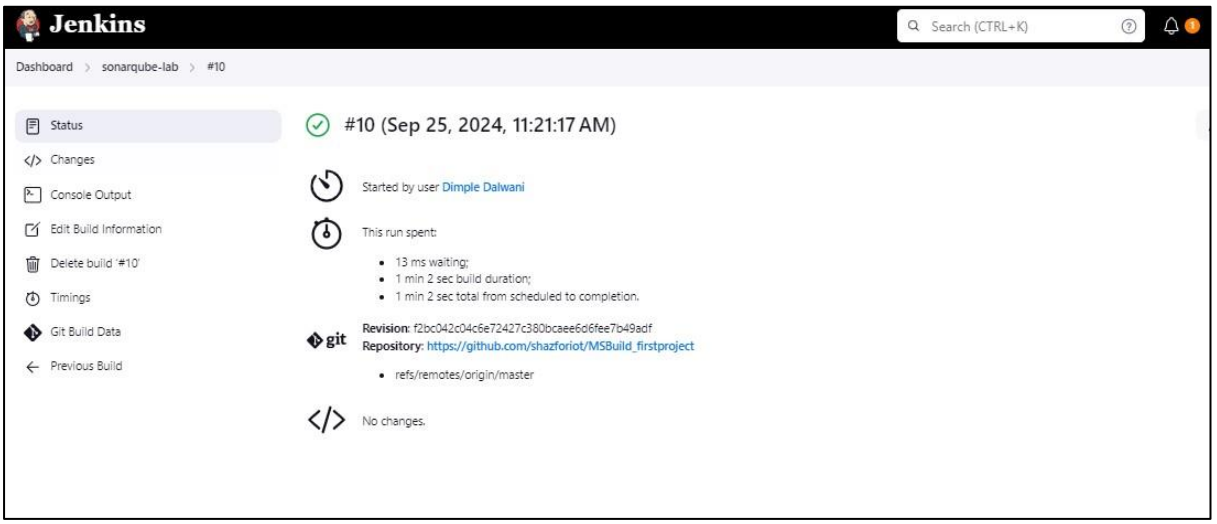
14. Use this GitHub repository in Source Code Management.
https://github.com/shazforiot/MSBuild_firstproject

15. In Analysis properties, mention the SonarQube Project Key, Login, Password, Source path and Host URL.

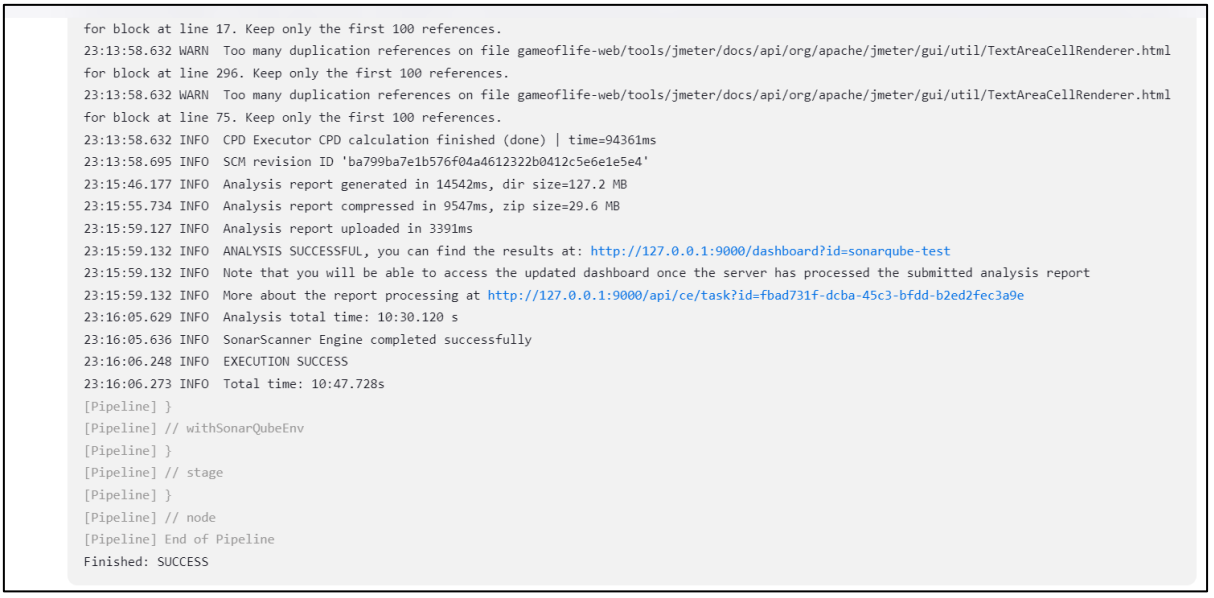
16. Now, you need to grant the local user (here admin user) permissions to Execute the Analysis stage on SonarQube. For this go to http://localhost:<port_number>/admin/permissions and check the ‘Execute Analysis’ checkbox under Administrator.



17. Go to the job you have just built and click on Build Now.

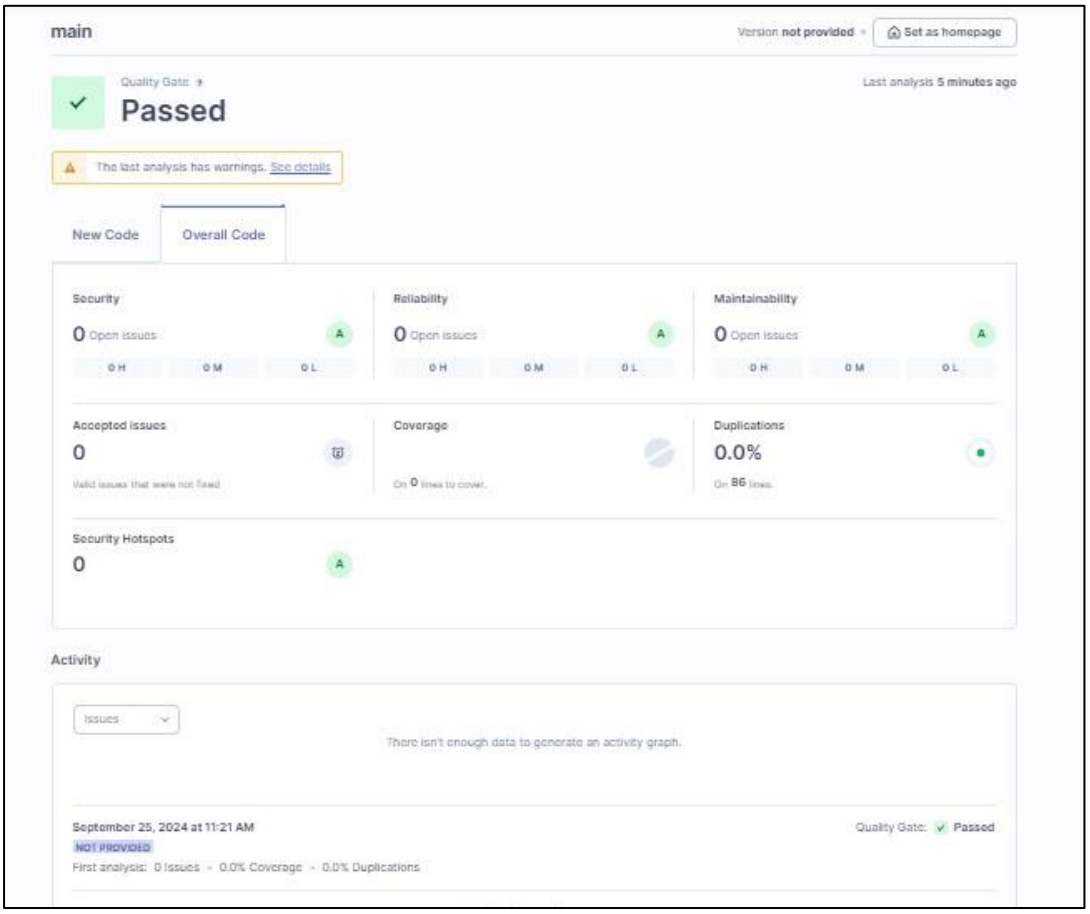


18. Check the console Output



```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO  CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO  Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO  Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO  Analysis report uploaded in 3391ms
23:15:59.132 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO  Analysis total time: 10:30.120 s
23:16:05.636 INFO  SonarScanner Engine completed successfully
23:16:06.248 INFO  EXECUTION SUCCESS
23:16:06.273 INFO  Total time: 10:47.728s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

19. Go back to SonarQube and check the project.



Conclusion: While performing this experiment there was an issue in creating sonarqube docker image and we resolved it by logging in to the docker desktop and performing it through the terminal. Other than this we created a freestyle project and entered the sonarqube credentials and then performed build.