**Experiment No 7**

**Name: Arya Manoj Madhavi**

**Div: D15B**

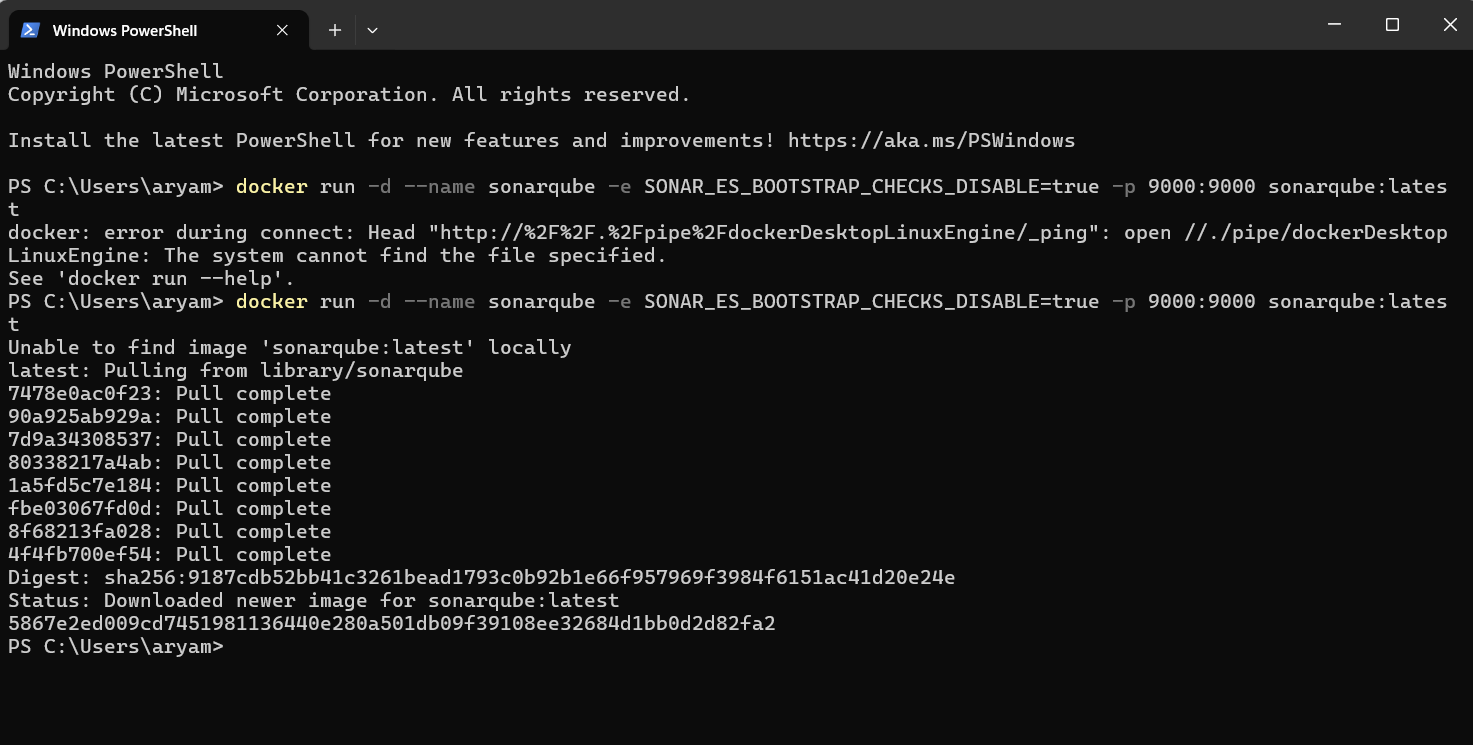
**Roll No.:31**

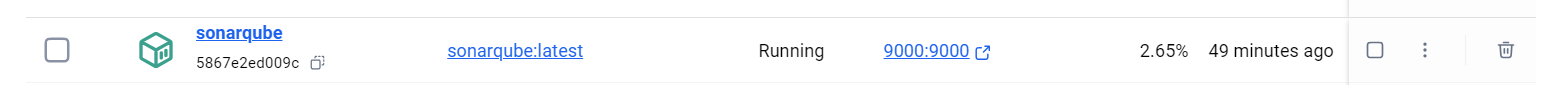
**Batch B**

**AIM: Installing SonarQube from the Docker Image**

$ docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000

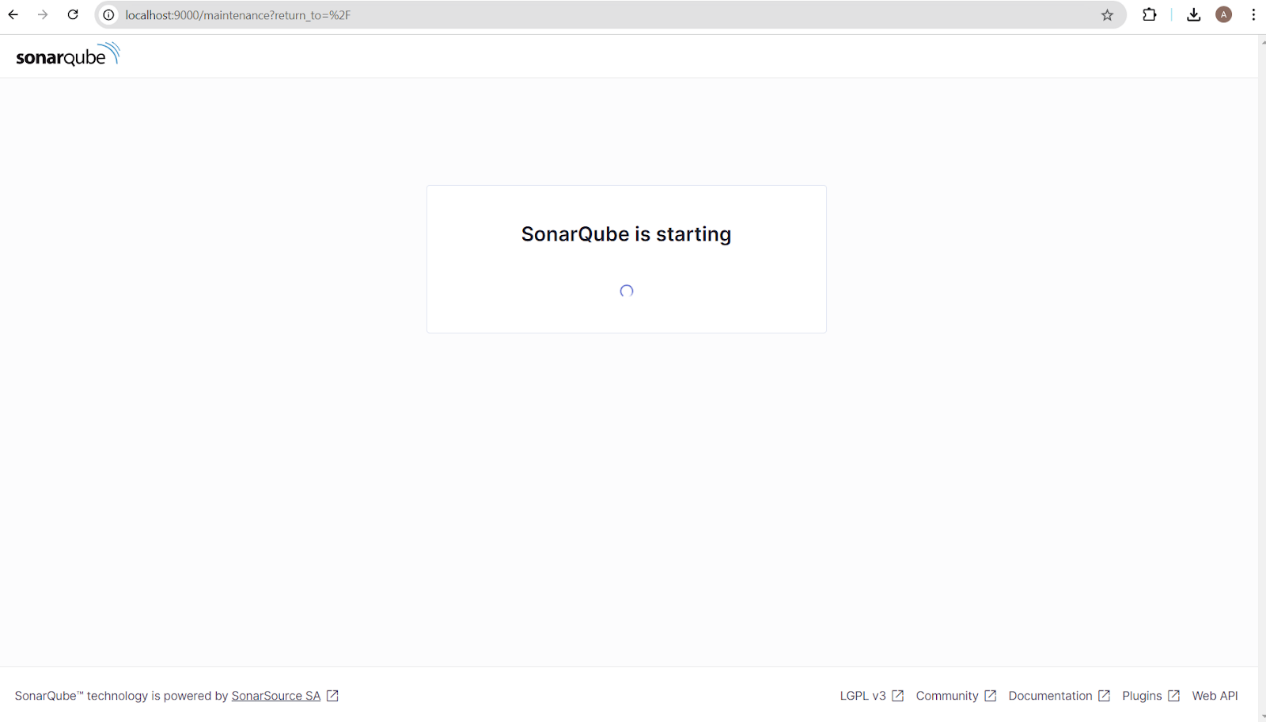
Sonarqube:latest



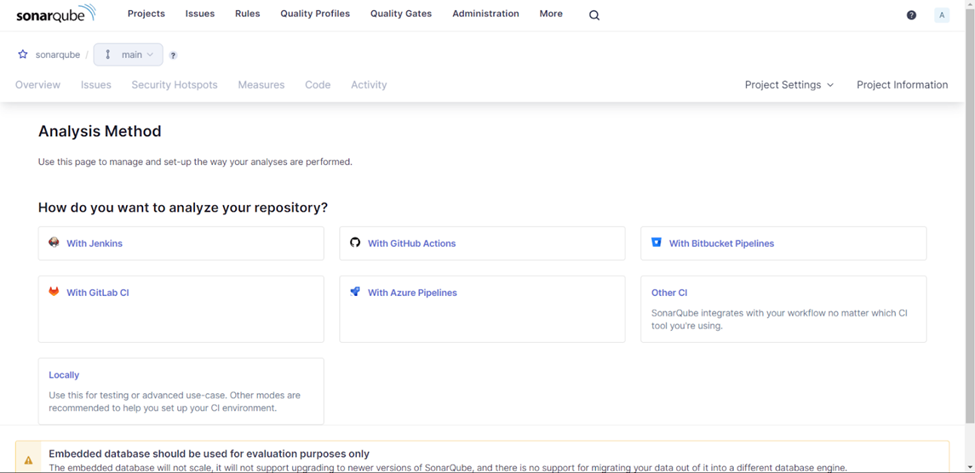


**go to the SonarQube page by typing:**

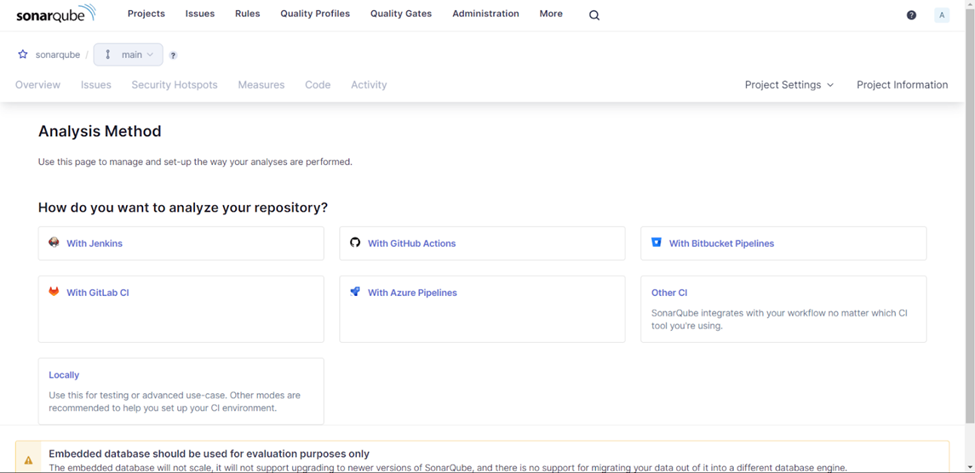
http://localhost:9000/ on your browser.



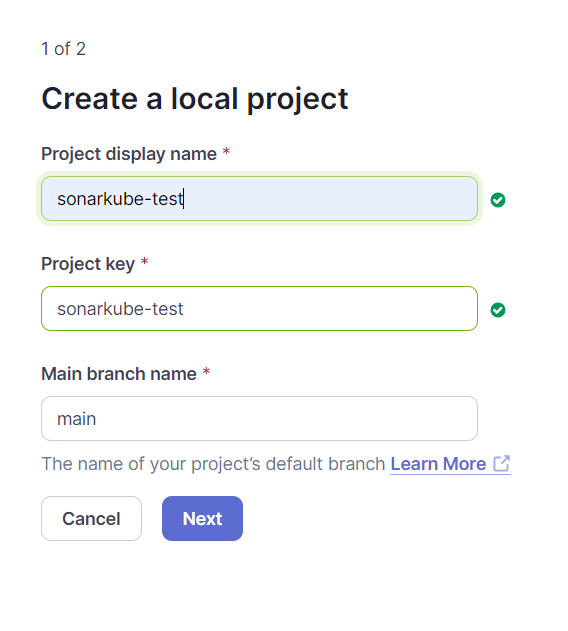
**Installation is successful if you see this page**

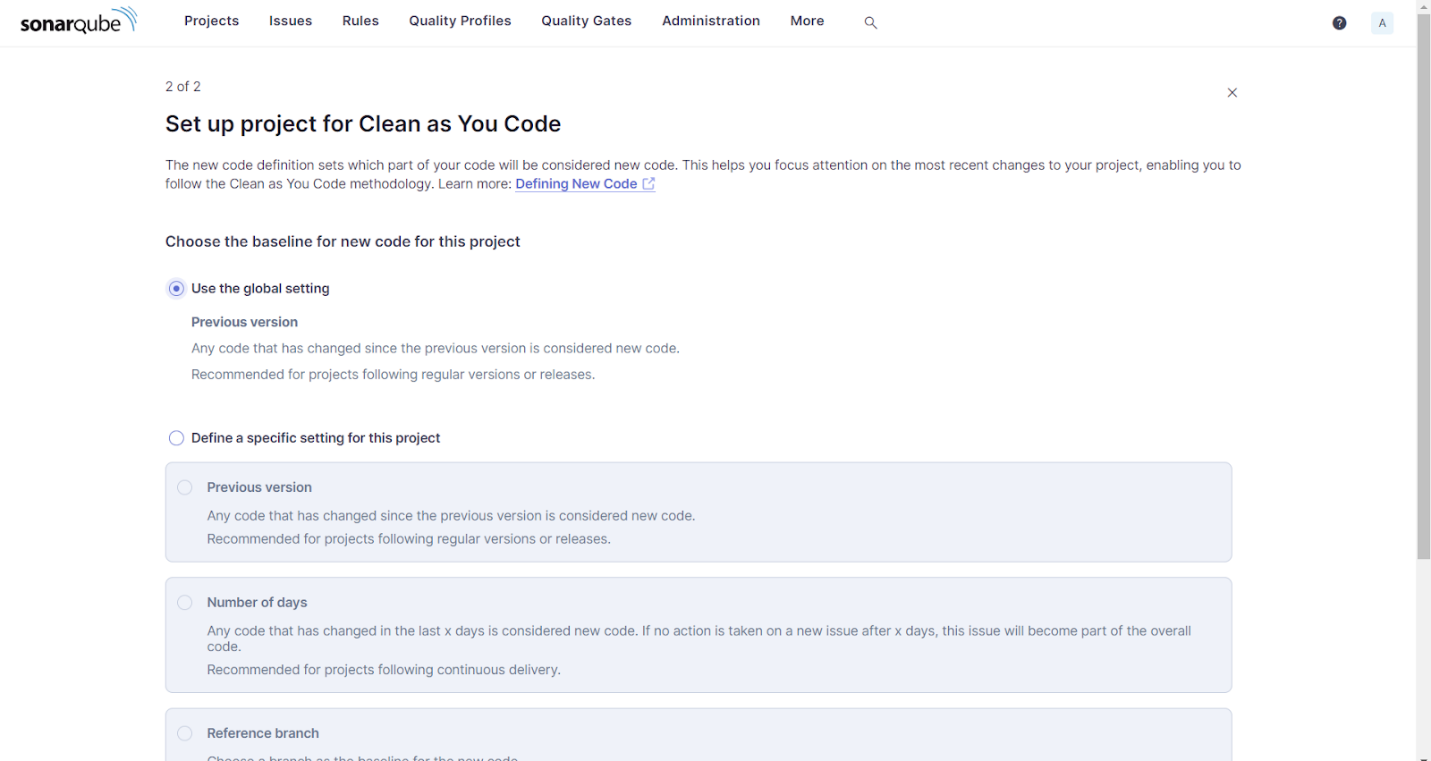
****

**Update to new password**

**Create project manually:**

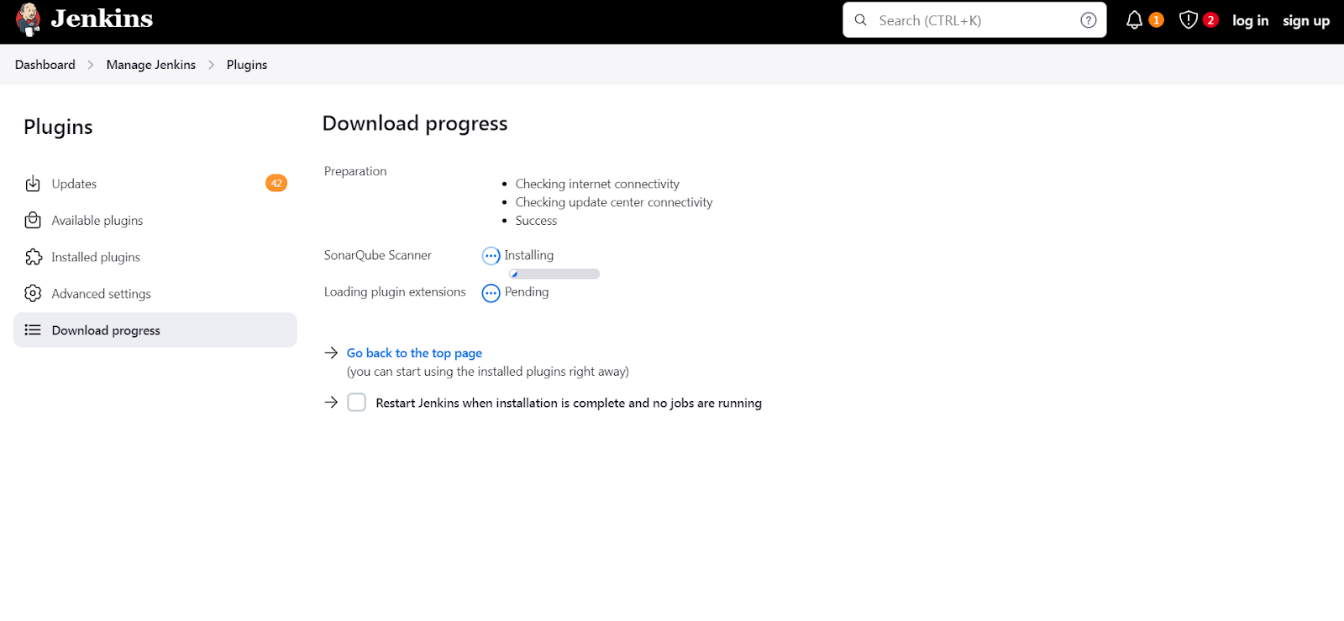
Here project name is “AdDevops”



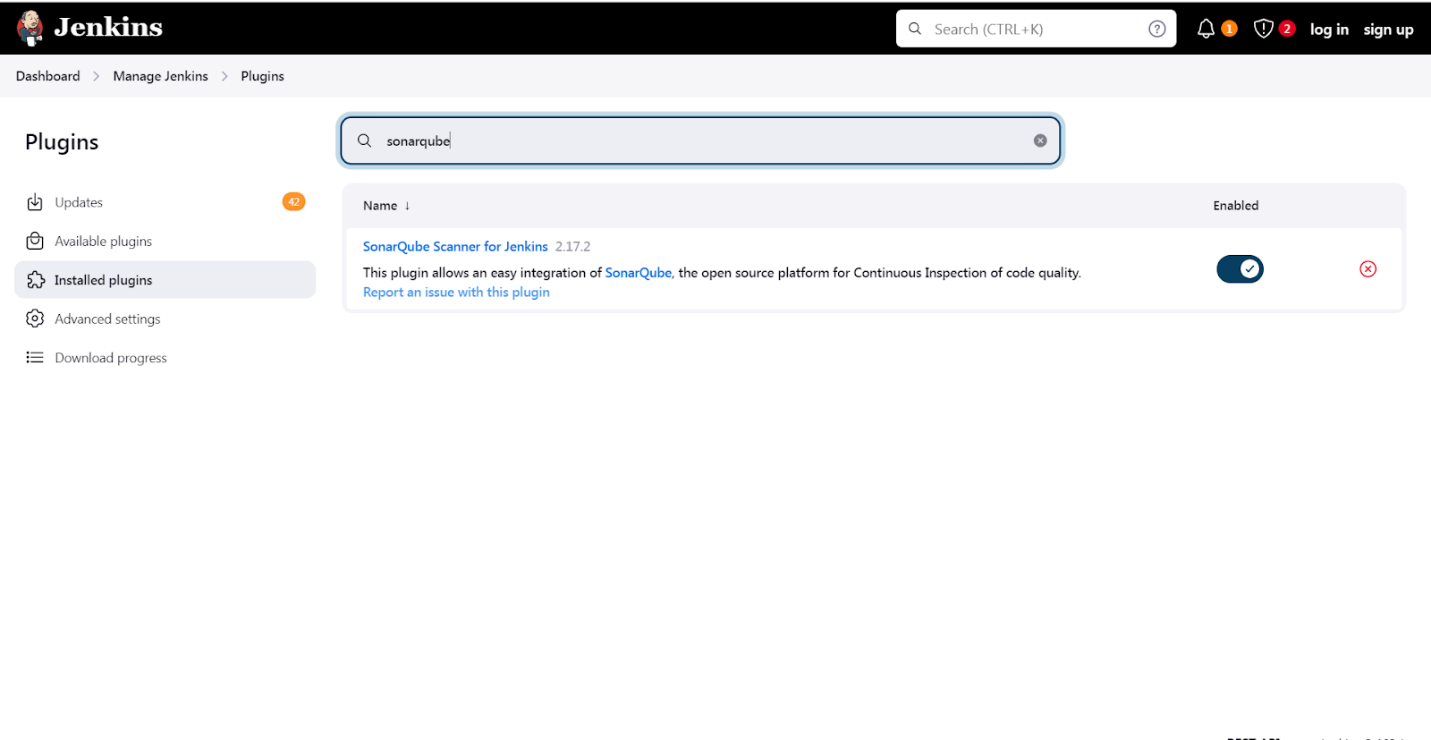


**open Jenkins**

Go to Dashboard ->Manage Jenkins -> Plugin Manager and search for



SonarQube Scanner under Available plugins for Jenkins and install without restart

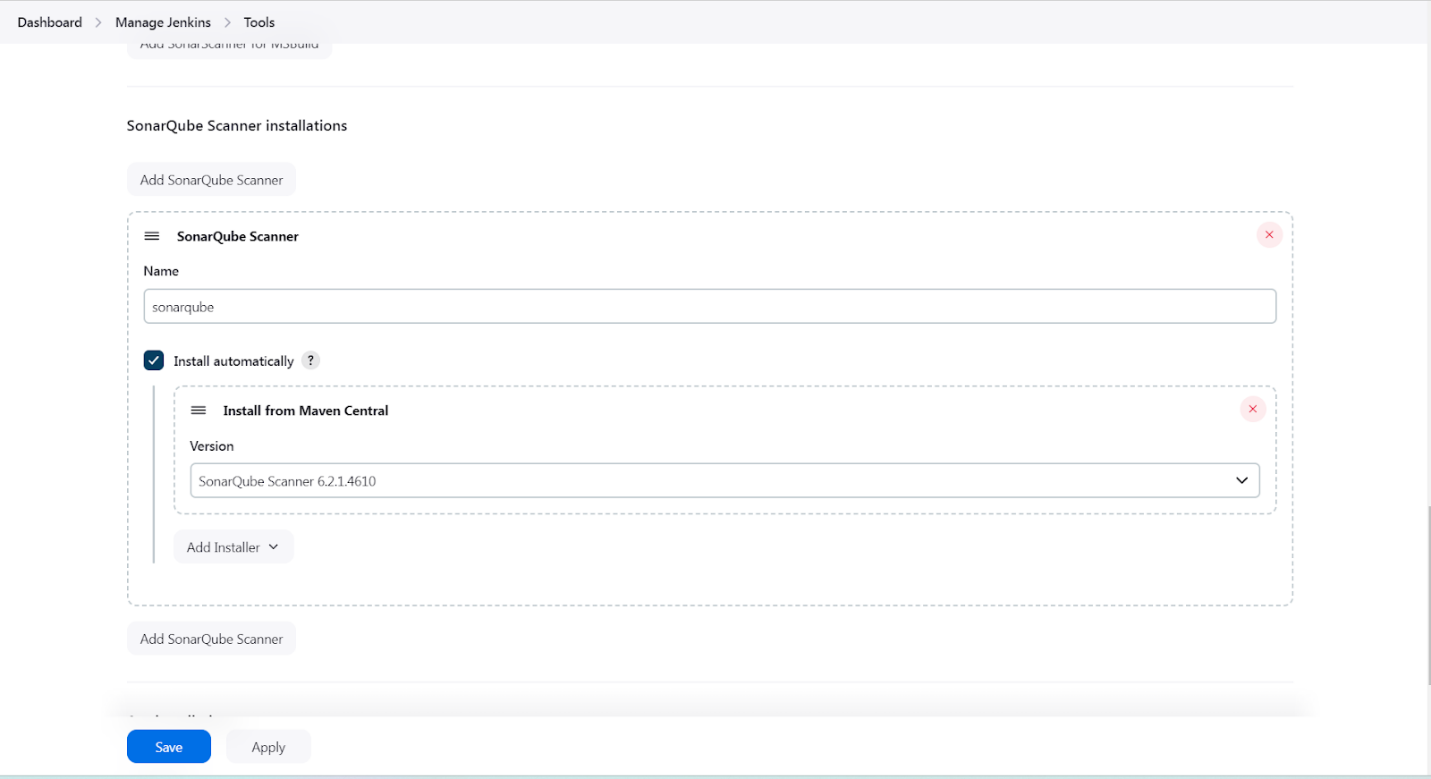


**Under Jenkins** ,

Dashboard -> Manage Jenkins -> Configure System ,

Look for SonarQube Servers and enter the details. Enter the Server Authentication Token

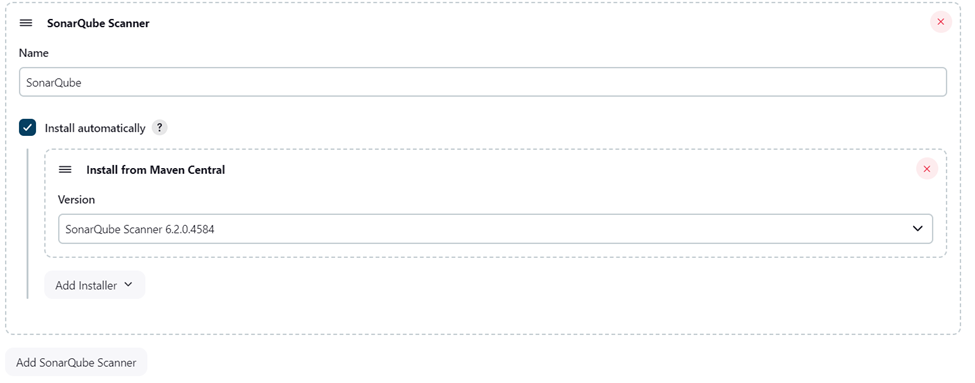
if needed.



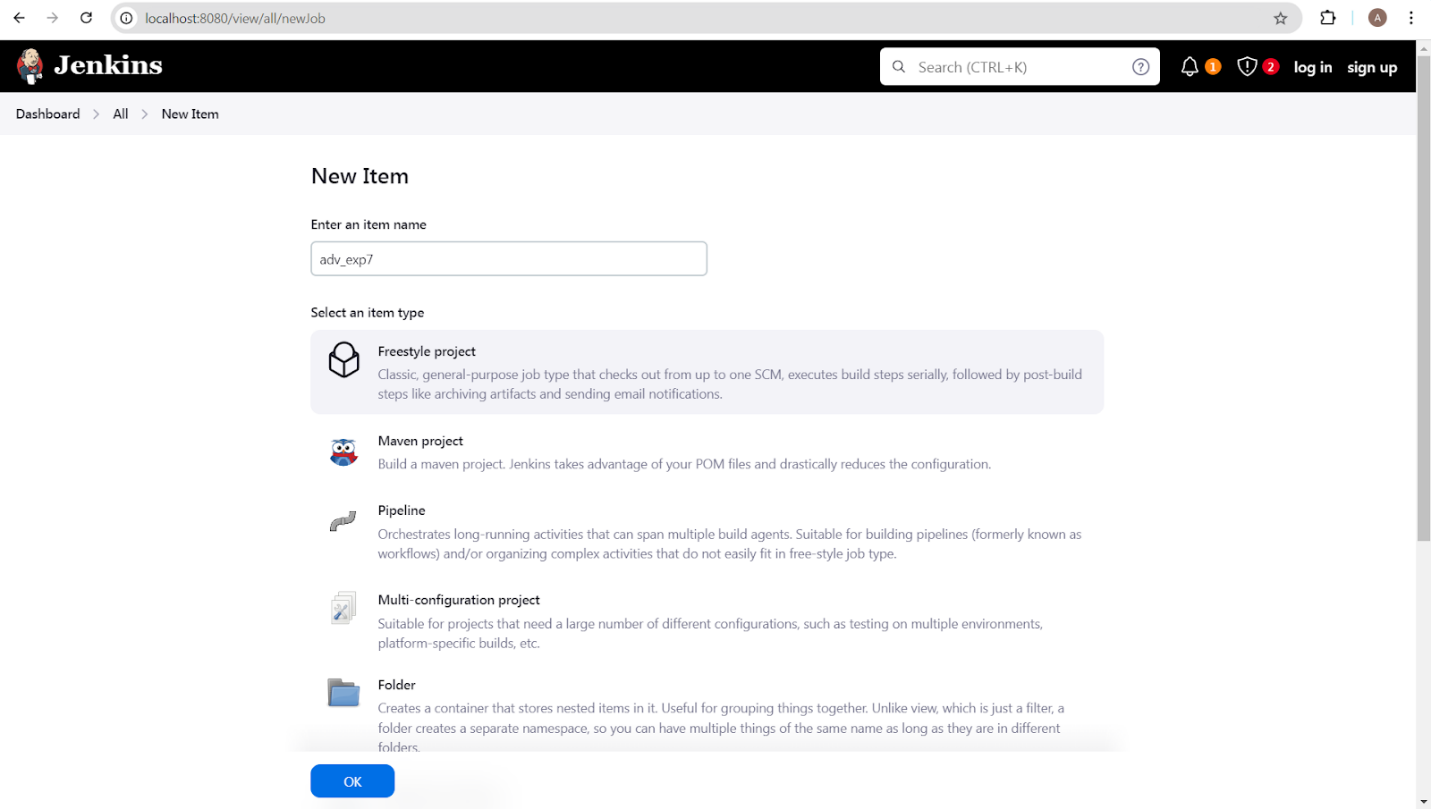
Search SonarQube Scanner under Dashboard -> Manage Jenkins -> Global

Tool Configuration.

**Choose the latest configuration and choose Install Automatically.**



**create a New Item in Jenkins, choose a freestyle project.**



**Choose this GitHub repository in Source Code Management.**

https://github.com/shazforiot/MSBuild\_firstproject.git

Under Build ->Execute SonarQube Scanner, enter these Analysis

properties. Mention the SonarQube Project Key, Login, Password, and Host

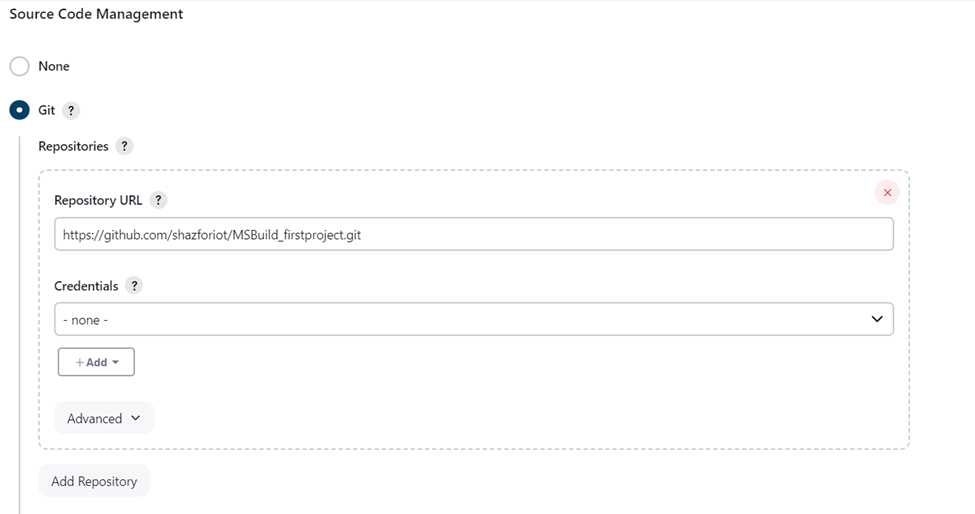
URL.

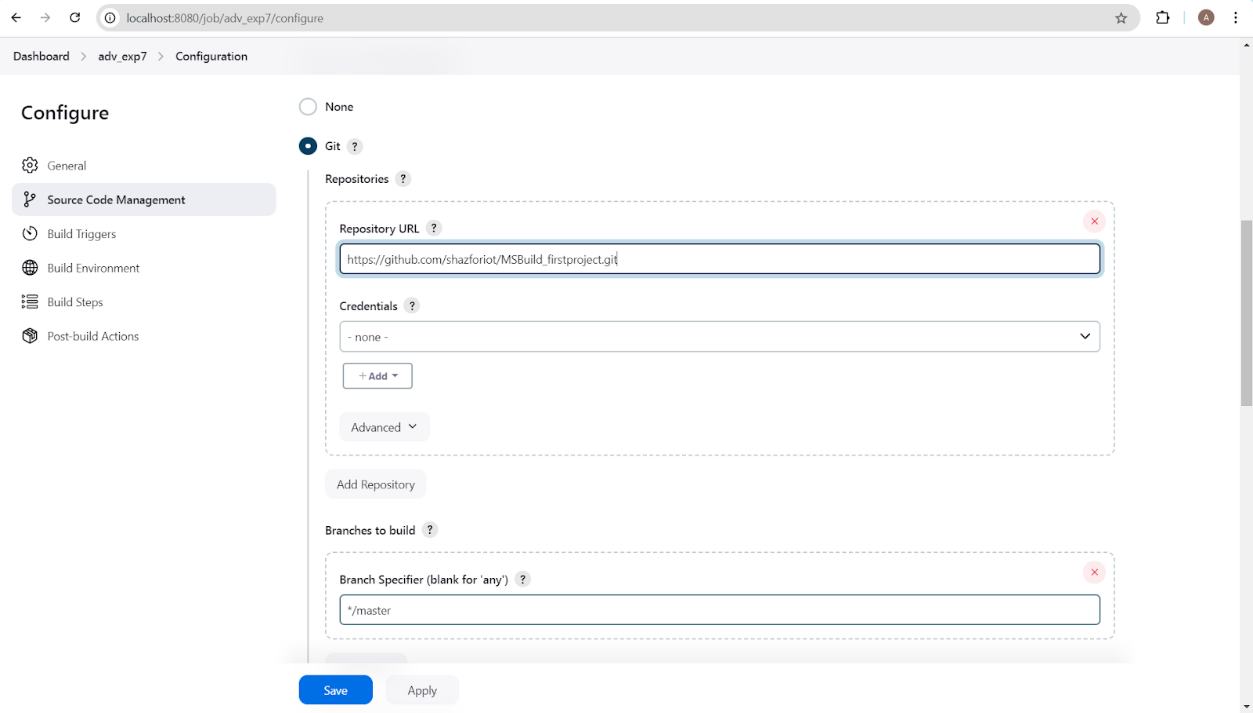
sonar.projectKey=AdDevops

sonar.login=admin

sonar.password=abc

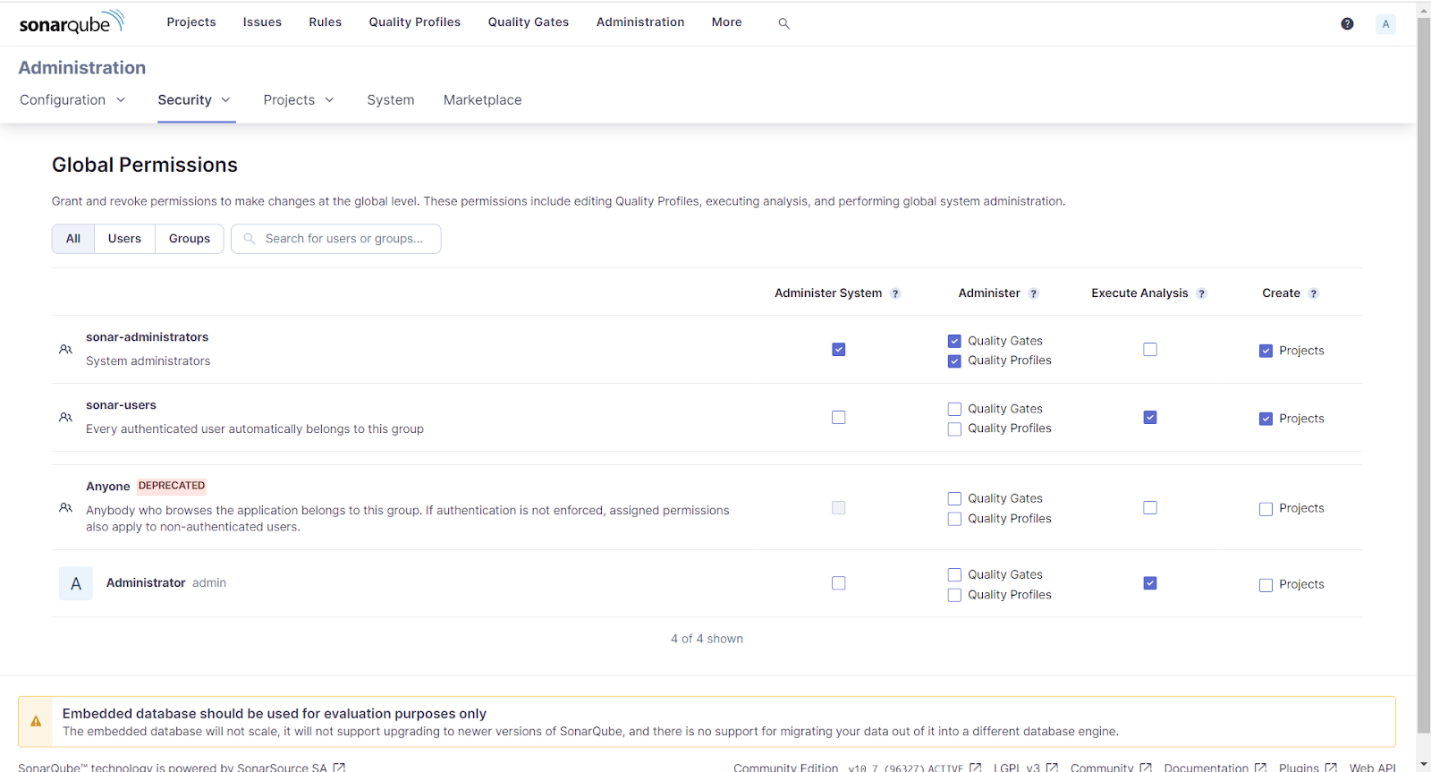
sonar.hosturl=<http://localhost:9000/>



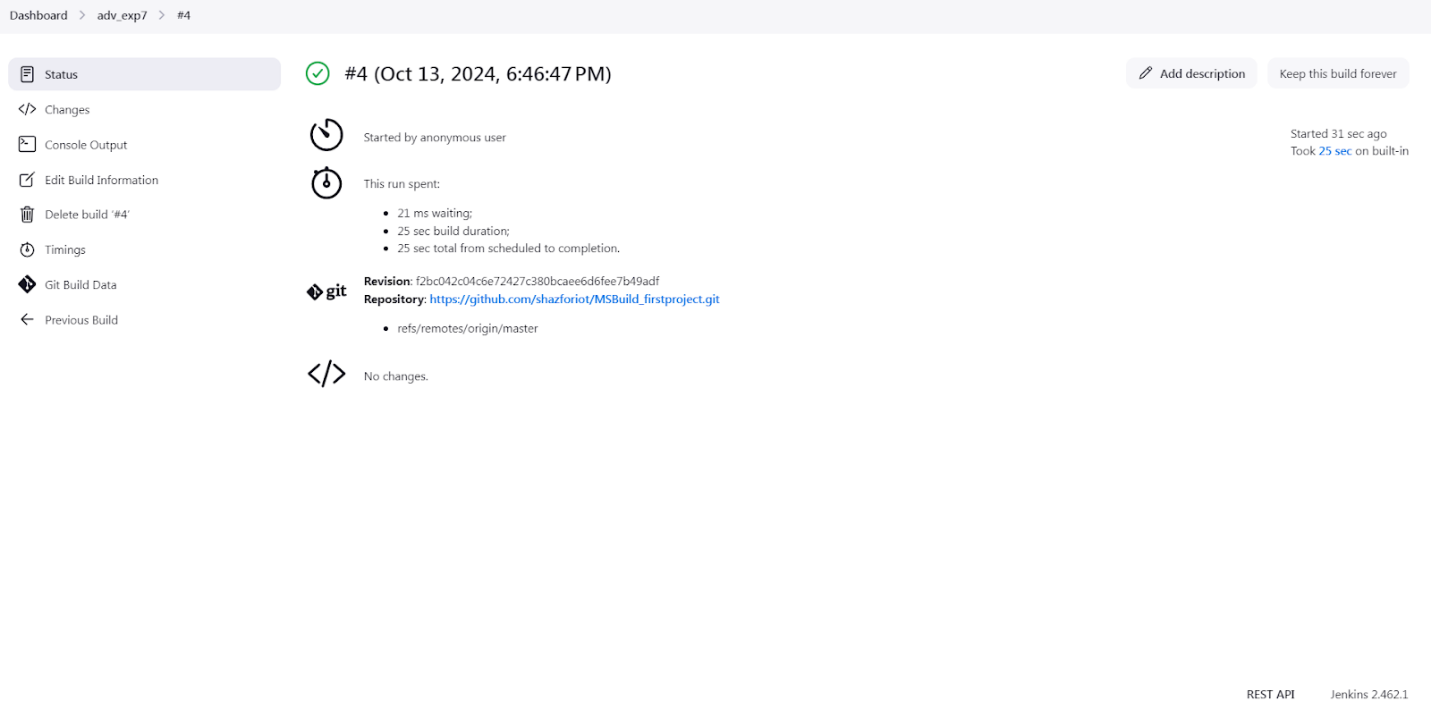


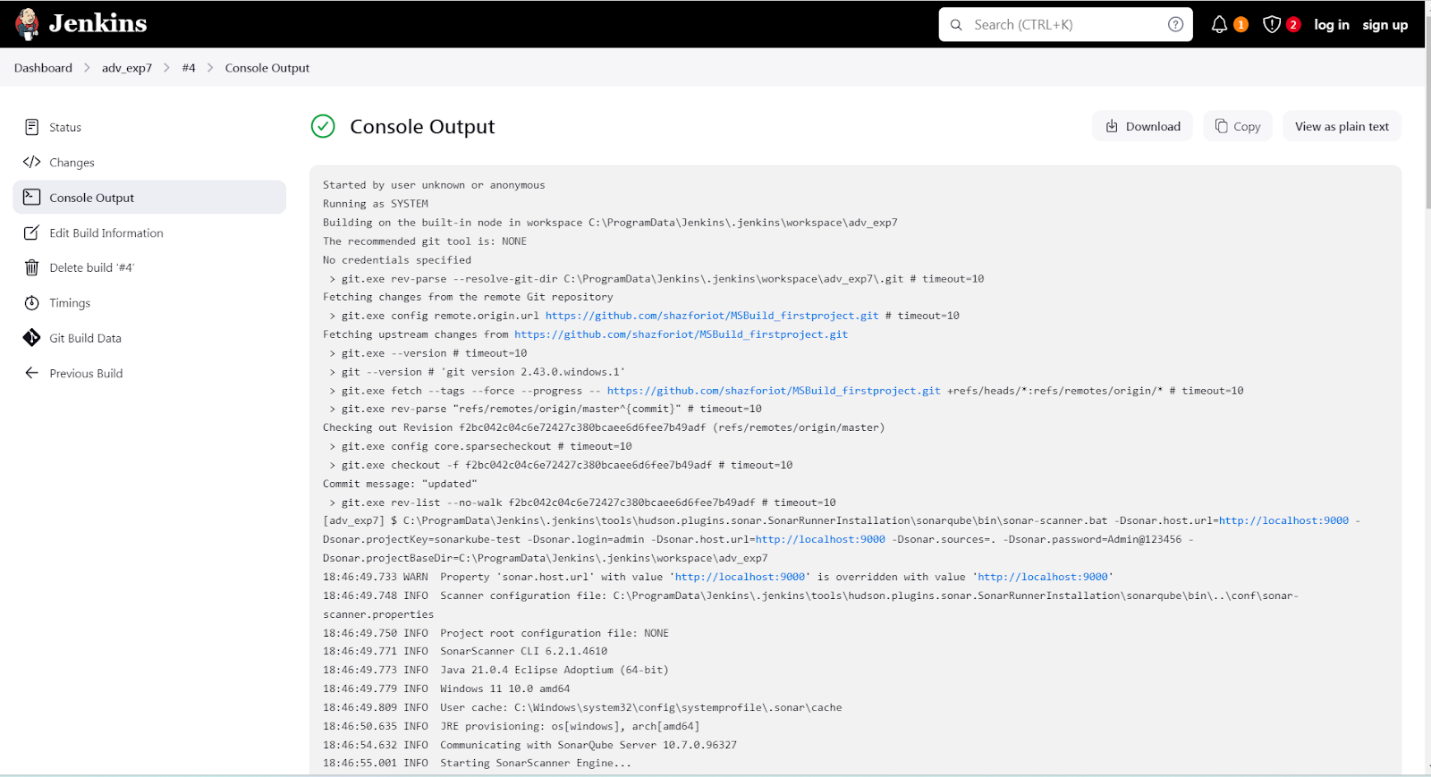
Go to http://localhost:9000/ and enter your previously created username.

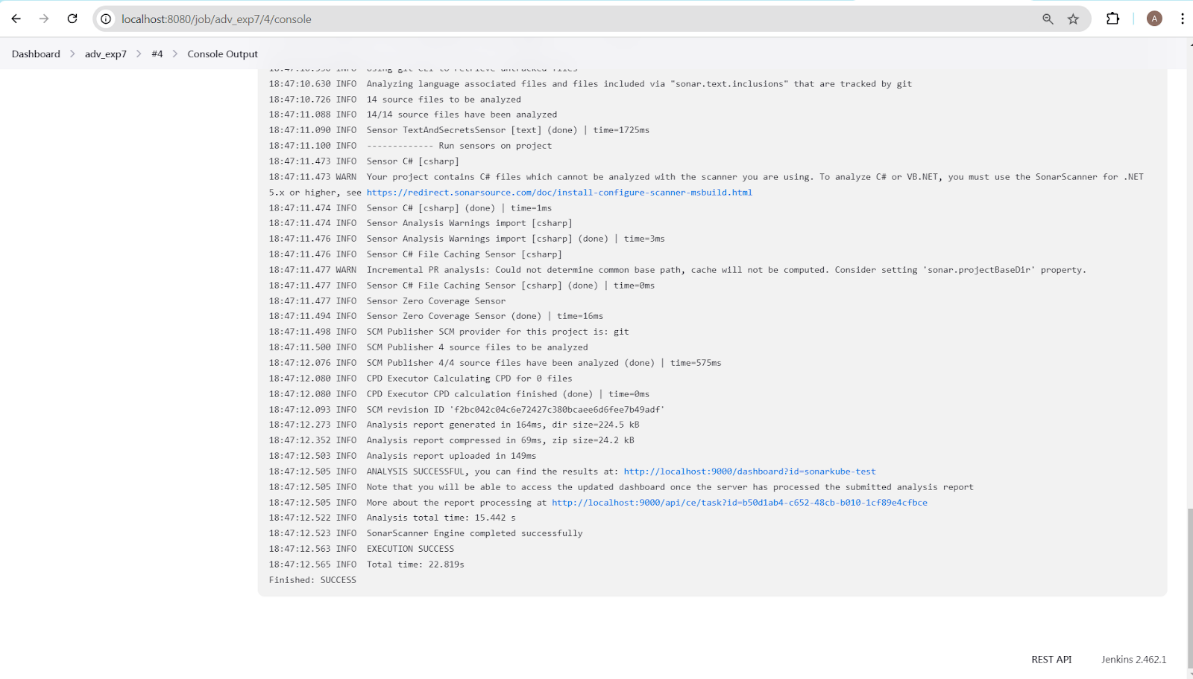
**Go to Permissions and grant the Admin user Execute Permissions.**



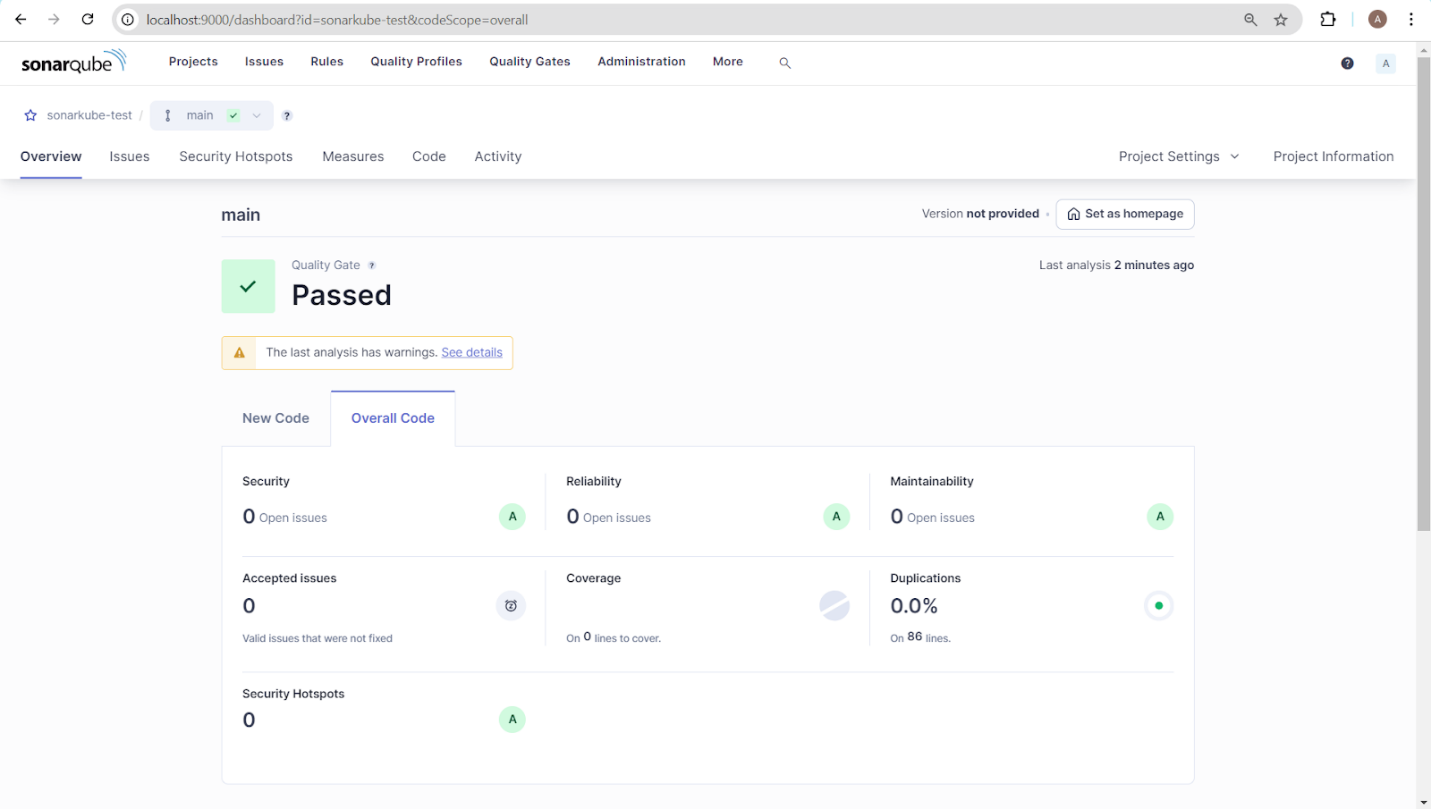
**Build and Run:**



**Console Output:** 



**Project on sonarqube:**



**Conclusion:**

Thus, we have successfully installed SonarQube using a Docker image, providing a seamless setup for continuous code quality analysis. This approach simplifies the deployment process, ensures easy scalability, and enhances development workflows with robust code review and testing capabilities in a containerized environment.