

# Experiment No. 7

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

# Theory –

1. **Static Application Security Testing (SAST):** SAST, also known as Static Code Analysis, is a security testing method that analyzes source code to identify vulnerabilities and weaknesses without executing the program. It helps developers detect security issues early in the development lifecycle by scanning the codebase for common vulnerabilities like SQL injection, cross-site scripting (XSS), buffer overflows, and more.
2. **SonarQube Overview:** SonarQube is a popular open-source platform for continuous code quality inspection. It supports static code analysis to detect bugs, code smells, and security vulnerabilities in multiple programming languages. SonarQube can be integrated with CI/CD pipelines to automate code scanning and reporting.

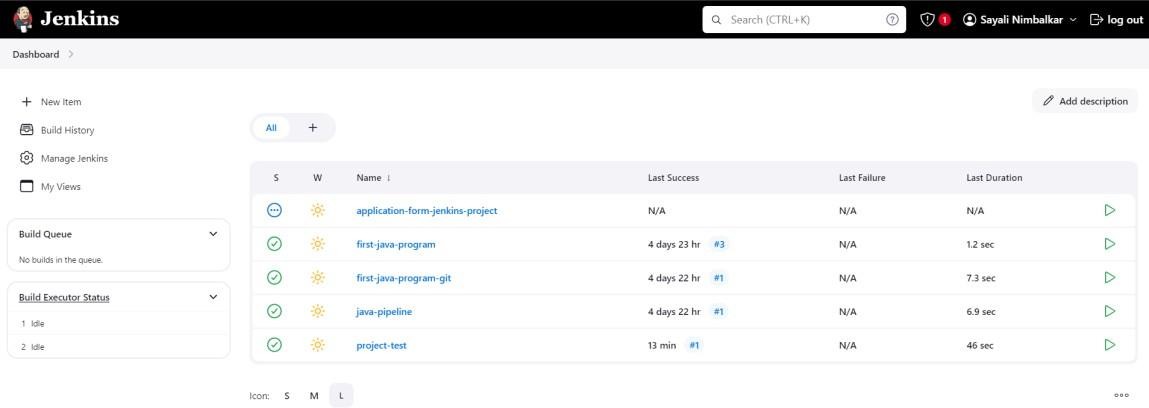
# Key Features of SonarQube:

* Supports multiple languages (Java, Python, JavaScript, etc.).
* Integrates with build tools and CI/CD systems (e.g., Jenkins, GitLab).
* Provides detailed reports with metrics on code quality, security vulnerabilities, and test coverage.

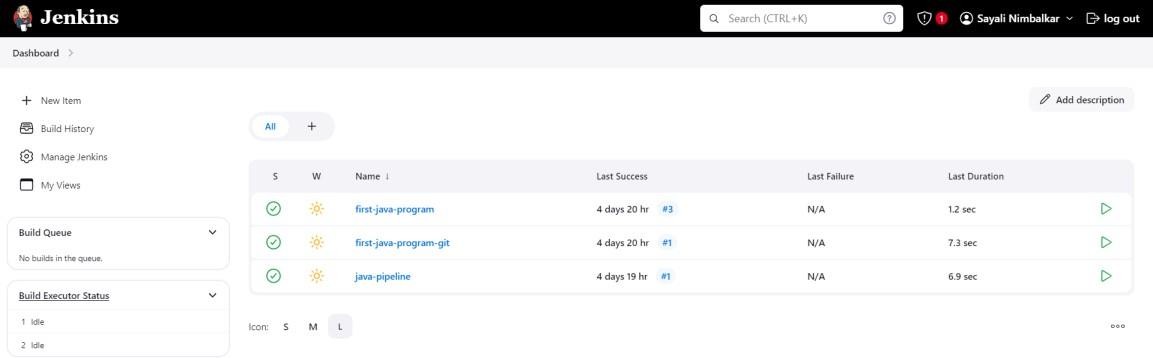
1. **Jenkins Overview:** Jenkins is an open-source automation server used for continuous integration and continuous delivery (CI/CD). It allows developers to automate the build, test, and deployment of applications. Jenkins can be extended with plugins, such as SonarQube, to integrate static analysis into the CI/CD process.
2. **GitLab Overview:** GitLab is a web-based DevOps platform that provides source code management (SCM), CI/CD pipeline automation, and code review features. GitLab can also be integrated with tools like SonarQube for automated code scanning.

# Steps to intergrate jenkins with sonarqube

Step 1: Start jekins by running by running cmd as asministartor run command ‘net start jenkins’ andthen open jekins port: localhost:808 & If Jenkins is not downloaded then download it from [https://www.jenkins.io/doc/book/installing/windows/.](https://www.jenkins.io/doc/book/installing/windows/)

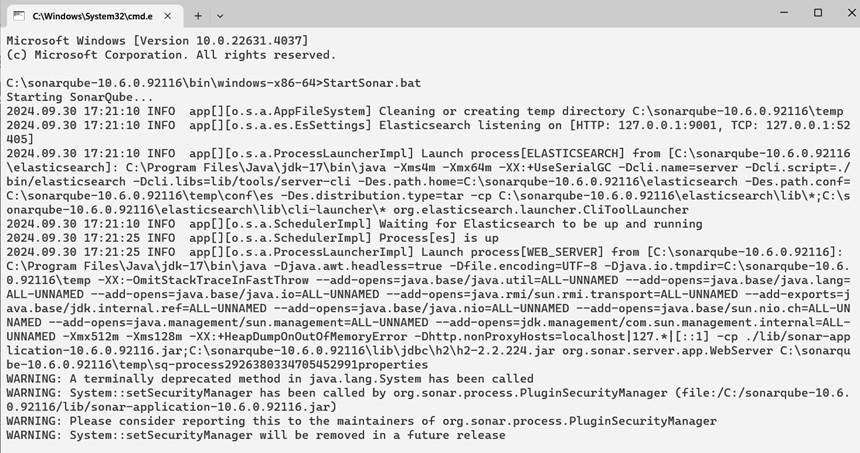


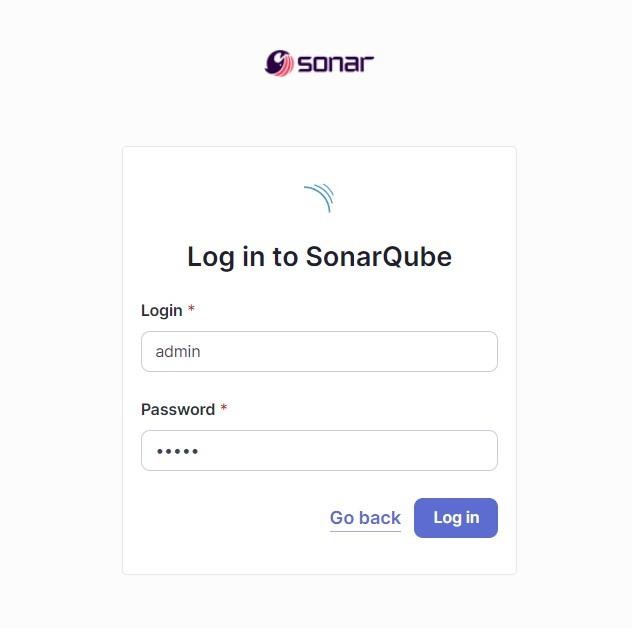
Step 2: Login to your jenkins dashboard.



Step 3: If Sonarqube is already downloaded then Start sonarqube by going to this path C:\Users\ADMIN\Downloads\sonarqube-10.6.0.92116\sonarqube 10.6.0.92116\bin\windows-x86-64 and then run StartSonar windows batch file as a administer and then open jekins port:

localhost:9000





If Sonarqube is not downaloded then Download sonarqube from website and install it, setting the environmental variable SONAR\_JAVA\_PATH on windows. Port – localhost:9000

# 1: Download and Install Oracle Java 17

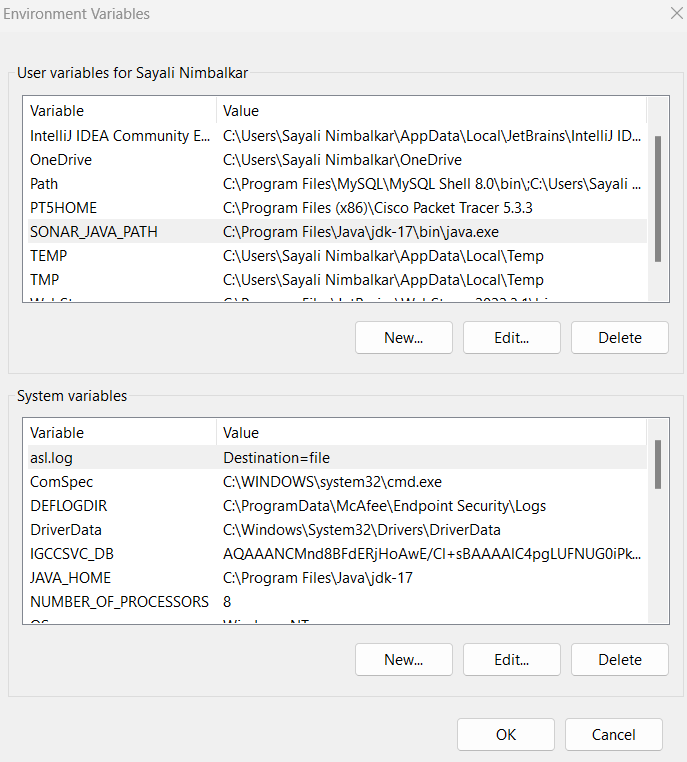
Before installing SonarQube, ensure you have Java installed. Follow these steps:

1. Download Oracle Java 17 from Oracle’s website.
2. Run the downloaded installer and follow the installation wizard.
3. Set the JAVA\_HOME environment variable to C:\Program Files\Java\jdk-17\bin.

# 2: Download and Install SonarQube Now,

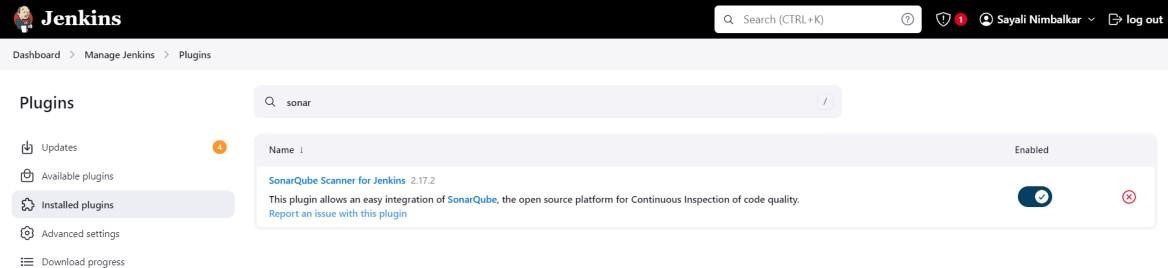
let’s install SonarQube:

1. Download the SonarQube zip file from SonarSource.
2. Extract the contents of the zip file to C:\sonarqube-10.4.1.88267. 3. Set the SONAR\_JAVA\_PATH environment variable using CMD: set "SONAR\_JAVA\_PATH=C:\ProgramFiles\Java\jdk-17\bin\java.exe"

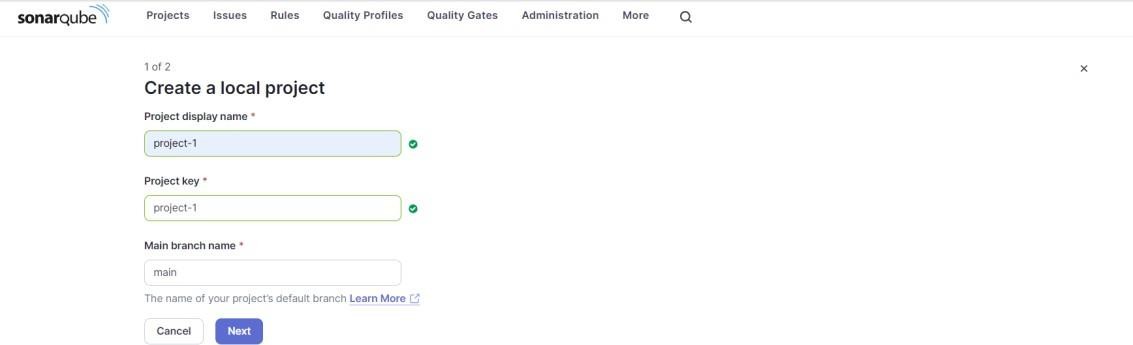


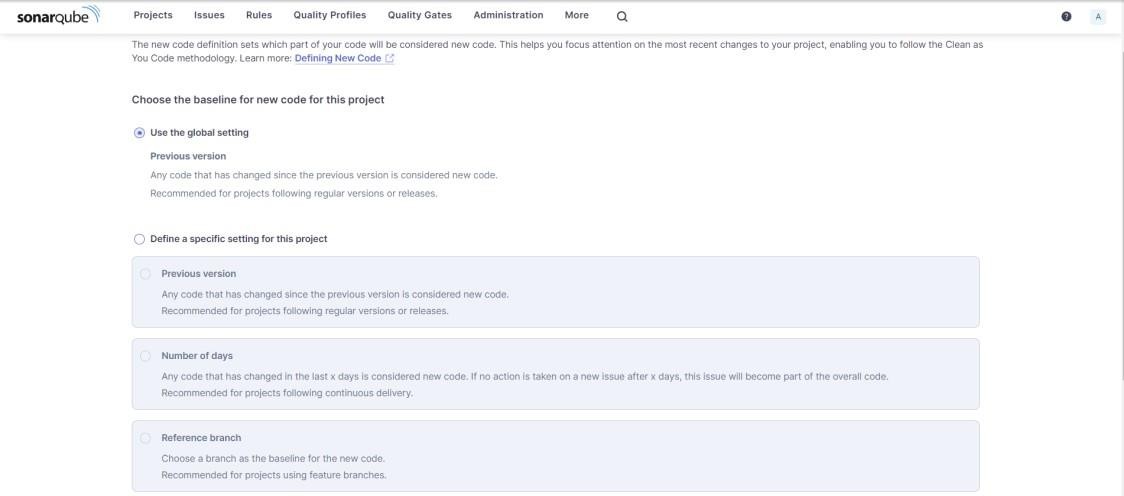
And then start it like shown in step 3.

Step 4: First go to Jenkins dashboard and download the SonarQube plugin. Then restart the Jenkins.

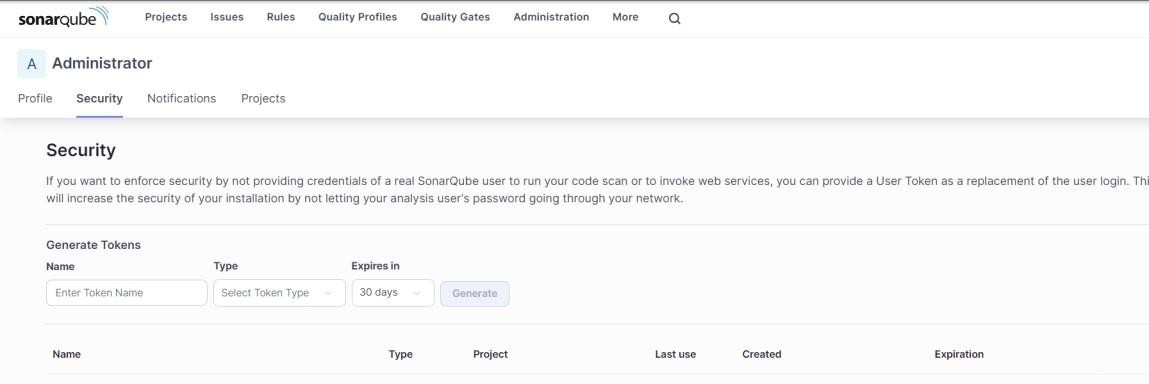


Step 5: Create a project in SonarQube.

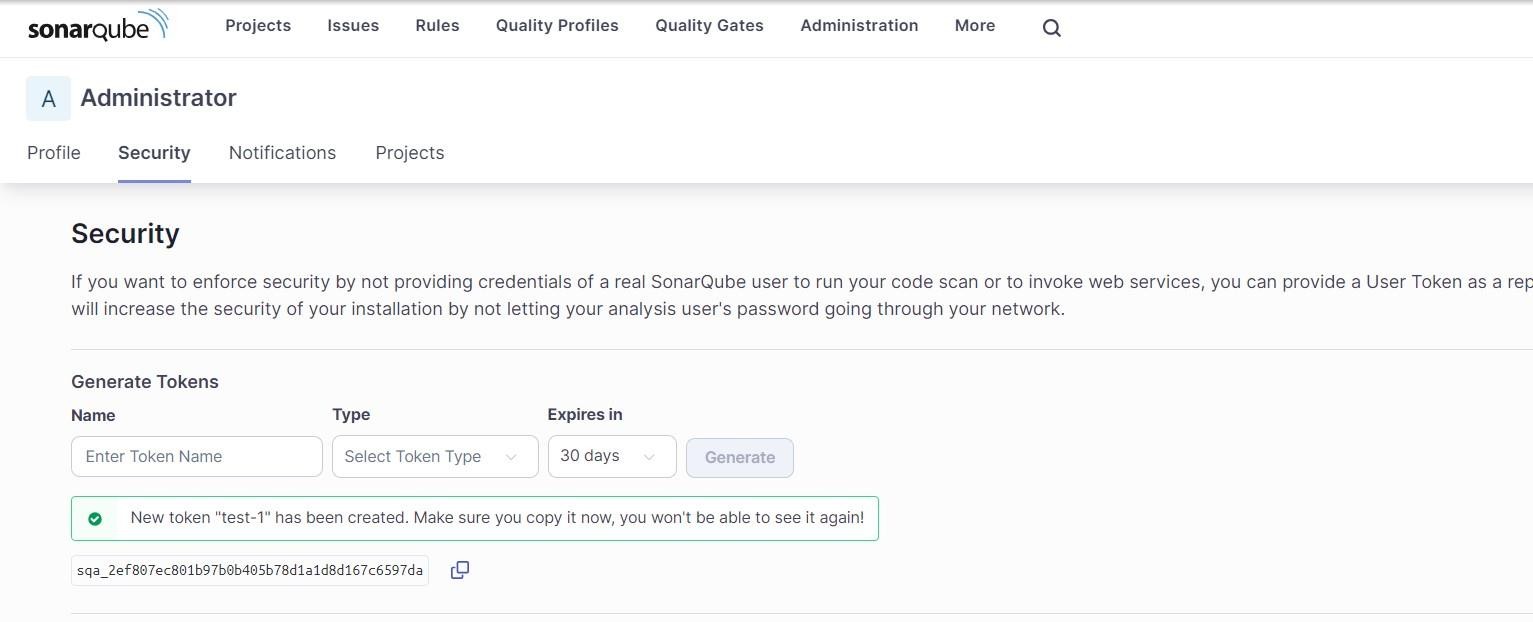




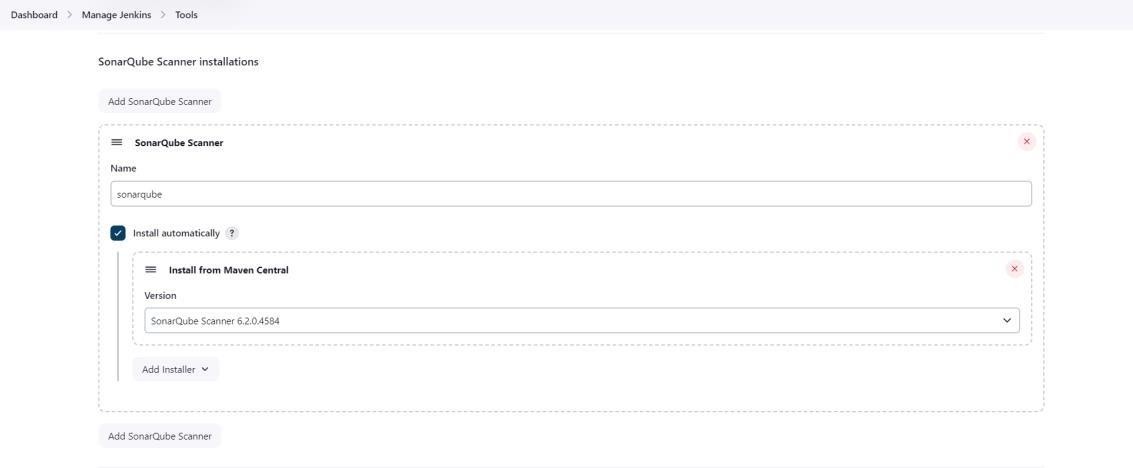
Step 6: Now go to sonarqube server. Here go to my account > security.



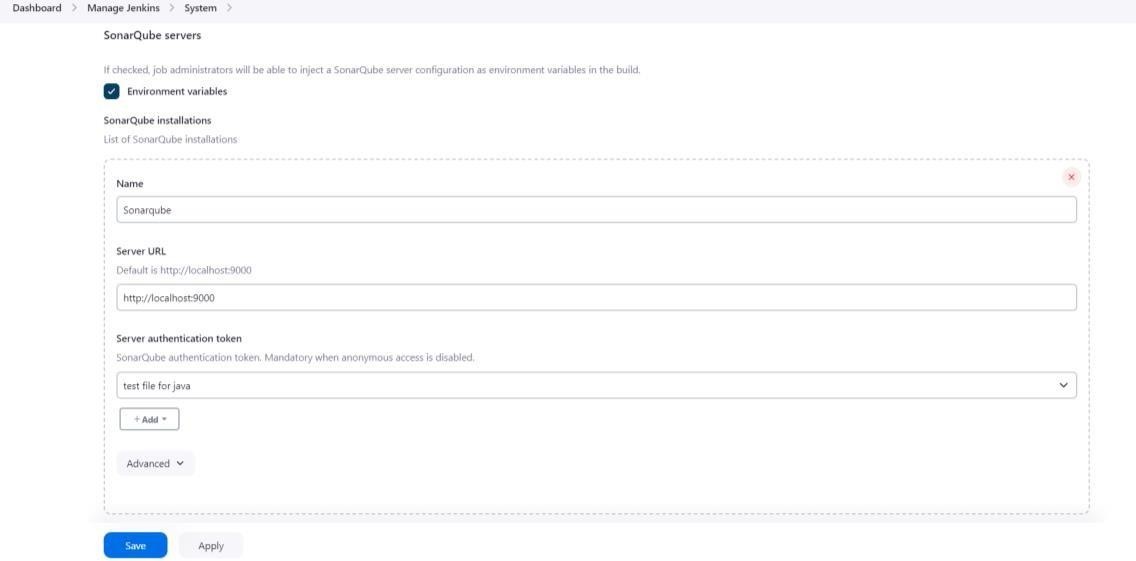
1. Now create a token for Jenkins by clicking on ‘create token’.



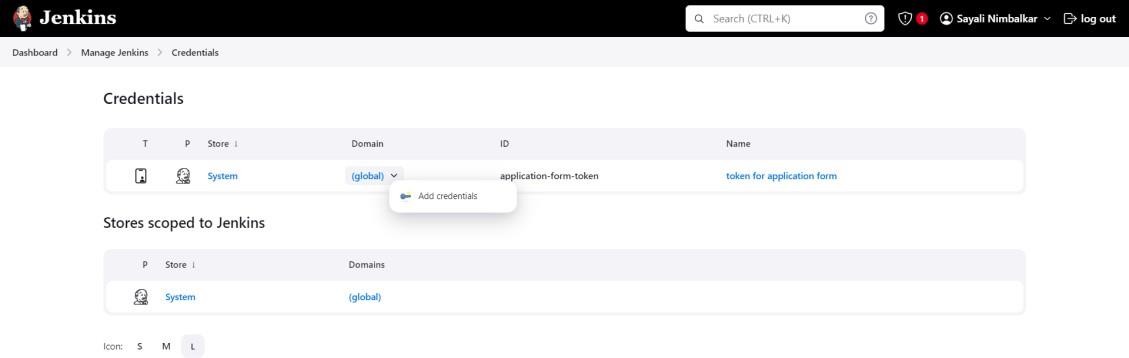
1. Go to Jenkins dashboard and go inside the tools section. Here you setup a sonar scanner which will scan the code, fill details and then apply > save.

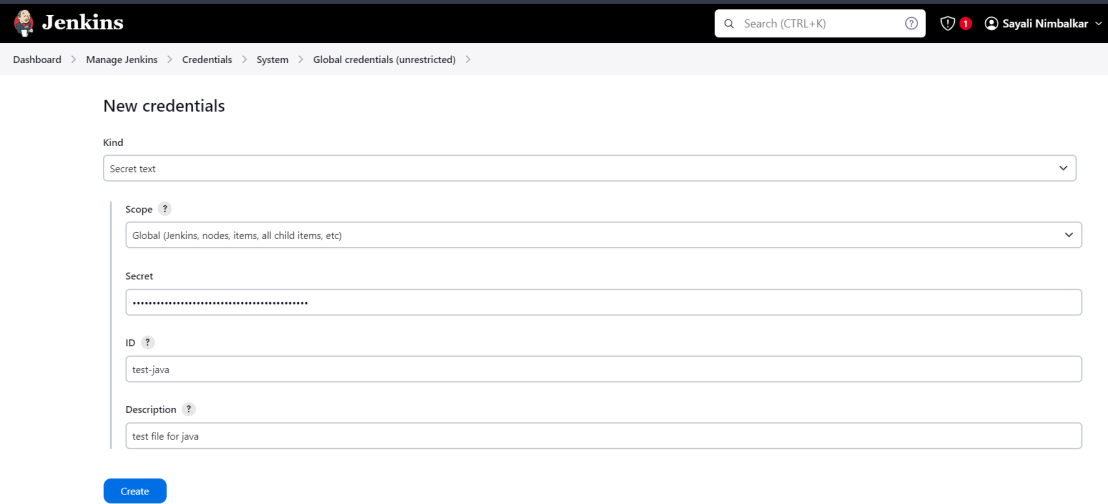


1. Now go into system section to configure sonarqube server and attach the credential to the sonarqube configuration.

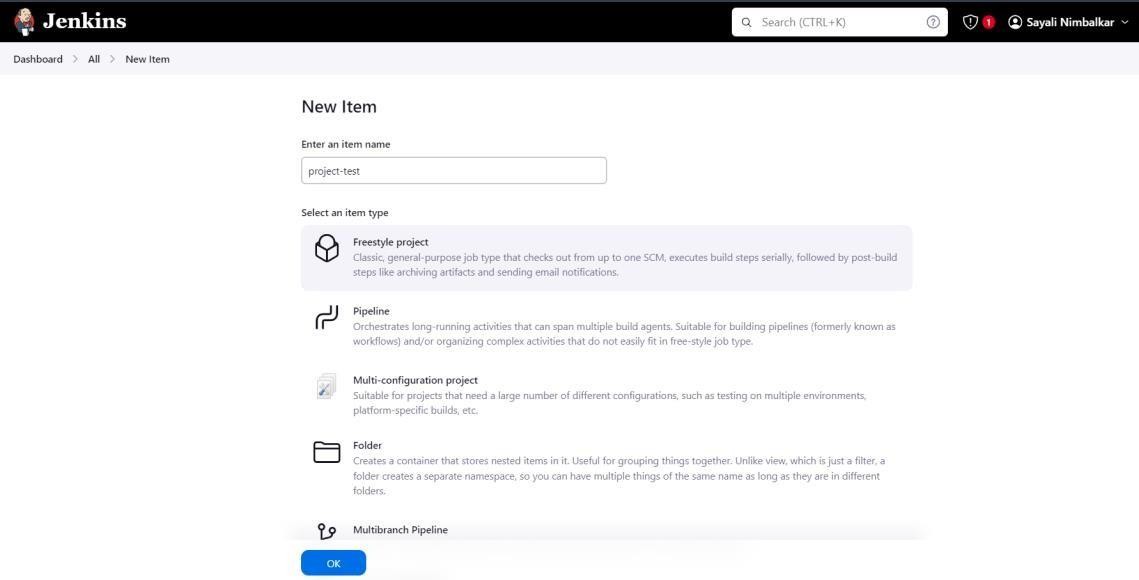


1. Create the secret text credential using token that is generated in step 3.

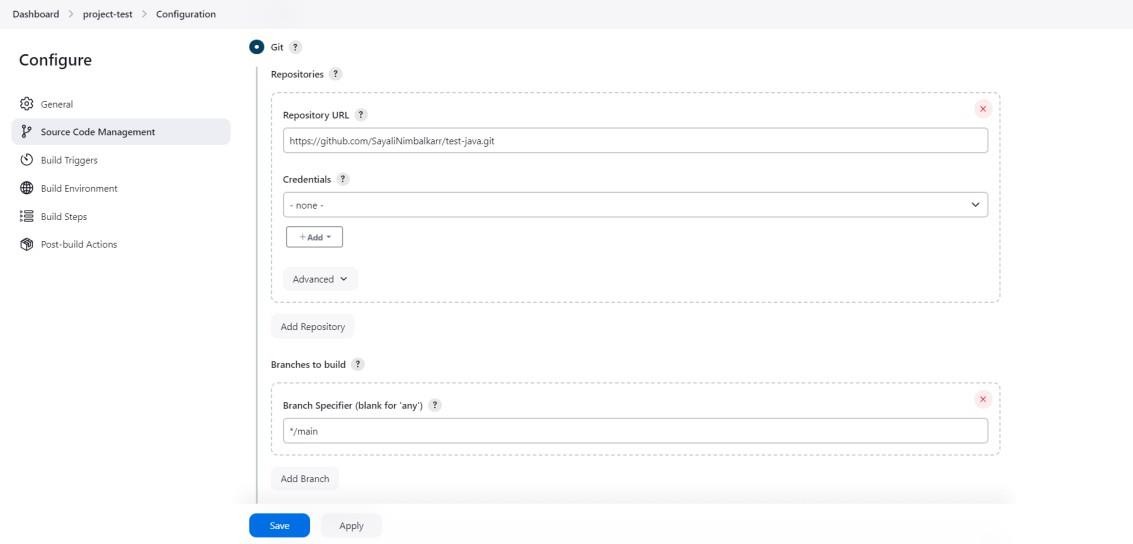




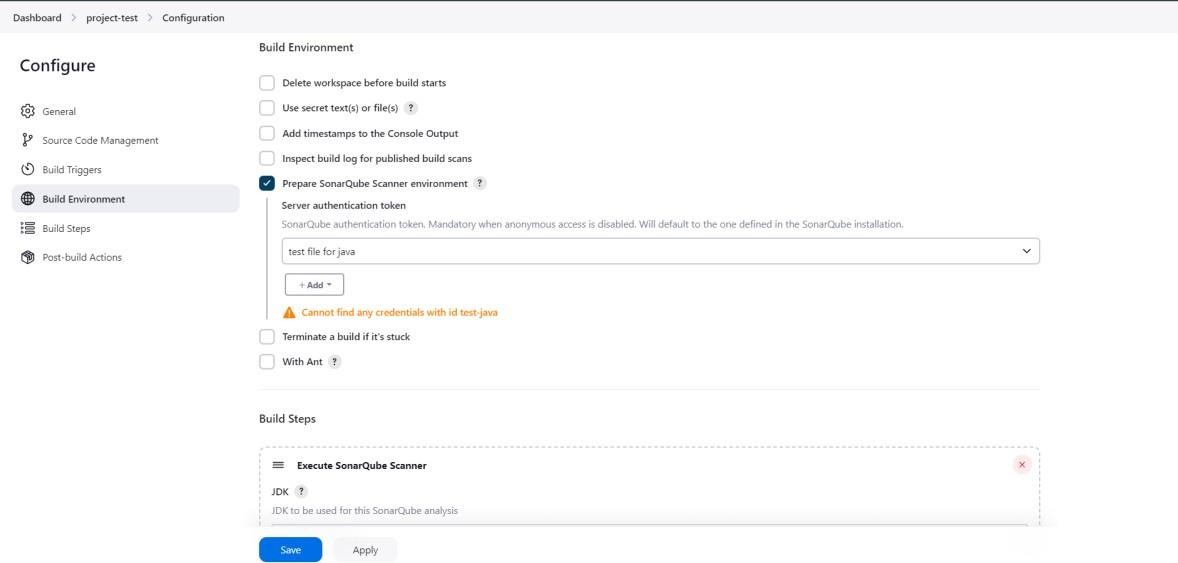
1. Create a pipeline. Give the freestyle project a name and select ok.



1. Select your source code repository if it on any git server.



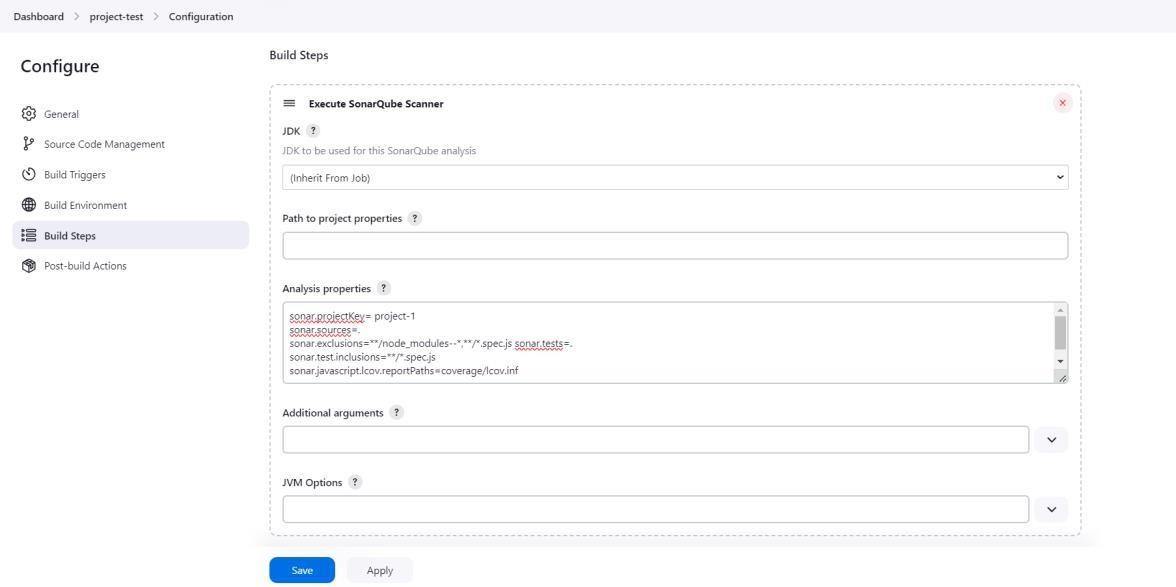
1. Select token we have created.



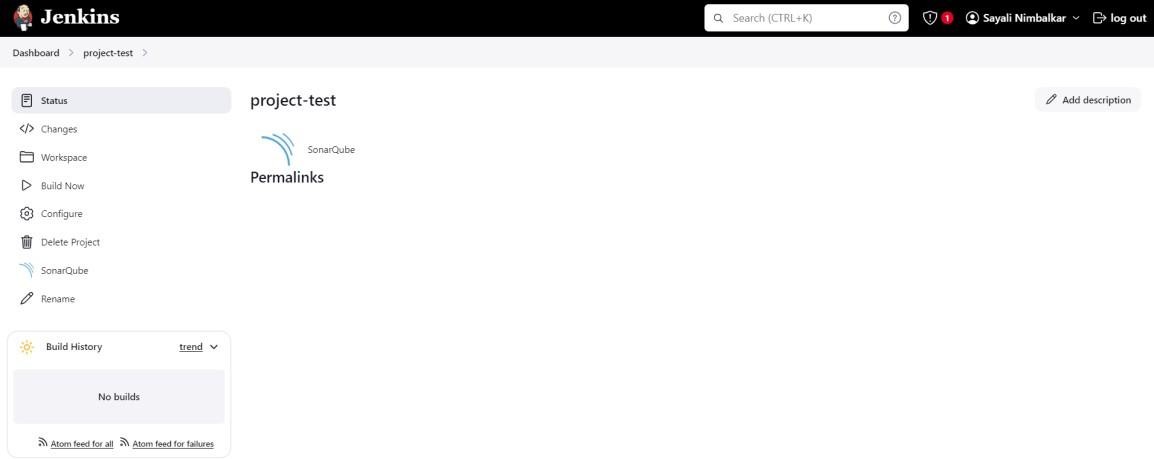
1. In Build step, just add this in analysis properties of Execute SonarQube scanner sonar.projectKey= <your\_project\_name>

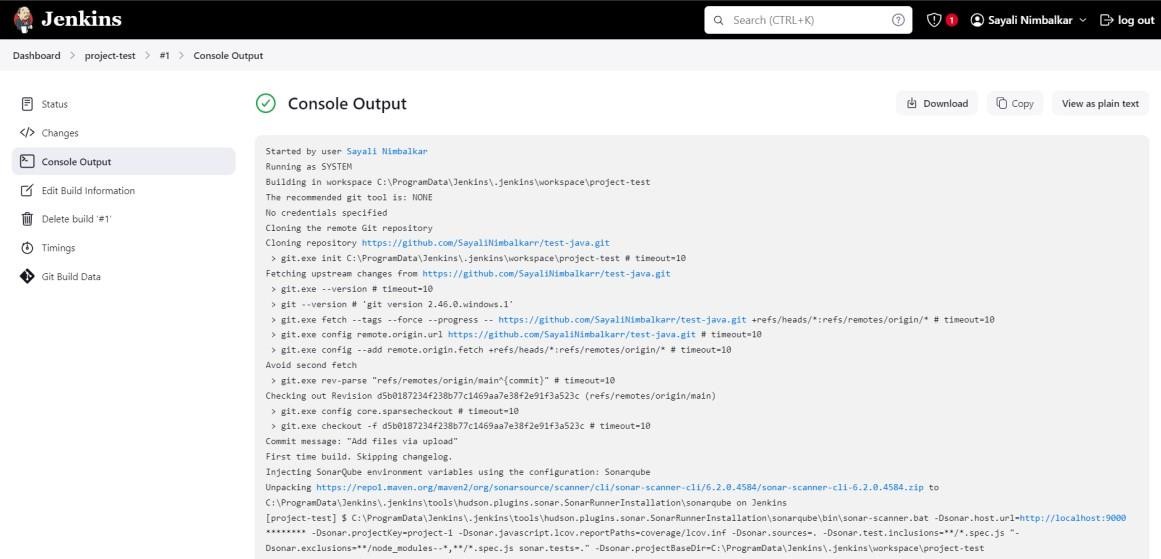
sonar.sources=.

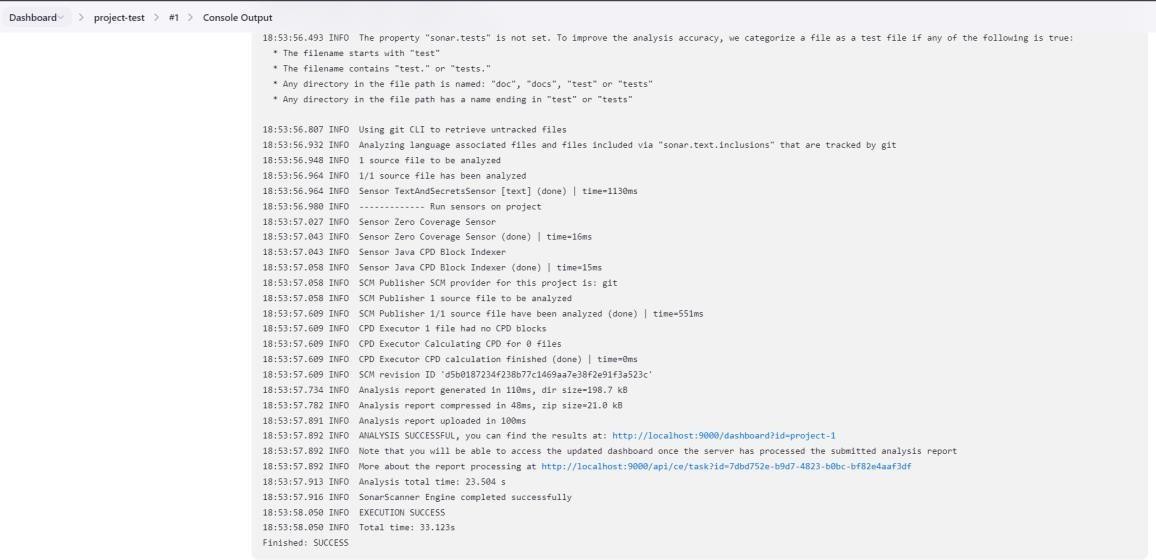
sonar.exclusions=\*\*/node\_modules--\*,\*\*/\*.spec.js sonar.tests=.sonar.test.inclusions=\*\*/\*.spec.js sonar.javascript.lcov.reportPaths=coverage/lcov.inf



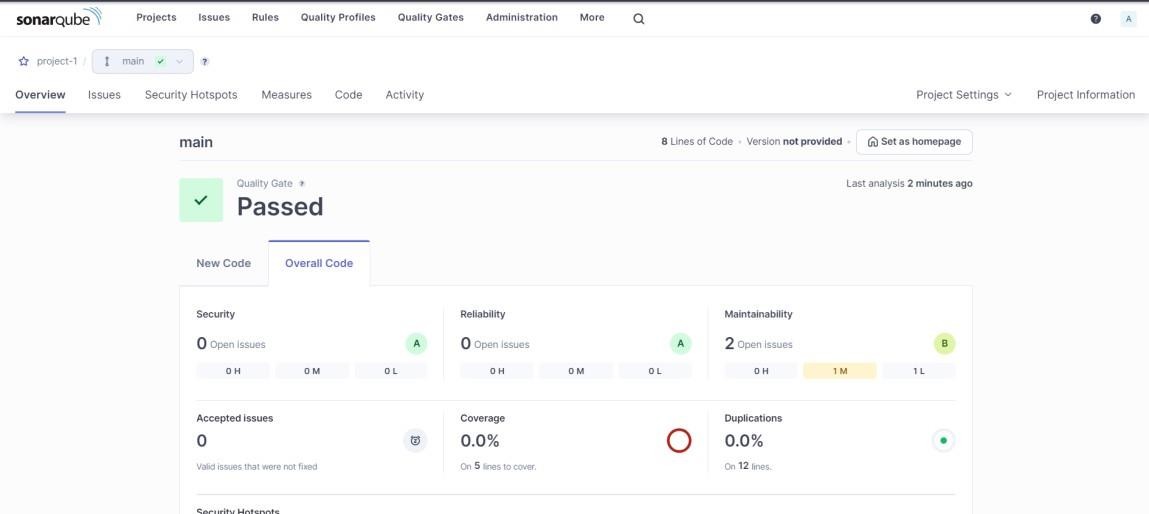
1. Click on ‘Build Now’ on tleft side and scroll down to see build.







1. If build is successful then go to sonarQube project.



# Conclusion –

Therefore, we completed the experiment successfully.