

Assignment - 3 - Integrate SonarQube scanner into Pipeline

What is sonarqube?

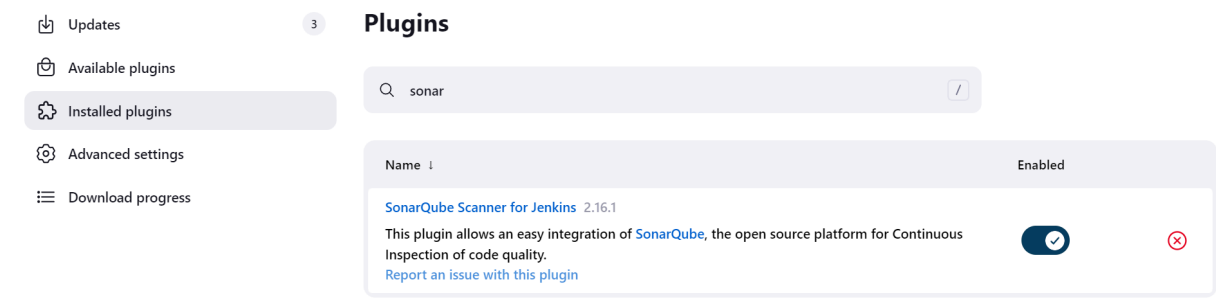
SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells on 29 programming languages

Pre-requisites and steps:

1. Install and run sonarqube in docker . Refer the doc
2. In the jenkins, install the following plugins sonarqube scanner for jenkins and pipeline maven integration

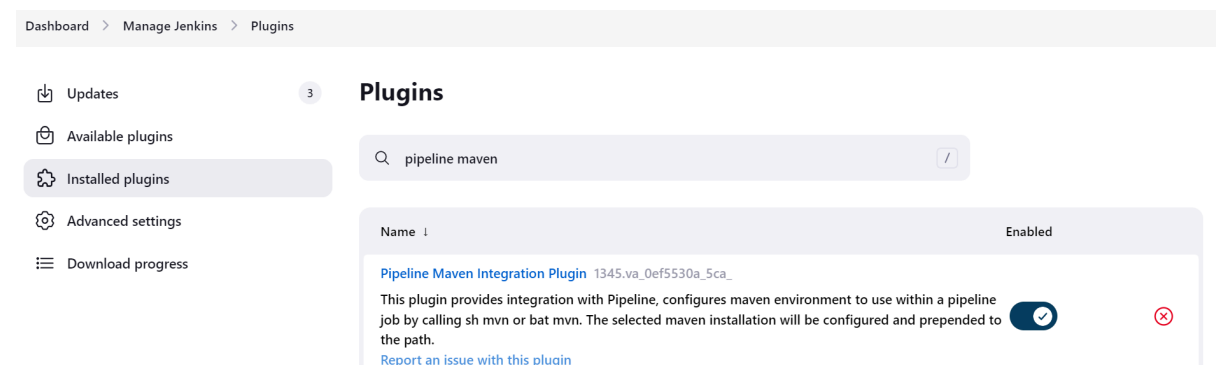
Plugin Manager

a:



The screenshot shows the Jenkins Plugin Manager interface. On the left, a sidebar contains links for Updates, Available plugins, Installed plugins (selected), Advanced settings, and Download progress. The main area is titled 'Plugins' and has a search bar with 'sonar' entered. Below the search bar, a table lists installed plugins. The first entry is 'SonarQube Scanner for Jenkins' version 2.16.1, which is enabled (indicated by a blue toggle switch) and has a red 'X' icon in the right column. The description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. Report an issue with this plugin'.

Name	Enabled
SonarQube Scanner for Jenkins 2.16.1 This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality. Report an issue with this plugin	<input checked="" type="checkbox"/>

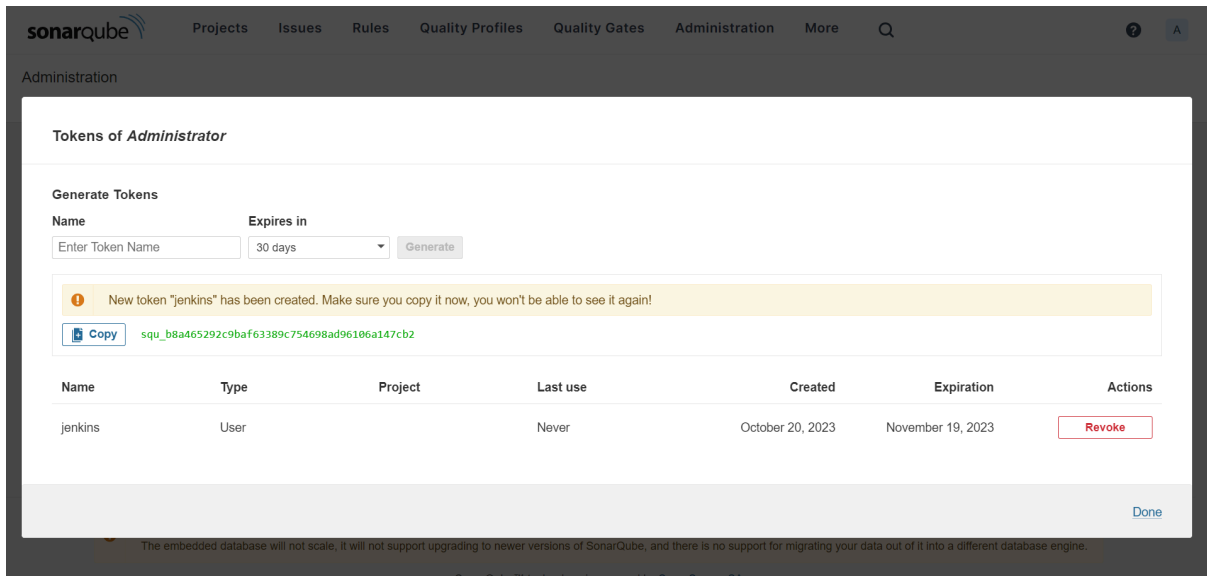


The screenshot shows the Jenkins Plugin Manager interface. On the left, a sidebar contains links for Updates, Available plugins, Installed plugins (selected), Advanced settings, and Download progress. The main area is titled 'Plugins' and has a search bar with 'pipeline maven' entered. Below the search bar, a table lists installed plugins. The first entry is 'Pipeline Maven Integration Plugin' version 1345.va_0ef5530a_5ca_, which is enabled (indicated by a blue toggle switch) and has a red 'X' icon in the right column. The description states: 'This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path. Report an issue with this plugin'.

Name	Enabled
Pipeline Maven Integration Plugin 1345.va_0ef5530a_5ca_ This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path. Report an issue with this plugin	<input checked="" type="checkbox"/>

3. Login to your sonarqube server -> Administration -> User -> Security -> tokens
4. Generate a token for jenkins and save it

A:

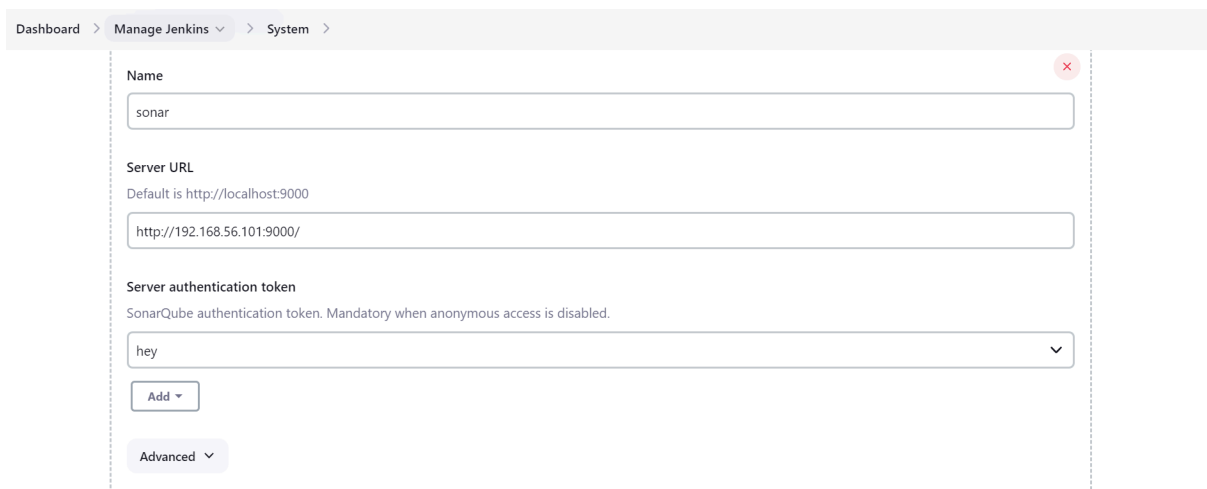


5. Navigate to Manage Jenkins -> configure system -> sonarqube servers

6. Provide the name sonar, your sonarqube server url and port, in the dropdown select the secret text that is generated from your server

(Note : You need to add sonarqube credentials in Manage Jenkins -> credentials -> system -> Global credentials . Select kind and copy sonarqube secret key, provide any id and description and add it)

A:



7. Now create a pipeline using the script placed in the link & Jenkins Script

8. Build the pipeline and verify the build status

A:

Dashboard > jenkins1_pipe > #5

- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete build '#5'
- Git Build Data
- See Fingerprints
- Test Result
- Maven
- Restart from Stage

Console Output

```
Started by user gujja pranay kumar
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/jenkins1_pipe
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Source)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/jenkins1_pipe/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/maraks-gradiious/maven-test1 # timeout=10
Fetching upstream changes from https://github.com/maraks-gradiious/maven-test1
> git --version # timeout=10
> git --version # 'git version 1.8.3.1'
> git fetch --tags --progress https://github.com/maraks-gradiious/maven-test1 +refs/heads/*:refs/remotes/origin/*
# timeout=10
```

9. Now navigate to sonarqube server to check the code quality results

A:

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Search for projects... Perspective Overall Status Sort by Name 1 project(s)

Filters

Quality Gate

- Passed 1
- Failed 0

Reliability

- A 1
- B 0
- C 0
- D 0
- E 0

java-maven-junit-helloworld PUBLIC Passed

Last analysis: 9 minutes ago · 210 Lines of Code · XML, Java

0 Bugs 0 Vulnerabilities A Hotspots Reviewed A 21 Code Smells 94.6% Coverage 0.0% Duplications

1 of 1 shown

10. You can click your project and see the code test results,

A:

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

java-maven-junit-helloworld / main Passed The last analysis has warnings. See details Version 2.0-SNAPSHOT

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Quality Gate Passed

Enjoy your sparkling clean code!

New Code Overall Code

Reliability 0 Bugs A

Maintainability 21 Code Smells A

Security 0 Vulnerabilities A

Security Review 0 Security Hotspots A

Coverage 94.6% Coverage

Duplications 0.0% Duplications

Script:

```
pipeline {
  agent any

  stages {
    stage('Source'){
      steps{
        git branch: 'main', url:'https://github.com/maraks-gradiious/maven-test1'

      }
    }
    stage('Build'){
      steps{
        withSonarQubeEnv('sonar'){
          // Optionally use a Maven environment you've configured already
          withMaven{
            sh "mvn clean verify sonar:sonar"
          }
        }
      }
    }
  }
}
```