

# **Jenkins and Sonarqube Integration**

## **Introduction:**

This project integrated Jenkins and SonarQube to enhance our CI/CD pipeline and ensure top-notch code quality!



### **Jenkins:**

Our trusted automation server, orchestrating builds, tests, and deployments seamlessly.



### **SonarQube:**

Our go-to tool for continuous code quality inspection, helping us identify and fix bugs, code smells, and security vulnerabilities early in the development process.

## **Project Highlights:**

Automated Quality Gates: Every commit is automatically analyzed, ensuring it meets our quality standards before merging.

Real-time Feedback: Developers receive instant insights on code quality, enabling quicker fixes and improvements.

Enhanced Collaboration: The integration fosters better communication and collaboration among team members, with clear visibility into code quality metrics.

## **Integration Steps:**

Installed SonarQube Scanner Plugin in Jenkins.

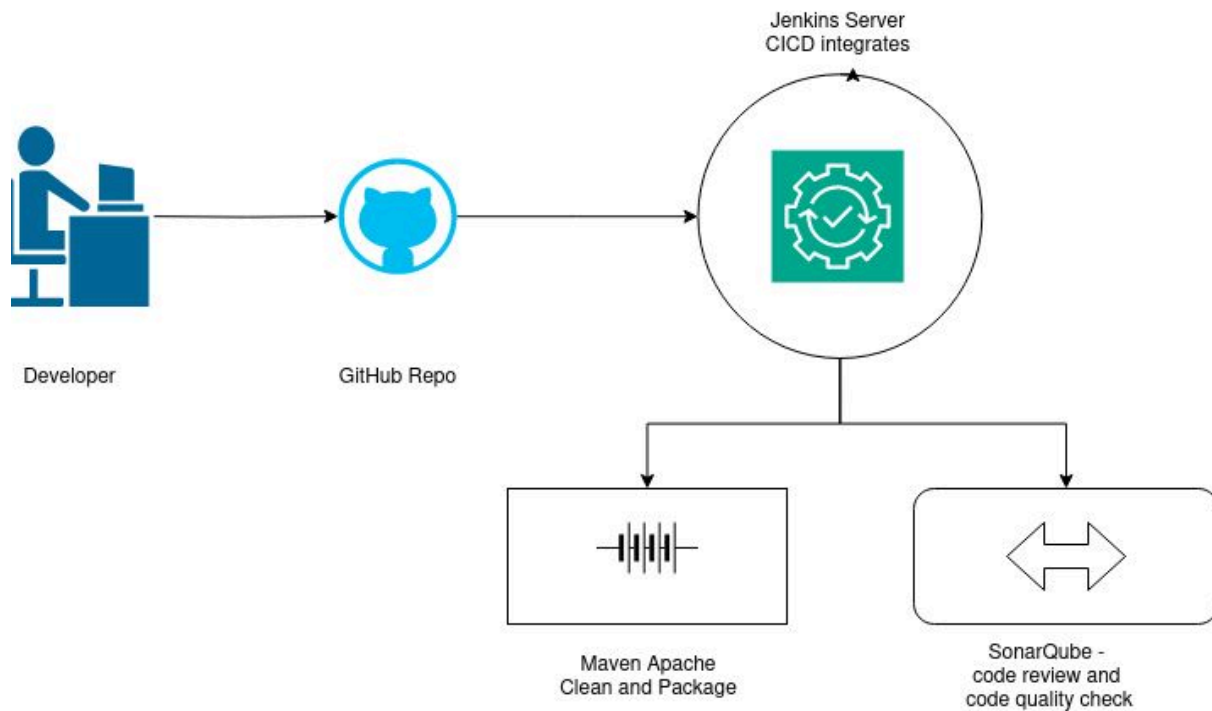
Generated Authentication Token in SonarQube for secure communication.

Configured SonarQube Server details in Jenkins.

Updated Jenkins Pipeline to include SonarQube analysis steps.

This integration has significantly improved our development workflow, ensuring we deliver high-quality, secure code faster and more efficiently.

## Diagram :



## Jenkins installation using script file :

```
Unknown > Full Stack project > $ 1.sh
1  sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
2  | https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
3  echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
4  | https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
5  /etc/apt/sources.list.d/jenkins.list > /dev/null
6  sudo apt-get update
7  sudo apt-get install jenkins -y
```

## SonarQube installation using Docker:

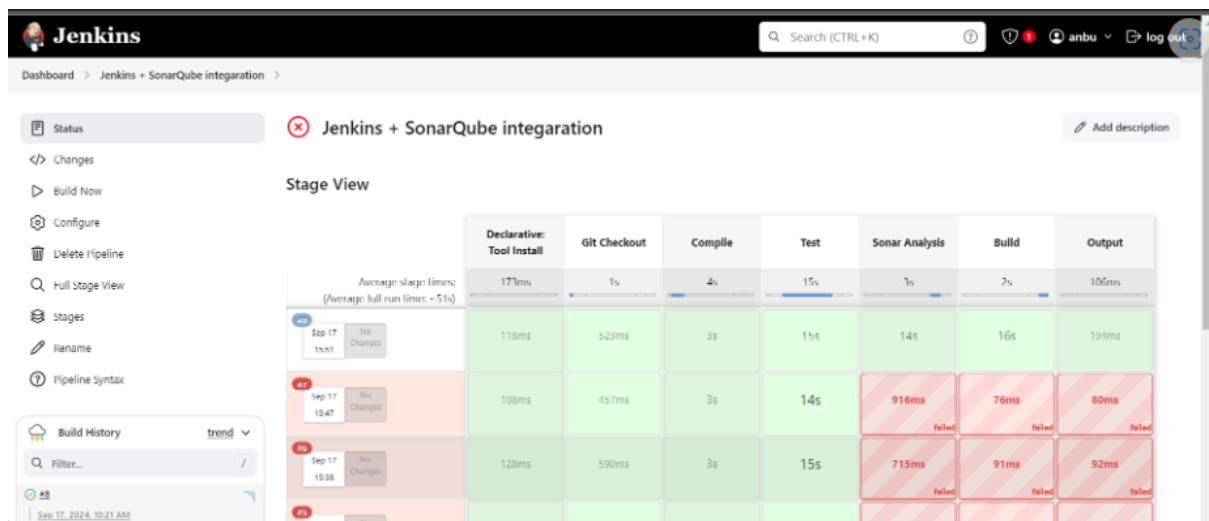
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-133:~$ sudo docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
e2765fa5e40d: Pull complete
c0be4e952492: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:b65d89bb7e3d61114df748614fdc02c3dce27b9d8af5dc9d44dcca8a3176be0
Status: Downloaded newer image for sonarqube:lts-community
9d0b8d50c02b28b9aca73c142cab43e3c2ab0e8524e122bc1fb1a1e94cd8885f
ubuntu@ip-172-31-82-133:~$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS
0d0b8d50c02b   sonarqube:lts-community  "/opt/sonarqube/dock..." 8 seconds ago  Up 5 seconds  0.0.0.0:9000->9000/tcp, :::9000->9000/tcp
sonar
```

## Creating Pipeline using Groovy sandbox :

Script ?

```
1 pipeline {
2   agent any
3
4   tools {
5     maven 'maven3'
6   }
7
8   environment {
9     SCANNER_HOME= tool 'sonar-scanner'
10  }
11
12  stages {
13    stage('Git Checkout') {
14      steps {
15        git branch: 'main', credentialsId: 'git', url: 'https://github.com/Anbudurai/Fullstack.git'
16      }
17    }
18    stage('Compile') {
19      steps {
20        sh 'mvn compile'
21      }
22    }
23    stage('Test') {
24      steps {
25        sh 'mvn test'
26      }
27    }
28    stage('Sonar Analysis') {
29      steps {
30        withSonarQubeEnv('sonar') {
31          sh '''$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Blogging-app -Dsonar.projectKey=Blogging-app \
32            -Dsonar.java.binaries=target '''
33        }
34      }
35    }
36    stage('Build') {
37      steps {
38        sh 'mvn package'
39      }
40    }
41    stage('Output') {
42      steps {
43        echo 'Jenkins and Sonarqube Intergration has been done'
44      }
45    }
46  }
47 }
48
```

## Jenkins File successfully execution :



## SonarQube server:

The screenshot displays the SonarQube web interface, specifically the 'Issues' tab. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is located on the right. The main content area shows a list of issues, each with a description, severity, and status. The sidebar on the left contains filters for Type, Severity, Scope, Resolution, and Status.

**Filters:**

- Type:** BUG (54), Vulnerability (0), Code Smell (24)
- Severity:** Blocker (0), Critical (0), Major (31), Minor (23), Info (0)
- Scope:** Main code (54), Test code (0)
- Resolution:** Unresolved (54), False Positive (0), Fixed (0), Removed (0), Won't Fix (0)
- Status:** Open (54), Reopened (0), Closed (0), Confirmed (0), Resolved (0)

**Issues List:**

Issue ID	Issue Description	Severity	Status	Effort	Created
1	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	6 days ago
2	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	6 days ago
3	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	6 days ago
4	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	6 days ago
5	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	3 minutes ago
6	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	3 minutes ago
7	Add "lang" and/or "xml:lang" attributes to this "html" element	Major	Open	2min effort	3 minutes ago

Thank you ! Happy Learning !