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.

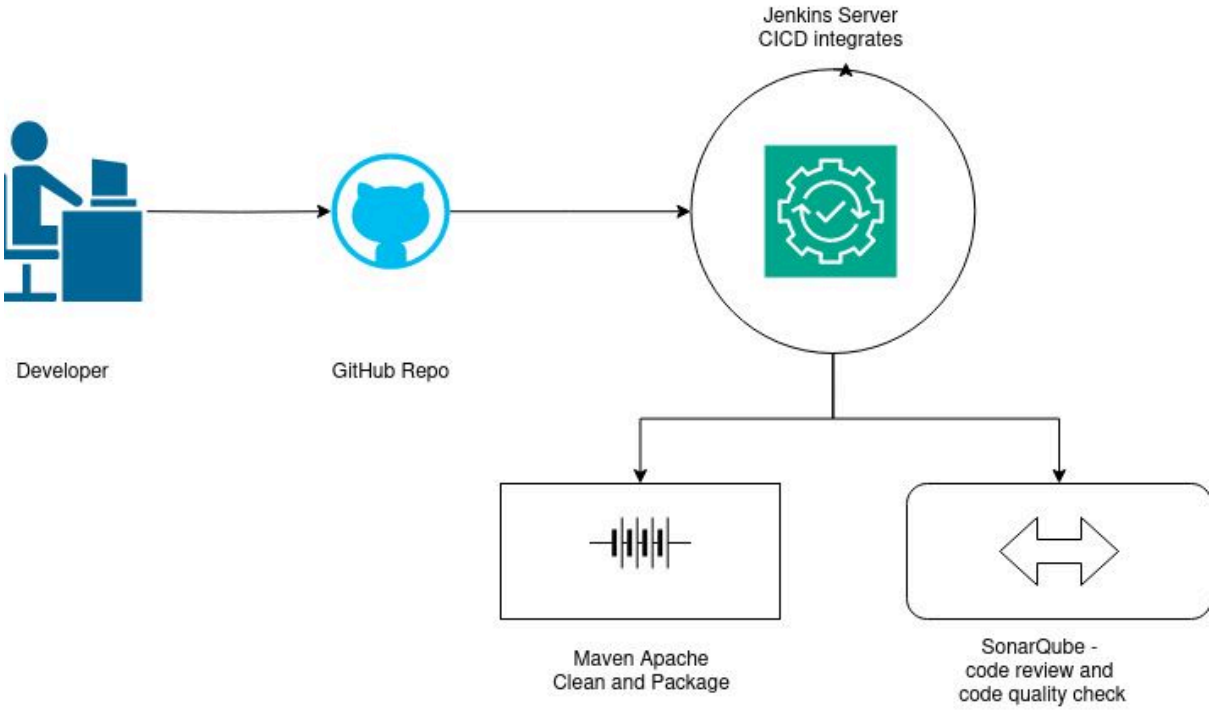
Generated Authentication Token in SonarQube for secure communication.

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

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
```

SonarQube installation using Docker:

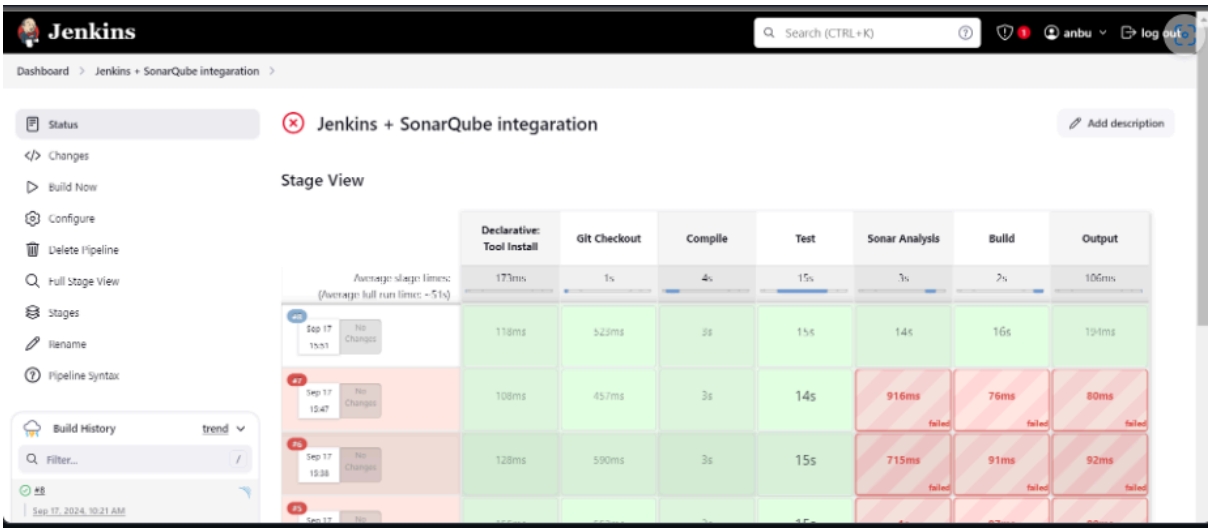
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-133:~$ sudo docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
e2765fa5e40d: Pull complete
c0be4e952492: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:b65d89bb7e3d61114df748614fdcf02c3dce27b9d8af5dc9d44dcca8a3176be0
Status: Downloaded newer image for sonarqube:lts-community
0d0b8d50c02b28b9aca73c142cab43e3c2ab0e8524e122bc1fb1a1e94cd8885f
ubuntu@ip-172-31-82-133:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS         PORTS
      NAMES
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 :

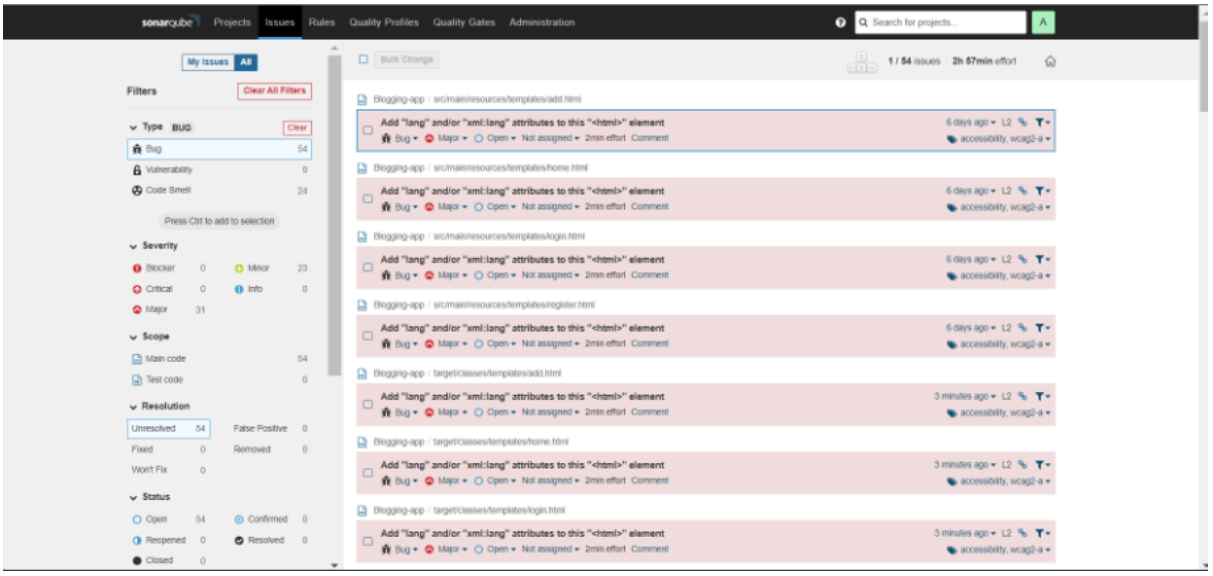
Script

```
pipeline {
    agent any

    tools {
        maven 'maven3'
    }

    environment {
        SCANNER_HOME= tool 'sonar-scanner'
    }

    stages {
        stage('Git Checkout') {
            steps {
                git branch: 'main', credentialsId: 'git', url: 'https://github.com/Anbudurai/Fullstack.git'
            }
        }
        stage('Compile') {
            steps {
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test'
            }
        }
        stage('Sonar Analysis') {
            steps {
                withSonarQubeEnv('sonar') {
                    sh '''$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Blogging-app -Dsonar.projectKey=Blogging-app \
                        -Dsonar.java.binaries=target '''
                }
            }
        }
        stage('Build') {
            steps {
                sh 'mvn package'
            }
        }
        stage('Output') {
            steps {
                echo 'Jenkins and Sonarqube intergration has been done'
            }
        }
    }
}

```

Jenkins File successfully execution :



SonarQube server:



Thank you ! Happy Learning !