SonarQube Integration Exercise – Hands-On Guide with Real-Time Example

# 🎯 Objective

This document walks through integrating SonarQube into a CI/CD pipeline to automate static code analysis, enforce quality gates, and detect vulnerabilities before deployment.

# 🔧 Required Tools

- Jenkins (or GitHub Actions)

- SonarQube Server (self-hosted or Docker)

- SonarQube Scanner CLI

- Java/Maven or Node.js Project

# 🛠️ Step-by-Step Integration

## 1. Install SonarQube Locally (Optional)

You can use Docker to run SonarQube locally:

docker run -d --name sonarqube -p 9000:9000 sonarqube

## 2. Configure SonarQube Project

- Log in to SonarQube (http://localhost:9000).

- Create a new project and generate a project token.

## 3. Install Sonar Scanner CLI

Download and configure SonarScanner CLI from: https://docs.sonarsource.com/

## 4. Update Your Project Configuration

Add a `sonar-project.properties` file at the root of your repo:

sonar.projectKey=secure-app  
sonar.sources=.  
sonar.host.url=http://localhost:9000  
sonar.login=YOUR\_TOKEN

## 5. Integrate with Jenkins Pipeline

Example `Jenkinsfile` stage for SonarQube:

stage('Static Code Analysis') {  
 steps {  
 sh 'sonar-scanner'  
 }  
}

- Add SonarQube server config in Jenkins: \*\*Manage Jenkins > Configure System > SonarQube Servers\*\*

- Use the correct token in Jenkins global environment or credentials store.

# ✅ Real-Time Example

Scenario:

A development team uses Jenkins to deploy a Java app. SonarQube is integrated into the pipeline.

When code is pushed:

- Jenkins checks out the code

- SonarScanner analyzes for bugs, vulnerabilities, and code smells

- Quality gates stop the pipeline if critical issues are found

# 📌 Benefits

- Automated detection of issues and technical debt

- Visibility of code health through dashboards

- Reduced risk of deploying vulnerable or buggy code

- Enforced development standards across teams

# 🔚 Conclusion

SonarQube integration enables teams to catch issues early in the CI/CD lifecycle. With a simple setup and strong visual reports, it adds immense value to DevOps workflows.