Hands-on: SonarQube – Run Scan → View Dashboard → Fix Issues → Re-scan

🎯 Objective:

Learn how to run a SonarQube code quality scan in Jenkins, analyze issues, fix them, and validate improvements through a re-scan.

✅ Prerequisites:

- **SonarQube server** running (local or remote)
- **SonarQube Scanner** installed on Jenkins
- **SonarQube plugin** configured in Jenkins
- SonarQube **project token** or **admin access**
- Jenkins agent with the build environment (e.g., Python, Java)
- Sample project with source code and `sonar-project.properties` file

🧭 Step-by-Step Instructions

🔹 Step 1: Configure SonarQube in Jenkins

1. Go to **Jenkins Dashboard** → **Manage Jenkins** → **Configure System**
2. Find the **SonarQube servers** section
3. Add:
   - **Name**: `SonarQube`
   - **Server URL**: `http://localhost:9000` or your SonarQube instance
   - **Authentication Token**: Add a credential with your SonarQube token
4. Click **Save**

🔹 Step 2: Add `sonar-project.properties` to Your Repo

In the root of your project directory:

```
sonar.projectKey=sample-app
sonar.projectName=Sample Application
sonar.projectVersion=1.0
sonar.sources=.
sonar.language=py
sonar.sourceEncoding=UTF-8

```

> Update `sonar.language` to `java`, `js`, etc., based on your tech stack.

🔹 Step 3: Jenkinsfile – Add SonarQube Scan Stage

For **Declarative Pipeline**, example:

```
pipeline {
    agent any

    tools {
        // For Java projects
        maven 'Maven 3.8.1'
    }

    environment {
        SONAR_SCANNER_HOME = tool 'SonarQube Scanner'
    }

    stages {
        stage('Checkout') {
            steps {
                git 'https://bitbucket.yourcompany.com/scm/devops/sample-app.git'
            }
        }

        stage('SonarQube Scan') {
            steps {
                withSonarQubeEnv('SonarQube') {
                    sh "${SONAR_SCANNER_HOME}/bin/sonar-scanner"
                }
            }
        }
    }
}

```

🔹 Step 4: Run the Pipeline Job

1. From Jenkins job, click **Build Now**
2. After scan is complete, go to your **SonarQube dashboard**
   - Visit: `http://<sonarqube-host>:9000`
   - Click on your project (e.g., *Sample Application*)
   - Explore tabs: **Issues**, **Code Smells**, **Duplications**, **Coverage**

🔹 Step 5: Fix Issues

1. Pick a few **code smells or bugs**
2. Fix them in your source code (e.g., variable naming, unused code, formatting)
3. Commit and push your changes

🔹 Step 6: Re-run Jenkins Job

- Trigger another build in Jenkins.
- Re-scan results will appear in SonarQube.
- Verify that previously flagged issues are resolved.

🧪 Outcome

✅ Code analyzed by SonarQube
✅ Issues visualized on Sonar dashboard
✅ Code quality improved via re-scan
✅ Integration of Jenkins ↔ SonarQube pipeline flow