

Jenkins-SonarQube Integration for Maven application

Prerequisite :

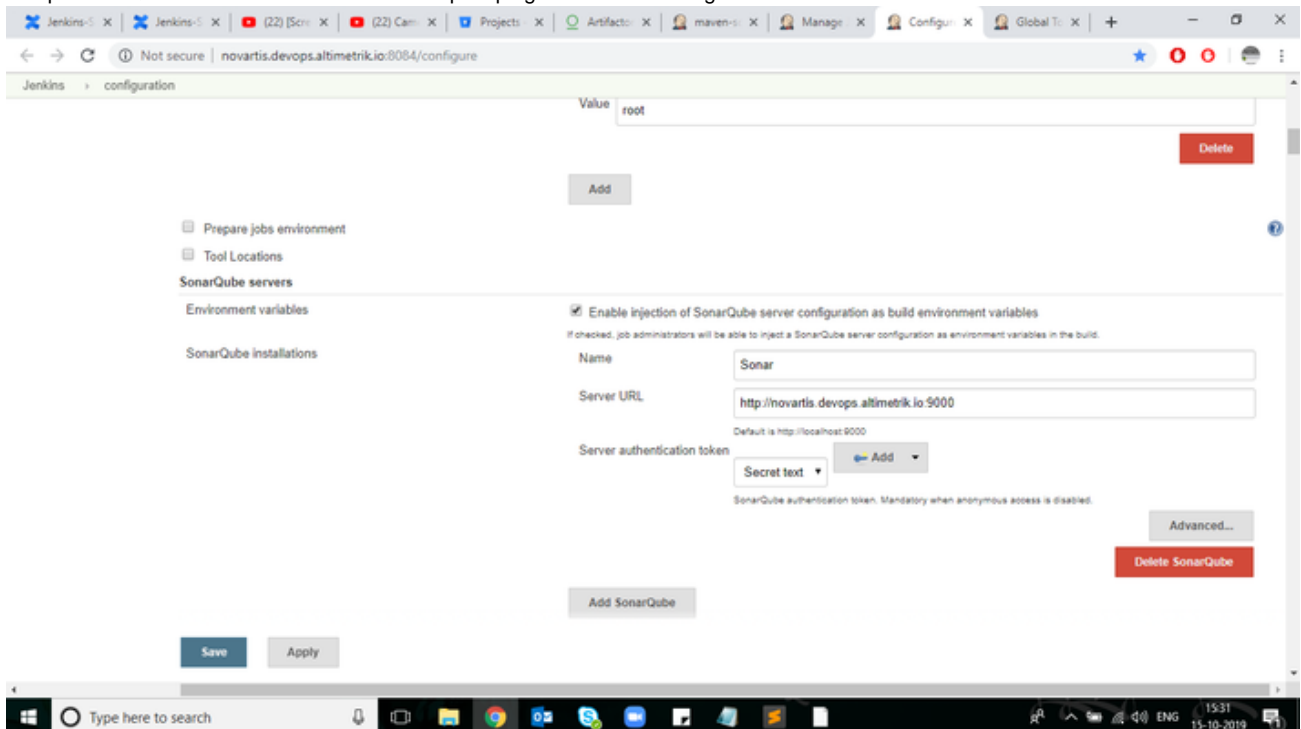
1. Jenkins is setup with Bitbucket repo.
2. SonarQube is running and you have Sonar Scanner setup in system.

Sonarqube configurations:-

1. go to sonarqube server and create a project.
2. go to sonarqube dashboard and create quality profiles and create quality gates for java application and assign the projects. reference this document <https://altimetrik.atlassian.net/wiki/spaces/DC/pages/209453145/How+to++Configuration+for+code+check+gates>.
3. now you can able to create the sonarqube project.

Jenkins configurations:-

1. Setup **SonarQube** with **Jenkins**- install sonarqube plugin and made configuration as below



The screenshot shows the Jenkins configuration page for SonarQube servers. The browser address bar indicates the URL is `novartis.devops.altimetrik.io:8084/configure`. The page title is "Jenkins > configuration".

At the top, there is a "Value" field set to "root" with a "Delete" button. Below it is an "Add" button.

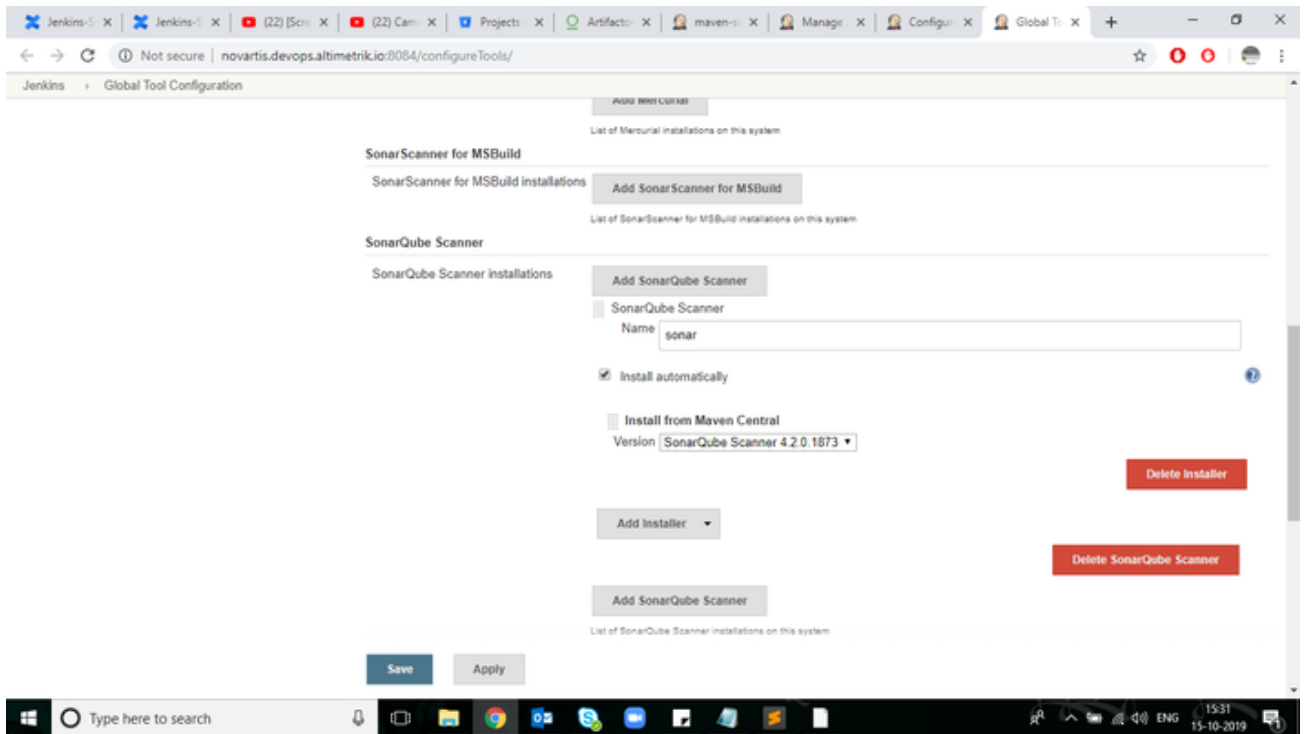
The main configuration area is titled "SonarQube servers". It includes a section for "Environment variables" with a checkbox labeled "Enable injection of SonarQube server configuration as build environment variables". Below this, a note states: "If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build."

Under "SonarQube installations", there are fields for "Name" (set to "Sonar"), "Server URL" (set to `http://novartis.devops.altimetrik.io:9000`), and "Server authentication token" (set to "Secret text"). A default value for the token is shown as `Default is http://localhost:9000`. There is an "Add" button next to the token field. A note below the token field states: "SonarQube authentication token. Mandatory when anonymous access is disabled."

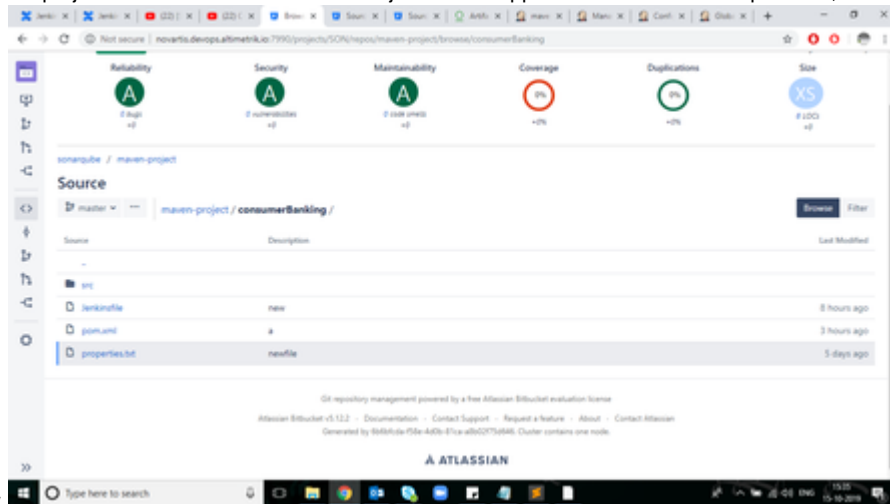
At the bottom of the configuration area, there is an "Advanced..." button and a "Delete SonarQube" button. Below the configuration area, there is an "Add SonarQube" button.

At the very bottom of the page, there are "Save" and "Apply" buttons.

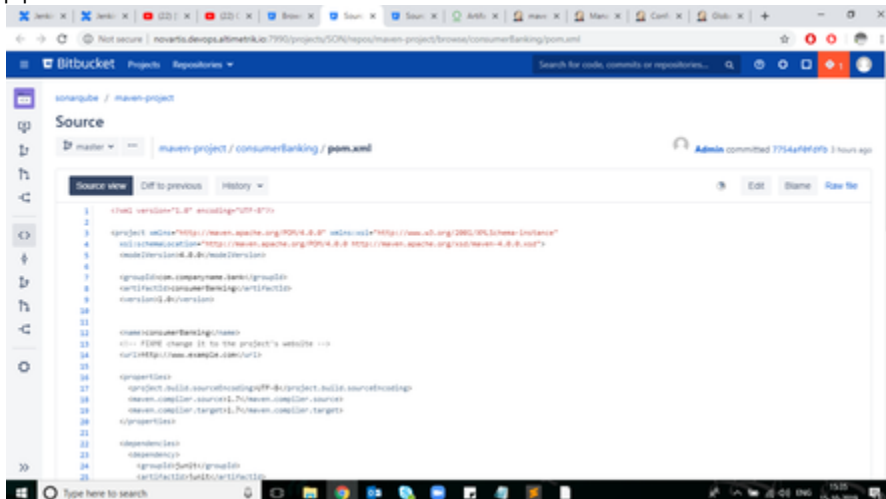
2.



- do the following configuration, sonar installation and sonar server instance configuration as above.
- go to Bitbucket and create project and create a source code for java based application which includes pom.xml, rc/main and src/test,



properties.xml as below.



GitLab Jenkins (22) [Sc] (22) Cn Browse Source Source Artifact ma X Jenkins Manag Config Global Slack + -

Not secure novartis.devops.altimetrik.io:8084/job/maven-sonar-jenkins/configure

Jenkins > maven-sonar-jenkins >

General Datadog Tagging Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Advanced...

☐ Inject environment variables

Properties File Path sonar-branch

Properties Content

☐ Invoke top-level Maven targets

Maven Version maven

Goals install sonar:sonar -Dsonar.branch.target=sonar-branch

POM consumerBanking/pom.xml

If your workspace has the top-level pom.xml in somewhere other than the 1st module's root directory, specify the path (relative to the module root) here, such as parent/pom.xml.
If left empty, defaults to pom.xml

Save Apply

sonar.sources=/var/lib/jenkins/workspace/

GitLab Jenkins (22) [Sc] (22) Cn Browse Source Source Artifact ma X Jenkins Manag Config Global Slack + -

Not secure novartis.devops.altimetrik.io:8084/job/maven-sonar-jenkins/configure

Jenkins > maven-sonar-jenkins >

General Datadog Tagging Source Code Management Build Triggers Build Environment **Build** Post-build Actions

☐ Invoke top-level Maven targets

Maven Version maven

Goals install sonar:sonar -Dsonar.branch.target=sonar-branch

POM consumerBanking/pom.xml

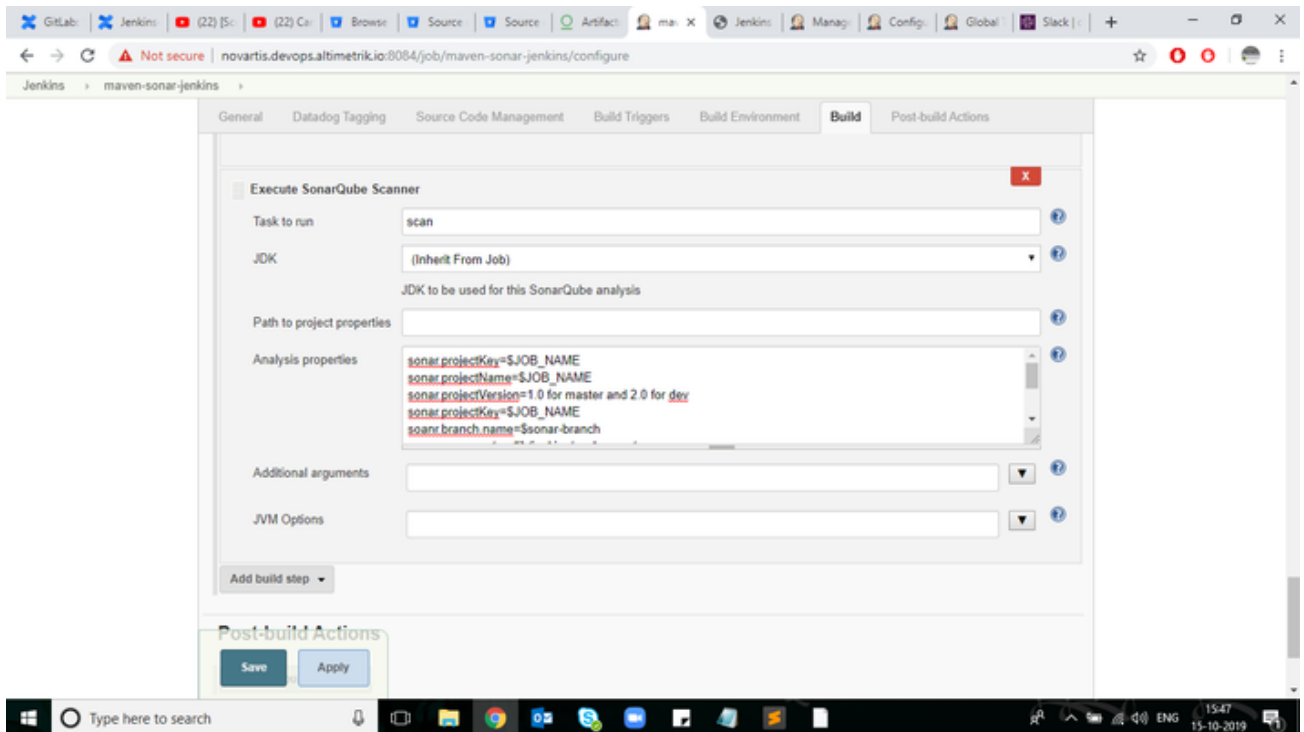
If your workspace has the top-level pom.xml in somewhere other than the 1st module's root directory, specify the path (relative to the module root) here, such as parent/pom.xml.
If left empty, defaults to pom.xml

Properties

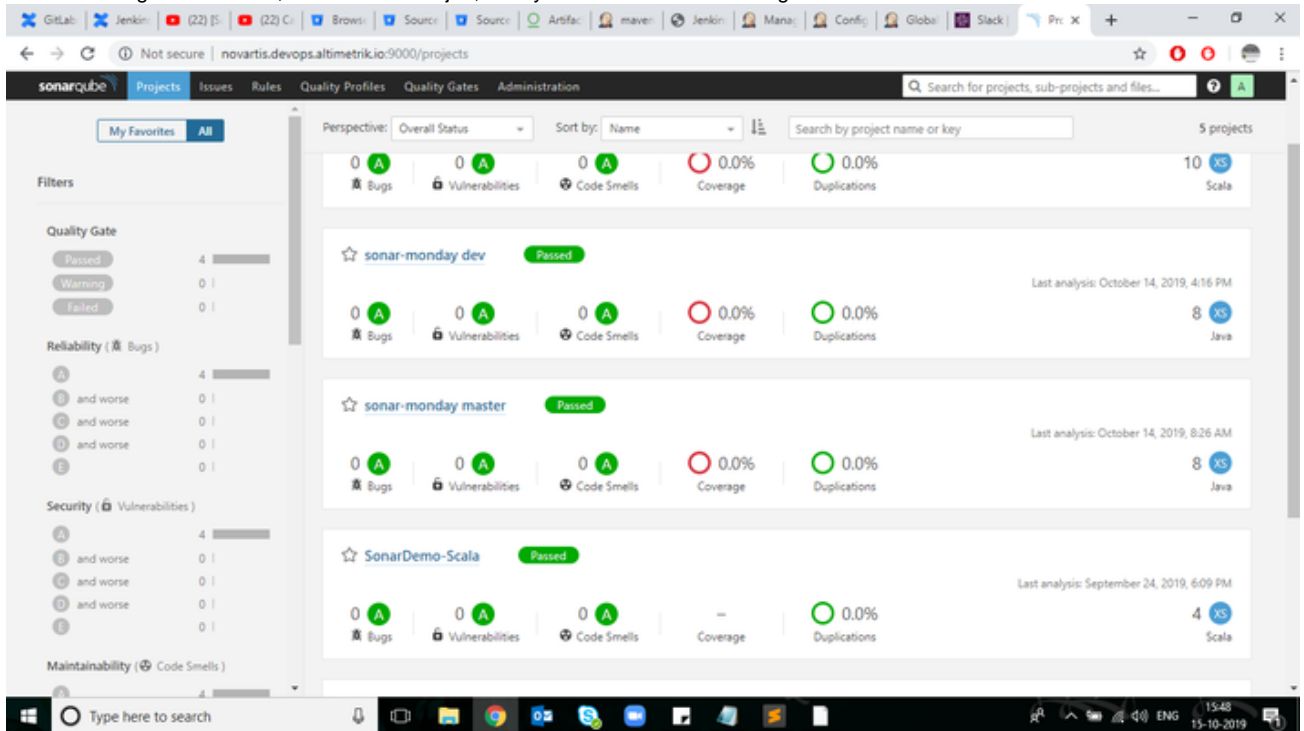
```
sonar.sources=/var/lib/jenkins/workspace/  
sonar.java.binaries=/target/classes  
sonar.branch.name=master/for devbranch I change here /devBranch  
sonar.sources=consumerBanking/src/main  
sonar.tests=consumerBanking/src/test
```

JVM Options

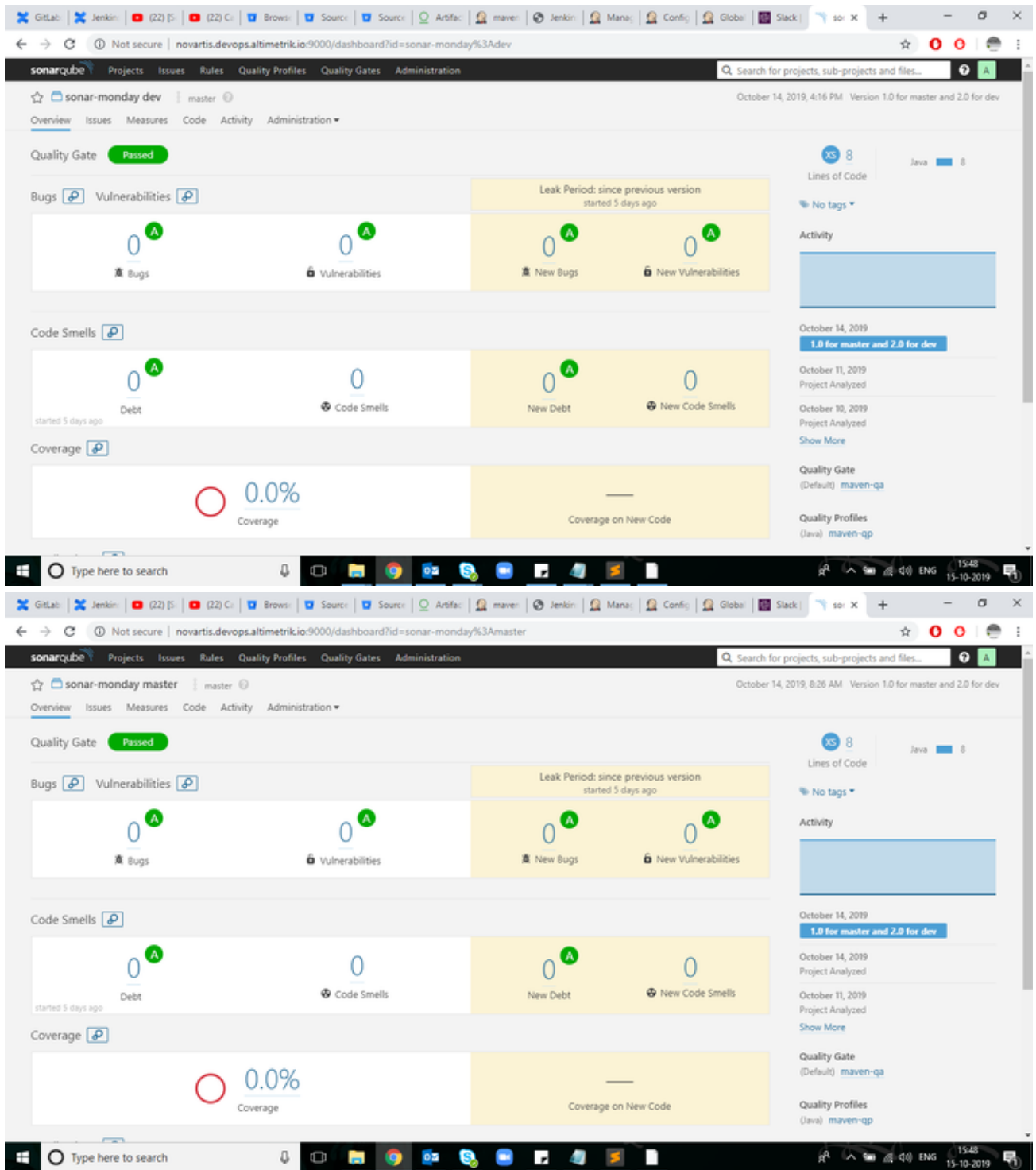
Save Apply



- once the configurations done, build the maven job, then you can observe the changes in sonar instance as below.



-



6. we have to insert the git branch env var, then we can see the same project name with different branches in the sonar.

thats all done !!!

