Sonatype CLM for Hudson and Jenkins

Contents

List of Figures

Chapter 1

Introduction

Eclipse Hudson and Jenkins are powerful and widely used open source continuous integration servers providing development teams with a reliable way to monitor changes in source control and trigger a variety of builds. They excel at integrating with almost every tool you can think of.

Historically the Hudson project and community split into two groups, with Jenkins as well as Hudson emerging as sibling products with a different focus going forward while sharing a common API for plugins. In general, with regard to the Sonatype CLM for CI functionality, the interaction will be near identical, with only a few differences, which are inherent to the CI, and not Sonatype CLM.

Sonatype CLM for Hudson and Jenkins evaluates the project workspace after a build for all supported component types, creates a summary file about all the components found and submits that to the Sonatype CLM service. The service uses that data to produce the analysis with the security and license information and send it back to the CI server. It will then use these results to render the analysis reports.

The file types supported for analysis are in tar/zip like format with the extensions tar, tar.bz2, tb2, tbz, tar.gz, tgz and zip or in Java archive formats of the type jar, ear, war, hpi, wsr, har, sar, rar, mar and nbm.

Chapter 2

Installation

Sonatype CLM for Hudson and Jenkins is distributed as a Hudson plugin package (`.hpi` file) and is compatible with Jenkins and Hudson.

In order to install the plugin you have to log into Jenkins or Hudson as administrator and then select to *Manage Jenkins/Manage Hudson* to get to the global configuration menu displayed in Figure 2.1 in the Jenkins look. The Hudson look will be similar in content, yet different in colors and styling.

Figure 2.1: Jenkins Global Configuration Menu

From the displayed configuration menu, select *Manage Plugins* and in the plugin management section, choose the *Advanced* tab.

The advanced plugin management allows you to upload a plugin distribution file (`.hpi`) in the section entitled *Manual Plugin Installation* on Hudson and *Upload Plugin* on Jenkins. Click on *Choose File* and select Sonatype CLM for Hudson and Jenkins hpi file named `sonatype-clm-ci-x.y.z.hpi` with `x.y.z` representing a version number like `2.11.2` in the file selection dialog. Then press the *Upload* button. Once the plugin has been uploaded to the server, you need to restart your continuous integration server.

Chapter 3

Global Configuration

After a successful installation of Sonatype CLM for Hudson and Jenkins, a new option will be available in the Jenkins/Hudson management area, *Configure Sonatype CLM for CI*. Follow these instructions to configure Jenkins or Hudson to connect to your Sonatype CLM Server.

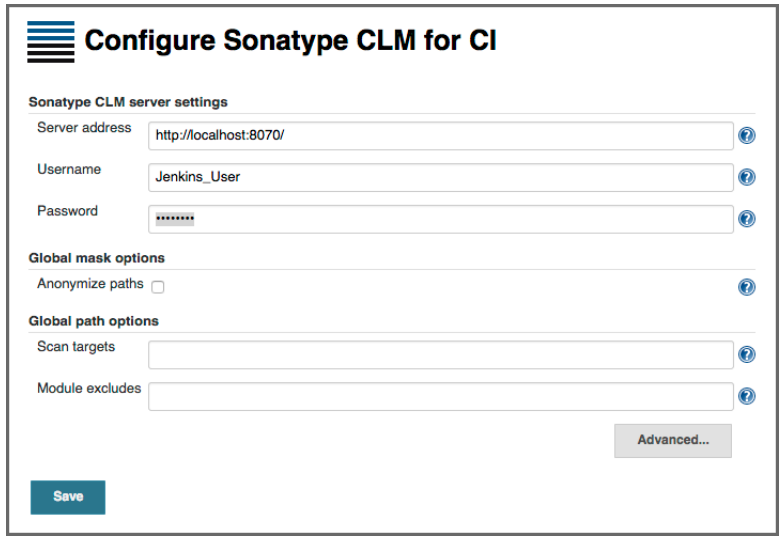

Figure 3.1: Global Configuration of Sonatype CLM for CI in Jenkins

Sonatype CLM server settings (required)

Server address

This is the address for the Sonatype CLM server as it can be reached from the Jenkins/Hudson server. By default, the Sonatype CLM Server address is `http://localhost:8070`.

If your Sonatype CLM server is behind a proxy server for serving HTTPS or other reasons, you have to use the public URL as it is reachable from the continuous integration server. Only the master Jenkins/Hudson server connects to the CLM server and you therefore only need to ensure connectivity in terms of open firewall ports and proxy server settings between the master CI server and the CLM server.

Username

This is the username you wish to use to connect to the CLM Server. Since these settings will be used across all projects for your Jenkins/Hudson installation, we suggest creating a single account on the Sonatype CLM Server for Jenkins/Hudson, and then associating that account with the Application Evaluator role for the Organizations or Applications you will be linking to Sonatype CLM for Hudson/Jenkins.

Password

The password for the username entered above.

Tip

Username and password can also be configured per job.

Global mask options

Anonymize paths

Enabling this feature will anonymize all paths before data is sent to the Sonatype CLM server. Ultimately, this prevents the CLM report from reporting the locations/occurrences of components.

Global path options

Scan targets

The scan targets setting allows you to control which files should be examined. The configuration uses an Apache Ant styled pattern, is relative to each project's workspace root directory, and has a useful default setting that includes all `jar`, `war`, `ear`, `zip` and `tar.gz` files. The default value is therefore

```
**/*.jar, **/*.war, **/*.ear, **/*.zip, **/*.tar.gz
```

Note

This default only applies if, and only if, neither global nor job config specify scan targets. Adding to this, if you are using a private Maven repository, our default pattern will include your entire Maven repo. This could greatly increase the time necessary for your evaluation, as well as skew evaluation results. To avoid this, consider using a more specialized pattern like **/target/*.jar.

Module excludes

If you are using Sonatype CLM for Maven, you may have noticed the creation of module information files. The process for excluding modules is documented in the Excluding Module Information Files in Continuous Integration Tools section of the Sonatype CLM for Maven Guide.

Advanced options

A number of additional parameters can be supplied to the plugin using this input field. Typically these parameters will be determined by Sonatype support.

Chapter 4

Job Configuration

After a completed installation (see Chapter 2) and global configuration (see Chapter 3) of Sonatype CLM for CI, you are ready to configure an invocation as part of a specific job.

Depending on your job type it will be available as pre and/or post-build step as well as a invocation as a main build step. The typical invocation would be as main build step, after the package that should be examined has been created. An example configuration from Jenkins is displayed in Figure 4.1. Alternatively a post-build step for example as displayed in Figure 4.2 can be used as well. A pre-build step or a main build step executed before your main build invocation step could be used to examine components existing in the workspace or being placed into the workspace by an earlier build step.

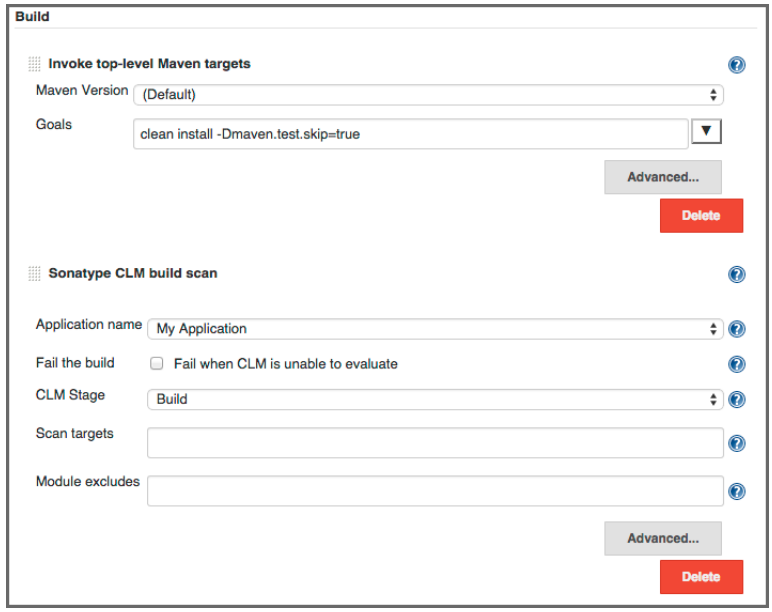

Figure 4.1: Sonatype CLM Build Scan Configuration for a Build Step

The configuration options for Sonatype CLM for CI invocations mimic the parameters from the global configuration described in Chapter 3 and are appended to the global parameters. The configuration parameters are:

Optional Job Specific Authentication

While username and password can be configured globally, in some cases you may want a certain job to be associated with a user who has permissions to specific Organization and/or Applications. Job Specific Authentication allows you to configure a user for this job and use the associated permissions to select the Application for the evaluation.

Username

the Sonatype CLM Server username you wish to use for this job.

Password

the password for the username above.

Sonatype CLM Job Configuration

Application name

The drop down for application name should be populated with the name of all applications configured in your Sonatype CLM server and allows you to select the desired application

scanning configuration. The policies associated to the application will be used for the analysis of this build job output.

Fail the build

Check this option if you want to fail the build when a CLM evaluation can't be performed. Once checked, if for any reason the evaluation is not generated, the build will be failed.

An example of this might be if the CLM server is inaccessible. In this scenario, the build would fail. In the same example, but where the *Fail the build* option is left unchecked, the build would be marked unstable.

CLM Stage

This corresponds to the stage you wish the policy evaluation of the application/project to be run against. Additionally, this will correspond to the stage location when viewing report information via the CLM Server (e.g. if you chose the Build stage, summary and dashboard violation results will be displayed accordingly).

Note

Depending on how your policies are configured, this may impact warning and fail actions.

Scan targets

The scan targets setting allows you to control which files should be examined with an Apache Ant styled pattern. The pattern is relative to the project workspace root directory and inherits the global configuration.

Module excludes

You can exclude modules from being scanned with module information files configured in this setting. The default value is inherited from the global configuration.

Advanced options

A number of additional parameters can be supplied to the plugin using this input field. Typically these parameters will be recommended to you by the Sonatype support team.

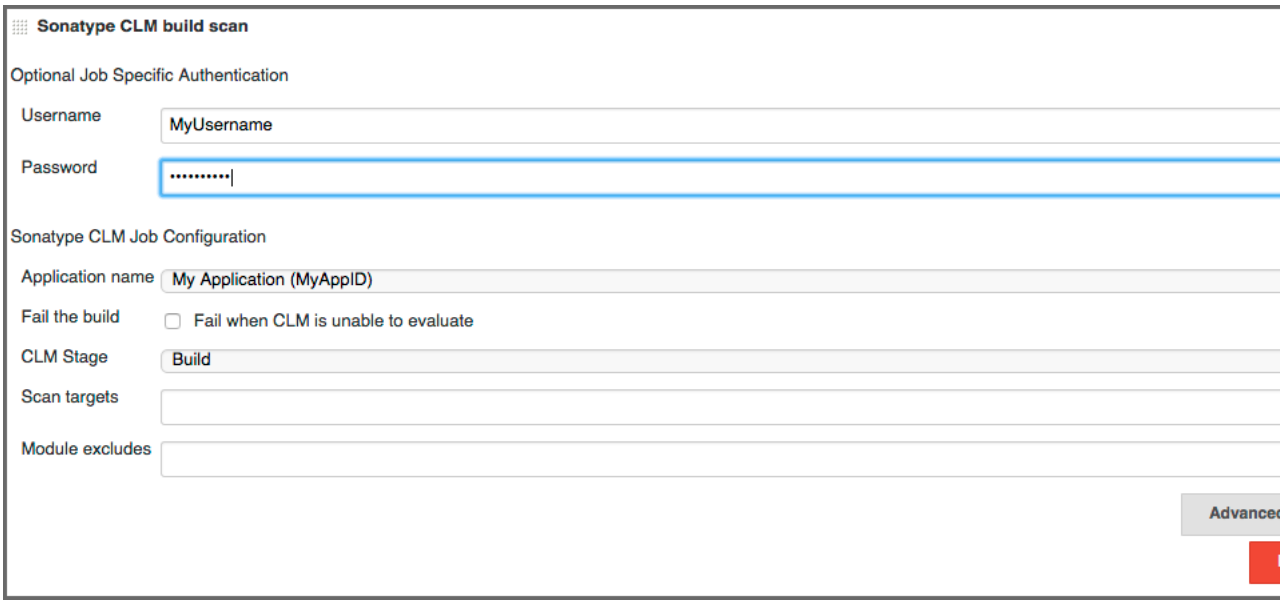

Figure 4.2: Post-build Action Configuration as Example for a Sonatype CLM for CI Configuration

Chapter 5

Inspecting Results

Once a specific build has successfully completed, Sonatype CLM for CI provides a link to the application composition report in the job list in the *Policy Violations* column as well as the project specific overview page. Clicking on the link *Application Composition Report*, will direct you to the display of the report within the Sonatype CLM Server. The three boxes (red, orange, and yellow) located below the link, give you counts for policy violations, and are based on the associated severities (high, medium, and low).

In addition to the link to the report, the left-hand menu for the job includes *Application Management*. Clicking on the link will take you directly to the specific application on the Sonatype CLM Server. In Figure 5.1 you can see both the link to the report, and the link to Application management.

Note
Accessing this information may require a login. Also, if you are using a version of Sonatype CLM for Hudson and Jenkins prior to version 2.11, and Sonatype CLM Server 1.7, a message will display indicating your report has been moved. Following this link will take you to the report on the Sonatype CLM Server.

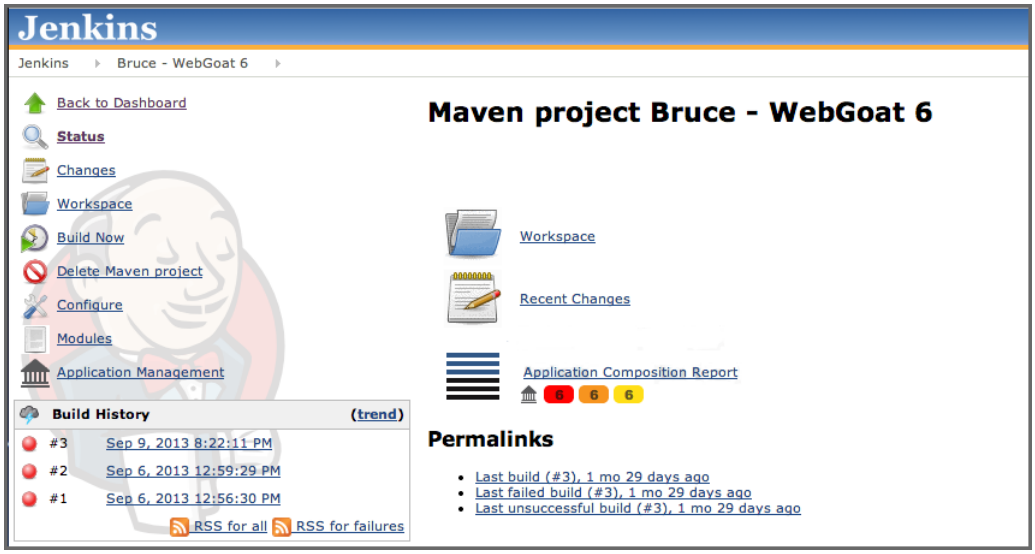

Figure 5.1: Job Overview Page with Links to the Application Composition Report and Application Management

If you are looking for previous report results, simply navigate to a specific build in the *Build History*. If you have previously scanned the application during that specific build, you will see a new item in the left menu, *Application Composition Report*. As with the report link above, you will be taken to the Sonatype CLM Server to review the results. An example is show in Figure 5.2 below.

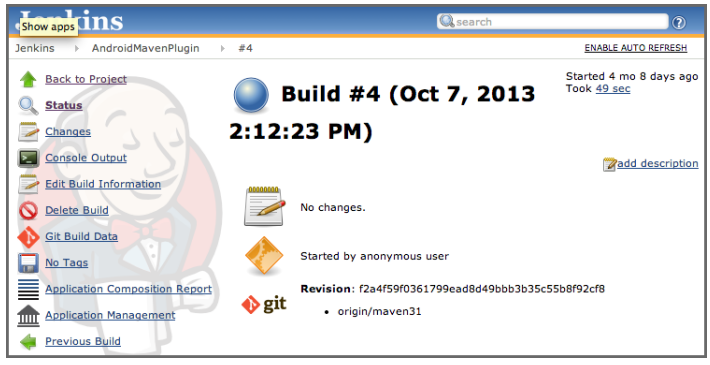

Figure 5.2: Left Menu with Link to the Application Composition Report

Chapter 6

Conclusion

You should now have Sonatype CLM for Hudson and Jenkins up and running. Just in case you missed something, here are some highlights if what was covered:

- Requirements, Installation and Configuration
- Job Configuration
- Inspecting Results

If you haven't already, you will want to take a look at the Policy Management guide. Plus don't forget if your organization uses multiple CI systems, many of these can be integrated using the Sonatype CLM for CLI or Maven tools.