Analyzing Source Code

Table of Contents

- Scope of Analysis: Types of Files and Data
  - Unrecognized files
- During Analysis
- Running Analysis
  - SonarCloud User?
- F.A.Q.
- For More Information

Once the SonarQube platform has been installed, you're ready to install an analyzer and begin creating projects. A project is created in the platform automatically on its first analysis. However, if you need to set some configuration on your project before its first analysis, you have the option of provisioning it.

Scope of Analysis: Types of Files and Data

SonarQube can perform analysis on 20+ different languages. The outcome of this analysis will be quality measures and issues (instances where coding rules were broken). However, what gets analyzed will vary depending on the language:

- On all languages, "blame" data will automatically be imported from supported SCM providers. Git and SVN are supported automatically. Other providers require additional plugins.
- On all languages, a static analysis of source code is performed (Java files, COBOL programs, etc.)
- A static analysis of compiled code can be performed for certain languages (*.class* files in Java, *.dll* files in C#, etc.)
- A dynamic analysis of code can be performed on certain languages.

Unrecognized files

By default, only files that are recognized by a language plugin are loaded into the project during analysis. For example if your SonarQube instance has the Java and JavaScript plugins on board, all *.java* and *.js* files will be loaded, but *.xml* files will be ignored.

During Analysis

During analysis, data is requested from the server, the files provided to the analysis are analyzed, and the resulting data is sent back to the server at the end in the form of a report, which is then analyzed asynchronously server-side.

Analysis reports are queued, and processed sequentially, so it is quite possible that for a brief period after your analysis log shows completion, the updated values are not visible in your SonarQube project. However, you will be able to tell what's going on because an icon will be added next to the project name. Mouse over it for more detail (and links if you're logged in with the proper permissions.)

The icon goes away once processing is complete, but if analysis report processing fails for some reason, the icon will change:

For more detail on analysis report processing, see Background Tasks.

Running Analysis

First, you should install the plugin(s) for the language(s) of the project to be analyzed, either by a direct download or through the update center.

Then, you need to choose an analysis method. The following are available:

- SonarQube Scanner for MSBuild: Launch analysis of .Net projects
- SonarQube Scanner for Maven: Launch analysis from Maven with minimal configuration
- SonarQube Scanner for Gradle: Launch Gradle analysis
- SonarQube Scanner for Ant: Launch analysis from Ant
- SonarQube Scanner For Jenkins: Launch analysis from Jenkins
- SonarQube Scanner: Launch analysis from the command line when none of the other analyzers is appropriate

Note that we do not recommend running an antivirus scanner on the machine where a SonarQube analysis runs, it could result in unpredictable behavior.

SonarCloud User?

If you are using SonarCloud and TravisCI, you can use the folllowing Travis Add-On to directly feed SonarCloud with the minimum of configuration: https://docs.travis-ci.com/user/sonarqube/

F.A.Q.

Q. Analysis errors out with `java.lang.OutOfMemoryError: GC overhead limit exceeded`. What do I do?

A. This means that your project is too large or too intricate for the scanner to analyze with the default memory allocation. To fix this you'll want to allocate a larger heap (using `-Xmx[numeric value here]`) to the process running the analysis. Some CI engines may give you an input to specify the necessary values, for instance if you're using a Maven Build Step in a Jenkins job to run analysis. Otherwise, use Java Options to set a higher value. Note that details of setting Java Options are omitted here because they vary depending on the environment.

For More Information

See also:

- Analysis Parameters
- Project Examples
- Generic Test Data