# SonarQube Integration

TeamCity integration with SonarQube is implemented via the open-source SonarQube plugin for TeamCity.

The plugin provides a simple user interface for configuring connection between TeamCity and SonarQube servers, and allows you to trigger analysis using the SonarQube Runner as a build step in TeamCity.

When the analysis is completed, the results are automatically published to the SonarQube server.

This page explains how to configure and use TeamCity integration with SonarQube.

On this page:



## Installing Plugin

Download the plugin from the public TeamCity server and install it as described here.

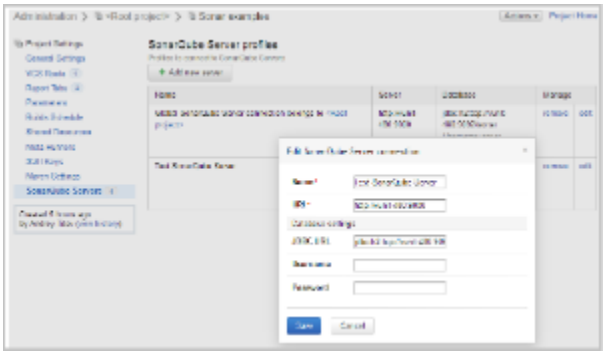
After the SonarQube Runner plugin is installed, the SonarQube Servers page appears in the project settings and the SonarQube Runner is added to the list of the available runners for a Build Step.

TeamCity 8.1 or greater is supported

## Configuring SonarQube Server Connection

To enable the integration, you need to create at least one connection to a SonarQube Server. The connection will be used by TeamCity to send data to the SonarQube Server.

SonarQube Servers connections are managed in the SonarQube Servers page of the Project Settings, where the SonarQube server name and URL as well as the database settings are specified.
For details on the database settings, see the SonarQube documentation. Example settings for different databases can be found here.



The connection defined in a project will be available to all its build configurations and subprojects.

## Configuring SonarQube Build Step

After the integration is enabled, you can add and configure the SonarQube Build Runner as the final step of the build you want to run analysis on.
Triggerring the analysis with Maven is not directly supported. To use the SonarQube runner with a Maven project, add the `sonar:sonar` goal to your project. See SonarQube documentation for detals.


In the runner settings, you need to select a connection to the server to send the data to. Other fields are to be configured as described in the SonarQube Runner documentation. `sonar-project.properties` are partially supported.

It is also recommended to provide paths to directories containing your sources root, test root, and binaries.

You can add more parameters for the SonarQube Runner in the "Additional parameters" field of the Advanced Options.


The plugin will trigger the SonarQube analysis with a SonarQube Runner and publish the results to the SonarQube server.

## Viewing analysis results

Once the build is finished, the View in sonar link appears on the Build Results page allowing you to navigate to the SonarQube dashboard to view the results of the analysis.

SonarQube Build Breaker is supported by the TeamCity plugin: the Build Breaker messages are parsed by TeamCity and added as Build Problems.

## Configuring SonarQube Runner with:

### ReSharper inspections (Inspections (.NET) runner)

1. Add `/output=%system.teamcity.build.tempDir%\inspectReport.xml` to the 'Additional inspectCode.exe arguments'
2. In SonarQube Runner step, add the following to the 'Additional parameters' field:

```
-Dsonar.resharper.cs.reportPath=%system.teamcity.build.tempDir%\inspectReport.xml
      -Dsonar.resharper.solutionFile=pathToSolutionFile
```

### dotCover results