

Viewing code scanning logs

In this article

- About your code scanning configuration
- About analysis and diagnostic information
- Viewing the logging output from code scanning

You can view the output generated during code scanning analysis in GitHub.com.

Who can use this feature

If you have write permissions to a repository, you can view the code scanning logs for that repository.

Code scanning is available for all public repositories on GitHub.com. Code scanning is also available for private repositories owned by organizations that use GitHub Enterprise Cloud and have a license for GitHub Advanced Security. For more information, see "[About GitHub Advanced Security](#)."

About your code scanning configuration

You can use a variety of tools to configure code scanning in your repository. For more information, see "[Configuring default setup for code scanning](#)" and "[Configuring advanced setup for code scanning](#)."

The log and diagnostic information available to you depends on the method you use for code scanning in your repository. You can check the type of code scanning you're using in the **Security** tab of your repository, by using the **Tool** drop-down menu in the alert list. For more information, see "[Managing code scanning alerts for your repository](#)."

About analysis and diagnostic information

You can see analysis and diagnostic information for code scanning run using CodeQL analysis on GitHub.

Analysis information is shown for the most recent analysis in a header at the top of the list of alerts. For more information, see "[Managing code scanning alerts for your repository](#)."

Diagnostic information is displayed in the Action workflow logs and consists of summary metrics and extractor diagnostics. For information about accessing code scanning logs on GitHub, see "[Viewing the logging output from code scanning](#)" below.

If you're using the CodeQL CLI outside GitHub, you'll see diagnostic information in the output generated during database analysis. This information is also included in the SARIF results file you upload to GitHub with the code scanning results.

For information about the CodeQL CLI, see "[Analyzing your code with CodeQL queries](#)."

About summary metrics [↗](#)

Summary metrics include:

- Lines of code in the codebase (used as a baseline), before creation and extraction of the CodeQL database
- Lines of code in the CodeQL database extracted from the code, including external libraries and auto-generated files
- Lines of code in the CodeQL database excluding auto-generated files and external libraries

About CodeQL source code extraction diagnostics [↗](#)

Extractor diagnostics only cover files that were seen during the analysis, metrics include:

- Number of files successfully analyzed
- Number of files that generated extractor errors during database creation
- Number of files that generated extractor warnings during database creation

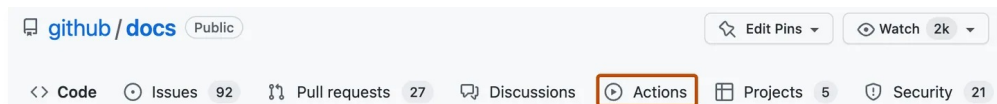
You can see more detailed information about CodeQL extractor errors and warnings that occurred during database creation by enabling debug logging. For more information, see "[Logs are not detailed enough](#)."

Viewing the logging output from code scanning [↗](#)

This section applies to code scanning run using GitHub Actions (CodeQL or third-party).

After configuring code scanning for your repository, you can watch the output of the actions as they run.


- 1 Under your repository name, click [⌵ Actions](#).



You'll see a list that includes an entry for running the code scanning workflow. The text of the entry is the title you gave your commit message.

All workflows

Showing runs from all workflows

1 workflow run		Event ▾	Status ▾	Branch ▾	Actor ▾
	Create .github/workflows/codeql.yml CodeQL #1: Commit 62aa401 pushed by octocat			main	now In progress

- 2 Click the entry for the code scanning workflow.

Note: If you are looking for the CodeQL workflow run triggered by enabling default setup, the text of the entry is "CodeQL."

- 3 Click the job name on the left. For example, **Analyze (LANGUAGE)**.

✓ Create .github/workflows/codeql.yml #1

Summary

Jobs

✓ Analyze (go)

Run details

Usage

Workflow file

Analyze (go)

succeeded 2 minutes ago in 1m 55s

Search logs

- > ✓ Set up job
- > ✓ Checkout repository
- > ✓ Initialize CodeQL
- > ✓ Autobuild

- 4 Review the logging output from the actions in this workflow as they run.
- 5 Optionally, to see more detail about the commit that triggered the workflow run, click the short commit hash. The short commit hash is 7 lowercase characters immediately following the commit author's username.
- 6 Once all jobs are complete, you can view the details of any code scanning alerts that were identified. For more information, see "[Managing code scanning alerts for your repository](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)