About the CodeQL CLI

In this article

You can use the CodeQL CLI to run CodeQL processes locally on software projects or to generate code scanning results for upload to GitHub Enterprise Cloud.

GitHub CodeQL is licensed on a per-user basis upon installation. You can use CodeQL only for certain tasks under the license restrictions. For more information, see "About the CodeQL CLI." If you have a GitHub Advanced Security license, you can use CodeQL for automated analysis, continuous integration, and continuous delivery. For more information, see "About GitHub Advanced Security."

About the CodeQL CLI

Software developers and security researchers can secure their code using CodeQL analysis. For more information about CodeQL, see "About code scanning with CodeQL."

The CodeQL CLI is a standalone, command-line tool that you can use to analyze code. Its main purpose is to generate a database representation of a codebase, a CodeQL database. Once the database is ready, you can query it interactively, or run a suite of queries to generate a set of results in SARIF format and upload the results to GitHub.com.

You can use the CodeQL CLI to:

- Run CodeQL analyses using queries provided by GitHub engineers and the open source community
- Generate code scanning alerts that you can upload to display in GitHub Enterprise Cloud
- Create CodeQL databases to use in the CodeQL for Visual Studio Code extension.
- Develop and test custom CodeQL queries to use in your own analyses

The CodeQL CLI can analyze:

- Dynamic languages, for example, JavaScript and Python.
- Compiled languages, for example, C/C++, C#, Go, and Java.
- Codebases written in a mixture of languages.

For information about setting up the CodeQL CLI, see "Setting up the CodeQL CLI."

About using the CodeQL CLI for code scanning

You can use the CodeQL CLI to run code scanning on code that you're processing in a third-party continuous integration (CI) system. Code scanning is a feature that you use to analyze the code in a GitHub repository to find security vulnerabilities and coding errors. Any problems identified by the analysis are shown in GitHub Enterprise Cloud. For an overview of using code scanning with external CI systems, see "Using code scanning with your existing CI system." For recommended specifications (RAM, CPU cores, and disk) for running CodeQL analysis, see "Recommended hardware resources for running CodeQL."

Alternatively, you can use GitHub Actions or Azure DevOps pipelines to scan code using the CodeQL CLI. For more information, see "Configuring default setup for code scanning" or Configure GitHub Advanced Security for Azure DevOps in Microsoft Learn.

For an overview of all the options for using CodeQL analysis for code scanning, see "About code scanning with CodeQL."

Notes:

- The CodeQL CLI is free to use on public repositories that are maintained on GitHub.com, and available to use on private repositories that are owned by customers with an Advanced Security license. For information, see "GitHub Enterprise Cloud CodeQL Terms and Conditions" and "CodeQL CLI."
- The CodeQL CLI is currently not compatible with non-glibc Linux distributions such as (musl-based) Alpine Linux.

About generating code scanning results with the CodeQL CLI

If you choose to run the CodeQL CLI directly, you first have to install the CodeQL CLI locally. If you are planning to use the CodeQL CLI with an external CI system, you need to make the CodeQL CLI available to servers in your CI system. For more information, see "Setting up the CodeQL CLI."

Once the CodeQL CLI is set up, you can use three different commands to generate results and upload them to GitHub Enterprise Cloud:

1. `database create` to create a CodeQL database to represent the hierarchical structure of each supported programming language in the repository. For more information, see "Preparing your code for CodeQL analysis."
2. `database analyze` to run queries to analyze each CodeQL database and summarize the results in a SARIF file. For more information, see "Analyzing your code with CodeQL queries."
3. `github upload-results` to upload the resulting SARIF files to GitHub Enterprise Cloud where the results are matched to a branch or pull request and displayed as code scanning alerts. For more information, see "Uploading CodeQL analysis results to GitHub."

Note: Uploading SARIF data to display as code scanning results in GitHub Enterprise Cloud is supported for organization-owned repositories with GitHub Advanced Security enabled, and public repositories on GitHub.com. For more information, see "Managing security and analysis settings for your repository."

Example CI configuration for CodeQL analysis

This is an example of the full series of commands for the CodeQL CLI that you might use to analyze a codebase with two supported languages and then upload the results to GitHub Enterprise Cloud.

```
# Create CodeQL databases for Java and Python in the 'codeql-dbs' directory
# Call the normal build script for the codebase: 'myBuildScript'

codeql database create codeql-dbs --source-root=src \
    --db-cluster --language=java,python --command=./myBuildScript

# Analyze the CodeQL database for Java, 'codeql-dbs/java'
# Tag the data as 'java' results and store in: 'java-results.sarif'

codeql database analyze codeql-dbs/java java-code-scanning.qls \
    --format=sarif-latest --sarif-category=java --output=java-results.sarif

# Analyze the CodeQL database for Python, 'codeql-dbs/python'
# Tag the data as 'python' results and store in: 'python-results.sarif'

codeql database analyze codeql-dbs/python python-code-scanning.qls \
    --format=sarif-latest --sarif-category=python --output=python-results.sarif

# Upload the SARIF file with the Java results: 'java-results.sarif'
# The GitHub App or personal access token created for authentication
# with GitHub's REST API is available in the `GITHUB_TOKEN` environment variable.

codeql github upload-results \
    --repository=my-org/example-repo \
    --ref=refs/heads/main --commit=deb275d2d5fe9a522a0b7bd8b6b6a1c939552718 \
    --sarif=java-results.sarif

# Upload the SARIF file with the Python results: 'python-results.sarif'

codeql github upload-results \
    --repository=my-org/example-repo \
    --ref=refs/heads/main --commit=deb275d2d5fe9a522a0b7bd8b6b6a1c939552718 \
    --sarif=python-results.sarif
```

About the GitHub CodeQL license

License notice: If you don’t have a GitHub Enterprise license then, by installing this product, you are agreeing to the GitHub CodeQL Terms and Conditions.

GitHub CodeQL is licensed on a per-user basis. Under the license restrictions, you can use CodeQL to perform the following tasks:

- To perform academic research.
- To demonstrate the software.
- To test CodeQL queries that are released under an OSI-approved License to confirm that new versions of those queries continue to find the right vulnerabilities.

Where "OSI-approved License" means an Open Source Initiative (OSI)-approved open source software license.

If you are working with an Open Source Codebase (that is, a codebase that is released under an OSI-approved License) you can also use CodeQL for the following tasks:

- To perform analysis of the Open Source Codebase.
- If the Open Source Codebase is hosted and maintained on GitHub.com, to generate CodeQL databases for or during automated analysis, continuous integration, or continuous delivery.

CodeQL can’t be used for automated analysis, continuous integration or continuous delivery, whether as part of normal software engineering processes or otherwise, except in the express cases set forth herein. For these uses, contact the sales team.