About code scanning with CodeQL

In this article

You can use CodeQL to identify vulnerabilities and errors in your code. The results are shown as code scanning alerts in GitHub.

Code scanning is available for all public repositories on GitHub.com. To use code scanning in a private repository owned by an organization, you must have a license for GitHub Advanced Security. For more information, see "About GitHub Advanced Security."

About code scanning with CodeQL

CodeQL is the code analysis engine developed by GitHub to automate security checks. You can analyze your code using CodeQL and display the results as code scanning alerts.

There are three main ways to use CodeQL analysis for code scanning:

- Use default setup to quickly configure CodeQL analysis for code scanning on your repository. Default setup automatically chooses the languages to analyze, query suite to run, and events that trigger scans. If you prefer, you can manually select the query suite to run and languages to analyze. After you enable CodeQL, GitHub Actions will execute workflow runs to scan your code. For more information, see "Configuring default setup for code scanning."
- Use advanced setup to add the CodeQL workflow to your repository. This generates a customizable workflow file which uses the github/codeql-action to run the CodeQL CLI. For more information, see "Configuring advanced setup for code scanning."
- Run the CodeQL CLI directly in an external CI system and upload the results to GitHub. For more information, see "Using code scanning with your existing CI system."

For information about code scanning alerts, see "About code scanning alerts."

About CodeQL

CodeQL treats code like data, allowing you to find potential vulnerabilities in your code with greater confidence than traditional static analyzers.

1. You generate a CodeQL database to represent your codebase.
2. Then you run CodeQL queries on that database to identify problems in the codebase.
3. The query results are shown as code scanning alerts in GitHub Enterprise Cloud

when you use CodeQL with code scanning.

CodeQL supports both compiled and interpreted languages, and can find vulnerabilities and errors in code that's written in the supported languages.

- C/C++
- C#
- Go
- Java/Kotlin
- JavaScript/TypeScript
- Python
- Ruby
- Swift

Notes:

- CodeQL analysis for Swift is currently in beta. During the beta, analysis of Swift will be less comprehensive than CodeQL analysis of other languages. Additionally, Swift 5.8 is not yet supported.
- CodeQL analysis for Kotlin is currently in beta. During the beta, analysis of Kotlin will be less comprehensive than CodeQL analysis of other languages.
- Use `java-kotlin` to analyze code written in Java, Kotlin or both.
- Use `javascript-typescript` to analyze code written in JavaScript, TypeScript or both.

For more information, see the documentation on the CodeQL website: "Supported languages and frameworks."

About CodeQL queries

GitHub experts, security researchers, and community contributors write and maintain the default CodeQL queries used for code scanning. The queries are regularly updated to improve analysis and reduce any false positive results. The queries are open source, so you can view and contribute to the queries in the `github/codeql` repository. For more information, see CodeQL on the CodeQL website. You can also write your own queries. For more information, see "About CodeQL queries" in the CodeQL documentation.

If you are scanning your code with advanced setup or an external CI system, you can run additional queries as part of your analysis. These queries must belong to a published CodeQL query pack (beta) or a CodeQL pack in a repository. CodeQL packs (beta) provide the following benefits over traditional QL packs:

- When a CodeQL query pack (beta) is published to the GitHub Container registry, all the transitive dependencies required by the queries and a compilation cache are included in the package. This improves performance and ensures that running the queries in the pack gives identical results every time until you upgrade to a new version of the pack or the CLI.
- QL packs do not include transitive dependencies, so queries in the pack can depend only on the standard libraries (that is, the libraries referenced by an `import LANGUAGE` statement in your query), or libraries in the same QL pack as the query.
- CodeQL query packs (beta) can be downloaded from multiple GitHub container registries. For more information, see "Customizing your advanced setup for code scanning."

For more information, see "Customizing analysis with CodeQL packs."

Note: The CodeQL package management functionality, including CodeQL packs, is currently in

beta and subject to change.