Running CodeQL code scanning in a container

In this article

You can run code scanning in a container by ensuring that all processes run in the same container.

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 a containerized build

If you're configuring code scanning for a compiled language, and you're building the code in a containerized environment, the analysis may fail with the error message "No source code was seen during the build." This indicates that CodeQL was unable to monitor your code as it was compiled.

You must run CodeQL inside the container in which you build your code. This applies whether you are using the CodeQL CLI or GitHub Actions. For the CodeQL CLI, see "Using code scanning with your existing CI system" for more information. If you're using GitHub Actions, configure your workflow to run all the actions in the same container. For more information, see "Example workflow."

Note: The CodeQL CLI is currently not compatible with non-glibc Linux distributions such as (musl-based) Alpine Linux.

Dependencies for CodeQL code scanning

You may have difficulty running code scanning if the container you're using is missing certain dependencies (for example, Git must be installed and added to the PATH variable). If you encounter dependency issues, review the list of software typically included on GitHub's runner images. For more information, see the version-specific `readme` files in these locations:

- Linux: https://github.com/actions/runner-images/tree/main/images/linux
- macOS: https://github.com/actions/runner-images/tree/main/images/macos
- Windows: https://github.com/actions/runner-images/tree/main/images/win

Example workflow

This sample workflow uses GitHub Actions to run CodeQL analysis in a containerized environment. The value of `container.image` identifies the container to use. In this example the image is named `codeql-container`, with a tag of `f0f91db`. For more information, see "Workflow syntax for GitHub Actions."

```yaml
name: "CodeQL"

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]
  schedule:
    - cron: '15 5 * * 3'

jobs:
  analyze:
    name: Analyze
    runs-on: ubuntu-latest
    permissions:
      security-events: write
      actions: read

    strategy:
      fail-fast: false
      matrix:
        language: [java-kotlin]

    # Specify the container in which actions will run
    container:
      image: codeql-container:f0f91db

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4
      - name: Initialize CodeQL
        uses: github/codeql-action/init@v2
        with:
          languages: ${{ matrix.language }}
      - name: Build
        run: |
          ./configure
          make
      - name: Perform CodeQL Analysis
        uses: github/codeql-action/analyze@v2
```