

Logs are not detailed enough

In this article

Enable step debug logging

Creating CodeQL debugging artifacts

If you'd like to increase the level of detail in your logs, try these steps.

If your logs are not detailed enough, there are several steps you can take to make them more useful.

Enable step debug logging [↗](#)

You can enable step debug logging in GitHub Actions to increase the verbosity of a job's logs during and after a job's execution. For more information, see "[Enabling debug logging](#)."

Creating CodeQL debugging artifacts [↗](#)

You can obtain artifacts to help you debug CodeQL. The debug artifacts will be uploaded to the workflow run as an artifact named `debug-artifacts`. The data contains the CodeQL logs, CodeQL database(s), and any SARIF file(s) produced by the workflow.

These artifacts will help you debug problems with CodeQL code scanning. If you contact GitHub support, they might ask for this data.

Creating CodeQL debugging artifacts by re-running jobs with debug logging enabled [↗](#)

You can create CodeQL debugging artifacts by enabling debug logging and re-running the jobs. For more information about re-running GitHub Actions workflows and jobs, see "[Re-running workflows and jobs](#)."

You need to ensure that you select **Enable debug logging**. This option enables runner diagnostic logging and step debug logging for the run. You'll then be able to download `debug-artifacts` to investigate further. You do not need to modify the workflow file when creating CodeQL debugging artifacts by re-running jobs.

Creating CodeQL debugging artifacts using a workflow flag [↗](#)

You can create CodeQL debugging artifacts by using a flag in your workflow. For this, you need to modify the `init` step of your CodeQL analysis workflow file and set `debug: true`.

```
- name: Initialize CodeQL
  uses: github/codeql-action/init@v2
  with:
    debug: true
```

Legal

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