GitHub Code Scanning Quickstart

CodeQL + Third-Party SAST (SonarCloud, Snyk) Cheat Sheet

TL;DR

- Run CodeQL *and* optionally a third-party SAST for broader coverage.
- Prefer tools that emit **SARIF** so findings appear in the repo Security tab and PR checks.
- Store vendor tokens in **Actions secrets**; never commit credentials.
- Avoid duplicate CodeQL by not enabling both the Default workflow and a custom CodeQL workflow.

Quickstart: Third-Party Scanning in GitHub

1. Choose a Marketplace Action or App (e.g., SonarCloud, Snyk).
2. Add required credentials as **Actions secrets** at repo or org level.
3. Add a workflow that sets env vars and runs the vendor Action.
4. Ensure a **SARIF** file gets uploaded (vendor does it automatically or add an explicit upload step).
5. Gate on PRs (required status checks) and schedule weekly scans on the default branch.
6. If using custom CodeQL YAML, *disable* the Default CodeQL in repo settings to prevent double runs.

CodeQL vs. Third-Party at a Glance

Comparison

CodeQL	First-class GitHub integration, query packs, customizable queries; config can be dense but deeply integrated.
Third-Party SAST	Often fast to start, rich dashboards/quality metrics; requires Action/App wiring and secrets; ensure SARIF upload for unified PR view.

Drop-In YAML Snippets

A. CodeQL (basic)

```yaml
name: CodeQL
on:
  push: { branches: ["main"] }
  pull_request: { branches: ["main"] }
  schedule:
    - cron: "0 3 * * 1"   # weekly
permissions:
  contents: read
  security-events: write
jobs:
  analyze:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: github/codeql-action/init@v3
        with:
          languages: 'javascript,python'  # adjust
      - uses: github/codeql-action/autobuild@v3
      - uses: github/codeql-action/analyze@v3
```

Note

If you use this custom workflow, disable the *Default* CodeQL configuration in the repo Security settings to avoid duplicate scans.

B. SonarCloud (vendor token in `SONAR_TOKEN`)

```yaml
name: SonarCloud Scan
on:
  pull_request:
  push: { branches: ["main"] }
permissions:
  contents: read
  id-token: write
  security-events: write
env:
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
  SONAR_TOKEN:  ${{ secrets.SONAR_TOKEN }}
jobs:
  scan:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with:
          fetch-depth: 0
      - uses: SonarSource/sonarcloud-github-action@v2
        with:
          args: >
            -Dsonar.projectKey=your_org_your_repo
            -Dsonar.organization=your_org
      # If your vendor does not auto-upload SARIF, add:
      # - uses: github/codeql-action/upload-sarif@v3
      #   with:
      #     sarif_file: path/to/report.sarif
```

C. Snyk (example with explicit SARIF upload)

```yaml
name: Snyk SAST
on:
  pull_request:
  push: { branches: ["main"] }
permissions:
  contents: read
  security-events: write
env:
  SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}
jobs:
  snyk:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: snyk/actions/setup@master
      - run: snyk code test --sarif > snyk.sarif
      - uses: github/codeql-action/upload-sarif@v3
        with:
          sarif_file: snyk.sarif
```

Using CodeQL Outside GitHub Actions

You can run CodeQL in other CI systems (e.g., Jenkins, Azure DevOps) and still publish results back to GitHub. Expect extra setup for CLI, auth, and SARIF upload, but the unified view in the repo Security tab and PR checks is preserved.

Gotchas & Best Practices

Checklist

- **Duplicate scans:** Do not run both Default and custom CodeQL.
- **Secrets hygiene:** Use `${{ secrets.* }}`; never hard-code tokens.
- **PR visibility:** Ensure tools emit SARIF; otherwise findings will not appear in PR checks/Security tab.
- **Least privilege:** Scope vendor tokens to what the scanner needs.
- **Schedules:** Add a weekly cron to catch drift on default branches.

Build Tips

- This document uses `minted`. Compile with `-shell-escape`.
- If your build system disallows `-shell-escape`, switch to `listings` or enable Pygments on your CI runner.
- Keep YAML blocks minimal to avoid unrecognized lexers and Unicode issues. Use ASCII quotes and hyphens when possible.

Cheat sheet version: January 19, 2026.