

Quick Start Guide for GitHub users

Add CodeDelta to a repository you already have: one workflow file, nothing to install, no secrets to set up. The first report arrives on your next pull request. New to GitHub altogether? Use the companion guide, [New to GitHub? Quick Start Guide](#).

What you need

A repository where you can add a workflow file, with GitHub Actions enabled. No licence file: the Action carries a built-in evaluation licence, valid to **31 December 2026**. Public and private repositories both work.

What it costs

It runs on your own GitHub Actions minutes; a free account includes 2,000 a month. The run on our public demo took 23 seconds. It runs entirely on your runner: no source leaves it and nothing phones home.

On an employer's GitHub, check first that you are allowed: organisations can switch Actions off or restrict who may add workflows and secrets. If so, ask the organisation's GitHub admin to enable Actions for the repository.

1 Add the workflow

Create `.github/workflows/codedelta.yml` on your default branch with the text below. Indentation matters; to avoid losing it when copying from a PDF, copy the same text from codedelta.app/github-quickstart.html.

```
name: CodeDelta
on: [pull_request]

permissions:
  contents: read
  pull-requests: write      # post the summary comment
  security-events: write    # SARIF to the Security tab (public repo or Advanced Security)

jobs:
  codedelta:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with: { fetch-depth: 0 }
      - uses: code-delta-app/action@v1
```

In words: whenever a pull request is opened or updated, fetch the code with its full history (`fetch-depth: 0`, needed to compare base and head), run CodeDelta, and post the result as a comment. The engine downloads itself from github.com/code-delta-app/releases.

2 Open a pull request

Any pull request from a branch of the same repository triggers it. CodeDelta compares the pull request's head with its base branch for churn, and scans the tree for AI agent code.

3 Read the report

A comment headed **CodeDelta report** appears on the pull request and is updated in place on each push. The same report is on the run's summary page under the **Actions** tab. This is the start of the real comment on our public demo, github.com/code-delta-app/demo/pull/4:

CodeDelta report
Churn: 47 logical statements (+25 / -14 / Δ8); REP_CHURN 0.83
Agent scan: 64 CRITICAL · 7 HIGH · 81 ELEVATED · 17 endpoint-level · 57 text-level

Churn	Logical statements the change touched: + added, – deleted, Δ changed in place. REP_CHURN = (added + deleted) ÷ churn, from 0 to 1. A high figure means code was replaced, not repaired.
Agent scan	Files that call AI models or launch AI agents, by severity. The demo's figures are large on purpose: it is full of inert fake agent files.

On a private repository the **Upload SARIF** step may show a warning. That step feeds the Security tab, which needs a public repository or GitHub Advanced Security. It is best-effort and does not affect the comment or the result of the run.

4 Options

Set these under **with:** on the `code-delta-app/action@v1` step. All are optional.

Input	Default	What it does
mode	<code>churn_agent</code>	<code>churn_agent</code> churn + Agent Scan · <code>churn</code> churn only · <code>both</code> adds the ML AI audit · <code>ai_audit</code> AI audit + Agent Scan, no churn · <code>ai</code> · <code>agent</code>
path	<code>.</code>	Directory to scan.
comment / sarif	<code>true</code>	Post the pull request comment / upload SARIF to the Security tab. Set <code>sarif: "false"</code> on a private repository without Advanced Security to skip that step.
gate	<code>false</code>	Fail the job on the default AI-governance policy: data sent to a model hosted in CN, RU, KP or IR, or a program launch driven by model output. <code>gate-policy</code> takes the path of your own policy file.
baseline + fail-on-new	—	Path to a committed <code>codedelta-baseline.json</code> ; with <code>fail-on-new: "true"</code> only findings that are new against the baseline fail the job.
bom	—	Write an AI Bill of Materials to this path; <code>bom-format</code> is <code>native</code> or <code>cyclonedx</code> .
threshold	<code>50</code>	AI audit sensitivity, 0 to 100. It moves the rating cut-offs only.
license	<code>built-in</code>	Your own licence: see step 6.

5 Block a merge

With `gate` or `fail-on-new` the job fails (exit code 3) when the policy is breached or new findings appear. To make that block the merge, mark the CodeDelta check as required in the branch protection rules of your base branch. On a pull request the baseline and any policy file are read from the **base branch**, never from the pull request, so a change cannot loosen its own gate: land rule changes on the base branch first.

6 Your own licence

To run past 31 December 2026, or to use your own licence sooner, store it as a repository secret and pass it in. Encode the file first: `base64 -i codedelta.lic | pbcopy` on macOS, `base64 -w0 codedelta.lic` on Linux. Then **Settings** → **Secrets and variables** → **Actions** → **New repository secret**, name `CODEDELTA_LICENSE` , and add to the step:

```
- uses: code-delta-app/action@v1
  with:
    license: ${ secrets.CODEDELTA_LICENSE }
```

The licence is verified on the runner. Pull requests from forks receive no secrets, so a stranger's pull request cannot read it.

7 Run without a pull request

For a manual button or a nightly run, use `on: workflow_dispatch` or `schedule` ; a ready-made file is `example-workflow-manual.yml` in github.com/code-delta-app/action. Without a pull request there is no base to compare, so churn is skipped and the run is a snapshot: Agent Scan, plus the AI audit in `both` mode.

If something does not look right

No comment appears	Open the run under Actions : the report is on its summary page and the log shows each step. The workflow needs <code>pull-requests: write</code> .
No run at all	The file must be at <code>.github/workflows/codedelta.yml</code> on the default branch, and the pull request opened or updated after it landed. Check Actions is enabled for the repository.
Churn is missing	The checkout needs <code>fetch-depth: 0</code> . If the base branch cannot be fetched the log says so and the run falls back to a snapshot.
"workflow is not valid"	Indentation was lost when pasting. Copy the text again from codedelta.app/github-quickstart.html .

Where to go next

Every way to run it	codedelta.app/cli.html : the Action, Docker, Codespaces, GitLab and other CI, and the command line. Questions and answers at codedelta.app/faq.html#github .
The desktop app	The same engine with full reports and a side-by-side code browser, for Windows, macOS and Linux: codedelta.app/download.html .
What the figures mean	codedelta.app/user-guide.html defines every metric with a worked example.
Questions	Use the chat on codedelta.app , or the contact button on the download page. A person replies.