

Review the change. Approve the fix.

GitHub review evidence, bounded source relationships and human-approved patches in one hosted workflow.

1. Inspect the reviewed snapshot

CodLab reviews the pull request against an immutable base and head. Findings distinguish deterministic rules from advisory AI output. Bounded JavaScript/TypeScript relationships and security observations include explicit coverage limits; static links do not establish runtime reachability or a verified exploit.

2. Inspect a proposed patch

Generate a bounded proposal, inspect the exact change and choose a new draft pull request or the current branch. Publishing requires current repository authority, an unchanged reviewed head and explicit patch approval under the workspace policy. Approval does not prove the patch is correct.

3. Read actual validation evidence

Required CI is checked against configured names, trusted producers and the candidate commit. Missing policy, unavailable checks or stale results remain unknown. Isolated execution of customer code is disabled until a suitable runtime is qualified; browser QA of CodLab is a separate activity.

4. Follow review delivery

The workspace distinguishes queued, processing, retrying, failed and completed reviews. Temporary provider failures need bounded recovery; ambiguous publication is reconciled using recorded provider identities. Cross-system effects are not advertised as exactly once.

Clear data boundaries. Inspectable controls.

Hosted processing and retained evidence

The hosted platform uses Cloudflare Workers, D1 and Workers AI. Full source context is bounded and transient during analysis. Stored reports can retain selected source excerpts; proposed fixes can retain old/new source with their separate expiry and cleanup rules. Pending publication payloads support recovery. This hosted product is not zero retention.

Authority and policy

Repository access, workspace roles and fresh GitHub authorization govern sensitive actions. Trusted-base `.codecr.yml` settings cannot be weakened by the pull request itself. Only qualifying deterministic critical findings block; AI and ordinary syntax candidates remain advisory. Merge requirements also depend on the repository branch protections.

Optional context and future deployment requirements

Supporting AI context requires both the operator gate and a current authorized repository administrator choice. It is disabled by default. VPC/on-premises deployment, air-gap parity, customer KMS/HSM, SAML/SCIM, residency and contractual service levels require separate feasibility, implemented controls and deployed evidence; this brief does not promise them as self-service capabilities.

Quality and support

Regression tests establish only their tested scope. Independent real-PR accuracy, production latency and customer patch execution must be evaluated separately. Public product details and pricing are at codecr.org; the hosted workflow guide is at docs.codecr.org. Start a private security report at hello@codecr.org without credentials or customer source.

Read the full data map at codecr.org/privacy/ and current control boundaries at codecr.org/security/. The HTML guides are the accessible reading alternative. Source builds and owner-qualified deployments are recorded separately.