CLAUDE CODE CENTER OF EXCELLENCE

AI-Powered Vulnerability Remediation

From scan to fixed MR — automatically. Claude agents triage reachability, generate dependency upgrades, patch SAST findings, and open a Jira-linked draft MR.

SCAN

Dep, SAST, Container, DAST, Secret

TRIAGE

Reachability analysis removes noise

FIX

AI-generated patches & version bumps

TRACK

Draft MR + Jira story per run

↑ ↓ to navigate

— WHAT'S NEW — V2.1.161 – V2.1.178

What's New with Claude Code

⚡

Fable 5 & Bedrock

Fable 5 now available — Mythos-class model
Bedrock auto-reads region from ~/.aws config
Credential caching now honors token Expiration

🤖

Agentic Workflows

Sub-agents nest up to 5 levels deep
fallbackModel setting for overload resilience
Parallel tool calls: failures no longer cancel the batch

🛠️

Commands & Permissions

/cd changes working directory mid-session
--safe-mode disables all customizations
Tool(param:value) permission rule syntax

— SECURITY CORNER

CVE Debt is Growing

📈

Volume & Velocity

Thousands of new CVEs published each month. Dependency trees mean a single package bump can resolve dozens of findings — but someone has to find, triage, and test each one.

💤

Noise Overload

Up to **60% of flagged CVEs** are unreachable in practice — dead code paths, test deps, unused imports. Without triage, teams waste cycles on phantom risk.

⏳

Slow Manual Loop

Scan → read report → identify fix → bump version → open PR → link Jira → get review. Repeated for every vuln, every repo, every sprint. Engineering time that doesn't ship features.

The opportunity: AI agents can automate the entire triage → fix → MR loop — freeing engineers to review rather than remediate.

ARCHITECTURE

The 5-Stage Pipeline

01

Scan

GitLab scanners produce report artifacts (dep, SAST, DAST)

02

Triage

AI reachability agent filters unreachable vulns (~40–60% noise removed)

03

MR + Jira

AI agent opens draft MR branch & creates Jira story linked to your epic

04

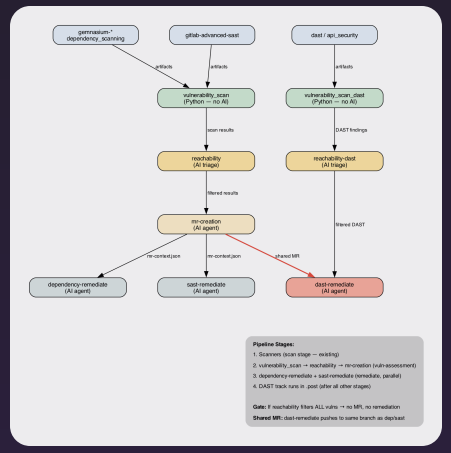
Remediate

Parallel agents fix dep, container, SAST & DAST — all push to same MR

05

Observe

cve-observe reports MR state, vuln counts & pipeline health in .post



↑ ↓ to navigate

— COVERAGE

Five Scan Types Supported

📦

Dependency

Vulnerable packages in manifests — Maven, npm, pip, and more

`DEPENDENCY_SCANNING: "true"`

📝

SAST

Static analysis findings in source code — code-level security issues

`SAST_SCANNING: "true"`

🐋

Container

Vulnerable base images & OS packages in container definitions

`CONTAINER_SCANNING: "true"`

🌐

DAST

Dynamic / API security findings from a running application

`DAST_SCANNING: "true"`

🔑

Secrets

Leaked tokens & credentials in source — detection only, no auto-fix yet

`SECRET_DETECTION_SCANNING: "true"`

Set at least one variable to `"true"` — no scan type enabled means no remediation jobs run

GETTING STARTED

Three Onboarding Levels

🤝

Level 1 — Assisted

Now. Logos team works 1:1 to configure & validate. We handle the YAML, verify test coverage, and trigger the first pipeline with you.

Reach out in Teams to get started

🛠️

Level 2 — Self-Service

Near-term. Follow the onboarding guide independently. Add stages, include the component, set scan type variables, and trigger a manual pipeline.

onboarding-guide.md in the repo

🏢

Level 3 — Org-Wide

Planned. DevSecOps applies the framework as a default include in shared pipeline templates. Teams opt-out rather than opt-in.

Qualification gates enforced automatically

PREREQUISITES

Unit tests in CI • Code coverage above threshold • Protected default branch • Integration/regression testing evidence

TRIGGER MODEL

Runs on **scheduled** or **manually triggered** pipelines only — not on every push

OUTPUT

Single draft MR on a `fix/cve-*` branch • One Jira story linked to your epic • All fix types in parallel

INTEGRATION

Add to Your .gitlab-ci.yml

.gitlab-ci.yml — required additions

```yaml
stages:
  # ...your existing stages...
  - vuln-assessment # after test
  - remediate # last before .post

include:
  - component: "${CI_SERVER_FQDN}/allyfinancial/shared-services/
    ai-marketplace/remediation-collaboration/
    vulnerability-remediation@main"

variables:
  DEPENDENCY_SCANNING: "true"
  SAST_SCANNING: "true" # optional
  JIRA_PROJECT: "ABC"
  EPIC_KEY: "ABC-468"
```

The component handles all job definitions. Your repo only needs **stages**, an **include**, and a few **variables**.

Optional inputs

Override via `inputs:` block:

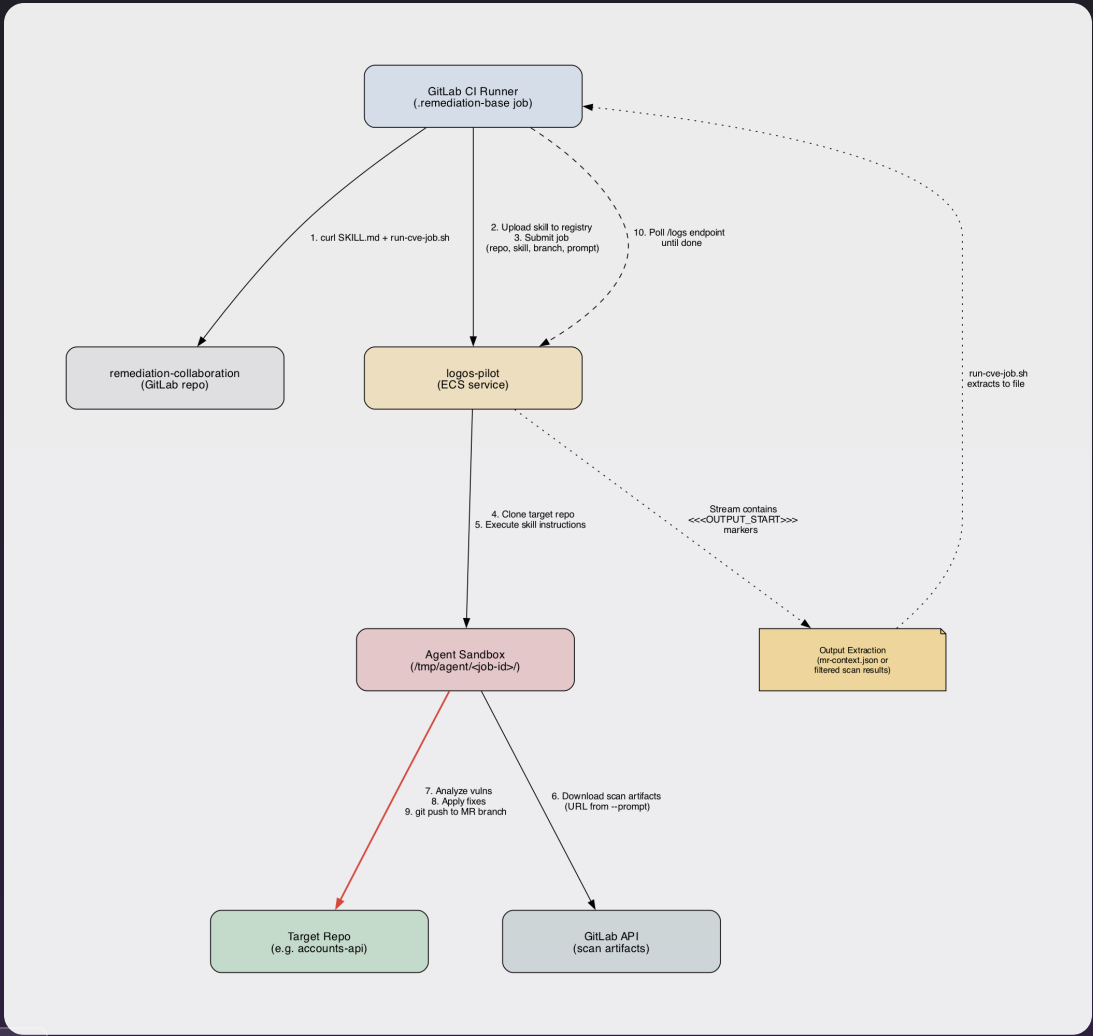
`cve-min-severity: "high"`
`cve-dry-run: "true"`
`SKIP_REACHABILITY: "true"`

DAST requires extra stage

Add `dast` stage between `test` and `vuln-assessment` only if your repo already has DAST scanning configured.

— UNDER THE HOOD

Agent Invocation Model



☁️

Remote Sandbox

Agents run in logos-pilot's ECS sandbox — not on your CI runner. Everything passes through `--prompt` as artifact URLs.

📦

Single MR Strategy

mr-creation opens the branch once. All remediation agents push to the same MR branch in parallel — one clean PR for review.

⚡

Reachability Gate

If triage filters out all findings, the pipeline short-circuits — no MR created, no remediation cost incurred.

STATUS & NEXT

Where We Are & What's Next

CURRENTLY WORKING

vulnerability_scan (artifact mode)	✓ Live
reachability triage	✓ Live
mr-creation + Jira story	✓ Live
dependency-remediate	✓ Live
container-remediate	✓ Live
sast-remediate	✓ Live
dast-remediate	✓ Live

ON THE ROADMAP

Post-remediation re-scan verification	In Progress
Secret detection auto-fix agent	Planned
Auto-merge for low-risk fixes	Planned
Unreachable-but-gating CVE workflow	Planned
Org-wide opt-out rollout (Level 3)	Future

Onboard now (Level 1): reach out to the Logos team in Teams — we'll configure it with you.