



Contributing


CI Pipeline

How the OpenClaw CI pipeline works

The CI runs on every push to `main` and every pull request. It uses smart scoping to skip expensive jobs when only docs or native code changed.

Job Overview

Job	Purpose	When it runs
docs-scope	Detect docs-only changes	Always
changed-scope	Detect which areas changed (node/macos/android)	Non-docs PRs
check	TypeScript types, lint, format	Non-docs changes
check-docs	Markdown lint + broken link check	Docs changed
code-analysis	LOC threshold check (1000 lines)	PRs only
secrets	Detect leaked secrets	Always
build-artifacts	Build dist once, share with other jobs	Non-docs, node changes
release-check	Validate npm pack contents	After build
checks	Node/Bun tests + protocol check	Non-docs, node changes
checks-windows	Windows-specific tests	Non-docs, node changes
macos	Swift lint/build/test + TS tests	PRs with macos changes

Job	Purpose	When it runs
 android	Gradle build + tests	Non-docs, android changes
>		

Fail-Fast Order

Jobs are ordered so cheap checks fail before expensive ones run:

- 1. docs-scope + code-analysis + check (parallel, ~1-2 min)
- 2. build-artifacts (blocked on above)
- 3. checks , checks-windows , macos , android (blocked on build)

Runners

Runner	Jobs
blacksmith-4vcpu-ubuntu-2404	Most Linux jobs
blacksmith-4vcpu-windows-2025	checks-windows
macos-latest	macos , ios
ubuntu-latest	Scope detection (lightweight)

Local Equivalents

```
pnpm check      # types + lint + format
pnpm test       # vitest tests
pnpm check:docs  # docs format + lint + broken links
pnpm release:check # validate npm pack
```



Powered by mintlify

>