Release notes

Release Checklist

Use `pnpm` (Node 22+) from the repo root. Keep the working tree clean before tagging/publishing.

Operator trigger

When the operator says “release”, immediately do this preflight (no extra questions unless blocked):

- Read this doc and `docs/platforms/mac/release.md`.
- Load env from `~/.profile` and confirm `SPARKLE_PRIVATE_KEY_FILE` + App Store Connect vars are set (SPARKLE_PRIVATE_KEY_FILE should live in `~/.profile`).
- Use Sparkle keys from `~/Library/CloudStorage/Dropbox/Backup/Sparkle` if needed.

1. **Version & metadata**

- [ ] Bump `package.json` version (e.g., `2026.1.29`).
- [ ] Run `pnpm plugins:sync` to align extension package versions + changelogs.
- [ ] Update CLI/version strings: `src/cli/program.ts` and the Baileys user agent in `src/provider-web.ts`.
- [ ] Confirm package metadata (name, description, repository, keywords, license) and `bin` map points to `openclaw.mjs` for

`openclaw .`

- [ ] If dependencies changed, run `pnpm install` so `pnpm-lock.yaml` is current.

2. **Build & artifacts**

- [ ] If A2UI inputs changed, run `pnpm canvas:a2ui:bundle` and commit any updated `src/canvas-host/a2ui/a2ui.bundle.js`.
- [ ] `pnpm run build` (regenerates `dist/`).
- [ ] Verify npm package `files` includes all required `dist/*` folders (notably `dist/node-host/**` and `dist/acp/**` for headless node + ACP CLI).
- [ ] Confirm `dist/build-info.json` exists and includes the expected `commit` hash (CLI banner uses this for npm installs).
- [ ] Optional: `npm pack --pack-destination /tmp` after the build; inspect the tarball contents and keep it handy for the GitHub release (do **not** commit it).

3. **Changelog & docs**

- [ ] Update `CHANGELOG.md` with user-facing highlights (create the file if missing); keep entries strictly descending by version.
- [ ] Ensure README examples/flags match current CLI behavior (notably new commands or options).

4. **Validation**

- [ ] `pnpm build`
- [ ] `pnpm check`
- [ ] `pnpm test` (or `pnpm test:coverage` if you need coverage output)
- [ ] `pnpm release:check` (verifies npm pack contents)
- [ ] `OPENCLAW_INSTALL_SMOKE_SKIP_NONROOT=1 pnpm test:install:smoke` (Docker install smoke test, fast path; required before release)

If the immediate previous npm release is known broken, set `OPENCLAW_INSTALL_SMOKE_PREVIOUS=<last-good-version>` or `OPENCLAW_INSTALL_SMOKE_SKIP_PREVIOUS=1` for the preinstall step.

- [ ] (Optional) Full installer smoke (adds non-root + CLI coverage): `pnpm test:install:smoke`
- [ ] (Optional) Installer E2E (Docker, runs `curl -fsSL https://openclaw.ai/install.sh | bash`, onboards, then runs real tool calls):
  - `pnpm test:install:e2e:openai` (requires `OPENAI_API_KEY`)
  - `pnpm test:install:e2e:anthropic` (requires `ANTHROPIC_API_KEY`)
  - `pnpm test:install:e2e` (requires both keys; runs both providers)
- [ ] (Optional) Spot-check the web gateway if your changes affect send/receive paths.

5. **macOS app (Sparkle)**

- [ ] Build + sign the macOS app, then zip it for distribution.
- [ ] Generate the Sparkle appcast (HTML notes via `scripts/make_appcast.sh`) and update `appcast.xml`.
- [ ] Keep the app zip (and optional dSYM zip) ready to attach to the GitHub release.
- [ ] Follow **macOS release** for the exact commands and required env vars.
  - `APP_BUILD` must be numeric + monotonic (no `-beta`) so Sparkle compares versions correctly.
  - If notarizing, use the `openclaw-notary` keychain profile created from App Store Connect API env vars (see **macOS release**).

6. **Publish (npm)**

- [ ] Confirm git status is clean; commit and push as needed.
- [ ] `npm login` (verify 2FA) if needed.

- [ ] `npm publish --access public` (use `--tag beta` for pre-releases).
- [ ] Verify the registry: `npm view openclaw version`, `npm view openclaw dist-tags`, and `npx -y openclaw@X.Y.Z --version` (or `--help`).

Troubleshooting (notes from 2.0.0-beta2 release)

`npm pack/publish` hangs or produces huge tarball: the macOS app bundle in `dist/OpenClaw.app` (and release zips) get swept into the package. Fix by whitelisting publish contents via `package.json` `files` (include dist subdirs, docs, skills; exclude app bundles). Confirm with `npm pack --dry-run` that `dist/OpenClaw.app` is not listed.

npm auth web loop for dist-tags: use legacy auth to get an OTP prompt:

```
NPM_CONFIG_AUTH_TYPE=legacy npm dist-tag add openclaw@X.Y.Z latest
```

`npx` **verification fails with** `ECOMPROMISED: Lock compromised`: retry with a fresh cache:

```
NPM_CONFIG_CACHE=/tmp/npm-cache-$(date +%s) npx -y openclaw@X.Y.Z --version
```

Tag needs repointing after a late fix: force-update and push the tag, then ensure the GitHub release assets still match:

```
git tag -f vX.Y.Z && git push -f origin vX.Y.Z
```

7. **GitHub release + appcast**

- [ ] Tag and push: `git tag vX.Y.Z && git push origin vX.Y.Z` (or `git push --tags`).
- [ ] Create/refresh the GitHub release for `vX.Y.Z` with **title** `openclaw X.Y.Z` (not just the tag); body should include the **full** changelog section for that version (Highlights + Changes + Fixes), inline (no bare links), and **must not repeat the title inside the body**.
- [ ] Attach artifacts: `npm pack` tarball (optional), `OpenClaw-X.Y.Z.zip`, and `OpenClaw-X.Y.Z.dSYM.zip` (if generated).

- [ ] Commit the updated `appcast.xml` and push it (Sparkle feeds from main).
- [ ] From a clean temp directory (no `package.json`), run `npx -y openclaw@X.Y.Z send --help` to confirm install/CLI entrypoints work.
- [ ] Announce/share release notes.

Plugin publish scope (npm)

We only publish **existing npm plugins** under the `@openclaw/*` scope. Bundled plugins that are not on npm stay **disk-tree only** (still shipped in `extensions/**`).

Process to derive the list:

1. `npm search @openclaw --json` and capture the package names.
2. Compare with `extensions/*/package.json` names.
3. Publish only the **intersection** (already on npm).

Current npm plugin list (update as needed):

- @openclaw/bluebubbles
- @openclaw/diagnostics-otel
- @openclaw/discord
- @openclaw/feishu
- @openclaw/lobster
- @openclaw/matrix
- @openclaw/msteams
- @openclaw/nextcloud-talk
- @openclaw/nostr
- @openclaw/voice-call
- @openclaw/zalo

@openclaw/zalouser

Release notes must also call out **new optional bundled plugins** that are **not on by default** (example: `tlon`).

‹ Credits Tests ›