Other install methods

Nix

The recommended way to run OpenClaw with Nix is via **nix-openclaw** — a batteries-included Home Manager module.

Quick Start

Paste this to your AI agent (Claude, Cursor, etc.):

```
I want to set up nix-openclaw on my Mac.
Repository: github:openclaw/nix-openclaw

What I need you to do:
1. Check if Determinate Nix is installed (if not, install it)
2. Create a local flake at ~/code/openclaw-local using templates/agent-first/flake
3. Help me create a Telegram bot (@BotFather) and get my chat ID (@userinfobot)
4. Set up secrets (bot token, Anthropic key) - plain files at ~/.secrets/ is fine
5. Fill in the template placeholders and run home-manager switch
6. Verify: launchd running, bot responds to messages

Reference the nix-openclaw README for module options.
```

> 📦 **Full guide:** The nix-openclaw repo is the source of truth for Nix installation. This page is just a quick overview.

What you get

- Gateway + macOS app + tools (whisper, spotify, cameras) — all pinned ›
- Launchd service that survives reboots
- Plugin system with declarative config
- Instant rollback: `home-manager switch --rollback`

Nix Mode Runtime Behavior

When `OPENCLAW_NIX_MODE=1` is set (automatic with nix-openclaw):

OpenClaw supports a **Nix mode** that makes configuration deterministic and disables auto-install flows. Enable it by exporting:

```
OPENCLAW_NIX_MODE=1
```

On macOS, the GUI app does not automatically inherit shell env vars. You can also enable Nix mode via defaults:

```
defaults write bot.molt.mac openclaw.nixMode -bool true
```

Config + state paths

OpenClaw reads JSON5 config from `OPENCLAW_CONFIG_PATH` and stores mutable data in `OPENCLAW_STATE_DIR`. When needed, you can also set `OPENCLAW_HOME` to control the base home directory used for internal path resolution.

- `OPENCLAW_HOME` (default precedence: `HOME` / `USERPROFILE` / `os.homedir()`)
- `OPENCLAW_STATE_DIR` (default: `~/.openclaw`)

`OPENCLAW_CONFIG_PATH` (default: `$OPENCLAW_STATE_DIR/openclaw.json`)

When running under Nix, set these explicitly to Nix-managed locations so runtime state and config stay out of the immutable store.

Runtime behavior in Nix mode

- Auto-install and self-mutation flows are disabled
- Missing dependencies surface Nix-specific remediation messages
- UI surfaces a read-only Nix mode banner when present

Packaging note (macOS)

The macOS packaging flow expects a stable Info.plist template at:

```
apps/macos/Sources/OpenClaw/Resources/Info.plist
```

copies this template into the app bundle and patches dynamic fields (bundle ID, version/build, Git SHA, Sparkle keys). This keeps the plist deterministic for SwiftPM packaging and Nix builds (which do not rely on a full Xcode toolchain).

Related

— full setup guide

— non-Nix CLI setup

— containerized setup

‹ Podman Ansible ›