Developer setup

Setup

If you are setting up for the first time, start with Getting Started. For wizard details, see Onboarding Wizard.

Last updated: 2026-01-01

TL;DR

- **Tailoring lives outside the repo:** `~/.openclaw/workspace` (workspace) + `~/.openclaw/openclaw.json` (config).
- **Stable workflow:** install the macOS app; let it run the bundled Gateway.
- **Bleeding edge workflow:** run the Gateway yourself via `pnpm gateway:watch` , then let the macOS app attach in Local mode.

Prereqs (from source)

- Node `>=22`
- `pnpm`
- Docker (optional; only for containerized setup/e2e — see Docker)

Tailoring strategy (so updates don't hurt)

If you want “100% tailored to me” *and* easy updates, keep your customization in:

- **Config:** `~/.openclaw/openclaw.json` (JSON/JSON5-ish)
- **Workspace:** `~/.openclaw/workspace` (skills, prompts, memories; make it a private git repo)

Bootstrap once:

```
openclaw setup
```

From inside this repo, use the local CLI entry:

```
openclaw setup
```

If you don’t have a global install yet, run it via `pnpm openclaw setup` .

Run the Gateway from this repo

After `pnpm build` , you can run the packaged CLI directly:

```
node openclaw.mjs gateway --port 18789 --verbose
```

Stable workflow (macOS app first)

1. Install + launch **OpenClaw.app** (menu bar).
2. Complete the onboarding/permissions checklist (TCC prompts).
3. Ensure Gateway is **Local** and running (the app manages it).
4. Link surfaces (example: WhatsApp):

```
openclaw channels login
```

5. Sanity check:

```
openclaw health
```

If onboarding is not available in your build:

> Run `openclaw setup`, then `openclaw channels login`, then start the Gateway manually (`openclaw gateway`).

Bleeding edge workflow (Gateway in a terminal)

Goal: work on the TypeScript Gateway, get hot reload, keep the macOS app UI attached.

0) (Optional) Run the macOS app from source too

If you also want the macOS app on the bleeding edge:

```
./scripts/restart-mac.sh
```

1) Start the dev Gateway

```
pnpm install
pnpm gateway:watch
```

`gateway:watch` runs the gateway in watch mode and reloads on TypeScript changes.

2) Point the macOS app at your running Gateway

In **OpenClaw.app**:

- Connection Mode: **Local** The app will attach to the running gateway on the configured port.

3) Verify

- In-app Gateway status should read **“Using existing gateway …”**
- Or via CLI:

```
openclaw health
```

Common footguns

- **Wrong port:** Gateway WS defaults to `ws://127.0.0.1:18789` ; keep app + CLI on the same port.
- **Where state lives:**
  - Credentials: `~/.openclaw/credentials/`
  - Sessions: `~/.openclaw/agents/<agentId>/sessions/`
  - Logs: `/tmp/openclaw/`

Credential storage map

Use this when debugging auth or deciding what to back up:

- **WhatsApp**: `~/.openclaw/credentials/whatsapp/<accountId>/creds.json`
- **Telegram bot token**: config/env or `channels.telegram.tokenFile`
- **Discord bot token**: config/env (token file not yet supported)
- **Slack tokens**: config/env (`channels.slack.*`)
- **Pairing allowlists**: `~/.openclaw/credentials/<channel>-allowFrom.json`

Model auth profiles: `~/.openclaw/agents/<agentId>/agent/auth-profiles.json`

Legacy OAuth import: `~/.openclaw/credentials/oauth.json` More detail: **Security**.

Updating (without wrecking your setup)

- Keep `~/.openclaw/workspace` and `~/.openclaw/` as "your stuff"; don't put personal prompts/config into the `openclaw` repo.
- Updating source: `git pull` + `pnpm install` (when lockfile changed) + keep using `pnpm gateway:watch` .

Linux (systemd user service)

Linux installs use a systemd **user** service. By default, systemd stops user services on logout/idle, which kills the Gateway. Onboarding attempts to enable lingering for you (may prompt for sudo). If it's still off, run:

```
sudo loginctl enable-linger $USER
```

For always-on or multi-user servers, consider a **system** service instead of a user service (no lingering needed). See for the systemd notes.

Related docs

- (flags, supervision, ports)
- (config schema + examples)
- and (reply tags + replyToMode settings)
- (gateway lifecycle)

‹ Session Management Deep Dive CI Pipeline ›