

[Start Here](#)

Onboarding

This doc describes the **current** first-run onboarding flow. The goal is a smooth “day 0” experience: pick where the Gateway runs, connect auth, run the wizard, and let the agent bootstrap itself.

Page order (current)

1. Welcome + security notice
2. **Gateway selection** (Local / Remote / Configure later)
3. **Auth (Anthropic OAuth)** – local only
4. **Setup Wizard** (Gateway-driven)
5. **Permissions** (TCC prompts)
6. **CLI** (optional)
7. **Onboarding chat** (dedicated session)
8. Ready

1) Local vs Remote

Where does the **Gateway** run?

Local (this Mac): onboarding can run OAuth flows and write credentials locally.

Remote (over SSH/Tailnet): onboarding does **not** run OAuth locally; credentials must exist on the gateway host.

Configure later: skip setup and leave the app unconfigured.

Gateway auth tip:



The wizard now generates a **token** even for loopback, so local WS clients must authenticate.

If you disable auth, any local process can connect; use that only on fully trusted machines.

Use a **token** for multi-machine access or non-loopback binds.

2) Local-only auth (Anthropic OAuth)

The macOS app supports Anthropic OAuth (Claude Pro/Max). The flow:

Opens the browser for OAuth (PKCE)

Asks the user to paste the `code#state` value

Writes credentials to `~/.openclaw/credentials/oauth.json`

Other providers (OpenAI, custom APIs) are configured via environment variables or config files for now.

3) Setup Wizard (Gateway-driven)

The app can run the same setup wizard as the CLI. This keeps onboarding in sync with Gateway-side behavior and avoids duplicating logic in SwiftUI.

4) Permissions

Onboarding requests TCC permissions needed for:

Notifications

Accessibility

Screen Recording

Microphone / Speech Recognition

Automation (AppleScript)

5) CLI (optional)

The app can install the global `openclaw` CLI via `npm/pnpm` so terminal workflows and `launchd` tasks work out of the box.

6) Onboarding chat (dedicated session)

After setup, the app opens a dedicated onboarding chat session so the agent can introduce itself and guide next steps. This keeps first-run guidance separate from your normal conversation.

Agent bootstrap ritual

On the first agent run, OpenClaw bootstraps a workspace (default `~/.openclaw/workspace`):

Seeds `AGENTS.md`, `BOOTSTRAP.md`, `IDENTITY.md`, `USER.md`

Runs a short Q&A ritual (one question at a time)

Writes identity + preferences to `IDENTITY.md`, `USER.md`, `SOUL.md`

Removes `BOOTSTRAP.md` when finished so it only runs once

Optional: Gmail hooks (manual)

Gmail Pub/Sub setup is currently a manual step. Use:

```
openclaw webhooks gmail setup --account you@gmail.com
```



See [/automation/gmail-pubsub](#) for details.

Remote mode notes

When the Gateway runs on another machine, credentials and workspace files live **on that host**. If you need OAuth in remote mode, create:



~/.openclaw/credentials/oauth.json
~/.openclaw/agents/<agentId>/agent/auth-profiles.json

on the gateway host.

< Docs Hubs

OpenClaw Lore >

Powered by [mintlify](#)