

# Best OpenClaw Tools and Extensions in 2026

February 6, 2026 · by Mark Bartlett

[Download this guide as PDF](#)

**Quick Answer:** Start with Crabwalk for live agent monitoring, Tokscale for cost tracking, and openclaw-docker for secure deployment. Add OpenClaw Copilot if you want browser integration. Skip ClawDeck and Claw Desktop unless you're running multiple agents or overnight tasks. The OpenClaw ecosystem has matured fast — the built-in dashboard covers basics, but these community tools fill real gaps in visibility, cost control, and operational security.

 **More on this topic:** [OpenClaw Setup Guide](#) · [OpenClaw Security Guide](#) · [Best Models for OpenClaw](#) · [OpenClaw Token Optimization](#)

OpenClaw's built-in dashboard gives you a chat interface, logs, and skill management at `http://127.0.0.1:18789`. It's functional. It's also the bare minimum for running an AI agent that has access to your shell, your files, and your API keys.

The community has built tools that solve the problems the built-in dashboard ignores: real-time visualization of what your agent is actually doing, cost tracking before your API bill hits triple digits, secure Docker deployments, and browser integration that turns OpenClaw into something you'd actually use daily.

This guide covers the third-party tools worth installing, what each one does, and which to skip. If you're not sold on OpenClaw itself, see our [OpenClaw alternatives comparison](#) — some lighter tools skip these problems entirely.

## Monitoring & Visualization

### Crabwalk

<b>What</b>	Real-time companion monitor with live node graph
<b>By</b>	<a href="#">@lucasveg</a>
<b>GitHub</b>	<a href="#">lucast/crabwalk</a>
<b>Site</b>	<a href="#">crabwalk.app</a>

Crabwalk is the tool most OpenClaw power users install first. It connects to your gateway via WebSocket and renders a live ReactFlow graph of everything your agent does – thinking states, tool calls, response chains, and the relationships between them.

The v1.0.5+ release added subagent support, which matters. When your agent spawns child agents, Crabwalk shows the parent-child hierarchy in real time with collapsible groups in the sidebar. Without this, you're blind to what subagents are doing and why your token count just spiked.

### What it shows you:

- Live activity graph across WhatsApp, Telegram, Discord, and Slack sessions
- Action tracing – expand any node to inspect tool arguments and payloads
- Multi-platform monitoring from a single view
- Subagent trees with status-colored edges

### Install:

```
git clone https://github.com/luccast/crabwalk
cd crabwalk
npm install && npm run dev
```

Point it at your gateway URL and it connects automatically.

**Who needs it:** Anyone running OpenClaw for real tasks. The built-in logs are text streams. Crabwalk turns them into something you can actually follow.

---

## Mission Control

<b>What</b>	Real-time task tracking dashboard
<b>By</b>	<a href="#">manish-raana</a>
<b>GitHub</b>	<a href="#">manish-raana/openclaw-mission-control</a>

Mission Control hooks into OpenClaw's lifecycle events via HTTP POST and syncs them to a Convex backend for real-time UI updates. No polling, no message queues. You get a clean React dashboard showing task state, agent activity, and live logs.

The key difference from Crabwalk: Mission Control focuses on task-level tracking rather than node-level visualization. It's better for answering "what did the agent accomplish?" rather than "what is the agent doing right now?"

### Install:

```
git clone https://github.com/manish-raana/openclaw-mission-control
cd openclaw-mission-control
# Follow README for Convex setup + env configuration
npm install && npm run dev
```

**Who needs it:** Users running multi-step task workflows who want a history of what completed, what failed, and what's still running.

## ClawDeck

<b>What</b>	Open-source agent management dashboard with REST API
<b>Site</b>	<a href="https://clawdeck.io">clawdeck.io</a>

ClawDeck gives you a single pane of glass for multiple OpenClaw agents. If you're running one agent on your home server and another on a VPS, ClawDeck shows both with status, health, and controls.

The REST API is the real draw – it lets you build your own integrations on top of your agent fleet without touching OpenClaw's internals.

**Who needs it:** Multi-agent operators. If you're running a single OpenClaw instance, Crabwalk or Mission Control is enough.

## Browser & Desktop

### OpenClaw Copilot (Chrome Extension)

<b>What</b>	Browser side panel that connects to your OpenClaw gateway
<b>Install</b>	<a href="https://chrome.google.com/webstore/detail/openclaw-copilot">Chrome Web Store</a>

<b>Docs</b>	<a href="https://docs.openclaw.ai/tools/chrome-extension">docs.openclaw.ai/tools/chrome-extension</a>
-------------	---

OpenClaw Copilot puts your agent in Chrome's side panel with direct WebSocket connection to your gateway. The useful part: real browser automation through Chrome DevTools Protocol. Your agent can click, type, navigate, read page content, and capture screenshots.

Security is handled through manual attach/detach – you explicitly grant tab access via the toolbar button. No automatic takeover. No data sent to third-party servers. Everything flows between the extension and your own gateway.

### Setup:

1. Install from Chrome Web Store
2. Click the settings icon in the side panel
3. Enter your gateway URL (default: `ws://127.0.0.1:18789` )
4. Navigate to any page, open the side panel, click Connect

**Who needs it:** Anyone who wants OpenClaw to interact with web pages – filling forms, extracting data, navigating workflows. The side panel is also just a convenient way to chat with your agent while browsing.

**Security note:** This extension has access to whatever tabs you attach. Review the [OpenClaw security guide](#) before giving your agent browser control. Don't attach tabs with banking, email, or credential management pages.

---

## Claw Desktop

<b>What</b>	Native macOS/Windows app for managing OpenClaw agents
<b>Site</b>	<a href="https://claw.so">claw.so</a>
<b>Price</b>	Free

Claw Desktop is a native app that pairs with your OpenClaw gateway for reviewing artifacts, seeing diffs, and resuming runs that started on Slack or the web. The pitch: send tasks that run overnight on dedicated infrastructure, review results in the morning.

It runs on macOS (Apple Silicon and Intel) and Windows.

**Who needs it:** Users running overnight or long-running agent tasks who want a polished native interface for reviewing output. If you're mostly doing interactive chat, the web dashboard or Copilot extension is fine.

---

## Cost & Token Tracking

---

OpenClaw's token consumption is a known pain point. Users regularly report \$10-50 overnight charges from agents left running. The built-in `/status` and `/usage` commands help, but they're per-session and don't give you historical trends or alerts.

For a deeper dive on reducing costs, see our [OpenClaw token optimization guide](#).

## Tokscale

<b>What</b>	CLI token usage tracker with visualization dashboard
<b>By</b>	<a href="#">junhoyeo</a>
<b>GitHub</b>	<a href="#">junhoyeo/tokscale</a>

Tokscale tracks token usage across OpenClaw and other AI coding tools (Claude Code, Codex, Gemini CLI, Cursor). It breaks down input, output, cache read, and cache write tokens with real-time pricing. The Rust core keeps it fast even with large usage histories.

The standout feature: a contributions graph (like GitHub's green squares) showing your AI tool spending over time. Useful for spotting patterns – like which days your agent burned through tokens and why.

### Install:

```
# Follow the GitHub README for platform-specific install
# Supports macOS, Linux, Windows
```

**Who needs it:** Anyone paying for API access. Tokscale catches spending spikes before they become billing surprises.

---

## Clawdbot Cost Monitor

<b>What</b>	Real-time OpenClaw spending tracker with history
<b>By</b>	<a href="#">bokonon23</a>
<b>GitHub</b>	<a href="#">bokonon23/clawdbot-cost-monitor</a>

More focused than Tokscale – this tracks OpenClaw specifically with historical data and real-time cost visibility. Claude’s platform dashboard shows overall API usage across all your projects. This shows what OpenClaw spent, when, and on which tasks.

**Who needs it:** Users who run OpenClaw as their primary API consumer and want granular cost attribution without the multi-tool scope of Tokscale.

## Deployment & Infrastructure

### openclaw-docker

<b>What</b>	Production-ready Docker Compose with security hardening
<b>By</b>	<a href="#">willbullen</a>
<b>GitHub</b>	<a href="#">willbullen/openclaw-docker</a>

OpenClaw’s official Docker support works, but it’s basic. This community project adds the security hardening that a production deployment needs:

- Runs as non-root user (UID 1000)
- Drops unnecessary Linux capabilities
- Prevents privilege escalation ( `no-new-privileges` )
- Read-only root filesystem where possible
- Explicit networks instead of default bridge
- Health checks for auto-restart on failure
- Secrets kept external (not baked into images)

### Setup:

```
git clone https://github.com/willbullen/openclaw-docker
cd openclaw-docker
cp .env.example .env
# Edit .env with your API keys
docker compose up -d
```

The repo includes docs on MCP tool isolation ( `docs/MCP_HARDENING.md` ) and additional security guidance ( `docs/SECURITY.md` ).

**Who needs it:** Anyone running OpenClaw on a server, VPS, or shared machine. The official Docker setup is fine for local testing. For anything exposed to the internet or running unattended, use this instead. Also read our [OpenClaw security guide](#) for the full hardening picture.

---

## OpenTelemetry Observability Plugin

<b>What</b>	Full agent lifecycle tracing with OpenTelemetry
<b>By</b>	<a href="#">henrikrexed</a>
<b>GitHub</b>	<a href="#">henrikrexed/openclaw-observability-plugin</a>

This plugin uses OpenClaw's typed plugin hooks to capture the entire agent lifecycle as hierarchical OpenTelemetry spans. The root span ( `openclaw.request` ) nests agent turns, which nest individual tool operations (file reads, shell commands, web searches).

Ships metrics to any OTLP-compatible backend – Dynatrace, Grafana, Jaeger, or your own collector. OpenClaw v2026.2+ has built-in OpenTelemetry support, but this plugin adds richer span structure and custom attributes.

**Who needs it:** DevOps and platform engineers running OpenClaw in team environments who already have observability infrastructure. Overkill for solo hobbyists.

---

## Skill Discovery

---

### awesome-openclaw-skills

<b>What</b>	Curated directory of 1,700+ community skills
<b>GitHub</b>	<a href="#">VoltAgent/awesome-openclaw-skills</a>

The go-to resource for finding skills beyond the 50+ bundled ones. Skills are categorized (Kubernetes, Confluence, web search, project management, automation) and sourced from ClawHub.

**The caveat:** “curated” doesn’t mean “vetted for security.” These skills are pulled from ClawHub, which has [well-documented security problems](#). Researchers found 341 malicious skills in a single audit – including Atomic Stealer malware and credential theft.

#### How to use safely:

1. Browse the list to find skills that match your needs
2. Read the source code of any skill before installing
3. Check the publisher’s GitHub account age and history
4. Never install skills that request unnecessary permissions
5. Review our [OpenClaw plugins & skills guide](#) for detailed vetting steps

## Comparison Table

---

Tool	Category	Install	Complexity	Best For
<b>Crabwalk</b>	Monitoring	npm	Low	Live agent visualization
<b>Mission Control</b>	Monitoring	npm + Convex	Medium	Task workflow tracking
<b>ClawDeck</b>	Management	Web	Low	Multi-agent fleets
<b>OpenClaw Copilot</b>	Browser	Chrome Store	Low	Browser automation + chat
<b>Claw Desktop</b>	Desktop	Download	Low	Overnight runs, artifact review
<b>Tokscale</b>	Cost	CLI	Low	Multi-tool token tracking
<b>Cost Monitor</b>	Cost	npm	Low	OpenClaw-specific spending

Tool	Category	Install	Complexity	Best For
<b>openclaw-docker</b>	Deployment	Docker	Medium	Secure production hosting
<b>OTel Plugin</b>	Observability	Plugin	High	Enterprise monitoring
<b>awesome-skills</b>	Discovery	Browse	None	Finding community skills

---

## What to Install First

---

If you're just getting started with OpenClaw, don't install everything. Start with this stack:

### Day 1: The essentials

1. **Tokscale** or **Cost Monitor** – Know what you're spending before anything else
2. **Crabwalk** – See what your agent is actually doing

**Week 1: When you're comfortable** 3. **OpenClaw Copilot** – If you want browser integration 4. **openclaw-docker** – If you're moving beyond local testing

**Later: When you need it** 5. **Mission Control** or **ClawDeck** – When you're running multiple agents or complex workflows 6. **OTel Plugin** – When you have observability infrastructure to send data to

The tools you don't need on day one: Claw Desktop (unless you're running overnight tasks), ClawDeck (unless you have multiple agents), and the OTel plugin (unless you have Grafana or Dynatrace already running).

---

## Related Guides

---

- [OpenClaw Setup Guide](#)
- [OpenClaw Security Guide](#)
- [OpenClaw ClawHub Security Alert](#)
- [Best Local Models for OpenClaw](#)
- [OpenClaw Token Optimization](#)
- [OpenClaw Plugins & Skills Guide](#)
- [How OpenClaw Actually Works](#)
- [OpenClaw Mac Setup Guide](#) – Mac-specific installation and optimization

- [Best OpenClaw Alternatives](#) – lighter options if you want less complexity
- 

Sources: [Crabwalk GitHub](#), [Mission Control GitHub](#), [ClawDeck](#), [OpenClaw Copilot Chrome Web Store](#), [Claw Desktop](#), [Tokscale GitHub](#), [Clawdbot Cost Monitor GitHub](#), [openclaw-docker GitHub](#), [OpenClaw Observability Plugin GitHub](#), [awesome-openclaw-skills GitHub](#), [Bitdefender ClawHub Research](#)

Get notified when we publish new guides.

[Subscribe](#) – free, no spam

---

Source: <https://insiderllm.com/guides/best-openclaw-tools-extensions/>

Free guides for running AI locally