

TÀI LIỆU HƯỚNG DẪN SỬ DỤNG

OpenClaw

Trợ lý AI mã nguồn mở chạy trên máy của bạn

The open-source AI assistant that runs on your own machine

Cài đặt · Kênh trò chuyện · Mô hình AI · Plugin

Gateway · Tự động hoá · Bảo mật · Xử lý sự cố

Song ngữ Việt – Anh · Vietnamese – English

Phiên bản 1.0 — Tháng 9 năm 2026

<https://globalmarket.vin>

M U C L U C

CONTENTS

Chương 1. OpenClaw là gì và dùng để làm gì — Chapter 1. What OpenClaw is and what it is for	6
1.1. OpenClaw khác gì các dịch vụ AI trên mạng — <i>How OpenClaw differs from cloud AI services</i>	<i>6</i>
1.2. Những việc OpenClaw làm được — <i>What OpenClaw can do</i>	<i>7</i>
1.3. Ba tên gọi của cùng một phần mềm — <i>Three names for the same software</i>	<i>8</i>
1.4. Nhân vật Molty và cách đọc tài liệu gốc — <i>Molty and how to read the original documentation</i>	<i>9</i>
1.5. Phiên bản, giấy phép và cách dự án được vận hành — <i>Version, licence and how the project is governed</i>	<i>9</i>
1.6. Phần mềm gửi đi những gì về nhà sản xuất — <i>What the software reports back</i>	<i>10</i>
Chương 2. Cài đặt OpenClaw trên máy tính — Chapter 2. Installing OpenClaw.....	11
2.1. Máy cần gì trước khi cài — <i>What your machine needs first.....</i>	<i>11</i>
2.2. Cài trên macOS và Linux — <i>Installing on macOS and Linux</i>	<i>12</i>
2.3. Cài trên Windows — <i>Installing on Windows</i>	<i>13</i>
2.4. Thiết lập ban đầu — <i>First-time setup.....</i>	<i>14</i>
2.5. Kiểm tra sau khi cài — <i>Checking the installation.....</i>	<i>14</i>
2.6. Gỡ cài và cài lại — <i>Uninstalling and reinstalling</i>	<i>15</i>
Chương 3. Bốn lối vào và cách làm quen — Chapter 3. Four ways in, and finding your feet.....	16
3.1. Bảng điều khiển web — <i>The web control panel</i>	<i>16</i>
3.2. Dòng lệnh — <i>The command line.....</i>	<i>17</i>
3.3. Cửa sổ trò chuyện trong terminal — <i>The chat window inside the terminal.....</i>	<i>17</i>
3.4. Ứng dụng nhắn tin — <i>Messaging apps</i>	<i>17</i>
3.5. Mười phút đầu tiên nên làm gì — <i>What to do in your first ten minutes</i>	<i>18</i>
Chương 4. Tập cấu hình và nơi phần mềm cất dữ liệu — Chapter 4. The configuration file and where data is kept.....	19
4.1. Các tệp quan trọng và công dụng — <i>The important files and what they do</i>	<i>20</i>
4.2. Tệp openclaw.json viết như thế nào — <i>How the openclaw.json file is written</i>	<i>20</i>
4.3. Sửa cấu hình mà không mở tệp — <i>Changing the configuration without opening the file.....</i>	<i>21</i>
4.4. Khai khoá bí mật cho đúng cách — <i>Declaring secrets the right way.....</i>	<i>21</i>
4.5. Chạy nhiều bản riêng biệt trên một máy — <i>Running separate instances on one machine</i>	<i>22</i>
Chương 5. Nối Telegram, Zalo và WhatsApp — Chapter 5. Connecting Telegram, Zalo and WhatsApp.....	23

5.1. Nói Telegram — cách dễ nhất để bắt đầu — <i>Connecting Telegram — the easiest way to start</i>	23
5.2. Nói Zalo — ba kênh khác nhau, chọn đúng kênh — <i>Connecting Zalo — three different channels, pick the right one</i>	24
5.3. Nói WhatsApp — <i>Connecting WhatsApp</i>	27
Chương 6. Nói Discord, Slack và các kênh còn lại — <i>Chapter 6. Connecting Discord, Slack and the remaining channels</i>.....	29
6.1. Nói Discord — <i>Connecting Discord</i>	29
6.2. Nói Slack — <i>Connecting Slack</i>	30
6.3. Các kênh còn lại — <i>The remaining channels</i>	30
6.4. Ai được nhắn tin cho trợ lý của bạn — <i>Who may message your assistant</i>	31
Chương 7. Mô hình AI và nhà cung cấp — <i>Chapter 7. AI models and providers</i>	33
7.1. Cách OpenClaw gọi tên mô hình AI: dạng provider/model — <i>How OpenClaw names a model: the provider/model reference</i>	33
7.2. Thứ tự chọn mô hình: mô hình chính, danh sách dự phòng và xoay vòng khoá — <i>Model selection order: primary, fallbacks and credential rotation</i>	34
7.3. Bốn cách khai khoá API cho OpenClaw — <i>Four ways to supply an API key to OpenClaw</i>	37
7.4. Danh sách nhà cung cấp mô hình AI được OpenClaw hỗ trợ sẵn — <i>The model providers OpenClaw supports out of the box</i>	39
7.5. Mô hình phụ: mô hình rẻ, mô hình quyết định, mô hình ảnh và mô hình PDF — <i>Secondary models: utility, decision, image and PDF models</i>	41
7.6. Chạy mô hình AI cục bộ trên máy của bạn với Ollama và LM Studio — <i>Running local AI models with Ollama and LM Studio</i>	43
7.7. Theo dõi hạn mức và chi phí gọi mô hình AI — <i>Tracking provider quota and model cost</i>	45
Chương 8. Agent, phiên làm việc và quyền hạn — <i>Chapter 8. Agents, sessions and permissions</i>	47
8.1. Một agent trong OpenClaw gồm những gì — <i>What one OpenClaw agent contains</i> ..	47
8.2. Các tệp định danh trong không gian làm việc của agent — <i>The identity files inside an agent workspace</i>	49
8.3. Chạy nhiều agent và định tuyến kênh về đúng agent — <i>Running several agents and routing each channel to the right one</i>	50
8.4. Phiên làm việc: khoá phiên, phạm vi, khởi tạo lại và nén ngữ cảnh — <i>Sessions: keys, scope, resets and context compaction</i>	52
8.5. Bốn chế độ quyền của phiên làm việc — <i>The four session permission modes</i>	55
8.6. Bộ công cụ theo hồ sơ: minimal, messaging, coding, full — <i>Tool profiles: minimal, messaging, coding and full</i>	56
8.7. Nhịp tim: cho trợ lý chủ động lên tiếng — <i>Heartbeat: letting the assistant speak first</i>	57
8.8. Bộ chạy agent: runtime nhúng, Codex, Claude CLI, Gemini CLI và Copilot — <i>Agent runtimes: the embedded runtime, Codex, Claude CLI, Gemini CLI and Copilot</i>	59

Chương 9. Công cụ và kỹ năng — Chapter 9. Tools and skills	61
9.1. Công cụ exec: agent chạy lệnh thật trên máy của bạn — <i>The exec tool: the agent runs real commands on your machine</i>	61
9.2. Chính sách công cụ: cho phép và cấm — <i>Tool policy: allow and deny.....</i>	63
9.3. Kỹ năng là gì và tệp SKILL.md viết ra sao — <i>What a skill is and how SKILL.md is written.....</i>	64
9.4. Các trường khai báo trong SKILL.md — <i>The frontmatter keys of SKILL.md</i>	66
9.5. Thứ tự nạp kỹ năng và mức ưu tiên — <i>Skill loading order and precedence</i>	67
9.6. Lệnh quản lý kỹ năng và kho ClawHub — <i>Managing skills from the command line and from ClawHub</i>	68
9.7. Lệnh gạch chéo dùng trong khung chat — <i>Slash commands you can type in the chat</i>	70
Chương 10. Phần mở rộng và kho ClawHub — Chapter 10. Plugins and the ClawHub registry.....	75
10.1. Plugin OpenClaw là gì và có những loại nào — <i>What an OpenClaw plugin is, and the kinds that exist</i>	75
10.2. Cách cài plugin OpenClaw bằng lệnh <code>openclaw plugins install</code> — <i>Installing an OpenClaw plugin with openclaw plugins install.....</i>	76
10.3. Kiểm tra tương thích plugin: <code>pluginApi</code> và <code>minGatewayVersion</code> — <i>Checking plugin compatibility: pluginApi and minGatewayVersion</i>	79
10.4. ClawHub là gì — kho đăng ký kỹ năng và plugin của OpenClaw — <i>What ClawHub is: the OpenClaw skill and plugin registry.....</i>	80
10.5. Dùng CLI <code>clawhub</code> để đăng nhập, tìm và công bố — <i>Using the clawhub CLI to log in, search and publish</i>	81
10.6. Cách tự viết một kỹ năng và công bố lên ClawHub — <i>Writing your own skill and publishing it to ClawHub</i>	83
10.7. Biến môi trường ClawHub và cách tắt telemetry — <i>ClawHub environment variables and turning telemetry off.....</i>	85
Chương 11. Tự động hoá — Chapter 11. Automation.....	86
11.1. Năm cơ chế tự động hoá trong OpenClaw và chọn cái nào — <i>The five OpenClaw automation mechanisms and which to choose</i>	86
11.2. Tác vụ hẹn giờ với lệnh <code>openclaw cron</code> — <i>Scheduled jobs with the openclaw cron command</i>	89
11.3. Nhịp tim heartbeat — tác vụ giám sát nền mặc định 30 phút — <i>Heartbeat: the background monitor that runs every 30 minutes.....</i>	92
11.4. Tác vụ nền <code>openclaw tasks</code> và luồng tác vụ <code>tasks flow</code> — <i>Background tasks with openclaw tasks and flows with tasks flow</i>	93
11.5. Móc sự kiện hooks và chỉ thị thường trực — <i>Event hooks and standing orders</i>	94
11.6. Webhook HTTP gọi vào gateway, trigger Gmail và email IMAP — <i>Inbound HTTP webhooks, Gmail triggers and IMAP email.....</i>	96
11.7. Ba ví dụ tự động hoá dùng được ngay — <i>Three automations you can use immediately</i>	98
Chương 12. Vận hành gateway — Chapter 12. Running the gateway	101

12.1. Gateway OpenClaw là gì và nó mở những cổng nào — <i>What the OpenClaw gateway is and what it exposes</i>	101
12.2. Khởi động, dừng và cài gateway làm dịch vụ nền — <i>Starting, stopping and installing the gateway as a background service</i>	102
12.3. Chạy gateway như dịch vụ trên macOS, Linux và Windows — <i>Running the gateway as a service on macOS, Linux and Windows</i>	105
12.4. Đọc nhật ký gateway OpenClaw — <i>Reading the OpenClaw gateway log</i>	107
12.5. Truy cập gateway từ xa qua Tailscale hoặc đường hầm SSH — <i>Remote gateway access over Tailscale or an SSH tunnel</i>	108
12.6. Chạy nhiều gateway trên cùng một máy — <i>Running several gateways on one machine</i>	110
12.7. Bộ lệnh vận hành gateway hằng ngày — <i>The everyday gateway operations command set</i>	113
12.8. Kênh cập nhật, sao lưu và quản nhiều gateway bằng fleet — <i>Release channels, backups and managing several gateways with fleet</i>	114

Chương 13. Bảo mật và ranh giới tin cậy — *Chapter 13. Security and the trust*

boundary	118
13.1. Hiểu đúng rủi ro — <i>13.1. Getting the risk right</i>	118
13.2. Một gateway là một ranh giới tin cậy — <i>13.2. One gateway is one trust boundary</i>	119
13.3. Các mặc định an toàn có sẵn — <i>13.3. The safe defaults you already have</i>	120
13.4. Sáu lớp phân quyền xếp chồng — <i>13.4. Six permission layers, stacked</i>	121
13.5. Sandbox — <i>13.5. The sandbox</i>	125
13.6. Xác thực gateway — <i>13.6. Gateway authentication</i>	127
13.7. Tiêm lệnh qua nội dung ngoài — <i>13.7. Prompt injection through external content</i>	129
13.8. Bí mật nằm trên đĩa — <i>13.8. Secrets on disk</i>	130
13.9. Danh mục kiểm tra trước khi cho người khác nhắn tin — <i>13.9. The checklist before letting other people message your assistant</i>	132

Chương 14. Xử lý sự cố — *Chapter 14. Troubleshooting*..... **135**

14.1. Thang lệnh sáu mươi giây đầu — <i>14.1. The first sixty seconds</i>	135
14.2. Chín nhánh của cây quyết định — <i>14.2. The nine branches of the decision tree</i> ...	136
14.3. Chữ ký nhật ký và nguyên nhân — <i>14.3. Log signatures and their causes</i>	140
14.4. Lỗi lúc cài đặt — <i>14.4. Installation errors</i>	143
14.5. Cấu hình sai làm gateway không khởi động — <i>14.5. A bad configuration stops the Gateway</i>	144
14.6. Sự cố theo từng kênh — <i>14.6. Channel-specific failures</i>	146
14.7. Cổng bị chiếm và nhiều bản gateway cùng chạy — <i>14.7. A taken port and duplicate gateways</i>	147
14.8. Khi nào nên gỡ cài và cài lại — <i>14.8. When to uninstall and reinstall</i>	148

Phụ lục A. Tra cứu lệnh — *Appendix A. Command reference*..... **150**

A.1. Cờ toàn cục — <i>A.1. Global flags</i>	150
A.2. Mười lăm nhóm lệnh — <i>A.2. The fifteen command groups</i>	151
A.3. Lệnh hay dùng theo đầu việc — <i>A.3. Everyday commands by task</i>	152

A.4. Lệnh gạch chéo trong cửa sổ chat — <i>A.4. Slash commands in the chat window</i>	160
A.5. Quy ước --json — <i>A.5. The --json convention</i>	161
Phụ lục B. Thuật ngữ Anh – Việt — <i>Appendix B. English – Vietnamese glossary</i>	163
B.1. Từ A đến C — <i>B.1. A to C</i>	163
B.2. Từ D đến L — <i>B.2. D to L</i>	165
B.3. Từ M đến R — <i>B.3. M to R</i>	166
B.4. Từ S đến Z — <i>B.4. S to Z</i>	168

Chương 1. OpenClaw là gì và dùng để làm gì — Chapter 1. What OpenClaw is and what it is for

OpenClaw là một trợ lý AI chạy trên chính máy tính của bạn. Thay vì mở trình duyệt vào một trang web để hỏi, bạn nhắn tin cho nó qua Telegram, Zalo, WhatsApp hay Slack — những ứng dụng bạn vẫn dùng hằng ngày. Trợ lý nhận tin, làm việc trên máy bạn, rồi nhắn lại kết quả.

OpenClaw is an AI assistant that runs on your own computer. Instead of opening a website to ask a question, you message it through Telegram, Zalo, WhatsApp or Slack — the apps you already use every day. The assistant receives the message, does the work on your machine, then messages the result back.

Tài liệu chính thức gọi nó là một cổng trung chuyển tự vận hành: phần mềm nối các ứng dụng nhắn tin với một tác nhân AI. Phần mềm mã nguồn mở, giấy phép MIT, do OpenClaw Foundation — một tổ chức phi lợi nhuận 501(c)(3) độc lập — bảo trợ và ký phát hành.

The official documentation describes it as a self-hosted gateway: software that connects messaging apps to an AI agent. It is open source under the MIT licence, stewarded and release-signed by the OpenClaw Foundation, an independent 501(c)(3) non-profit.



Hình 1. Tin nhắn đi từ ứng dụng chat qua gateway, tới agent chạy trên máy bạn

Figure 1. A message travels from the chat app through the gateway to the agent on your machine

1.1. OpenClaw khác gì các dịch vụ AI trên mạng — How OpenClaw differs from cloud AI services

Khác biệt nằm ở chỗ phần mềm chạy ở đâu và nó được phép động vào những gì. Một dịch vụ AI trên mạng chỉ đọc được thứ bạn dán vào ô chat. OpenClaw chạy trên máy bạn nên nó mở được tệp trong ổ đĩa, chạy lệnh, gửi thư, và nhắn tin thay bạn — trong phạm vi bạn cho phép.

The difference is where the software runs and what it is allowed to touch. A cloud AI service can only read what you paste into the chat box. OpenClaw runs on your machine, so it can open files on disk, run commands, send email and message people on your behalf — within the limits you set.

Tiêu chí <i>Aspect</i>	Dịch vụ AI trên mạng <i>Cloud AI service</i>	OpenClaw
Nơi chạy <i>Where it runs</i>	Máy chủ của nhà cung cấp <i>The provider's servers</i>	Máy tính của bạn <i>Your own computer</i>
Dữ liệu đi đâu <i>Where your data goes</i>	Gửi lên máy chủ, theo chính sách của họ <i>Uploaded to their servers under their policy</i>	Ở lại máy bạn; chỉ câu hỏi gửi tới mô hình, và có thể dùng mô hình cục bộ để không gửi gì <i>Stays on your machine; only the prompt goes to the model, and a local model sends nothing</i>
Đọc được tệp trong máy <i>Can read local files</i>	Không, trừ khi bạn tải lên từng tệp <i>No, unless you upload each file</i>	Có, trong thư mục bạn cho phép <i>Yes, within the folders you allow</i>
Chạy được lệnh trên máy <i>Can run commands</i>	Không <i>No</i>	Có, theo chính sách công cụ bạn đặt <i>Yes, under the tool policy you set</i>
Nhắn tin thay bạn <i>Can message people for you</i>	Không <i>No</i>	Có, qua kênh bạn đã nối <i>Yes, through the channels you connected</i>
Chi phí <i>Cost</i>	Trả theo thuê bao của dịch vụ <i>A subscription to the service</i>	Phần mềm miễn phí; trả tiền cho mô hình AI bạn chọn <i>The software is free; you pay for the AI model you choose</i>

GHI CHÚ / NOTE — OpenClaw không tự nó biết suy nghĩ. Nó cần một mô hình AI đứng sau — của OpenAI, Anthropic, Google, hoặc một mô hình chạy ngay trên máy bạn. Chương 7 nói kỹ phần này.

OpenClaw does not think on its own. It needs an AI model behind it — from OpenAI, Anthropic, Google, or a model running locally on your machine. Chapter 7 covers this.

1.2. Những việc OpenClaw làm được — *What OpenClaw can do*

Dưới đây là những việc người dùng giao cho trợ lý nhiều nhất. Mỗi việc đều thực hiện bằng cách nhắn một câu tiếng Việt bình thường, không cần nhớ cú pháp.

These are the tasks people hand to the assistant most often. Each one is done by sending an ordinary sentence, with no syntax to memorise.

- Đọc và sắp xếp tệp trong máy: tìm hợp đồng ký tháng trước, gom ảnh công trình theo ngày, đổi tên hàng loạt tệp quét.

Reading and organising local files: find last month's signed contract, group site photos by date, batch-rename scanned files.

- Tóm tắt tài liệu dài: đọc một tệp PDF trăm trang rồi trả lời câu hỏi về nội dung trong đó.
Summarising long documents: read a hundred-page PDF, then answer questions about it.
- Theo dõi hộp thư: sáng nào cũng tóm tắt thư đến trong đêm và chỉ báo việc cần trả lời.
Watching your inbox: each morning, summarise overnight mail and flag only what needs a reply.
- Nhắc việc theo lịch: đặt lịch nhắc, nhấn đúng giờ vào cuộc trò chuyện bạn đang dùng.
Scheduled reminders: set a reminder and get the message at the right time in the conversation you already use.
- Chạy việc lặp đi lặp lại: mỗi chiều thứ sáu lập báo cáo tuần từ nhật ký công việc.
Running recurring jobs: every Friday afternoon, build the weekly report from your work log.
- Làm việc với mã nguồn: đọc một kho mã, sửa lỗi, chạy thử, báo kết quả.
Working with source code: read a repository, fix a bug, run the tests, report back.

CẢNH BÁO / WARNING — Trợ lý có quyền tới đâu là do bạn đặt. Mặc định nó chạy lệnh ngay trên máy thật, không có hộp cát. Đọc Chương 13 trước khi cho người khác nhắn tin cho trợ lý của bạn.

How much the assistant may do is up to you. By default it runs commands directly on the host, with no sandbox. Read Chapter 13 before letting anyone else message your assistant.

1.3. Ba tên gọi của cùng một phần mềm — *Three names for the same software*

Tìm tài liệu về OpenClaw trên mạng, bạn sẽ gặp hai tên cũ. Đây là cùng một dự án, đổi tên hai lần trong hai tháng đầu năm 2026. Bài viết mang tên cũ vẫn đúng về ý tưởng nhưng sai về tên lệnh và đường dẫn tệp.

Searching for OpenClaw material online, you will meet two older names. It is the same project, renamed twice in the first two months of 2026. Articles using the old names are still right about the ideas but wrong about command names and file paths.



Hình 2. Dự án đổi tên hai lần: Clawd, rồi Moltbot, rồi OpenClaw

Figure 2. The project was renamed twice: Clawd, then Moltbot, then OpenClaw

Tên đầu tiên là Clawd, sống trong phần mềm tên Clawdbot từ ngày 25 tháng 11 năm 2025. Tháng 1 năm 2026, Anthropic gửi thư yêu cầu đổi tên vì trùng nhãn hiệu Claude. Ngày 27 tháng 1, cộng đồng bình chọn tên Moltbot. Ba ngày sau, ngày 30 tháng 1, dự án chốt tên OpenClaw và chuyển toàn bộ kho mã, tên miền tài liệu, gói phát hành trong vòng ba giờ.

The first name was Clawd, living inside software called Clawdbot from 25 November 2025. In January 2026 Anthropic wrote asking for a rename over the Claude trademark. On 27 January the community voted for Moltbot. Three days later, on 30 January, the project settled on

OpenClaw and moved the repository, the documentation domain and the published packages within three hours.

MẸO / TIP — Khi tra cứu, thêm chữ OpenClaw vào từ khoá và bỏ qua bài viết trước tháng 2 năm 2026. Tên lệnh cũ `clawdbot` và `moltbot` không còn chạy.

When searching, add the word OpenClaw to your query and skip articles written before February 2026. The old commands `clawdbot` and `moltbot` no longer work.

1.4. Nhân vật Molty và cách đọc tài liệu gốc — *Molty and how to read the original documentation*

Tài liệu chính thức có giọng văn đùa quanh hình con tôm hùm. Molty là tên nhân vật trợ lý — một con tôm hùm vũ trụ. Các từ như claw, molt, ClawHub đều xoay quanh hình ảnh đó. Biết vậy để khỏi mất thì giờ tìm ý nghĩa kỹ thuật trong những chữ chỉ là tên riêng.

The official documentation has a running joke about lobsters. Molty is the name of the assistant character, a space lobster. Words like claw, molt and ClawHub all come from that image. Knowing this saves you from hunting for technical meaning in what are only names.

Tên riêng <i>Name</i>	Nghĩa trong tài liệu <i>What it means in the documentation</i>
Molty	Nhân vật trợ lý mà phần mềm được viết cho <i>The assistant character the software was written for</i>
Gateway	Tiến trình luôn chạy, điều phối mọi thứ — phần cốt lõi của phần mềm <i>The always-on process that coordinates everything — the core of the software</i>
ClawHub	Kho công khai chứa kỹ năng và phần mở rộng, ở địa chỉ clawhub.ai <i>The public registry of skills and plugins, at clawhub.ai</i>
Claw	Gói thiết lập có đánh số phiên bản để dựng một agent mới <i>A versioned setup bundle for creating a new agent</i>
Moltiverse	Cộng đồng người dùng quanh dự án <i>The user community around the project</i>

1.5. Phiên bản, giấy phép và cách dự án được vận hành — *Version, licence and how the project is governed*

Phiên bản đánh số theo năm và tháng. Bản mới nhất khi tài liệu này biên soạn là 2026.9.5 — bản phát hành thứ năm của tháng 9 năm 2026, ra ngày 19 tháng 9. Cách đánh số này cho biết ngay bản bạn đang chạy cũ bao nhiêu tháng.

Versions are numbered by year and month. The latest release as this guide was written is 2026.9.5 — the fifth release of September 2026, published on 19 September. The scheme tells you at a glance how many months old your build is.

Mục <i>Item</i>	Giá trị <i>Value</i>
Phiên bản mới nhất <i>Latest version</i>	2026.9.5 (19/9/2026)
Cách đánh số <i>Versioning</i>	Năm.tháng.số thứ tự trong tháng <i>Year.month.sequence</i>
Giấy phép <i>Licence</i>	MIT — dùng tự do, kể cả cho việc kinh doanh <i>MIT — free to use, including commercially</i>
Bản quyền <i>Copyright</i>	OpenClaw Foundation, 2026
Kho mã <i>Source code</i>	github.com/openclaw/openclaw
Tài liệu gốc <i>Documentation</i>	docs.openclaw.ai
Kho kỹ năng <i>Skill registry</i>	clawhub.ai

Quỹ OpenClaw Foundation trả lương cho nhóm lõi và ký các bản phát hành. Các nhà tài trợ đóng góp tiền và hạ tầng nhưng không sở hữu và không điều khiển dự án; tài liệu ghi rõ OpenAI là nhà tài trợ, không phải chủ sở hữu.

The OpenClaw Foundation employs the core team and signs releases. Sponsors contribute money and infrastructure but neither own nor direct the project; the documentation states plainly that OpenAI is a donor, not an owner.

1.6. Phần mềm gửi đi những gì về nhà sản xuất — *What the software reports back*

Mặc định phần mềm chỉ gọi về một lần mỗi ngày để kiểm tra bản cập nhật. Lần gọi đó mang theo phiên bản OpenClaw, tên hệ điều hành, phiên bản Node và kiến trúc vi xử lý. Thống kê tính năng ẩn danh mặc định tắt.

By default the software phones home once a day to check for updates. That check carries the OpenClaw version, the operating system, the Node version and the CPU architecture. Anonymous feature statistics are off by default.

Nội dung hội thoại, tin nhắn, tên mô hình, khoá API, đường dẫn tệp và định danh máy không bao giờ được gửi đi. Muốn tắt hẳn cả lần kiểm tra cập nhật, đặt một khoá trong tệp cấu hình.

Conversation content, messages, model names, API keys, file paths and machine identifiers are never sent. To switch off even the update check, set one key in the configuration file.

```
// ~/.openclaw/openclaw.json
{ update: { checkOnStart: false } }
```

Chương 2. Cài đặt OpenClaw trên máy tính — Chapter 2. Installing OpenClaw

2.1. Máy cần gì trước khi cài — What your machine needs first

Phần mềm chạy trên Node.js. Cần bản 24.16 trở lên, hoặc bản 26.1 trở lên; tài liệu khuyến nghị dùng Node 26. Nếu máy chưa có Node, trình cài đặt tự cài giúp: bản 26 trên macOS, bản 24 LTS trên Linux.

The software runs on Node.js. You need version 24.16 or later, or 26.1 or later; the documentation recommends Node 26. If your machine has no Node, the installer provides it: version 26 on macOS, version 24 LTS on Linux.

Hệ điều hành <i>Operating system</i>	Cách cài <i>How to install</i>	Ghi chú <i>Note</i>
macOS	Lệnh một dòng trong Terminal, hoặc tải tệp .dmg <i>One line in Terminal, or download the .dmg</i>	Chạy tốt trên cả chip Apple và Intel <i>Runs on both Apple silicon and Intel</i>
Linux	Lệnh một dòng trong terminal <i>One line in the terminal</i>	Cài được cả trên máy chủ không màn hình <i>Installs on headless servers too</i>
Windows	Ứng dụng Windows Hub, hoặc lệnh PowerShell <i>The Windows Hub app, or a PowerShell command</i>	Người dùng thông thường nên chọn Windows Hub <i>Ordinary users should choose Windows Hub</i>
Windows qua WSL2 <i>Windows via WSL2</i>	Dùng đúng lệnh của Linux <i>Use the Linux command</i>	Hợp với người đã quen WSL2 <i>Suits those already using WSL2</i>
Raspberry Pi	Lệnh của Linux <i>The Linux command</i>	Tối thiểu 1 GB RAM, 500 MB đĩa trống, hệ điều hành 64-bit <i>At least 1 GB RAM, 500 MB free disk, a 64-bit OS</i>

GHI CHÚ / NOTE — Tài liệu chính thức không công bố mức RAM và dung lượng đĩa tối thiểu cho máy để bàn. Hai con số duy nhất có trong tài liệu: 6 GB RAM khi tự dựng ảnh Docker từ mã nguồn, và mức của Raspberry Pi nêu trong bảng trên.

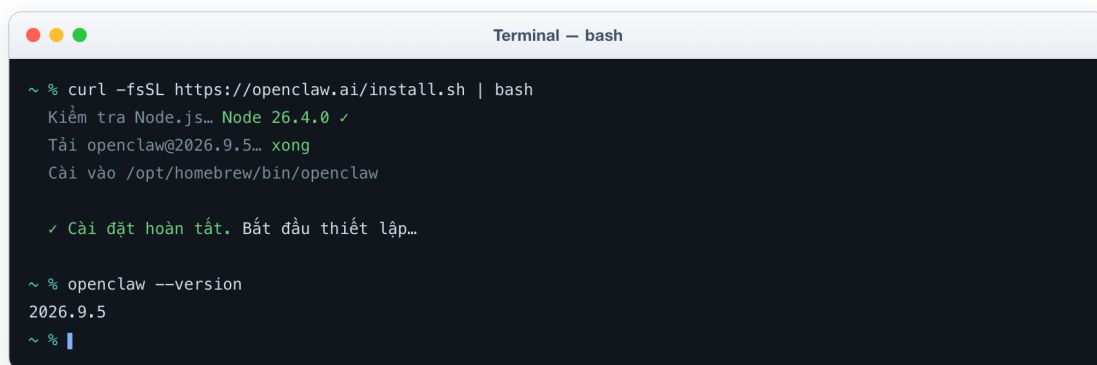
The official documentation does not publish minimum RAM or disk figures for desktop machines. The only two numbers given are 6 GB RAM for building a Docker image from source, and the Raspberry Pi figures in the table above.

2.2. Cài trên macOS và Linux — *Installing on macOS and Linux*

Mở Terminal và chạy một dòng. Trình cài đặt tải phần mềm, kiểm tra Node, rồi mở luôn trợ lý thiết lập ban đầu.

Open Terminal and run a single line. The installer downloads the software, checks Node, then opens the setup wizard.

```
curl -fsSL https://openclaw.ai/install.sh | bash
```



Hình 3. Cài đặt bằng một dòng lệnh trên macOS, mất khoảng một phút

Figure 3. A one-line install on macOS, taking about a minute

Muốn cài xong mà chưa thiết lập ngay thì thêm tham số bỏ qua trợ lý. Cách này hợp khi bạn cài sẵn cho người khác dùng, hoặc cài trên máy chủ để cấu hình sau.

To install without running setup immediately, add the flag that skips the wizard. This suits installing on someone else's behalf, or on a server you will configure later.

```
curl -fsSL https://openclaw.ai/install.sh | bash -s -- --no-onboard
```

Cài bằng npm hoặc trình quản lý gói khác — *Installing with npm or another package manager*

Máy đã có Node thì cài thẳng bằng npm. Từ npm 12, trình quản lý gói chặn các kịch bản chạy sau khi cài, nên phải thêm tham số cho phép; npm 11.15 trở về trước thì bỏ tham số đó đi.

If your machine already has Node, install straight from npm. Since npm 12 the package manager blocks post-install scripts, so the allow flag is required; on npm 11.15 and earlier, drop it.

```
npm install -g openclaw@latest --allow-scripts=openclaw  
openclaw onboard --install-daemon
```

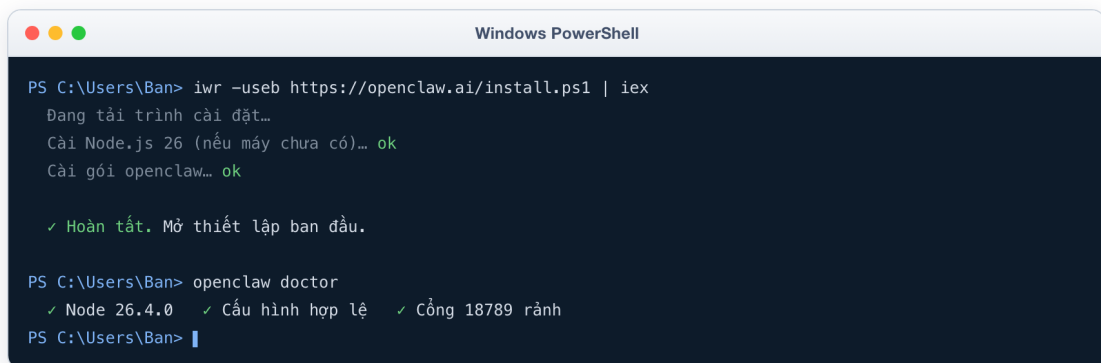
Cách cài <i>Method</i>	Lệnh <i>Command</i>	Khi nào dùng <i>When to use</i>
Trình cài đặt <i>Installer</i>	<code>curl ... install.sh bash</code>	Mặc định, hợp với đa số <i>The default, suits most people</i>
npm	<code>npm install -g openclaw@latest</code>	Máy đã có Node và bạn quen npm <i>You already have Node and know npm</i>
pnpm	<code>pnpm add -g --allow-build=openclaw openclaw@latest</code>	Dự án của bạn đã dùng pnpm <i>Your projects already use pnpm</i>
Thử nhanh <i>Quick try</i>	<code>npx openclaw@latest</code>	Chạy thử một lần, không cài vào máy <i>Run once without installing</i>
Docker	<code>ghcr.io/openclaw/openclaw:latest</code>	Chạy tách biệt khỏi hệ điều hành <i>Run isolated from the host system</i>
Từ mã nguồn <i>From source</i>	<code>git clone ... && pnpm install && pnpm build</code>	Muốn sửa mã hoặc theo nhánh mới nhất <i>You want to modify the code or follow the latest branch</i>

2.3. Cài trên Windows — Installing on Windows

Windows có ba đường. Đơn giản nhất là ứng dụng Windows Hub: tải tệp cài đặt rồi chạy như mọi phần mềm khác. Người quen dòng lệnh chạy một lệnh PowerShell. Người đã dùng WSL2 thì cài trong WSL2 bằng đúng lệnh của Linux.

Windows offers three routes. The simplest is the Windows Hub app: download the installer and run it like any other program. Those comfortable with the command line run a single PowerShell command. Those already using WSL2 install inside WSL2 with the Linux command.

```
iwr -useb https://openclaw.ai/install.ps1 | iex
```



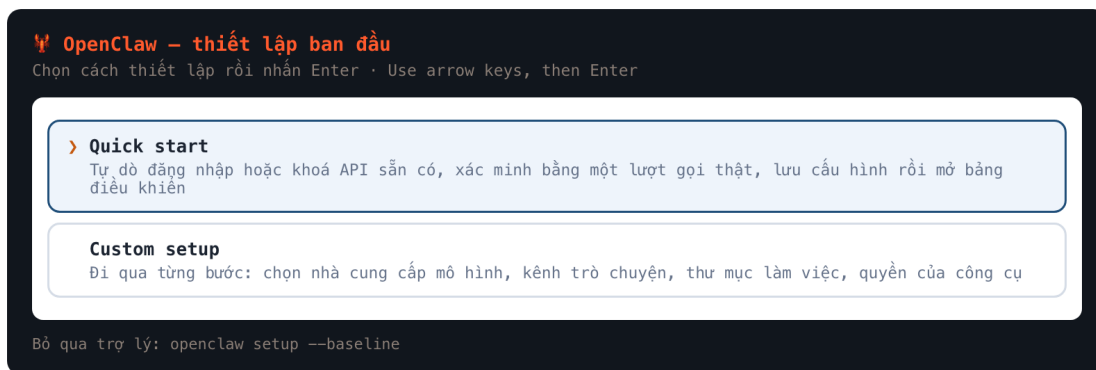
Hình 4. Cài đặt bằng PowerShell trên Windows rồi kiểm tra ngay bằng lệnh doctor

Figure 4. Installing with PowerShell on Windows, then checking straight away with the doctor command

2.4. Thiết lập ban đầu — First-time setup

Cài xong, trợ lý thiết lập tự mở. Có hai lối: Quick start tự dò xem máy đã có sẵn đăng nhập hay khoá API của nhà cung cấp mô hình nào, xác minh bằng một lượt gọi thật, lưu cấu hình rồi mở bảng điều khiển. Custom setup dẫn bạn qua từng bước.

After installation the setup wizard opens by itself. There are two routes: Quick start detects any model-provider login or API key already on the machine, verifies it with a real call, saves the configuration and opens the dashboard. Custom setup walks you through every step.



Hình 5. Màn hình đầu tiên của trợ lý thiết lập: chọn Quick start hoặc Custom setup

Figure 5. The first screen of the setup wizard: choose Quick start or Custom setup

1. Chọn Quick start nếu máy đã có sẵn đăng nhập của một nhà cung cấp mô hình. Chọn Custom setup nếu bạn muốn tự tay chọn từng thứ.
Choose Quick start if the machine already has a model provider login. Choose Custom setup if you want to pick everything yourself.
2. Khai nơi lấy trí tuệ cho trợ lý: khoá API của OpenAI, Anthropic, Google, hoặc một mô hình chạy cục bộ. Chương 7 nói kỹ từng lựa chọn.
Declare where the intelligence comes from: an API key from OpenAI, Anthropic or Google, or a locally running model. Chapter 7 covers each option.
3. Nối kênh trò chuyện đầu tiên. Tài liệu chính thức khuyên bắt đầu bằng Telegram vì chỉ cần một mã bot, không phải cài thêm gì.
Connect your first channel. The official documentation recommends starting with Telegram because it needs only a bot token and no extra installation.
4. Quick start giữ gateway chạy ngay trong cửa sổ terminal. Nhấn Ctrl+C để dừng, rồi cài nó thành dịch vụ chạy nền.
Quick start keeps the gateway running in the terminal window. Press Ctrl+C to stop it, then install it as a background service.

```
openclaw gateway install
```

2.5. Kiểm tra sau khi cài — Checking the installation

Chạy lần lượt bốn lệnh sau. Tất cả đều chỉ đọc, không đổi gì trong máy.

Run these four commands in order. All of them only read; none change anything.

```
openclaw --version      # phiên bản đang cài
openclaw doctor         # soát cấu hình và môi trường
openclaw gateway status # gateway có đang chạy không
openclaw dashboard      # mở bảng điều khiển trong trình duyệt
```

Lệnh <i>Command</i>	Kết quả đúng là thế nào <i>What a healthy result looks like</i>
openclaw --version	In ra số phiên bản, ví dụ 2026.9.5 <i>Prints a version number such as 2026.9.5</i>
openclaw doctor	Không có dòng lỗi chặn <i>No blocking errors reported</i>
openclaw gateway status	Runtime hiện running, dòng probe hiện ok <i>Runtime shows running and the probe line shows ok</i>
openclaw dashboard	Trình duyệt mở bảng điều khiển ở cổng 18789 <i>The browser opens the dashboard on port 18789</i>

CẢNH BÁO / WARNING — Gặp báo lỗi `openclaw: command not found` thì gần như chắc chắn là chuyện đường dẫn: thư mục chứa lệnh chưa nằm trong biến PATH. Mục 14.4 hướng dẫn xử lý.

If you see `openclaw: command not found`, it is almost always a PATH problem: the folder holding the command is not in your PATH. Section 14.4 explains the fix.

2.6. Gỡ cài và cài lại — *Uninstalling and reinstalling*

Gỡ phần mềm bằng lệnh gỡ có sẵn. Trước khi gỡ, sao lưu lại cấu hình và dữ liệu phiên nếu bạn định cài lại — thư mục trạng thái nằm riêng nên gỡ phần mềm không đụng tới nó.

Remove the software with the built-in uninstall command. Back up your configuration and session data first if you plan to reinstall — the state folder is separate, so uninstalling does not touch it.

```
openclaw backup create      # sao lưu trước
openclaw gateway stop --disable
openclaw uninstall
```

Chương 3. Bốn lối vào và cách làm quen — Chapter 3. Four ways in, and finding your feet

Cùng một trợ lý, bốn cách nói chuyện với nó. Cả bốn dùng chung một phiên làm việc và một bộ nhớ, nên bạn hỏi trong Telegram rồi hỏi tiếp trong cửa sổ dòng lệnh, nó vẫn nhớ câu trước.

One assistant, four ways to talk to it. All four share one session and one memory, so you can ask something in Telegram, continue in the command line, and it still remembers.



Hình 6. Bốn lối vào cùng dẫn tới một gateway, dùng chung phiên làm việc và bộ nhớ

Figure 6. Four entry points to the same gateway, sharing one session and one memory

3.1. Bảng điều khiển web — The web control panel

Bảng điều khiển là trang web do chính gateway phục vụ, mở ở địa chỉ máy nhà tại cổng 18789. Đây là nơi dễ nhìn nhất để xem trợ lý đang làm gì, kênh nào đang nối, mô hình nào đang dùng.

The control panel is a web page served by the gateway itself, at your local address on port 18789. It is the easiest place to see what the assistant is doing, which channels are connected and which model is in use.

```
openclaw dashboard # mở sẵn trong trình duyệt
# hoặc gõ thẳng địa chỉ: http://127.0.0.1:18789
```



Hình 7. Bảng điều khiển web: danh sách phiên bên trái, cửa sổ trò chuyện ở giữa

Figure 7. The web control panel: sessions on the left, the conversation in the middle

3.2. Dòng lệnh — The command line

Mọi việc quản trị đều làm được bằng lệnh `openclaw`. Cách đặt tham số theo quy tắc quen thuộc: tùy chọn riêng của lệnh đặt sau tên lệnh, tùy chọn chung đặt trước.

Everything administrative can be done with the `openclaw` command. Options follow a familiar rule: command-specific options go after the command name, global options before it.

```
openclaw status --json          # tùy chọn riêng đặt sau
openclaw --profile thu status   # tùy chọn chung đặt trước
```

MẸO / TIP — Không nhớ tên lệnh con thì thêm `--help` vào sau bất kỳ lệnh nào. Phụ lục A liệt kê đủ các nhóm lệnh để tra nhanh.

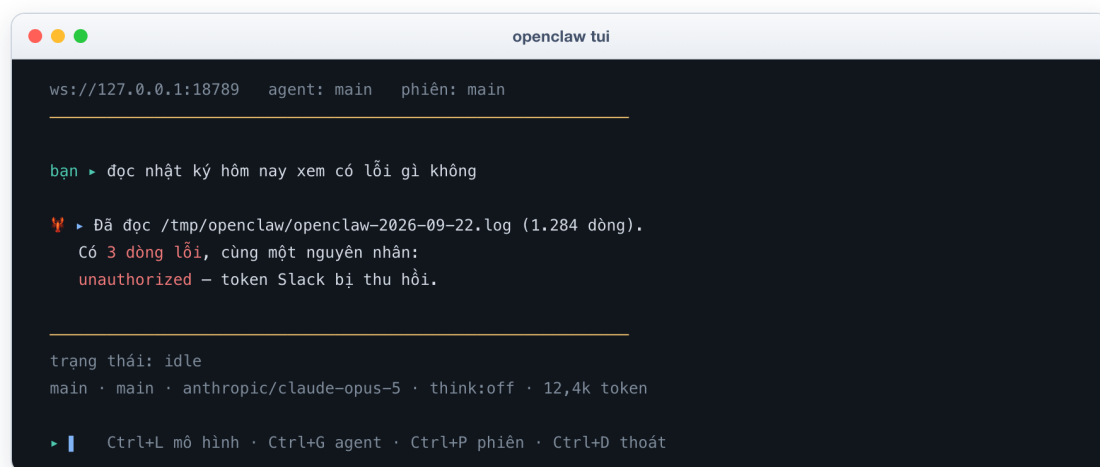
If you cannot remember a sub-command, add `--help` to any command. Appendix A lists every command group for quick reference.

3.3. Cửa sổ trò chuyện trong terminal — The chat window inside the terminal

Muốn nói chuyện với trợ lý mà không rời terminal thì mở giao diện chat dạng cửa sổ. Nó hiện phiên đang dùng, mô hình đang chạy, và nhận cả lệnh gạch chéo.

To talk to the assistant without leaving the terminal, open the windowed chat interface. It shows the current session, the model in use, and accepts slash commands.

```
openclaw tui          # hoặc viết tắt: openclaw chat
```



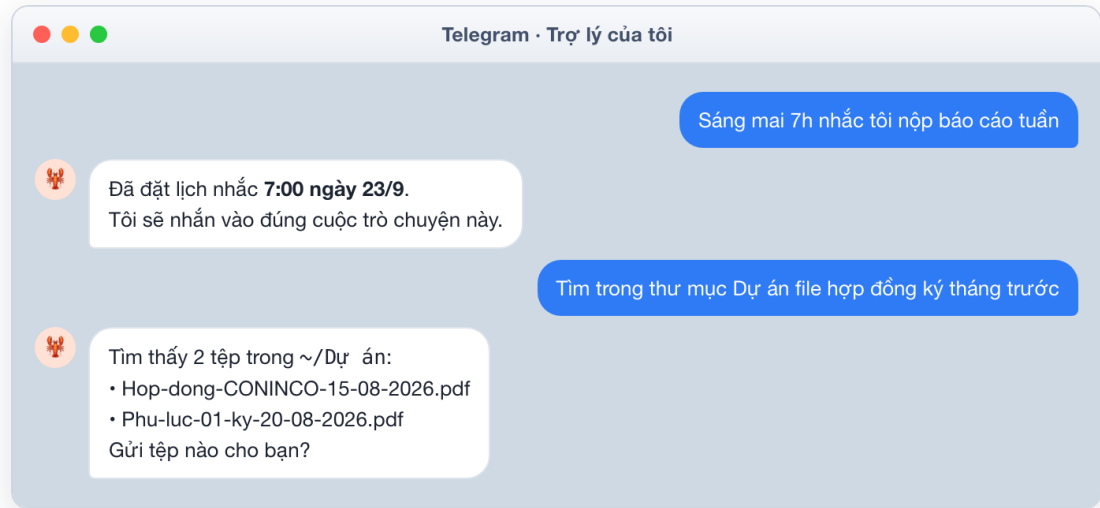
Hình 8. Cửa sổ trò chuyện chạy ngay trong terminal, hiện phiên và mô hình đang dùng

Figure 8. The chat window running inside the terminal, showing the session and the model in use

3.4. Ứng dụng nhắn tin — Messaging apps

Đây là lối vào được dùng nhiều nhất, vì nó cho phép giao việc từ điện thoại, ở bất cứ đâu. Máy ở nhà vẫn chạy, bạn ngồi ngoài công trường vẫn nhắn được. Chương 5 và 6 hướng dẫn nối từng kênh.

This is the most used entry point, because it lets you hand over work from your phone, anywhere. The machine at home keeps running while you message from the site. Chapters 5 and 6 cover connecting each channel.



Hình 9. Giao việc cho trợ lý qua Telegram, ngay trên điện thoại

Figure 9. Handing work to the assistant over Telegram, straight from your phone

3.5. Mười phút đầu tiên nên làm gì — What to do in your first ten minutes

1. Nhấn một câu hỏi đơn giản để chắc chắn đường dây thông: hỏi hôm nay là ngày mấy.
Send a simple question to confirm the line works: ask what today's date is.
2. Giao một việc chỉ đọc, không sửa gì: bảo nó liệt kê các tệp trong một thư mục.
Give it a read-only task: ask it to list the files in a folder.
3. Xem trợ lý tự giới thiệu năng lực: hỏi nó đang dùng mô hình nào và có những công cụ gì.
Let it introduce itself: ask which model it is using and what tools it has.
4. Thử một lệnh gạch chéo: gõ /status để xem trạng thái phiên và hạn mức còn lại.
Try a slash command: type /status to see the session state and remaining quota.
5. Đọc Chương 13 trước khi giao việc có sửa đổi tệp hoặc trước khi cho người khác nhắn tin cho trợ lý.
Read Chapter 13 before giving it tasks that modify files, or before letting anyone else message the assistant.

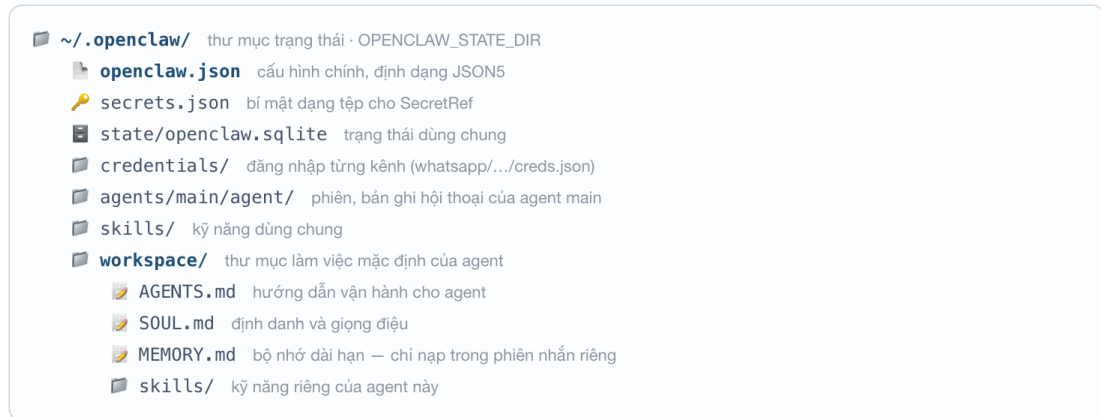
GHI CHÚ / NOTE — Trợ lý trả lời bằng tiếng Việt nếu bạn hỏi bằng tiếng Việt. Riêng trợ lý thiết lập ban đầu chỉ có tiếng Anh và hai bản tiếng Trung, chưa có tiếng Việt.

The assistant answers in Vietnamese if you write in Vietnamese. The setup wizard itself, however, exists only in English and two Chinese variants; Vietnamese is not available.

Chương 4. Tập cấu hình và nơi phần mềm cất dữ liệu — Chapter 4. The configuration file and where data is kept

Mọi thứ OpenClaw nhớ đều nằm trong một thư mục duy nhất: `~/.openclaw`. Biết thư mục này chứa gì thì bạn sao lưu được, chuyển máy được, và biết chỗ nào tuyệt đối không được chia sẻ.

Everything OpenClaw remembers lives in a single folder: `~/.openclaw`. Knowing what it holds lets you back it up, move it to another machine, and know which parts must never be shared.



Nhật ký nằm riêng: `/tmp/openclaw/openclaw-YYYY-MM-DD.log`

Hình 10. Thư mục `~/.openclaw` và thư mục làm việc của agent

Figure 10. The `~/.openclaw` folder and the agent workspace

4.1. Các tệp quan trọng và công dụng — *The important files and what they do*

Đường dẫn <i>Path</i>	Chứa gì <i>What it holds</i>	Chia sẻ được không <i>Shareable</i>
~/.openclaw/openclaw.json	Cấu hình chính <i>The main configuration</i>	Chỉ khi đã xóa hết khoá bí mật <i>Only after removing every secret</i>
~/.openclaw/secrets.json	Khoá API và bí mật dạng tệp <i>API keys and file-based secrets</i>	Không bao giờ <i>Never</i>
~/.openclaw/credentials/	Đăng nhập từng kênh <i>Per-channel logins</i>	Không bao giờ <i>Never</i>
~/.openclaw/state/openclaw.sqlite	Trạng thái dùng chung <i>Shared state</i>	Không <i>No</i>
~/.openclaw/agents/<id>/	Phiên và bản ghi hội thoại <i>Sessions and transcripts</i>	Không <i>No</i>
~/.openclaw/workspace/	Thư mục làm việc của agent <i>The agent workspace</i>	Được, nếu bạn chủ động kiểm tra nội dung <i>Yes, if you check the contents first</i>
/tmp/openclaw/openclaw-<ngày>.log	Nhật ký theo ngày <i>Daily log files</i>	Cân nhắc — nhật ký có thể lộ thông tin <i>Think twice — logs can leak information</i>

CẢNH BÁO / WARNING — Đừng đưa thư mục ~/.openclaw lên kho mã nguồn hay dịch vụ đồng bộ. Nó chứa khoá API và phiên đăng nhập của mọi kênh bạn đã nối.
Never commit ~/.openclaw to a code repository or a sync service. It holds API keys and the login sessions of every channel you connected.

4.2. Tệp openclaw.json viết như thế nào — *How the openclaw.json file is written*

Tệp cấu hình theo định dạng JSON5 — giống JSON nhưng cho phép ghi chú và dấu phẩy thừa ở cuối danh sách. Hai đặc điểm đó khiến tệp dễ đọc và dễ sửa hơn JSON thuần.

The configuration file uses JSON5 — like JSON but allowing comments and trailing commas. Those two differences make the file easier to read and edit than plain JSON.



Hình 11. Một tập cấu hình đủ dùng: thư mục làm việc, mô hình chính, hai kênh và cổng gateway

Figure 11. A working configuration: workspace, primary model, two channels and the gateway port

Cấu hình chia làm hai nhóm. Nhóm gốc là hạ tầng: gateway, kênh, mô hình, phần mở rộng, công cụ, phiên, nhật ký. Nhóm `agents.defaults` là cách agent làm việc: thư mục làm việc, mô hình, nhịp tim, hộp cát, thời gian chờ.

The configuration has two buckets. The root holds infrastructure: gateway, channels, models, plugins, tools, sessions, logging. The `agents.defaults` bucket holds agent behaviour: workspace, model, heartbeat, sandbox, timeouts.

4.3. Sửa cấu hình mà không mở tệp — *Changing the configuration without opening the file*

Sửa tay dễ gõ nhầm dấu ngoặc. Dùng lệnh thì phần mềm tự kiểm tra trước khi ghi.

Hand-editing invites a missing bracket. The commands validate before writing.

```
openclaw config get gateway.port
openclaw config set gateway.port 18789
openclaw config validate                # soát toàn bộ tệp
openclaw config patch --file ./sua.json5 --dry-run  # thử trước khi áp
```

CẢNH BÁO / WARNING — Cấu hình sai khoá hoặc sai kiểu dữ liệu làm gateway TỪ CHỐI khởi động. Khi đó chỉ còn bốn lệnh chạy được: `doctor`, `logs`, `health`, `status`. Chạy `openclaw doctor --fix` để sửa.

A wrong key or wrong data type makes the gateway REFUSE to start. Only four commands still work: `doctor`, `logs`, `health`, `status`. Run `openclaw doctor --fix` to repair it.

4.4. Khai khoá bí mật cho đúng cách — *Declaring secrets the right way*

Đừng ghi thẳng khoá API vào tệp cấu hình. Khai một tham chiếu, phần mềm sẽ đi lấy giá trị thật lúc chạy. Bốn nguồn lấy được: biến môi trường, tệp riêng, lệnh bên ngoài, kho khoá.

Do not write API keys directly into the configuration. Declare a reference and the software fetches the real value at run time. Four sources are supported: environment variables, a separate file, an external command, or a key store.

```
{ source: "env", provider: "default", id: "OPENAI_API_KEY" }
```

Nguồn <i>Source</i>	Lấy khoá từ đâu <i>Where the value comes from</i>
env	Biến môi trường của tiến trình gateway <i>An environment variable of the gateway process</i>
file	Tệp ~/.openclaw/secrets.json <i>The ~/.openclaw/secrets.json file</i>
exec	Kết quả của một lệnh bên ngoài <i>The output of an external command</i>
store	Kho khoá của hệ điều hành <i>The operating system key store</i>

4.5. Chạy nhiều bản riêng biệt trên một máy — *Running separate instances on one machine*

Muốn tách hẳn một bản thử nghiệm khỏi bản đang dùng thật, thêm tham số hồ sơ. Mỗi hồ sơ có thư mục trạng thái riêng, cấu hình riêng, không đụng nhau.

To keep a test instance fully separate from your working one, add a profile flag. Each profile has its own state folder and configuration, and they never touch.

```
openclaw --profile thu gateway      # trạng thái ở ~/.openclaw-thu
openclaw --dev gateway              # bản phát triển, cổng 19001
```

Chương 5. Nối Telegram, Zalo và WhatsApp — Chapter 5.

Connecting Telegram, Zalo and WhatsApp

OpenClaw nối được 32 kênh trò chuyện. Ba kênh trong chương này là ba kênh người Việt dùng nhiều nhất. Tài liệu chính thức khuyên bắt đầu bằng Telegram: chỉ cần một mã bot, không phải cài thêm gì, và không đụng tới tài khoản cá nhân của bạn.

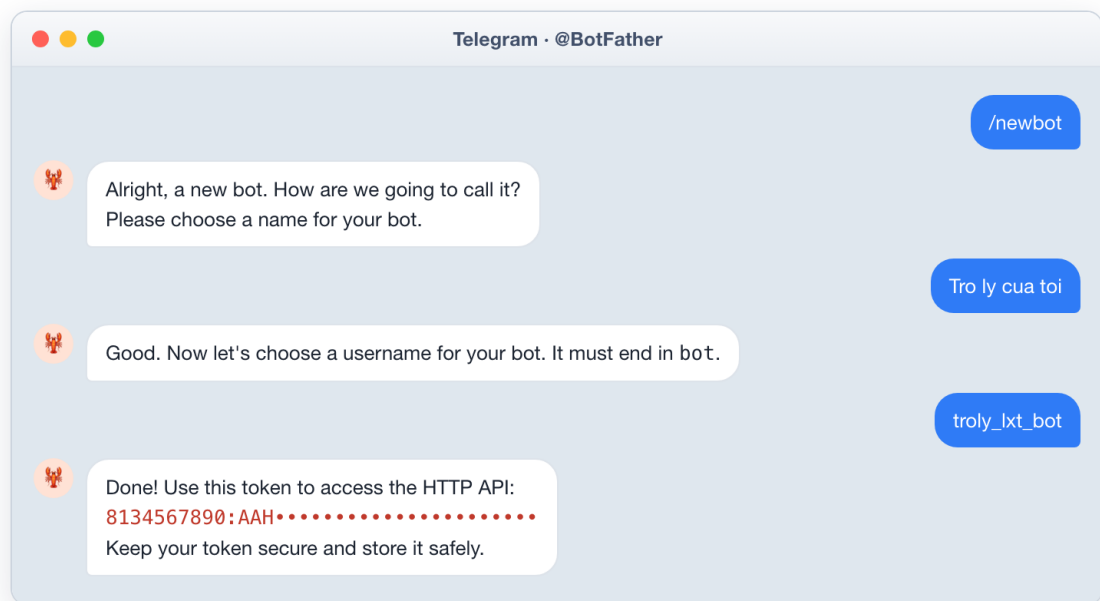
OpenClaw connects to 32 chat channels. The three in this chapter are the ones most used in Vietnam. The official documentation recommends starting with Telegram: it needs only a bot token, no extra installation, and it does not touch your personal account.

5.1. Nối Telegram — cách dễ nhất để bắt đầu — Connecting Telegram — the easiest way to start

Telegram nằm sẵn trong bản cài, không phải cài thêm phần mở rộng. Việc duy nhất phải làm là tạo một con bot và lấy mã của nó.

Telegram ships with the installation; no plugin to add. The only task is creating a bot and collecting its token.

1. Mở Telegram, tìm tài khoản @BotFather — chú ý viết đúng tên, có nhiều tài khoản giả mạo tên gần giống.
Open Telegram and find the account @BotFather — check the spelling; many fake accounts use similar names.
2. Gõ /newbot, đặt tên hiển thị, rồi đặt tên người dùng kết thúc bằng chữ bot.
Send /newbot, give a display name, then a username ending in the word bot.
3. BotFather trả về một chuỗi mã. Giữ kín chuỗi này như giữ mật khẩu.
BotFather returns a token string. Keep it as private as a password.



Hình 12. Tạo bot Telegram và nhận mã: toàn bộ việc gói trong ba câu nhắn

Figure 12. Creating a Telegram bot and receiving the token: the whole job in three messages

4. Nạp mã vào OpenClaw bằng một lệnh.

Load the token into OpenClaw with one command.

```
openclaw channels add --channel telegram --token <mã-bot>
openclaw channels status --probe
```

5. Nhấn cho bot một câu bất kỳ. Lần đầu nó không trả lời mà sinh ra một mã ghép đôi — đây là hàng rào chặn người lạ. Duyệt mã đó rồi bot mới nói chuyện.

Message the bot anything. The first time it does not reply but issues a pairing code — the barrier that keeps strangers out. Approve the code and the bot starts talking.

```
openclaw pairing list telegram
openclaw pairing approve telegram <Mã>
```

CẢNH BÁO / WARNING — Mã ghép đôi hết hạn sau một giờ. Quá hạn thì nhấn lại cho bot để lấy mã mới.

The pairing code expires after one hour. If it lapses, message the bot again for a new one.

Dùng bot trong nhóm Telegram — *Using the bot in a Telegram group*

Bot Telegram mặc định bật chế độ riêng tư, nghĩa là trong nhóm nó chỉ thấy tin nhắn có nhắc tên nó. Muốn nó đọc mọi tin thì tắt chế độ đó ở BotFather bằng lệnh /setprivacy, hoặc cho bot làm quản trị viên nhóm. Đôi xong phải xoá bot khỏi nhóm rồi thêm lại.

Telegram bots have privacy mode on by default, so in a group they only see messages that mention them. To let it read everything, turn privacy off in BotFather with /setprivacy, or make the bot a group administrator. After changing it, remove the bot from the group and add it again.

5.2. Nối Zalo — ba kênh khác nhau, chọn đúng kênh — *Connecting Zalo — three different channels, pick the right one*

Zalo có tới ba cách nối, tên gần giống nhau nhưng bản chất khác hẳn. Chọn nhầm là mất công làm lại, nặng hơn là mất tài khoản.

Zalo offers three routes with similar names but very different natures. Picking the wrong one costs time at best, and your account at worst.

Kênh <i>Channel</i>	Bản chất <i>What it is</i>	Nên dùng khi <i>Use it when</i>
zalo	Bot chính thức tạo trên nền tảng bot của Zalo, có sẵn trong bản cài <i>An official bot created on Zalo's bot platform, bundled with the installation</i>	Bạn muốn một con bot riêng, không đụng tài khoản cá nhân <i>You want a separate bot that never touches your personal account</i>
zalouser	Tự động hoá chính tài khoản Zalo cá nhân, KHÔNG chính thức <i>Automation of your own personal Zalo account, UNOFFICIAL</i>	Chấp nhận rủi ro bị khoá tài khoản để trợ lý trả lời ngay trong Zalo của bạn <i>You accept the risk of a ban so the assistant answers inside your own Zalo</i>
zaloclawbot	Bot cá nhân gắn với chủ, đăng nhập bằng quét mã QR trong Mini App <i>An owner-bound personal bot, logged in by scanning a QR code in a Mini App</i>	Muốn nhanh gọn và chỉ mình bạn nhắn cho bot <i>You want something quick where only you message the bot</i>

CẢNH BÁO / WARNING — Tài liệu chính thức ghi nguyên văn về kênh zalouser: đây là tích hợp không chính thức và có thể dẫn tới việc tài khoản bị đình chỉ hoặc bị cấm. Cần nhắc kỹ trước khi nối tài khoản Zalo cá nhân đang dùng cho công việc.

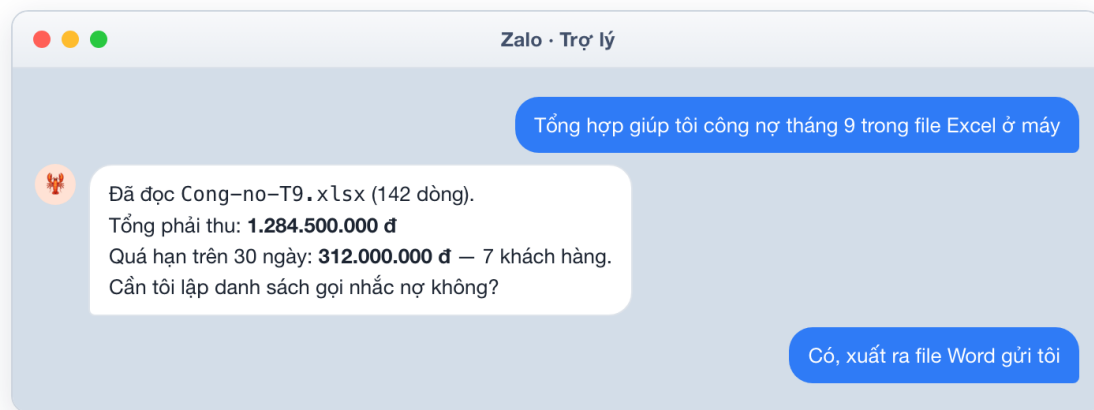
The official documentation states plainly about the zalouser channel: this is an unofficial integration and may result in account suspension or ban. Think carefully before connecting the personal Zalo account you use for work.

Cách 1: bot Zalo chính thức — *Route 1: the official Zalo bot*

- Vào trang bot.zaloplatforms.com, đăng nhập rồi tạo một bot mới.
Go to bot.zaloplatforms.com, sign in and create a new bot.
- Lấy mã bot. Mã có dạng một dãy số, dấu hai chấm, rồi phần bí mật.
Collect the bot token. It looks like a number, a colon, then a secret part.
- Khai mã vào cấu hình rồi kiểm tra kết nối.
Put the token into the configuration, then check the connection.

```
{ channels: { zalo: { enabled: true, accounts: { default: {
  botToken: "12345689:abc-xyz", dmPolicy: "pairing" } } } }
```

```
openclaw channels status --probe
openclaw pairing list zalo
openclaw pairing approve zalo <MÃ>
```



Hình 13. Trợ lý đọc bảng tính trên máy rồi trả lời ngay trong Zalo

Figure 13. The assistant reads a spreadsheet on the machine and answers inside Zalo

Kênh Zalo đang ở trạng thái thử nghiệm và có những giới hạn do nền tảng Zalo đặt ra, không phải do OpenClaw. Biết trước để khỏi mất công tìm cách chỉnh.

The Zalo channel is experimental and carries limits set by the Zalo platform itself, not by OpenClaw. Knowing them saves you hunting for settings that do not exist.

Giới hạn <i>Limit</i>	Giá trị <i>Value</i>
Độ dài một tin nhắn <i>Length of one message</i>	2.000 ký tự, dài hơn sẽ bị cắt thành nhiều tin <i>2,000 characters; longer text is split</i>
Dung lượng tệp gửi kèm <i>Attachment size</i>	5 MB theo mặc định <i>5 MB by default</i>
Trong nhóm <i>In groups</i>	Bắt buộc nhắc tên bot mới kích hoạt, không tắt được <i>Mentioning the bot is required and cannot be turned off</i>
Không hỗ trợ <i>Not supported</i>	Thả cảm xúc, luồng trả lời, bình chọn, lệnh gốc <i>Reactions, threads, polls, native commands</i>
Danh sách cho phép <i>Allowlist</i>	Chỉ nhận mã số người dùng Zalo, không nhận tên <i>Accepts numeric Zalo user IDs only, not names</i>

Cách 2: nối tài khoản Zalo cá nhân — *Route 2: connecting a personal Zalo account*

Cách này cho trợ lý trả lời ngay trong Zalo cá nhân của bạn, không cần bot. Đổi lại, đây là tích hợp không chính thức: Zalo không cấp phép cho nó, và tài khoản có thể bị khoá.

This route lets the assistant answer inside your own Zalo without a bot. In exchange, it is an unofficial integration: Zalo does not sanction it, and the account can be banned.

```
openclaw plugins install @openclaw/zalouser
openclaw channels login --channel zalouser # hiện mã QR, quét bằng Zalo
```

Tìm mã số người dùng và mã nhóm để điền vào danh sách cho phép bằng nhóm lệnh tra danh bạ. Luôn dùng mã số, đừng dùng tên — tên chỉ tra được khi bật một tùy chọn rủi ro.

Find user and group IDs for your allowlists with the directory commands. Always use the numeric IDs, never names — name matching only works behind a risky option.

```
openclaw directory self --channel zalouser
openclaw directory peers list --channel zalouser --query "tên"
openclaw directory groups list --channel zalouser --query "công ty"
```

CẢNH BÁO / WARNING — Với kênh này, `groupAllowFrom` không tự lấy theo `allowFrom`. Để trống `groupAllowFrom` trong một nhóm đã cho phép nghĩa là mở nhóm đó cho bất kỳ ai nhắn vào.

On this channel, `groupAllowFrom` does not fall back to `allowFrom`. Leaving it empty on an allowlisted group opens that group to any sender.

Cách 3: Zalo ClawBot — *Route 3: Zalo ClawBot*

Cách nhanh nhất: chạy trợ lý thiết lập, chọn Zalo ClawBot trong danh sách kênh, quét mã QR hiện ra trong terminal, đồng ý điều khoản trong Mini App. Bot sinh ra gắn với đúng tài khoản Zalo của bạn và chỉ nói chuyện với bạn; tin nhắn của người khác bị chặn ngay ở nền tảng.

The quickest route: run the setup wizard, choose Zalo ClawBot from the channel list, scan the QR code shown in the terminal and accept the terms in the Mini App. The bot created is bound to your Zalo account and talks only to you; messages from anyone else are dropped at the platform level.

```
openclaw plugins install "@zalo-platforms/openclaw-zaloclawbot@0.1.4"
openclaw plugins enable openclaw-zaloclawbot
openclaw channels login --channel openclaw-zaloclawbot
```

GHI CHÚ / NOTE — Mã QR đăng nhập hết hạn sau 5 phút. Quá hạn thì chạy lại lệnh đăng nhập.

The login QR code expires after 5 minutes. If it lapses, run the login command again.

5.3. Nói WhatsApp — *Connecting WhatsApp*

WhatsApp nói qua WhatsApp Web, nghĩa là điện thoại của bạn phải có tài khoản đó và bạn quét mã QR để liên kết thiết bị, giống như khi dùng WhatsApp trên máy tính.

WhatsApp connects through WhatsApp Web, so the account must exist on a phone and you link the device by scanning a QR code, exactly as you would to use WhatsApp on a computer.

1. Cài phần mở rộng WhatsApp.

Install the WhatsApp plugin.

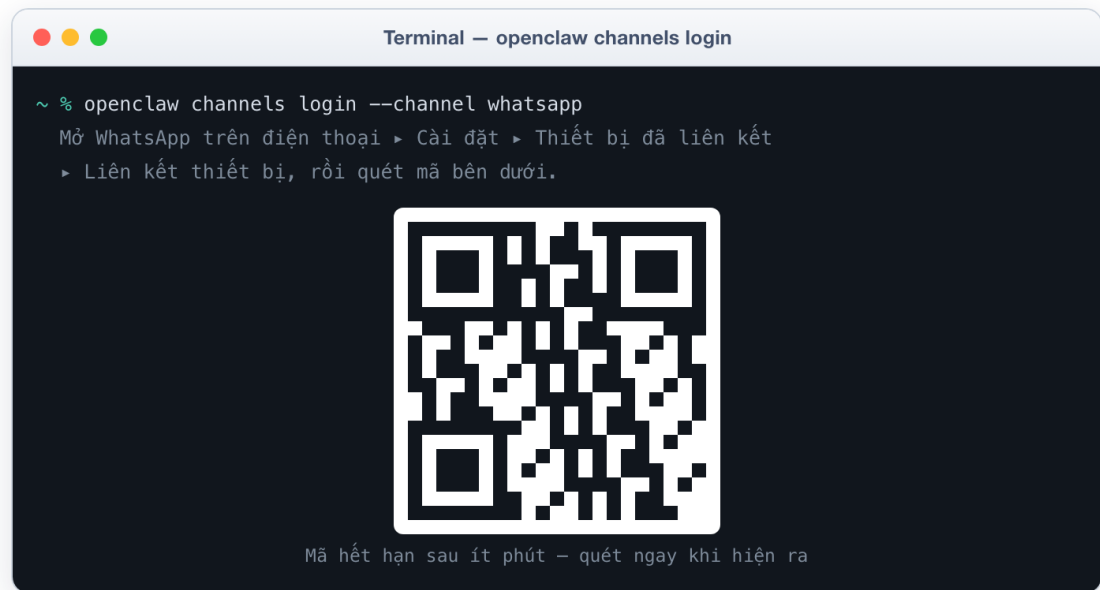
2. Khai danh sách số được phép nhắn cho trợ lý TRƯỚC khi đăng nhập.

Declare which numbers may message the assistant BEFORE logging in.

3. Chạy lệnh đăng nhập rồi quét mã QR bằng điện thoại.

Run the login command and scan the QR code with your phone.

```
openclaw plugins install @openclaw/whatsapp
openclaw channels login --channel whatsapp
```



Hình 14. Quét mã QR để liên kết WhatsApp với gateway (mã trong hình chỉ để minh họa)

Figure 14. Scanning the QR code to link WhatsApp to the gateway (the code shown is for illustration)

CẢNH BÁO / WARNING — Tài liệu chính thức khuyên dùng một số điện thoại RIÊNG cho trợ lý. Nới số cá nhân đồng nghĩa mọi tin nhắn gửi cho bạn đều trở thành dữ liệu đầu vào của agent.

The official documentation recommends a SEPARATE phone number for the assistant. Linking your personal number means every message sent to you becomes agent input.

Trên máy chủ không màn hình, chuẩn bị sẵn đường đưa mã QR tới điện thoại trước khi chạy lệnh. Mã hết hạn nhanh, chụp màn hình rồi gửi qua chat thường là không kịp.

On a headless server, arrange a way to get the QR code to your phone before running the command. The code expires quickly, and screenshotting then sending it over chat is usually too slow.

Chương 6. Nối Discord, Slack và các kênh còn lại — Chapter 6.

Connecting Discord, Slack and the remaining channels

6.1. Nối Discord — Connecting Discord

1. Vào Discord Developer Portal, tạo một ứng dụng mới, rồi mở mục Bot và đặt tên.
Open the Discord Developer Portal, create a new application, then open the Bot section and set a name.
2. Bật các quyền đặc biệt. Thiếu quyền đọc nội dung tin nhắn thì bot câm trong kênh chung.
Turn on the privileged intents. Without the message content intent the bot stays mute in shared channels.



Hình 15. Bật quyền đặc biệt cho bot Discord: thiếu quyền đọc nội dung là bot không nghe thấy gì

Figure 15. Enabling privileged intents for a Discord bot: without message content it hears nothing

3. Bấm Reset Token để lấy mã bot — lần đầu bấm chính là lần sinh mã.
Press Reset Token to get the bot token — the first press is what generates it.
4. Sang mục OAuth2, chọn URL Generator, đánh dấu phạm vi bot và applications.commands, rồi chọn các quyền tối thiểu: xem kênh, gửi tin, đọc lịch sử, nhúng liên kết, đính kèm tệp.
Go to OAuth2, open URL Generator, tick the bot and applications.commands scopes, then the minimum permissions: view channels, send messages, read message history, embed links, attach files.
5. Mở đường dẫn sinh ra, chọn máy chủ để thêm bot vào.
Open the generated URL and choose the server to add the bot to.
6. Khai mã bot qua biến môi trường, đừng ghi thẳng vào tệp cấu hình.
Declare the token through an environment variable rather than writing it into the configuration file.

```
export DISCORD_BOT_TOKEN="mã-bot-của-bạn"
openclaw config patch --file ./discord.patch.json5 --dry-run
openclaw config patch --file ./discord.patch.json5
openclaw gateway restart
```

GHI CHÚ / NOTE — Mỗi kênh Discord là một phiên làm việc riêng. Tệp MEMORY.md chỉ tự nạp trong phiên nhắn riêng, không nạp trong kênh của máy chủ.

Each Discord channel is its own session. The MEMORY.md file loads automatically only in direct-message sessions, not in server channels.

6.2. Nối Slack — Connecting Slack

Slack cần hai mã khác nhau, lấy ở hai chỗ khác nhau. Nhầm chỗ là lỗi hay gặp nhất khi nối kênh này.

Slack needs two different tokens from two different places. Mixing them up is the most common mistake when connecting this channel.

1. Cài phần mở rộng Slack.

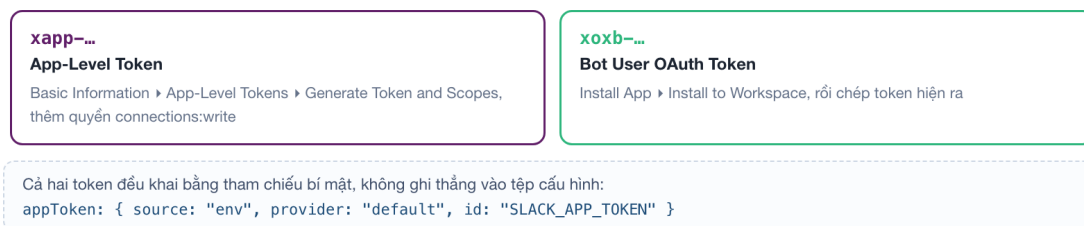
Install the Slack plugin.

2. Tạo ứng dụng Slack từ bản khai mẫu mà tài liệu chính thức cung cấp — cách này đặt sẵn toàn bộ quyền cần thiết, khỏi tích tay từng mục.

Create the Slack app from the manifest the official documentation provides — this sets every required scope at once instead of ticking them by hand.

3. Lấy hai mã theo đúng hai chỗ trong hình dưới.

Collect the two tokens from the two places shown below.



Hình 16. Slack có hai mã: một mã cấp ứng dụng và một mã của bot, lấy ở hai trang khác nhau

Figure 16. Slack has two tokens — one app-level, one bot — taken from two different pages

```
openclaw plugins install @openclaw/slack
openclaw gateway restart
openclaw channels status --probe
```

6.3. Các kênh còn lại — The remaining channels

Hai mươi chín kênh còn lại cài theo cùng một khuôn: cài phần mở rộng, khai thông tin đăng nhập, kiểm tra, rồi duyệt mã ghép đôi. Bảng dưới nêu điểm riêng của những kênh hay dùng.

The other twenty-nine channels follow the same pattern: install the plugin, declare the credentials, check, then approve the pairing code. The table below notes what is specific to the more common ones.

Kênh <i>Channel</i>	Điểm cần biết trước khi nối <i>What to know before connecting</i>
Signal	Cần công cụ signal-cli và nên dùng một số điện thoại riêng; đăng ký số bằng signal-cli có thể làm đăng xuất ứng dụng Signal chính của số đó <i>Needs the signal-cli tool and a dedicated number; registering a number with signal-cli can log out the main Signal app for that number</i>
iMessage	Chỉ chạy trên máy Mac đã đăng nhập Messages, cần cấp quyền truy cập toàn ổ đĩa <i>Runs only on a Mac signed in to Messages, and needs Full Disk Access</i>
Matrix	Hỗ trợ mã hoá đầu cuối; mặc định bot không tự vào phòng khi được mời <i>Supports end-to-end encryption; by default the bot does not auto-join rooms</i>
Microsoft Teams	Đăng ký ứng dụng trong Azure, phù hợp môi trường doanh nghiệp <i>Requires an app registration in Azure; suits corporate environments</i>
Google Chat	Cấu hình qua Google Cloud, hợp với tổ chức đã dùng Google Workspace <i>Configured through Google Cloud; suits organisations already on Workspace</i>
WebChat	Có sẵn trong bản lõi, không phải cài gì; là tab trò chuyện của bảng điều khiển <i>Built into the core with nothing to install; it is the control panel's chat tab</i>

6.4. Ai được nhắn tin cho trợ lý của bạn — *Who may message your assistant*

Mọi kênh đều đi qua cùng một hàng rào. Hiểu bốn lớp này rồi thì cấu hình kênh nào cũng giống nhau.

Every channel passes through the same barrier. Once you understand these four layers, configuring any channel is the same job.



Hình 17. Đường đi của một tin nhắn lạ: qua đủ bốn cổng mới tới được agent

Figure 17. The path of a message from a stranger: four gates before it reaches the agent

Chính sách <i>Policy</i>	Nghĩa là gì <i>What it means</i>	Khi nào dùng <i>When to use</i>
pairing	Người lạ nhắn thì nhận mã chờ duyệt, không được trả lời <i>A stranger gets a code awaiting approval and no reply</i>	Mặc định; hợp với hầu hết trường hợp <i>The default; right for most cases</i>
allowlist	Chỉ người trong danh sách được nhắn <i>Only listed people may message</i>	Khi đã biết trước chính xác ai được dùng <i>When you know exactly who may use it</i>
open	Ai nhắn cũng được trả lời <i>Anyone who messages gets a reply</i>	Gần như không bao giờ nên dùng <i>Almost never appropriate</i>
disabled	Tắt hẳn nhắn riêng trên kênh đó <i>Direct messages are off for that channel</i>	Kênh chỉ dùng trong nhóm <i>Channels used only in groups</i>

Trong nhóm còn thêm một cổng nữa: mặc định trợ lý chỉ trả lời khi có ai nhắc tên nó. Bỏ cổng này thì nó đọc và trả lời mọi tin trong nhóm — tốn tiền gọi mô hình và phiền người khác.

Groups add one more gate: by default the assistant replies only when mentioned. Remove that gate and it reads and answers every message in the group — burning model credits and annoying everyone.

MẸO / TIP — Trước khi cho một nhóm đông người nhắn cho trợ lý, đặt `dmPolicy` là `pairing`, giữ cổng nhắc tên bật, và đọc Chương 13.

Before letting a busy group message the assistant, set `dmPolicy` to `pairing`, keep the mention gate on, and read Chapter 13.

Chương 7. Mô hình AI và nhà cung cấp — Chapter 7. AI models and providers

OpenClaw không tự sinh ra câu trả lời. Nó nhận tin nhắn từ kênh trò chuyện, gói lại thành một lượt hỏi rồi gửi tới một mô hình ngôn ngữ do nhà cung cấp bên ngoài vận hành — OpenAI, Anthropic, Google, hoặc một máy chủ chạy ngay trong nhà bạn. Chọn mô hình nào, đăng nhập bằng khoá gì, khi mô hình chính hỏng thì rơi xuống đâu: ba câu hỏi đó quyết định trợ lý chạy nhanh hay chậm, tốn bao nhiêu tiền, và có ngừng trả lời giữa chừng hay không.

OpenClaw does not produce answers by itself. It takes a message from a chat channel, packages it into a turn and sends that turn to a language model run by an outside provider — OpenAI, Anthropic, Google, or a server running inside your own house. Which model, which credential, and where the system lands when the main model fails: those three questions decide how fast the assistant answers, how much it costs, and whether it stops replying halfway through.

GHI CHÚ / NOTE — Quy ước ký hiệu trong chương: chữ trong **dấu nháy ngược** là tên khoá cấu hình, tên biến môi trường, tên mô hình hoặc lệnh gõ vào; chữ **in đậm** là tên nút hoặc mục trên giao diện web của OpenClaw; dấu **→** chỉ đường đi qua các cấp trình đơn.

*Notation used in this chapter: text in **backticks** is a configuration key, an environment variable, a model name or a command to type; **bold** text names a button or a section in OpenClaw's web interface; the arrow **→** shows a path through menu levels.*

7.1. Cách OpenClaw gọi tên mô hình AI: dạng provider/model — How OpenClaw names a model: the provider/model reference

Mỗi mô hình được gọi bằng một chuỗi hai phần ngăn nhau bằng dấu gạch chéo: phần trước là mã nhà cung cấp, phần sau là mã mô hình. Chuỗi `openai/claude-opus-4-6` nghĩa là mô hình `claude-opus-4-6` truy cập qua nhà cung cấp `openai`. Cùng một mô hình có thể mua qua nhiều nhà cung cấp khác nhau, nên phần đầu là bắt buộc, không được lược bỏ.

Every model is named by a two-part string separated by a slash: the provider id first, the model id second. The reference `openai/claude-opus-4-6` means the model `claude-opus-4-6` reached through the provider `openai`. The same model can be bought through several providers, so the first part is mandatory and must not be dropped.

Chuỗi này chỉ chọn nhà cung cấp và mô hình. Nó không chọn bộ chạy agent cấp thấp — phần đó là khoá `agentRuntime.id` đặt trên từng mục nhà cung cấp hoặc từng mô hình, trình bày ở mục 8.8. Riêng `openai/*` có một ngoại lệ: chính sách định tuyến của OpenAI chỉ chọn bộ chạy Codex khi tuyến đi đúng một điểm cuối Responses chính thức chạy HTTPS và không có yêu cầu tự soạn đề lên. Tiền tố `openai/` một mình không bao giờ chọn Codex.

The reference selects the provider and the model only. It does not select the low-level agent runtime — that is the `agentRuntime.id` key placed on a provider or model entry, covered in section 8.8. One exception applies to `openai/*`: OpenAI's route policy selects the Codex

runtime only for an exact official HTTPS Responses endpoint with no authored request override. The `openai/` prefix alone never selects Codex.

```
openclaw models list
openclaw models list --provider openai
openclaw models list --all
openclaw models list --refresh
openclaw models set anthropic/claude-opus-5
openclaw models status
```

Lệnh Command	Việc nó làm What it does
<code>models list</code>	Liệt kê mô hình đang dùng được theo khoá đã khai; thêm <code>--provider <id></code> để lọc một nhà cung cấp <i>Lists the models available under the configured credentials; add <code>--provider <id></code> to filter to one provider</i>
<code>models list --all</code>	Mở trọn danh mục dựng sẵn, gồm cả các dòng bị đánh dấu ngừng hỗ trợ <i>Opens the full built-in catalog, including rows marked deprecated or disabled</i>
<code>models list --refresh</code>	Hỏi lại nhà cung cấp danh sách mô hình mới phát hành <i>Asks the providers again for newly released models</i>
<code>models set <ref></code>	Đặt mô hình chính cho toàn bộ agent <i>Sets the primary model for the agent</i>
<code>models status</code>	Cho biết mô hình nào đang hiệu lực và khoá nào đang đứng sau nó <i>Reports which model is in effect and which credential is behind it</i>
<code>models aliases add</code>	Đặt tên tắt cho một mô hình để gọi nhanh trong khung chat <i>Gives a model a short alias so it can be called quickly from chat</i>

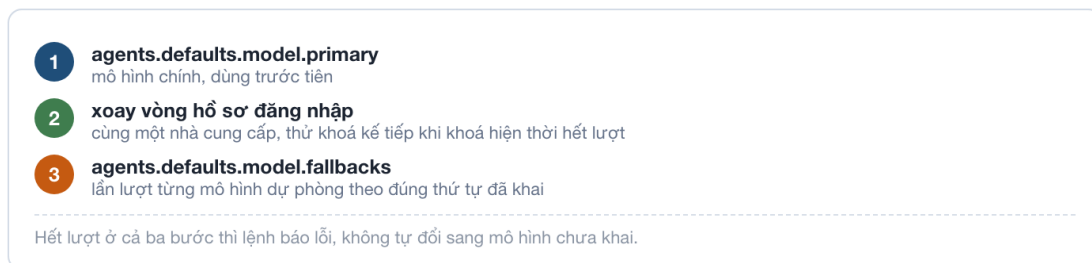
MẸO / TIP — Tên mô hình thay đổi theo thời gian, nhà cung cấp phát hành bản mới rồi khai tử bản cũ. Đừng chép tên mô hình từ một bài viết trên mạng. Chạy `openclaw models list --refresh` rồi chọn trong danh sách máy trả về.

Model names change over time; providers release new ones and retire old ones. Do not copy a model name out of a blog post. Run `openclaw models list --refresh` and pick from the list the machine returns.

7.2. Thứ tự chọn mô hình: mô hình chính, danh sách dự phòng và xoay vòng khoá — *Model selection order: primary, fallbacks and credential rotation*

Khi một lượt hỏi bắt đầu, OpenClaw đi qua ba bậc theo đúng thứ tự. Bậc một là mô hình chính khai ở `agents.defaults.model.primary` — khoá này cũng nhận trực tiếp một chuỗi, viết `agents.defaults.model: "openai/gpt-6-astra"` là hợp lệ. Bậc hai là danh sách dự phòng `agents.defaults.model.fallbacks`, thử lần lượt từ trên xuống. Bậc ba nằm bên trong một nhà cung cấp: nếu bạn khai nhiều khoá cho cùng nhà cung cấp đó, OpenClaw xoay vòng hết các khoá trước khi nhảy sang mô hình dự phòng kế tiếp.

When a turn starts, OpenClaw walks three levels in order. Level one is the primary model at `agents.defaults.model.primary` — the key also accepts a plain string, so `agents.defaults.model: "openai/gpt-6-astra"` is valid. Level two is the fallback list `agents.defaults.model.fallbacks`, tried from top to bottom. Level three sits inside a single provider: if several credentials are configured for that provider, OpenClaw rotates through all of them before moving on to the next fallback model.



Hình 18. Thứ tự chọn mô hình: mô hình chính, rồi danh sách dự phòng, và trong mỗi nhà cung cấp là vòng xoay khoá đăng nhập

Figure 18. Model selection order: primary first, then the fallback list, with credential rotation inside each provider

Cùng một chuỗi `provider/model` lại cư xử khác nhau tùy vào nơi nó được đặt. Bảng dưới là điểm hay gây hiểu nhầm nhất của cả hệ thống mô hình.

The same `provider/model` reference behaves differently depending on where it came from. The table below covers the most misunderstood part of the whole model system.

Nguồn đặt mô hình <i>Where it came from</i>	Cách xử lý khi mô hình hỏng <i>Behaviour on failure</i>
Mặc định trong cấu hình <i>Configured default</i>	Điểm xuất phát bình thường; dùng đúng danh sách <code>fallbacks</code> đã khai <i>The normal starting point; uses the configured fallbacks list</i>
Tự rơi dự phòng <i>Automatic fallback</i>	Trạng thái hồi phục tạm thời, ghi dấu <code>modelOverrideSource: "auto"</code> . OpenClaw dò lại mô hình chính theo chu kỳ, khi mô hình chính sống lại thì tự bỏ lựa chọn tạm và báo một lần mỗi lần đổi trạng thái <i>A temporary recovery state stored as modelOverrideSource: "auto". OpenClaw reprobes the original primary periodically, clears the automatic selection on recovery, and announces each transition once</i>
Người dùng chọn trong phiên <i>User session selection</i>	Chính xác và nghiêm ngặt. Lệnh <code>/model</code> , bộ chọn mô hình trên giao diện web đều ghi <code>modelOverrideSource: "user"</code> . Mô hình đó chết thì lượt chạy báo lỗi ra mặt, không lặng lẽ rơi sang mô hình khác <i>Exact and strict. /model and the model picker both store modelOverrideSource: "user". If that model becomes unreachable the run fails visibly instead of silently falling through to another model</i>
Tác vụ hẹn giờ đặt riêng <i>Cron job model</i>	Làm mô hình chính cho riêng lần chạy đó, vẫn dùng <code>fallbacks</code> chung trừ khi công việc tự khai danh sách riêng; khai <code>fallbacks: []</code> là chạy nghiêm ngặt một mô hình <i>Acts as the primary for that run and still uses the configured fallbacks unless the job supplies its own list; fallbacks: [] forces a strict single-model run</i>

CẢNH BÁO / WARNING — Đổi `agents.defaults.model.primary` không viết lại mô hình đã ghim trong các phiên đang mở. Nếu `openclaw status` báo phiên này đang ghim mô hình X trong khi cấu hình đã là Y, gõ `/model default` trong khung chat để gỡ ghim.
Changing agents.defaults.model.primary does not rewrite the model pinned into sessions that are already open. If openclaw status reports that the session is pinned to X while the configuration says Y, type /model default in the chat to clear the pin.

Danh sách dự phòng sửa được bằng lệnh, không cần mở tệp cấu hình.

The fallback list can be edited by command; there is no need to open the configuration file.

```
openclaw models fallbacks list
openclaw models fallbacks add google/gemini-3.5-flash
openclaw models fallbacks remove google/gemini-3.5-flash
openclaw models fallbacks clear
```

MẸO / TIP — Đặt mô hình chính là mô hình mạnh nhất bạn mua được, và để dành danh sách dự phòng cho những mô hình rẻ và nhanh. Với agent có quyền chạy lệnh hoặc nhận tin nhắn từ người lạ, tránh xếp các mô hình đời cũ và yếu vào danh sách dự phòng.
Set the primary to the strongest model you have access to and keep the fallback list for cheaper, faster

models. For an agent that can run commands or receives messages from strangers, avoid putting older, weaker model tiers on the fallback list.

7.3. Bốn cách khai khoá API cho OpenClaw — Four ways to supply an API key to OpenClaw

Khoá API là chuỗi ký tự nhà cung cấp cấp cho bạn để tính tiền và nhận diện. OpenClaw nhận khoá theo bốn đường, dùng đường nào cũng được và có thể trộn lẫn.

An API key is the string a provider issues so it can identify and bill you. OpenClaw accepts keys through four routes; any of them works and they can be mixed.

1. Trình hướng dẫn ban đầu: chạy `openclaw onboard` hoặc `openclaw configure`, chọn nhà cung cấp, dán khoá. Đây là đường ngắn nhất cho người mới.
The setup wizard: run `openclaw onboard` or `openclaw configure`, pick a provider and paste the key. This is the shortest route for a first-time user.
2. Giao diện web: mở bảng điều khiển ở `http://127.0.0.1:18789`, vào **Settings** → **Models**, thêm khoá cho từng nhà cung cấp.
*The web interface: open the control panel at `http://127.0.0.1:18789`, go to **Settings** → **Models** and add a key for each provider.*
3. Biến môi trường: đặt `OPENAI_API_KEY`, `ANTHROPIC_API_KEY`... trong môi trường của tiến trình gateway, hoặc ghi vào tệp `~/.openclaw/.env`.
Environment variables: set `OPENAI_API_KEY`, `ANTHROPIC_API_KEY` and so on in the gateway process environment, or write them into `~/.openclaw/.env`.
4. Tham chiếu bí mật (SecretRef): trong tệp cấu hình chỉ ghi một con trỏ tới nơi chứa khoá, không ghi chính khoá.
A SecretRef: the configuration file holds only a pointer to where the key lives, never the key itself.

Ba cách đầu ghi khoá ở dạng chữ thường đọc được. Cách thứ tư sinh ra để tránh điều đó. Một SecretRef là một đối tượng ba trường, dán vào đúng chỗ đáng lẽ ghi khoá.

The first three routes leave the key in readable plain text. The fourth exists to avoid that. A SecretRef is a three-field object placed exactly where the key would otherwise sit.

```
{ source: "env", provider: "default", id: "OPENAI_API_KEY" }

// dạng viết tắt cung được, đặt thang vào o khoá:
"${OPENAI_API_KEY}"
```

Nguồn <i>Source</i>	Khoá nằm ở đâu <i>Where the secret lives</i>	Dạng mã định danh <i>Id grammar</i>
env	Biến môi trường của tiến trình gateway <i>An environment variable of the gateway process</i>	Chữ in hoa, bắt đầu bằng chữ cái, tối đa 128 ký tự <i>Upper case, starts with a letter, up to 128 characters</i>
file	Một tệp JSON trên đĩa, ví dụ ~/.openclaw/secrets.json <i>A JSON file on disk, for example</i> ~/.openclaw/secrets.json	Con trỏ JSON tuyệt đối, ví dụ /providers/openai/apiKey <i>An absolute JSON pointer such as</i> /providers/openai/apiKey
exec	Kết quả của một lệnh bên ngoài: 1Password, Bitwarden, HashiCorp Vault, pass, sops <i>The output of an external command:</i> 1Password, Bitwarden, HashiCorp Vault, pass or sops	Đường dẫn kèm bộ chọn, ví dụ providers/openai/apiKey#value <i>A path with an optional selector such as</i> providers/openai/apiKey#value
store	Kho bí mật dùng chung của gateway trong SQLite <i>The Gateway-wide shared secret store in SQLite</i>	Cùng dạng tên biến môi trường <i>The same environment-variable name grammar</i>

Nhà cung cấp SecretRef khai dưới khoá `secrets.providers`, mỗi mục ghi rõ `source` và tham số riêng của nó: `path` và `mode` cho nguồn tệp, `command`, `args`, `passEnv` cho nguồn lệnh. Khoá `secrets.defaults` chọn tên mặc định cho từng nguồn. Sau khi đổi, nạp lại bằng `openclaw secrets reload` và soát lại bằng `openclaw secrets audit`.

SecretRef providers are declared under secrets.providers; each entry names its source and its own parameters: path and mode for a file source, command, args and passEnv for an exec source. The secrets.defaults key picks the default alias for each source. After a change, reload with openclaw secrets reload and review with openclaw secrets audit.

CẢNH BÁO / WARNING — Khoá ghi thẳng dạng chữ thường trong `openclaw.json`, trong `.env`, hoặc trong các tệp `models.json` sinh ra theo từng agent đều nằm trong tầm đọc của chính agent. Chuyển hết sang SecretRef rồi chạy `openclaw secrets audit --check` cho tới khi không còn báo tồn dư khoá dạng chữ thường.
Plain-text keys in openclaw.json, in .env, or in the generated per-agent models.json files are all readable by the agent itself. Migrate them to SecretRefs and run openclaw secrets audit --check until it reports no plain-text residue.

GHI CHÚ / NOTE — Nhiều nhà cung cấp chấp nhận nhiều khoá cùng lúc để chia tải và né giới hạn tần suất. Khai biến số nhiều `OPENAI_API_KEYS` hoặc đánh số

OPENAI_API_KEY_1, OPENAI_API_KEY_2; Anthropic dùng ANTHROPIC_API_KEYS và ANTHROPIC_API_KEY_1, ANTHROPIC_API_KEY_2 theo cùng quy tắc.

Several providers accept more than one key so load can be spread and rate limits avoided. Use the plural form OPENAI_API_KEYS, or numbered names OPENAI_API_KEY_1 and OPENAI_API_KEY_2; Anthropic uses ANTHROPIC_API_KEYS with ANTHROPIC_API_KEY_1 and ANTHROPIC_API_KEY_2 on the same rule.

7.4. Danh sách nhà cung cấp mô hình AI được OpenClaw hỗ trợ sẵn — The model providers OpenClaw supports out of the box

Các nhà cung cấp dưới đây đi kèm bản cài, không phải cài thêm gì. Cột giữa là biến môi trường OpenClaw đọc khoá; cột phải là một mô hình có thật của nhà cung cấp đó, chép đúng như tài liệu gốc ghi, dùng làm ví dụ chứ không phải khuyến nghị.

The providers below ship with the install and need nothing extra. The middle column is the environment variable OpenClaw reads the key from; the right column is one real model from that provider, copied as the official documentation writes it, given as an example rather than a recommendation.

```

Terminal — openclaw models

~ % openclaw models list

NHÀ CUNG CẤP      MÔ HÌNH          TRẠNG THÁI
anthropic         claude-opus-5    ● chính
anthropic         claude-haiku-4-5 phụ trợ
openai            gpt-6-astra      dự phòng 1
google            gemini-3.1-pro-preview dự phòng 2
ollama            qwen3:14b        cục bộ

~ % openclaw models set openai/gpt-6-astra
✓ Mô hình chính đổi sang openai/gpt-6-astra
  
```

Hình 19. Màn hình danh sách mô hình: mỗi dòng là một chuỗi provider/model dùng được với khoá đang khai

Figure 19. The model list screen: each row is a provider/model reference usable with the configured credentials

Nhà cung cấp <i>Provider</i>	Biến môi trường <i>Auth environment variable</i>	Mô hình ví dụ <i>Example model</i>
OpenAI	OPENAI_API_KEY	openai/gpt-6-astra
Anthropic	ANTHROPIC_API_KEY	anthropic/claude-opus-5
Google Gemini	GEMINI_API_KEY (dự phòng GOOGLE_API_KEY) GEMINI_API_KEY (falls back to GOOGLE_API_KEY)	google/gemini-3.1-pro-preview
Google Vertex	Thông tin đăng nhập mặc định của gcloud <i>gcloud Application Default Credentials</i>	—

Nhà cung cấp <i>Provider</i>	Biến môi trường <i>Auth environment variable</i>	Mô hình ví dụ <i>Example model</i>
OpenRouter	OPENROUTER_API_KEY hoặc đăng nhập OAuth OPENROUTER_API_KEY or OAuth sign-in	openrouter/auto
OpenCode	OPENCODE_API_KEY	opencode/claude-opus-4-6
DeepSeek	DEEPSEEK_API_KEY	deepseek/deepseek-v4-flash
Groq	GROQ_API_KEY	groq/llama-3.3-70b-versatile
Mistral	MISTRAL_API_KEY	mistral/mistral-large-latest
Moonshot	MOONSHOT_API_KEY	moonshot/kimi-k2.6
Z.AI (GLM)	ZAI_API_KEY	zai/glm-5.2
xAI	XAI_API_KEY hoặc OAuth SuperGrok XAI_API_KEY or SuperGrok OAuth	xai/grok-4.7
Cerebras	CEREBRAS_API_KEY	cerebras/zai-glm-4.7
Cohere	COHERE_API_KEY	cohere/command-a-plus-05-2026
Together	TOGETHER_API_KEY	together/meta-llama/Llama-3.3-70B-Instruct-Turbo
MiniMax	MINIMAX_API_KEY	minimax/MiniMax-M3
GitHub Copilot	COPILOT_GITHUB_TOKEN	—
Vercel AI Gateway	AI_GATEWAY_API_KEY	vercel-ai-gateway/anthropic/claude-opus-4.6

Ngoài bảng trên còn Arcee, BytePlus, Chutes, ClawRouter, DeepInfra, Featherless, GMI Cloud, Hugging Face, NVIDIA, NovitaAI, Ollama Cloud, Qianfan, Tencent TokenHub, Venice, Volcano Engine và Xiaomi. Xem danh sách hiện hành bằng `openclaw models auth list`.

Beyond the table there are Arcee, BytePlus, Chutes, ClawRouter, DeepInfra, Featherless, GMI Cloud, Hugging Face, NVIDIA, NovitaAI, Ollama Cloud, Qianfan, Tencent TokenHub, Venice, Volcano Engine and Xiaomi. Read the current list with `openclaw models auth list`.

Một số nhà cung cấp cho đăng nhập bằng tài khoản thuê bao thay vì khoá trả theo lượt: ChatGPT hoặc Codex của OpenAI, tài khoản Claude, SuperGrok của xAI, GitHub Copilot. Đường đăng nhập chung là `openclaw models auth login`.

Some providers allow a subscription sign-in instead of a pay-per-token key: OpenAI's ChatGPT or Codex plan, a Claude account, xAI's SuperGrok, GitHub Copilot. The common entry point is `openclaw models auth login`.

```
openclaw models auth list
openclaw models auth login --provider anthropic
openclaw models auth login --provider openai --set-default
openclaw models auth paste-api-key
openclaw models auth order get
openclaw models auth order set
```

CẢNH BÁO / WARNING — Thêm một khoá hay một tài khoản mới KHÔNG đổi mô hình chính đang chạy. Tài liệu gốc ghi rõ điểm này vì nó gây hiểu nhầm thường xuyên. Muốn đổi mô hình chính thì chạy `openclaw models set <provider/model>`, hoặc thêm cờ `--set-default` vào lệnh đăng nhập.

Adding a credential or a new account does NOT change the primary model in use. The official documentation calls this out because it is a frequent misunderstanding. To change the primary, run `openclaw models set <provider/model>` or add `--set-default` to the sign-in command.

7.5. Mô hình phụ: mô hình rẻ, mô hình quyết định, mô hình ảnh và mô hình PDF — *Secondary models: utility, decision, image and PDF models*

Không phải việc nào cũng đáng gọi mô hình mạnh nhất. Đặt tiêu đề cho một phiên, tóm tắt tiến độ, đọc một trang PDF — những việc đó có mô hình riêng, rẻ hơn nhiều lần.

Not every task deserves the strongest model. Naming a session, narrating progress, reading a page of PDF — each has its own, far cheaper, model slot.

Khoá cấu hình <i>Configuration key</i>	Dùng vào việc gì <i>What it is used for</i>
<code>agents.defaults.utilityModel</code>	Việc nội bộ ngắn: đặt tiêu đề phiên, đặt tên luồng trên kênh, thuật lại tiến độ, gom tóm tắt hoạt động. Bỏ trống thì OpenClaw lấy mô hình nhỏ do nhà cung cấp chính khai (OpenAI → <code>gpt-5.6-luna</code> , Anthropic → <code>claude-haiku-4-5</code>). Đặt chuỗi rỗng để tắt hẳn <i>Short internal tasks: session titles, channel thread titles, progress narration and activity recaps. Left unset, OpenClaw takes the small-model default declared by the primary provider (OpenAI → <code>gpt-5.6-luna</code>, Anthropic → <code>claude-haiku-4-5</code>). Set it to an empty string to disable it</i>
<code>agents.defaults.decisionModel</code>	Mô hình quyết định có kiểu, trả về lựa chọn, điểm số và xác suất đúng/sai cho phần mở rộng. Không khai thì tắt. Mô hình loại này không bao giờ hiện trong bộ chọn mô hình của khung chat <i>A typed decision model returning choices, scores and boolean probabilities for plugins. Disabled when unset. Decision models never appear among chat, primary, fallback or utility choices</i>
<code>agents.defaults.imageModel</code>	Chỉ dùng khi mô hình chính không đọc được ảnh <i>Used only when the primary model cannot accept images</i>
<code>agents.defaults.pdfModel</code>	Dùng cho công cụ pdf. Không khai thì rơi xuống <code>imageModel</code> , rồi tới mô hình của phiên <i>Used by the pdf tool. If unset it falls back to <code>imageModel</code>, then to the session model</i>
<code>agents.defaults.mediaModels</code>	Ba ô con <code>image</code> , <code>music</code> , <code>video</code> cho các công cụ sinh phương tiện <i>Three sub-keys <code>image</code>, <code>music</code> and <code>video</code> backing the media generation tools</i>
<code>agents.defaults.modelPolicy.allow</code>	Danh sách trắng giới hạn mô hình được phép chọn. Nhận chuỗi chính xác hoặc ký tự đại diện đuôi như <code>provider/*</code> . Bỏ trống hoặc để <code>[]</code> là cho phép mọi mô hình <i>An allowlist limiting which models may be selected. Accepts exact refs or trailing wildcards such as <code>provider/*</code>. Omit it or set <code>[]</code> to allow any model</i>

Các khoá trên đặt ở `agents.defaults` là áp cho mọi agent. Muốn khác nhau theo từng agent thì đặt lại ở `agents.entries.<id>` với cùng tên khoá; mục đặt riêng thay thế mục mặc định chứ không trộn vào.

Placed under `agents.defaults`, these keys apply to every agent. To differ per agent, set the same key under `agents.entries.<id>`; a per-agent entry replaces the default rather than merging with it.

GHI CHÚ / NOTE — Việc nội bộ giao cho mô hình rẻ vẫn là lượt gọi thật tới nhà cung cấp, vẫn gửi đi một phần nội dung công việc và vẫn bị tính tiền. Đặt `utilityModel` là chuỗi rỗng

nếu bạn không muốn nội dung chạy qua nhà cung cấp thứ hai.

A task handed to the cheap model is still a real provider call: it sends bounded task content and is still billed. Set `utilityModel` to an empty string if you do not want content passing through a second provider.

7.6. Chạy mô hình AI cục bộ trên máy của bạn với Ollama và LM Studio — *Running local AI models with Ollama and LM Studio*

Mô hình cục bộ chạy ngay trên máy bạn, không gửi gì ra ngoài. Đây là lựa chọn duy nhất khi dữ liệu không được phép rời khỏi máy, và là cách cắt trọt chi phí theo lượt gọi. Đổi lại, chất lượng trả lời thấp hơn mô hình thương mại cùng thời điểm, tốc độ phụ thuộc phần cứng, và máy phải đủ bộ nhớ để nạp trọn mô hình.

A local model runs on your own machine and sends nothing outside. It is the only option when data may not leave the host, and it removes per-call cost entirely. In exchange, answer quality is below a commercial model of the same date, speed depends on the hardware, and the machine must have enough memory to hold the whole model.

Máy chủ cục bộ <i>Local server</i>	Mã nhà cung cấp <i>Provider id</i>	Địa chỉ mặc định và khoá <i>Default address and key</i>
Ollama	ollama	<code>http://127.0.0.1:11434</code> ; không cần khoá, bật dò tự động bằng <code>OLLAMA_API_KEY</code> <code>http://127.0.0.1:11434</code> ; <i>no key required, opt into discovery with</i> <code>OLLAMA_API_KEY</code>
LM Studio	lmstudio	<code>http://localhost:1234/v1</code> ; khoá <code>LM_API_TOKEN</code> <code>http://localhost:1234/v1</code> ; <i>key</i> <code>LM_API_TOKEN</code>
llama.cpp	llama-cpp	Cài phần mở rộng <code>@openclaw/llama-cpp-provider</code> ; tham chiếu dạng <code>llama-cpp/<model></code> <i>Install the plugin</i> <code>@openclaw/llama-cpp-provider</code> ; <i>references take the form</i> <code>llama-cpp/<model></code>
vLLM	vllm	<code>http://127.0.0.1:8000/v1</code> ; khoá tùy máy chủ, bật dò bằng <code>VLLM_API_KEY</code> <code>http://127.0.0.1:8000/v1</code> ; <i>key depends on your server, opt into discovery with</i> <code>VLLM_API_KEY</code>
SGLang	sglang	<code>http://127.0.0.1:30000/v1</code> ; bật dò bằng <code>SGLANG_API_KEY</code> <code>http://127.0.0.1:30000/v1</code> ; <i>opt into discovery with</i> <code>SGLANG_API_KEY</code>
llmman	llmman	<code>http://127.0.0.1:17434/v1</code> ; khai api: <code>"openai-completions"</code> <code>http://127.0.0.1:17434/v1</code> ; <i>declare</i> api: <code>"openai-completions"</code>
LiteLLM và các cầu nối khác <i>LiteLLM and other proxies</i>	Tự đặt <i>Your own id</i>	Khai trong <code>models.providers</code> với <code>baseUrl</code> và api: <code>"openai-completions"</code> <i>Declared under</i> <code>models.providers</code> <i>with a</i> <code>baseUrl</code> <i>and</i> <code>api: "openai-completions"</code>

Ollama là đường ngắn nhất. Cài Ollama, tải mô hình về, rồi trỏ mô hình chính sang nó.
Ollama is the shortest route. Install Ollama, pull a model, then point the primary at it.

```
ollama pull llama3.3
openclaw models set ollama/llama3.3
openclaw models list --provider ollama
```

Cầu nối tương thích OpenAI khai bằng tay trong `models.providers`, mỗi mô hình ghi rõ mã, tên hiển thị, cửa sổ ngữ cảnh và chi phí. Khai chi phí bằng 0 để bảng theo dõi chi phí không cộng nhảm tiền cho mô hình chạy trong nhà.

An OpenAI-compatible proxy is declared by hand under `models.providers`, *with each model carrying its id, display name, context window and cost. Declare the cost as zero so the cost report does not bill you for a model running in your own house.*

```
{
  models: {
    providers: {
      lmstudio: {
        baseUrl: "http://localhost:1234/v1",
        apiKey: "${LM_API_TOKEN}",
        api: "openai-completions",
        timeoutSeconds: 300,
        models: [
          { id: "my-local-model", name: "Local Model",
            contextWindow: 200000, maxTokens: 8192 },
        ],
      },
    },
  },
}
```

MẸO / TIP — Cách dùng thực tế nhất là trộn: mô hình cục bộ làm `utilityModel` cho việc vặt, mô hình thương mại làm mô hình chính cho việc cần đúng. Tiền giảm mà chất lượng câu trả lời chính không đổi.

The most practical arrangement is a mix: a local model as the `utilityModel` for small chores and a commercial model as the primary for work that must be right. Cost drops while the quality of the real answers stays the same.

7.7. Theo dõi hạn mức và chi phí gọi mô hình AI — *Tracking provider quota and model cost*

Một trợ lý chạy suốt ngày đêm tiêu khoá nhanh hơn người dùng tưởng, nhất là khi bật nhíp tim chủ động ở mục 8.7. Kiểm hạn mức bằng một lệnh.

An assistant running around the clock burns through a plan faster than expected, especially with the proactive heartbeat of section 8.7 turned on. One command checks the quota.

```
openclaw status --usage
openclaw gateway usage-cost
openclaw status --all
```

Lệnh `openclaw status --usage` quy mọi nhà cung cấp về cùng một cách đọc: phần trăm hạn mức còn lại. Những nhà cung cấp báo được hạn mức gồm Anthropic, Gemini CLI, GitHub Copilot, MiniMax, OpenAI Codex, Xiaomi và z.ai. Nhà cung cấp ngoài danh sách này chỉ trả về trạng thái kết nối, không trả về số dư.

`openclaw status --usage` *normalises every provider into one reading: the percentage of quota left. The providers that report quota are Anthropic, Gemini CLI, GitHub Copilot, MiniMax, OpenAI Codex, Xiaomi and z.ai. Providers outside that list return connection status only, not a balance.*

Nơi xem <i>Where to look</i>	Cho biết gì <i>What it tells you</i>
<code>openclaw status --usage</code>	Phần trăm hạn mức còn lại theo từng nhà cung cấp <i>Percentage of quota left, per provider</i>
<code>openclaw gateway usage-cost</code>	Chi phí ước tính do gateway ghi nhận <i>The estimated cost the gateway has recorded</i>
Lệnh <code>/status</code> trong khung chat <i>The /status command in chat</i>	Trạng thái chạy, thời gian hoạt động của gateway, sức khỏe phần mở rộng, kèm hạn mức nhà cung cấp <i>Execution status, Gateway and system uptime, plugin health, plus provider usage and quota</i>
Lệnh <code>/usage</code> trong khung chat <i>The /usage command in chat</i>	Bật tắt dòng chân trang đếm token sau mỗi câu trả lời: <code>off</code> , <code>tokens</code> , <code>full</code> , <code>cost</code> <i>Controls the per-response usage footer: off, tokens, full, cost</i>
Giao diện web <i>The web interface</i>	Cùng số liệu, xem theo bảng <i>The same figures, presented as a table</i>

CẢNH BÁO / WARNING — Chạy hết hạn mức thì trợ lý ngừng trả lời chứ không báo cho bạn một tiếng. Nếu trợ lý đang nhận tin của khách, hãy khai danh sách dự phòng ở mục 7.2 sang một nhà cung cấp khác — khác nhà cung cấp mới có tác dụng, cùng nhà cung cấp thì cùng hết hạn mức.

When the quota runs out the assistant simply stops replying without warning you. If it is answering customers, configure the fallback list from section 7.2 to a different provider — it only helps if the provider differs, because the same provider runs out at the same moment.

Chương 8. Agent, phiên làm việc và quyền hạn — *Chapter 8. Agents, sessions and permissions*

Agent là nhân vật trợ lý: có tên, có tính cách, có thư mục làm việc riêng, có trí nhớ riêng và có quyền hạn riêng. Một tiến trình gateway chạy được nhiều agent cùng lúc, mỗi agent phục vụ một mục đích — một agent cho việc riêng, một agent cho nhóm công ty, một agent cho khách hàng. Phiên làm việc là một mạch hội thoại của agent đó; quyền hạn quyết định agent được đọc được ghi ở đâu và ai duyệt khi nó xin chạy lệnh.

An agent is the assistant persona: it has a name, a character, its own working folder, its own memory and its own permissions. One gateway process runs several agents at once, each for a purpose — one for personal work, one for the company group, one for customers. A session is one thread of conversation belonging to that agent; permissions decide where it may read and write and who approves when it asks to run a command.

8.1. Một agent trong OpenClaw gồm những gì — *What one OpenClaw agent contains*

Agent không phải một dòng cấu hình. Nó là năm thứ đi liền nhau, tách hẳn khỏi các agent khác.

An agent is not one line of configuration. It is five things travelling together, kept separate from every other agent.

Thành phần <i>Component</i>	Đường dẫn mặc định <i>Default path</i>	Chứa gì <i>What it holds</i>
Không gian làm việc <i>Workspace</i>	<code>~/.openclaw/workspace</code>	Tệp định danh, trí nhớ, kỹ năng riêng; cũng là thư mục hiện hành khi agent chạy lệnh <i>Identity files, memory, workspace skills; also the working directory when the agent runs commands</i>
Thư mục trạng thái <i>State directory (agentDir)</i>	<code>~/.openclaw/agents/<agentId>/agent</code>	Hồ sơ đăng nhập, danh mục mô hình, cấu hình riêng của agent <i>Auth profiles, the model registry and per-agent configuration</i>
Hồ sơ đăng nhập <i>Auth profiles</i>	Trong <code>openclaw-agent.sqlite</code> <i>Inside openclaw-agent.sqlite</i>	Khoá và phiên đăng nhập nhà cung cấp của riêng agent này <i>The provider keys and sign-ins belonging to this agent alone</i>
Danh mục mô hình <i>Model registry</i>	Trong thư mục trạng thái <i>Inside the state directory</i>	Mô hình chính, danh sách dự phòng và tên tắt của riêng agent <i>The agent's own primary model, fallback list and aliases</i>
Kho phiên <i>Session store</i>	<code><agentDir>/openclaw-agent.sqlite</code>	Hàng phiên, bản ghi hội thoại, trạng thái định tuyến <i>Session rows, transcripts and routing state</i>

Không cấu hình gì thì OpenClaw chạy đúng một agent: mã `main`, khoá phiên `agent:main:main`, không gian làm việc `~/.openclaw/workspace`, trạng thái ở `~/.openclaw/agents/main/agent`.

With nothing configured, OpenClaw runs exactly one agent: id main, session key agent:main:main, workspace ~/.openclaw/workspace, state under ~/.openclaw/agents/main/agent.

CẢNH BÁO / WARNING — Không gian làm việc là thư mục mặc định, KHÔNG phải một hàng rào cứng. Đường dẫn tương đối quy về trong đó, nhưng đường dẫn tuyệt đối vẫn chạm tới chỗ khác trên máy nếu chưa bật hộp cát. Muốn cách ly thật thì bật `agents.defaults.sandbox`.

The workspace is the default working directory, NOT a hard boundary. Relative paths resolve inside it, but absolute paths still reach elsewhere on the host unless sandboxing is enabled. For real isolation, turn on agents.defaults.sandbox.

8.2. Các tệp định danh trong không gian làm việc của agent — *The identity files inside an agent workspace*

Mỗi lần mở phiên, OpenClaw đọc một nhóm tệp trong không gian làm việc và tiêm nội dung vào đầu cuộc hội thoại. Đây là chỗ bạn dạy trợ lý: nó tên gì, nói năng ra sao, đang làm dở việc gì, và biết gì về bạn.

Each time a session opens, OpenClaw reads a set of files in the workspace and injects them at the start of the conversation. This is where you teach the assistant: what it is called, how it speaks, what work is in progress and what it knows about you.

Tệp File	Vai trò Role	Nạp khi nào When it is loaded
AGENTS.md	Hướng dẫn vận hành: quy tắc, thứ tự ưu tiên, cách cư xử <i>Operating instructions: rules, priorities and how to behave</i>	Mọi phiên, kể cả phiên của agent con <i>Every session, including subagent sessions</i>
SOUL.md	Tính cách, giọng nói, giới hạn <i>Persona, tone and boundaries</i>	Mọi phiên <i>Every session</i>
IDENTITY.md	Tên, phong thái, biểu tượng của agent <i>The agent's name, vibe and emoji</i>	Mọi phiên; tạo trong nghi thức chạy lần đầu <i>Every session; created during the first-run ritual</i>
USER.md	Mô hình người dùng: thói quen, cách liên lạc, việc đang làm <i>The user model: preferences, communication style, active projects</i>	Mọi phiên, có ngân sách riêng 4.000 ký tự <i>Every session, with its own 4,000-character budget</i>
MEMORY.md	Trí nhớ dài hạn đã chất lọc <i>Curated long-term memory</i>	Chỉ phiên chính, riêng tư; nhóm và kênh chung không nạp <i>The main private session only; groups and shared channels do not load it</i>
memory/YYYY-MM-DD.md	Nhật ký trí nhớ theo ngày, mỗi ngày một tệp <i>A daily memory log, one file per day</i>	Theo yêu cầu, qua công cụ tìm trí nhớ <i>On demand, through the memory search tools</i>
BOOTSTRAP.md	Nghi thức chạy lần đầu, xong thì xoá <i>The one-time first-run ritual; delete it once complete</i>	Chỉ tạo cho không gian làm việc hoàn toàn mới <i>Created only for a brand-new workspace</i>

Tệp nào cũng có thể dài, nhưng phần tiêm vào hội thoại bị cắt theo hạn mức. Mặc định mỗi tệp tối đa 20.000 ký tự, đặt ở `agents.defaults.bootstrapMaxChars`; tổng tất

cả các tệp tối đa 60.000 ký tự, đặt ở `agents.defaults.bootstrapTotalMaxChars`. Riêng `USER.md` giữ ngân sách riêng 4.000 ký tự, không tính chung.

Any of these files may be long, but the part injected into the conversation is truncated by a budget. The default is 20,000 characters per file, set by `agents.defaults.bootstrapMaxChars`, and 60,000 characters in total, set by `agents.defaults.bootstrapTotalMaxChars`. `USER.md` keeps its own separate 4,000-character cap outside those totals.

CẢNH BÁO / WARNING — Nội dung vượt hạn mức bị cắt lửng lể. Một tệp `AGENTS.md` dài 50.000 ký tự nghĩa là quá nửa số quy tắc bạn viết không bao giờ tới tay mô hình. Viết ngắn, đưa chi tiết xuống `memory/YYYY-MM-DD.md` để công cụ tìm trí nhớ lấy khi cần.

Content over the budget is cut silently. A 50,000-character `AGENTS.md` means more than half the rules you wrote never reach the model. Keep these files short and push the detail down into `memory/YYYY-MM-DD.md`, where the memory tools can retrieve it on demand.

Nếu bạn tự quản các tệp này, tắt việc tạo tự động bằng `agents.defaults.skipBootstrap: true`. Thiếu một tệp bắt buộc thì OpenClaw chèn một dấu báo thiếu tệp rồi chạy tiếp; thiếu `USER.md` hoặc `MEMORY.md` thì bỏ qua, không báo gì.

If you manage these files yourself, turn off automatic creation with `agents.defaults.skipBootstrap: true`. When a required file is missing, OpenClaw injects a missing-file marker and carries on; when the optional `USER.md` or `MEMORY.md` is missing, it is simply omitted.

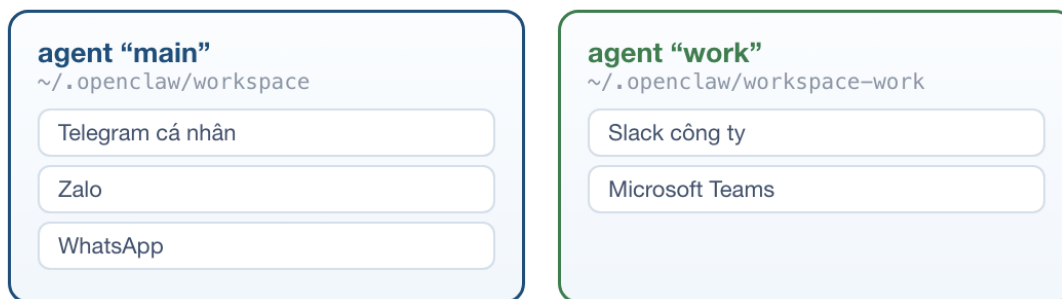
MẸO / TIP — Không gian làm việc là trí nhớ của trợ lý. Đặt nó trong một kho git RIÊNG TƯ để có bản lưu và lấy lại được. Tuyệt đối không đưa bất cứ thứ gì dưới `~/.openclaw` vào kho git — chỗ đó chứa khoá, phiên đăng nhập và bản ghi hội thoại.

The workspace is the assistant's memory. Keep it in a PRIVATE git repository so it is backed up and recoverable. Never commit anything under `~/.openclaw` — that directory holds credentials, sign-ins and transcripts.

8.3. Chạy nhiều agent và định tuyến kênh về đúng agent — *Running several agents and routing each channel to the right one*

Một tiến trình gateway chạy nhiều agent cách ly nhau, mỗi agent một không gian làm việc, một thư mục trạng thái và một kho phiên. Tin nhắn vào được đưa tới đúng agent bằng ràng buộc — mảng `bindings` đặt ở cấp gốc tệp cấu hình.

One gateway process runs several isolated agents, each with its own workspace, state directory and session store. Inbound messages reach the right agent through bindings — the `bindings` array at the top level of the configuration file.



Ràng buộc khai trong mảng bindings: { agentId: "work", match: { channel: "slack" } }

Hình 20. Nhiều agent trong một gateway: mỗi tài khoản kênh được ràng buộc về một agent, mỗi agent giữ không gian làm việc và kho phiên riêng

Figure 20. Several agents in one gateway: each channel account is bound to one agent, and each agent keeps its own workspace and session store

Thêm một agent bằng một lệnh. Trình hướng dẫn hỏi luôn có muốn tạo ràng buộc hay không.

One command adds an agent. The wizard offers to create the binding at the same time.

```
openclaw agents add work
openclaw agents add work --workspace ~/cong-viec \
  --model anthropic/claude-opus-5
openclaw agents add work --bind telegram:default --non-interactive
openclaw agents list --bindings
openclaw agents list --tree
```

Cờ lệnh Flag	Tác dụng Effect
<code>--role <role></code>	Gán vai trò có sẵn kèm bản mô tả công việc <i>Assigns a predefined role with its written role contract</i>
<code>--workspace <dir></code>	Chỉ định không gian làm việc riêng <i>Names a separate workspace</i>
<code>--model <id></code>	Đặt mô hình chính cho riêng agent này <i>Sets this agent's own primary model</i>
<code>--agent-dir <dir></code>	Chỉ định thư mục trạng thái riêng <i>Names a separate state directory</i>
<code>--bind <channel[:accountId]></code>	Tạo ràng buộc ngay lúc thêm; lặp lại được nhiều lần <i>Creates a binding at the same time; may be repeated</i>
<code>--non-interactive</code>	Không hỏi gì; bắt buộc có <code>--workspace</code> trừ khi đã cho vai trò <i>Asks nothing; requires --workspace unless a role is supplied</i>

Một ràng buộc là một đối tượng gồm mã agent và điều kiện khớp. Điều kiện khớp có thể là cả một kênh, một tài khoản cụ thể của kênh, hoặc một người gửi cụ thể.

A binding is an object holding an agent id and a match condition. The condition can name a whole channel, one specific account of that channel, or one specific sender.

```
{
  bindings: [
    { agentId: "main",
      match: { channel: "telegram", accountId: "default" } },
    { agentId: "work",
      match: { channel: "slack", accountId: "*" } },
  ],
}
```

- Ràng buộc chi tiết hơn luôn thắng. Thứ tự từ hẹp tới rộng: người gửi chính xác, người gửi cha, ký tự đại diện người gửi, máy chủ kèm vai trò, máy chủ, nhóm làm việc, tài khoản, kênh, rồi mới tới agent mặc định.

The more specific binding always wins. The order runs from narrow to wide: exact peer, parent peer, peer wildcard, guild with roles, guild, team, account, channel, and finally the default agent.

- Nhiều ràng buộc cùng mức độ chi tiết thì cái đứng trước trong tệp cấu hình thắng.
When several bindings match at the same tier, the first one in configuration order wins.
- Ràng buộc bỏ trống `accountId` chỉ khớp tài khoản mặc định, KHÔNG khớp mọi tài khoản. Muốn bao cả kênh thì ghi `accountId: "*"` .
A binding that omits `accountId` matches only the default account, NOT every account. Use `accountId: "*"` for a channel-wide rule.
- Một ràng buộc khai nhiều điều kiện thì mọi điều kiện phải khớp đồng thời.
When a binding sets several match fields, all of them must match.

CẢNH BÁO / WARNING — TUYỆT ĐỐI không cho hai agent dùng chung `agentDir`.

Hai agent chung một thư mục trạng thái sẽ giẫm lên hồ sơ đăng nhập và kho phiên của nhau, sinh lỗi rất khó truy.

NEVER let two agents share an `agentDir`. Two agents on one state directory collide over auth profiles and session storage, producing faults that are very hard to trace.

GHI CHÚ / NOTE — Quyền truy cập tin nhắn riêng — ghép đôi hay danh sách trắng — đặt theo tài khoản kênh, không đặt theo agent. Hai agent gán vào cùng một số WhatsApp vẫn dùng chung một danh sách người được phép nhắn.

Direct-message access control — pairing or allowlist — is set per channel account, not per agent. Two agents bound to the same WhatsApp number still share one list of permitted senders.

8.4. Phiên làm việc: khoá phiên, phạm vi, khởi tạo lại và nén ngữ cảnh — *Sessions: keys, scope, resets and context compaction*

Phiên là nơi giữ mạch hội thoại. OpenClaw đưa mỗi tin nhắn vào một phiên dựa trên nơi tin nhắn đến từ đó. Mọi trạng thái phiên do gateway sở hữu; giao diện web chỉ hỏi gateway chứ không tự giữ bản riêng.

A session is where the thread of conversation lives. OpenClaw routes every inbound message into a session based on where it came from. All session state is owned by the gateway; the web interface queries the gateway rather than keeping its own copy.

Nguồn tin nhắn <i>Message source</i>	Phiên nào nhận <i>Which session receives it</i>
Tin nhắn riêng <i>Direct messages</i>	Dùng chung một phiên theo mặc định <i>Share one session by default</i>
Nhóm trò chuyện <i>Group chats</i>	Mỗi nhóm một phiên riêng <i>Isolated per group</i>
Phòng, kênh <i>Rooms and channels</i>	Mỗi phòng một phiên riêng <i>Isolated per room</i>
Tác vụ hẹn giờ <i>Cron jobs</i>	Phiên mới cho mỗi lần chạy <i>A fresh session per run</i>
Webhook	Mỗi móc một phiên riêng <i>Isolated per hook</i>

CẢNH BÁO / WARNING — Mặc định mọi tin nhắn riêng dồn chung một phiên. Nếu nhiều người nhắn được cho trợ lý của bạn, tất cả họ chung một mạch hội thoại — tin của người này hiện trong ngữ cảnh của người kia. Đặt `session.dmScope: "per-channel-peer"` trước khi mở cho người thứ hai.

By default every direct message lands in one shared session. If more than one person can message your assistant, they all share one conversation — one person's messages appear in another's context. Set `session.dmScope: "per-channel-peer"` before opening it to a second person.

Giá trị session.dmScope <i>session.dmScope value</i>	Cách chia phiên <i>How sessions are split</i>
<code>main</code> (mặc định) <code>main</code> (default)	Mọi tin nhắn riêng chung một phiên chính <i>All direct messages share the main session</i>
<code>per-peer</code>	Tách theo người gửi, gộp các kênh lại <i>Isolated by sender, across channels</i>
<code>per-channel-peer</code>	Tách theo kênh và người gửi — lựa chọn được khuyến nghị <i>Isolated by channel and sender — the recommended setting</i>
<code>per-account-channel-peer</code>	Tách theo tài khoản, kênh và người gửi <i>Isolated by account, channel and sender</i>

Khởi tạo lại phiên — *Resetting a session*

Phiên dùng mãi cho tới khi bạn tự khởi tạo lại hoặc bật chính sách khởi tạo tự động. Mặc định là không tự khởi tạo lại: phiên giữ nguyên mã, việc nén ngữ cảnh lo phần hội thoại phình ra.

A session is reused until you reset it yourself or opt into an automatic reset policy. The default is no automatic reset: the session keeps its id and compaction manages the growing context.

Chế độ <i>Mode</i>	Hành vi <i>Behaviour</i>
<code>none</code> (mặc định) <code>none</code> (default)	Không bao giờ tự khởi tạo lại <i>Never resets on its own</i>
<code>daily</code>	Sang phiên mới vào một giờ địa phương cố định, <code>session.reset.atHour</code> , mặc định 4 giờ sáng; tính theo lúc phiên hiện tại bắt đầu <i>Rolls into a new session at a fixed local hour, <code>session.reset.atHour</code>, default 4; measured from when the current session started</i>
<code>idle</code>	Sang phiên mới sau <code>session.reset.idleMinutes</code> phút không có tương tác thật của người dùng <i>Rolls into a new session after <code>session.reset.idleMinutes</code> of no real user interaction</i>
Bảng tay <i>Manual</i>	Gõ <code>/new</code> hoặc <code>/reset</code> trong khung chat; <code>/new <model></code> vừa mở phiên mới vừa đổi mô hình <i>Type <code>/new or /reset</code> in the chat; <code>/new <model></code> starts a new session and switches model at once</i>

Khai đồng thời cả hai chế độ tự động thì cái nào tới hạn trước sẽ thắng. Lướt chạy của nhịp tim, của tác vụ hẹn giờ và của sự kiện chạy lệnh có ghi dữ liệu phiên nhưng KHÔNG làm phiên trở lại — nghĩa là một trợ lý chỉ chạy nhịp tim suốt đêm vẫn bị khởi tạo lại đúng giờ.

With both automatic modes configured, whichever expires first wins. Heartbeat, cron and exec system-event turns write session metadata but do NOT refresh reset freshness — so an assistant that only ran heartbeats overnight is still reset on schedule.

```
{
  session: {
    dmScope: "per-channel-peer",
    reset: { mode: "daily", atHour: 4 },
    resetByType: { group: { mode: "idle", idleMinutes: 120 } },
  },
}
```

Nén ngữ cảnh bằng lệnh `/compact` — *Compacting context with `/compact`*

Hội thoại dài dần sẽ chạm trần của sổ ngữ cảnh của mô hình. Lệnh `/compact` tóm lại phần đã nói thành một bản gọn rồi chạy tiếp trên bản gọn đó, và báo lại ngân sách còn dư. Thêm chỉ dẫn phía sau, ví dụ `/compact` giữ lại các quyết định và số liệu, để nói rõ phần nào không được bỏ.

A long conversation eventually hits the model's context window. `/compact` summarises what has been said into a shorter form, continues from that form and reports the remaining budget. Add instructions after the command, for example `/compact keep the decisions and the figures, to say what must not be dropped.`

GHI CHÚ / NOTE — Hàng phiên và bản ghi hội thoại nằm trong

`~/.openclaw/agents/<agentId>/agent/openclaw-agent.sqlite`. Bản lưu trữ cũ nằm ở `~/.openclaw/agents/<agentId>/sessions/`. Chép máy khác thì chép tệp SQLite, đừng chép thư mục lưu trữ.

Session rows and transcripts live in `~/.openclaw/agents/<agentId>/agent/openclaw-agent.sqlite`. Older archives sit in `~/.openclaw/agents/<agentId>/sessions/`. To move to another machine, copy the SQLite file rather than the archive folder.

8.5. Bốn chế độ quyền của phiên làm việc — The four session permission modes

Chế độ quyền quyết định hai việc cho một phiên: agent được đọc ghi trong phạm vi nào, và ai duyệt khi nó xin chạy một lệnh ngoài phạm vi cho phép. Ranh giới là thư mục gốc của phiên — `sessionRoot` đã ghi nhận, hoặc không gian làm việc của agent khi phiên chưa ghi nhận gốc nào.

A permission mode decides two things for a session: where the agent may read and write, and who reviews an escalation when it asks to run a command outside that boundary. The boundary is the session root — the recorded `sessionRoot`, or the agent's workspace when no root has been recorded.

CHẾ ĐỘ	TRUY CẬP TỆP	AI DUYỆT KHI CẦN CHẠY LỆNH
read-only	Chỉ đọc trong thư mục phiên	Không — mọi lệnh bị từ chối
guarded	Đọc và ghi trong thư mục phiên	Người duyệt, sau bước danh sách cho phép
workspace	Đọc và ghi trong thư mục phiên	Mô hình xét duyệt; ba lần từ chối liên tiếp thì chuyển cho người
full	Toàn bộ ổ đĩa, không giới hạn	Không ai duyệt — chạy thẳng

Hình 21. Bốn chế độ quyền của phiên, từ chỉ đọc tới toàn quyền

Figure 21. The four session permission modes, from read-only to full access

Chế độ Mode	Quyền với tệp Filesystem access	Ai duyệt khi xin chạy lệnh Exec reviewer
read-only	Chỉ đọc trong phạm vi gốc phiên; các công cụ ghi bị gỡ khỏi danh mục <i>Reads under the session root only; the managed write tools are omitted</i>	Không ai; lệnh bị từ chối thẳng <i>None; exec is denied</i>
guarded	Đọc và ghi trong phạm vi gốc phiên <i>Reads and writes under the session root</i>	Một CON NGƯỜI, sau khi qua đường tắt danh sách trắng <i>A HUMAN, after the allowlist fast path</i>
workspace	Đọc và ghi trong phạm vi gốc phiên <i>Reads and writes under the session root</i>	Một mô hình ngôn ngữ soát lại: cho phép, từ chối, hoặc chuyển hỏi người <i>An LLM reviewer: allow, deny, or ask a human</i>
full	Toàn quyền hệ thống tệp <i>Unrestricted filesystem access</i>	Không ai <i>None</i>

Chế độ `full` đòi phạm vi quyền `operator.admin`; ba chế độ còn lại đòi `operator.write`. Chọn chế độ ngay trong khung chat qua trình đơn **Execution permissions**, đổi được giữa chừng công việc.

The full mode requires the operator.admin scope; the other three require operator.write. Pick a mode from the chat composer's Execution permissions menu; it can be changed in the middle of a task.

- Ở chế độ `workspace`, khi bộ soát bằng mô hình từ chối, lý do được trả về cho agent mà không sinh thẻ hỏi người. Agent phải chọn cách khác an toàn hơn hoặc hỏi bạn, không được tìm đường lách.

In workspace mode, a reviewer denial returns a reason to the agent without creating a human approval card. The agent must choose a materially safer alternative or ask you; it must not work around the denial.

- Ba lần bộ soát từ chối liên tiếp thì lệnh thứ ba được đẩy lên cho người duyệt. Một lần được duyệt sẽ đặt lại bộ đếm.

Three consecutive reviewer denials escalate the third command to a human. One approval resets the counter.

- Đổi chế độ giữa chừng sẽ huỷ các yêu cầu duyệt còn treo — huỷ chứ không phải cho qua. Việc đã ghi xong và tiến trình đã chạy thì không quay lại được.

Changing the mode mid-task cancels pending approvals — cancels them, not grants them. Writes already completed and processes already started are not rolled back.

CẢNH BÁO / WARNING — Không cấu hình gì, không bật hộp cát, thì mặc định là toàn quyền. Trước khi nối trợ lý vào một kênh có người lạ nhắn được, đặt chế độ quyền cho rõ ràng.

With nothing configured and no sandboxing, the default is full access. Set an explicit permission mode before connecting the assistant to a channel strangers can message.

8.6. Bộ công cụ theo hồ sơ: `minimal`, `messaging`, `coding`, `full` — Tool profiles: `minimal`, `messaging`, `coding` and `full`

Khoá `tools.profile` đặt danh sách trắng nền cho agent trước khi `tools.allow` và `tools.deny` can thiệp. Đây là cách thô nhất và nhanh nhất để giới hạn agent làm được những gì.

The tools.profile key sets a base allowlist for the agent before tools.allow and tools.deny are applied. It is the bluntest and fastest way to limit what an agent can do.

Hồ sơ <i>Profile</i>	Gồm những gì <i>What it includes</i>
<code>minimal</code>	Chỉ <code>session_status</code> và <code>gateway</code> với riêng quyền cập nhật <i>Only session_status and gateway with the update action alone</i>
<code>messaging</code>	Nhóm nhắn tin và nhóm phiên: gửi tin, liệt kê phiên, tra lịch sử, gọi agent con; hẹp, chỉ dành cho trò chuyện <i>The messaging and session groups: sending messages, listing sessions, reading history, spawning subagents; narrow, chat only</i>
<code>coding</code>	Nhóm tệp, nhóm thực thi, nhóm web, nhóm phiên, nhóm trí nhớ, hẹn giờ, mục tiêu, sinh ảnh nhạc video <i>The filesystem, runtime, web, session and memory groups, plus cron, goals and image, music and video generation</i>
<code>full</code>	Không lọc công cụ lõi và bật luôn công cụ tùy chọn của phần mở rộng. Đây là hồ sơ trình hướng dẫn đặt cho bản cài trên máy cá nhân <i>No core filtering, and optional plugin tools are selected as well. This is the profile local onboarding sets</i>

Bỏ trống `tools.profile` thì công cụ lõi không bị lọc, nhưng công cụ của phần mở rộng cũng không tự bật. Chọn `full` là chọn công cụ, KHÔNG phải cấp quyền thực thi — trình đơn **Execution permissions** ở mục 8.5 mới quyết định công cụ đó được làm gì.

Leaving tools.profile unset also leaves core tools unfiltered, but does not opt into optional plugin tools. Choosing full selects tools; it does NOT grant execution permissions — the Execution permissions menu from section 8.5 decides what those tools may do.

GHI CHÚ / NOTE — Ba hồ sơ hẹp đều kèm công cụ `gateway` nhưng chỉ với một hành động: chạy cập nhật. Chủ máy vẫn xin cập nhật OpenClaw qua chat được mà không mở quyền đọc cấu hình.

The three narrow profiles all include the gateway tool with one action only: running an update. The owner can still ask for an OpenClaw update from chat without opening up configuration reads.

8.7. Nhịp tim: cho trợ lý chủ động lên tiếng — *Heartbeat: letting the assistant speak first*

Nhịp tim là lượt chạy định kỳ trong phiên chính, để mô hình tự rà xem có gì cần báo mà không phải chờ bạn hỏi. Đây là thứ biến OpenClaw từ một hộp hỏi đáp thành một trợ lý theo dõi công việc.

The heartbeat is a periodic turn in the main session, letting the model check whether anything needs raising without waiting to be asked. It is what turns OpenClaw from a question box into an assistant that watches the work.

Khoá <i>Key</i>	Mặc định <i>Default</i>	Ý nghĩa <i>Meaning</i>
<code>heartbeat.every</code>	30m, hoặc 1h khi dùng đăng nhập Anthropic 30m, or 1h with Anthropic OAuth or token auth	Khoảng cách giữa hai lượt chạy; đặt "0m" để tắt nhịp định kỳ <i>The interval between runs; set "0m" to disable the recurring cadence</i>
<code>heartbeat.target</code>	owner	Nơi gửi thông báo: mặc định là tin nhắn riêng của chủ máy <i>Where alerts go: by default the owner's direct message</i>
<code>heartbeat.directPolicy</code>	allow	Đặt block để chặn hẳn việc gửi tới đích tin nhắn riêng <i>Set it to block to suppress sending to direct-message targets</i>
<code>heartbeat.lightContext</code>	Tắt <i>Off</i>	Bỏ qua các tệp định danh của không gian làm việc trong lượt nhịp tim <i>Skips the workspace bootstrap files for heartbeat runs</i>
<code>heartbeat.isolateSession</code>	Tắt <i>Off</i>	Mỗi lượt một phiên mới, không mang theo lịch sử hội thoại <i>A fresh session each run, carrying no conversation history</i>
<code>heartbeat.activeHours</code>	Không đặt <i>Unset</i>	Giới hạn nhịp tim trong khung giờ địa phương, ví dụ từ 08:00 tới 24:00 <i>Restricts heartbeats to a local time window, for example 08:00 to 24:00</i>

Nhịp tim gửi thông báo về tin nhắn riêng của chủ máy, nên phải khai chủ máy là ai. Danh sách chỉ toàn ký tự đại diện không xác định được chủ.

The heartbeat sends its alerts to the owner's direct message, so the owner must be declared. A wildcard-only allowlist does not identify one.

```
openclaw config set commands.ownerAllowFrom '["telegram:123456789"]'
openclaw cron list --all
openclaw system heartbeat last
openclaw system heartbeat disable
```

- Đặt "0m" chỉ tắt nhịp định kỳ. Một sự kiện có đích vẫn đánh thức được một lượt chạy, ví dụ khi một lệnh chạy nền kết thúc. Muốn tắt cả phần đó thì đặt `tools.exec.notifyOnExit: false`.
Setting "0m" disables the recurring cadence only. A targeted event can still wake one turn, for example when a background command finishes. To stop that too, set
`tools.exec.notifyOnExit: false`.
- Lượt đánh thức theo sự kiện bị chặn tần suất: tối thiểu 30 giây giữa hai lượt, và một chốt chống dồn dập khi có năm lượt khởi động trong vòng 60 giây.

Event-driven wakes are rate limited: a 30-second minimum between turns and a flood guard after five starts within 60 seconds.

- Trong bản ghi hội thoại, lượt nhịp tim được đánh dấu `[OpenClaw heartbeat poll]`, phân biệt với lượt do lệnh kết thúc, tác vụ hẹn giờ hay sự kiện phiên.
In the transcript, a heartbeat turn is marked `[OpenClaw heartbeat poll]`, distinct from an exec completion, a cron wake or a session event.
- Agent trả về `NO_REPLY` thì tin nhắn đi ra bị chặn — đây là cách trợ lý im lặng khi không có gì đáng báo.
When the agent returns `NO_REPLY`, the outbound message is blocked — this is how the assistant stays quiet when there is nothing worth raising.

CẢNH BÁO / WARNING — Nhịp tim là lượt gọi mô hình có tính tiền, chạy đều đặn cả ngày lẫn đêm. Trong lúc còn đang thử nghiệm, đặt `agents.defaults.heartbeat.every: "0m"` rồi bật lại khi đã yên tâm về chi phí.
Every heartbeat is a billed model call, running day and night. While still evaluating, set `agents.defaults.heartbeat.every: "0m"` and turn it back on once the cost is understood.

8.8. Bộ chạy agent: runtime nhúng, Codex, Claude CLI, Gemini CLI và Copilot — Agent runtimes: the embedded runtime, Codex, Claude CLI, Gemini CLI and Copilot

Bộ chạy agent là lớp thực sự điều khiển vòng lặp mô hình: nhận lời nhắc, đẩy mô hình sinh chữ, xử lý các lượt gọi công cụ, rồi trả lượt đã xong về cho OpenClaw. Nó dễ bị lẫn với nhà cung cấp vì cả hai đều hiện ra quanh phần cấu hình mô hình, nhưng đó là hai tầng khác nhau.

The agent runtime is the layer that actually drives the model loop: it takes the prompt, pushes the model to produce output, handles native tool calls and returns the finished turn to OpenClaw. It is easily confused with the provider because both appear near the model configuration, but they are different layers.

Tầng Layer	Ví dụ Examples	Nghĩa là gì What it means
Nhà cung cấp Provider	<code>anthropic, openai, github-copilot</code>	Cách OpenClaw đăng nhập, dò mô hình và gọi tên mô hình <i>How OpenClaw authenticates, discovers models and names model refs</i>
Mô hình Model	<code>claude-opus-4-6, gpt-6-astra</code>	Mô hình được chọn cho lượt chạy <i>The model selected for the turn</i>
Bộ chạy agent Agent runtime	<code>openclaw, codex, claude-cli, copilot</code>	Vòng lặp cấp thấp thực thi lượt đã chuẩn bị <i>The low-level loop that executes the prepared turn</i>
Kênh Channel	Discord, Slack, Telegram, WhatsApp	Nơi tin nhắn đi vào và đi ra khỏi OpenClaw <i>Where messages enter and leave OpenClaw</i>

Có hai họ bộ chạy. Họ nhúng chạy bên trong vòng lặp agent của OpenClaw: `openclaw` dựng sẵn, cộng các phần mở rộng như `codex` và `copilot`. Họ dòng lệnh chạy một tiến trình CLI cục bộ trong khi tên mô hình vẫn giữ nguyên dạng chuẩn: khai `anthropic/claude-opus-5` kèm `agentRuntime.id: "claude-cli"` nghĩa là chọn mô hình của Anthropic nhưng thực thi qua Claude CLI.

There are two runtime families. Embedded harnesses run inside OpenClaw's own agent loop: the built-in `openclaw`, plus plugin harnesses such as `codex` and `copilot`. CLI backends run a local CLI process while the model reference stays canonical: `anthropic/claude-opus-5` with `agentRuntime.id: "claude-cli"` means select the Anthropic model but execute it through Claude CLI.

CẢNH BÁO / WARNING — Khoá `agentRuntime.id` đặt trên từng nhà cung cấp hoặc từng mô hình. Các khoá bộ chạy đặt cho cả agent là dạng cũ và bị bỏ qua. Chạy `openclaw doctor --fix` để gỡ chúng và viết lại tham chiếu mô hình theo dạng chuẩn.

The `agentRuntime.id` key belongs on a provider or a model entry. Whole-agent runtime keys are legacy and are ignored. Run `openclaw doctor --fix` to remove them and rewrite the model references into canonical form.

GHI CHÚ / NOTE — Chọn bộ chạy không quyết định việc tính tiền. Khoá API của OpenAI và tài khoản thuê bao ChatGPT hay Codex vẫn là hai nguồn thanh toán riêng biệt, dù cùng chạy qua một bộ chạy.

Runtime selection does not determine billing. An OpenAI API key and a ChatGPT or Codex subscription remain separate payment sources even when both run through the same runtime.

Chương 9. Công cụ và kỹ năng — Chapter 9. Tools and skills

Mô hình ngôn ngữ chỉ sinh ra chữ. Muốn trợ lý đọc được tệp, chạy được lệnh, gửi được tin nhắn hay tra được web, nó phải gọi công cụ. Công cụ là các hàm OpenClaw phơi ra cho mô hình; kỹ năng là các tệp văn bản dạy mô hình khi nào và dùng công cụ nào. Hai thứ này quyết định trợ lý làm được việc hay chỉ nói chuyện.

A language model only produces text. For the assistant to read a file, run a command, send a message or search the web, it must call a tool. Tools are the functions OpenClaw exposes to the model; skills are text files that teach the model when and how to use them. Together they decide whether the assistant does work or merely talks.

9.1. Công cụ exec: agent chạy lệnh thật trên máy của bạn — The exec tool: the agent runs real commands on your machine

Trong toàn bộ danh mục công cụ, `exec` là thứ mạnh nhất và cũng nguy hiểm nhất. Nó chạy một lệnh shell trong không gian làm việc, và lệnh shell thì tạo, sửa hoặc xoá được tệp ở bất cứ đâu mà máy chủ hoặc hộp cát cho phép. Tất các công cụ tệp như `write`, `edit`, `apply_patch` KHÔNG làm `exec` trở thành chỉ đọc.

Of all the tools, exec is the most powerful and the most dangerous. It runs a shell command in the workspace, and a shell command can create, edit or delete files anywhere the selected host or sandbox filesystem permits. Disabling file tools such as write, edit and apply_patch does NOT make exec read-only.

Chế độ chạy lệnh <i>Exec mode</i>	Hành vi <i>Behaviour</i>
<code>deny</code>	Từ chối mọi lệnh <i>Exec is denied outright</i>
<code>allowlist</code>	Chỉ lệnh nằm trong danh sách trắng được chạy; phần còn lại bị chặn, không hỏi ai <i>Only allowlisted or safe-bin commands run; everything else is blocked without asking</i>
<code>ask</code>	Lệnh trong danh sách trắng chạy thẳng; mọi lệnh khác hỏi một con người <i>Allowlisted commands run directly; everything else asks a human</i>
<code>auto</code>	Lệnh trong danh sách trắng chạy thẳng; lệnh lạ được một mô hình soát và trả về cho phép, từ chối, hoặc chuyển hỏi người <i>Allowlisted commands run directly; a miss is reviewed by a model that returns allow, deny, or ask a human</i>
<code>full</code>	Không hỏi gì theo chính sách thông thường <i>No ordinary policy prompts at all</i>

Mặc định của máy chủ gateway và của nút thiết bị là `full`, nghĩa là không hỏi duyệt. Đây là điểm người mới hay bỏ sót. Khóa `tools.exec.host` chọn nơi lệnh chạy: `auto` quy về `sandbox` khi hộp cát đang bật và về `gateway` khi không; đặt thẳng `gateway` hoặc `node` để ép. Hạn giờ mặc định mỗi lệnh là 1.800 giây, đổi bằng `tools.exec.timeoutSeconds`.

The default for the gateway host and for a node is full, which means no approval prompts. New users regularly miss this. The tools.exec.host key picks where a command runs: auto resolves to sandbox when a sandbox is active and to gateway otherwise; set gateway or node to force it. The default per-command timeout is 1,800 seconds, changed with tools.exec.timeoutSeconds.

Trong khung chat, chỉ thị `/exec` cho bạn xem và đổi thiết lập của riêng lượt chạy đó.

In the chat, the /exec directive shows and changes the settings for a single run.

```
/exec
/exec host=node node=worker-1
/exec security=allowlist ask=always kiểm tra kết quả build
```

- Gửi `/exec host=... node=...` một mình thì phân chọn nơi chạy được nhớ lại cho cả phiên.
Sending /exec host=... node=... on its own persists the placement choice for the session.
- Hai tham số `security` và `ask` chỉ áp cho ĐÚNG tin nhắn đó, không giữ sang lượt sau. Muốn đặt chính sách cho cả phiên thì dùng chế độ quyền ở mục 8.5.
The security and ask options apply to THAT message only and never carry to later turns. For a session-wide policy, use the permission modes of section 8.5.
- Chỉ thị này chỉ nghe lời người gửi đã được uỷ quyền qua danh sách trắng kênh, qua ghép đôi hoặc qua nhóm quyền truy cập.
The directive is honoured only for senders authorised through channel allowlists, pairing or access groups.
- Khi phiên đã có chế độ quyền, `/exec` chỉ SIẾT thêm được chứ không nói ra. Gõ `/exec security=deny` là chặn lệnh cho lượt đó ngay cả trong phiên toàn quyền.
When a session already has a permission mode, /exec can only TIGHTEN it, never loosen it. /exec security=deny blocks exec for that turn even inside a full-access session.

CẢNH BÁO / WARNING — Muốn tắt hẳn khả năng chạy lệnh thì phải chặn bằng chính sách công cụ: `tools.deny: ["exec"]`. Đặt `tools.exec.mode` thành `deny` là chính sách chạy lệnh, không phải là gỡ công cụ khỏi tay agent.

To hard-disable command execution, deny it through the tool policy: tools.deny: ["exec"]. Setting tools.exec.mode to deny is an execution policy, not a removal of the tool from the agent's hands.

GHI CHÚ / NOTE — Khóa `tools.elevated` là cửa thoát cho phép chạy lệnh NGOÀI hộp cát, khai kèm danh sách người được dùng theo từng kênh. Chỉ thị `/elevated on|off|ask|full` bật tắt theo phiên. Đây là quyền cao nhất trong hệ thống; chỉ mở cho

chính bạn.

The `tools.elevated` key is the escape hatch that allows commands to run OUTSIDE the sandbox, declared together with a per-channel list of who may use it. The `/elevated on|off|ask|full` directive toggles it per session. This is the highest privilege in the system; open it to yourself only.

9.2. Chính sách công cụ: cho phép và cấm — Tool policy: allow and deny

Trên nền hồ sơ công cụ ở mục 8.6, hai khoá `tools.allow` và `tools.deny` chỉnh lại từng công cụ một. Luật đơn giản: cấm thắng cho phép. Hai khoá này không phân biệt hoa thường và nhận ký tự đại diện `*`. Chúng được áp ngay cả khi hộp cát đang tắt, và áp TRƯỚC luật của hộp cát — công cụ đã bị cấm thì hộp cát không mở lại được.

On top of the tool profile from section 8.6, the `tools.allow` and `tools.deny` keys adjust individual tools. The rule is simple: deny wins. Both keys are case-insensitive and accept the `*` wildcard. They apply even when the sandbox is off, and they apply BEFORE the sandbox rules — a denied tool cannot be reopened by the sandbox.

Công cụ được gom thành nhóm để khai cho gọn. Cấm cả một nhóm bằng một dòng.

Tools are collected into groups so policy stays short. One line denies a whole group.

Nhóm <i>Group</i>	Công cụ trong nhóm <i>Tools in the group</i>
<code>group:runtime</code>	<code>exec, process, code_execution</code>
<code>group:fs</code>	<code>read, write, edit, apply_patch</code>
<code>group:web</code>	<code>web_search, x_search, web_fetch</code>
<code>group:memory</code>	<code>memory_search, memory_get</code>
<code>group:messaging</code>	<code>message</code>
<code>group:media</code>	<code>view_image, image_generate, music_generate, video_generate, tts, pdf</code>
<code>group:ui</code>	<code>browser, screen, theme, dashboard, terminal, portal, canvas, show_widget</code>
<code>group:nodes</code>	<code>nodes, computer</code>
<code>group:plugins</code>	Công cụ của mọi phần mở rộng đang nạp <i>Every tool owned by a loaded plugin</i>

```
{
  tools: {
    profile: "coding",
    deny: ["browser", "canvas"],
    byProvider: { anthropic: { profile: "minimal" } },
  },
}
```

Thứ tự áp dụng là: hồ sơ nền, rồi hồ sơ theo nhà cung cấp `tools.byProvider`, rồi cho phép và cấm. Khoá `tools.toolsBySender` siết thêm theo người gửi của lượt hiện tại, khai theo dạng `channel:<channelId>:<senderId>, id:<senderId>, e164:<số điện thoại>, username:<tên> hoặc "*"`.

The order is: base profile, then the per-provider profile in `tools.byProvider`, then allow and deny. The `tools.toolsBySender` key narrows further by the requester of the current turn, keyed as `channel:<channelId>:<senderId>, id:<senderId>, e164:<phone>, username:<handle> or "*"`.

CẢNH BÁO / WARNING — Hai công cụ `write` và `apply_patch` là hai mã riêng. Cấm `write` KHÔNG cấm `apply_patch`. Muốn chặn mọi đường ghi tệp thì cấm `group:fs`, hoặc liệt kê đủ cả ba: `write, edit, apply_patch`.
`write` and `apply_patch` are separate tool ids. Denying `write` does NOT deny `apply_patch`. To block every path to a file write, deny `group:fs`, or list all three: `write, edit, apply_patch`.

GHI CHÚ / NOTE — Trong cùng một phạm vi cấu hình, không được khai đồng thời `allow` và `alsoAllow`; bộ kiểm cấu hình sẽ từ chối và gateway không khởi động. Gộp `alsoAllow` vào `allow`, hoặc bỏ `allow` và dùng cặp `profile` cộng `alsoAllow`.
Within one configuration scope, `allow` and `alsoAllow` cannot both be set; validation rejects it and the gateway refuses to start. Merge `alsoAllow` into `allow`, or drop `allow` and use `profile` together with `alsoAllow`.

9.3. Kỹ năng là gì và tệp SKILL.md viết ra sao — *What a skill is and how SKILL.md is written*

Kỹ năng là tệp văn bản dạy agent cách và lúc nào dùng công cụ. Mỗi kỹ năng là một THƯ MỤC chứa tệp `SKILL.md`: phần đầu là khối thông tin YAML kẹp giữa hai dòng ba dấu gạch ngang, phần sau là nội dung hướng dẫn viết bằng markdown thường. OpenClaw theo chuẩn mở AgentSkills, nên kỹ năng viết cho OpenClaw dùng lại được ở nơi khác và ngược lại.

A skill is a text file that teaches the agent how and when to use a tool. Each skill is a DIRECTORY containing a `SKILL.md` file: a YAML frontmatter block between two triple-dash lines first, then the instructions as ordinary markdown. OpenClaw follows the open AgentSkills spec, so a skill written for OpenClaw can be reused elsewhere and the other way round.



Hình 22. Cấu trúc một tệp SKILL.md: khối YAML khai tên và mô tả, thân markdown là phần hướng dẫn agent

Figure 22. The structure of a SKILL.md file: a YAML block declaring the name and description, with the markdown body holding the instructions

Bản tối thiểu chỉ cần hai trường: `name` và `description`.

A minimal skill needs two fields only: `name` and `description`.

```

---
name: image-lab
description: Generate or edit images via a provider-backed
  image workflow
---

When the user asks to generate an image, use the
`image_generate` tool...

Reference files stored beside this skill with {baseDir}.

```

Trường `description` là thứ mô hình đọc để quyết định có gọi kỹ năng này hay không. Viết nó như một câu tả rõ việc và hoàn cảnh sử dụng, đừng viết một cụm danh từ cụt lùn.

The `description` field is what the model reads to decide whether to invoke the skill. Write it as a sentence naming the task and the circumstances, not as a bare noun phrase.

CẢNH BÁO / WARNING — Tên kỹ năng và lệnh gạch chéo của nó lấy từ trường `name`, hoặc từ tên thư mục khi thiếu `name`. Đường dẫn thư mục chỉ để sắp xếp cho gọn, không tham gia đặt tên.

The skill's name and its slash command come from the `name` field, or from the directory name when `name` is missing. The folder path is for organisation only and plays no part in naming.

9.4. Các trường khai báo trong SKILL.md — *The frontmatter keys of SKILL.md*

Trường Key	Mặc định Default	Tác dụng Effect
<code>name</code>	Bắt buộc <i>Required</i>	Tên kỹ năng, cũng là tên lệnh gạch chéo <i>The skill name, which is also the slash command name</i>
<code>description</code>	Bắt buộc <i>Required</i>	Mô tả để mô hình biết khi nào dùng <i>The description the model reads to know when to use it</i>
<code>homepage</code>	Không đặt <i>Unset</i>	Địa chỉ web hiện ở mục Website trong giao diện kỹ năng <i>A URL shown as Website in the skills interface</i>
<code>user-invocable</code>	<code>true</code>	Phơi kỹ năng thành một lệnh gạch chéo cho người dùng gọi <i>Exposes the skill as a user-invocable slash command</i>
<code>disable-model-invocation</code>	<code>false</code>	Đặt <code>true</code> để giữ hướng dẫn của kỹ năng ra ngoài lời nhắc thường; kỹ năng vẫn gọi được bằng lệnh gạch chéo <i>Set it to true to keep the skill's instructions out of the normal prompt; the skill remains available as a slash command</i>
<code>command-dispatch</code>	Không đặt <i>Unset</i>	Đặt <code>tool</code> để lệnh gạch chéo đi thẳng tới một công cụ, bỏ qua mô hình <i>Set it to tool so the slash command dispatches straight to a tool, bypassing the model</i>
<code>os</code>	Không đặt <i>Unset</i>	Lọc cứng theo hệ điều hành: <code>darwin</code> , <code>linux</code> , <code>win32</code> <i>A hard platform filter: darwin, linux or win32</i>

Khởi `metadata.openclaw` lọc kỹ năng ngay lúc nạp. Kỹ năng thiếu thứ nó cần sẽ không được nạp thay vì nạp rồi hỏng giữa chừng.

The `metadata.openclaw` block filters skills at load time. A skill missing what it needs is not loaded, rather than loading and failing halfway.

```
metadata:
{
  "openclaw":
  {
    "requires": { "bins": ["uv"],
                  "env": ["GEMINI_API_KEY"],
                  "config": ["browser.enabled"] },
    "primaryEnv": "GEMINI_API_KEY",
  },
}
```

Điều kiện <i>Requirement</i>	Phải thỏa gì <i>What must be true</i>
<code>requires.bins</code>	Mọi chương trình liệt kê phải có trên đường dẫn <code>PATH</code> <i>Every listed binary must exist on PATH</i>
<code>requires.anyBins</code>	Ít nhất một chương trình trong danh sách có trên <code>PATH</code> <i>At least one listed binary must exist on PATH</i>
<code>requires.env</code>	Mọi biến môi trường liệt kê phải tồn tại <i>Every listed environment variable must exist</i>
<code>requires.config</code>	Mọi đường dẫn khoá trong <code>openclaw.json</code> phải có giá trị đúng <i>Every listed openclaw.json path must be truthy</i>
<code>primaryEnv</code>	Tên biến môi trường gắn với <code>skills.entries.<name>.apiKey</code> <i>The environment variable name tied to skills.entries.<name>.apiKey</i>
<code>always</code>	Đặt <code>true</code> để bỏ qua mọi điều kiện trên, trừ lọc hệ điều hành <i>Set it to true to bypass all of the above, except the OS filter</i>

MẸO / TIP — Trong thân kỹ năng, viết `{baseDir}` để trỏ tới chính thư mục chứa kỹ năng đó. Nhờ vậy kỹ năng kèm theo tệp mẫu, tệp dữ liệu hay script phụ mà không phải ghi đường dẫn tuyệt đối — đường dẫn tuyệt đối sẽ sai ngay khi chuyển sang máy khác.

Inside the skill body, write {baseDir} to point at the skill's own folder. A skill can then ship with templates, data files or helper scripts without hard-coding an absolute path — an absolute path breaks the moment the skill moves to another machine.

9.5. Thứ tự nạp kỹ năng và mức ưu tiên — Skill loading order and precedence

OpenClaw quét nhiều thư mục để tìm kỹ năng. Trùng tên thì nguồn có mức ưu tiên cao hơn thắng, nguồn thấp hơn bị che. Nhờ vậy bạn ghi đè được một kỹ năng đóng gói sẵn bằng bản sửa riêng đặt trong không gian làm việc, mà không phải sửa vào bản cài.

OpenClaw scans several directories for skills. When a name appears twice, the higher-precedence source wins and the lower one is shadowed. This lets you override a bundled skill with your own edited copy in the workspace, without touching the installation.