

[Help](#)

FAQ

Quick answers plus deeper troubleshooting for real-world setups (local dev, VPS, multi-agent, OAuth/API keys, model failover). For runtime diagnostics, see [Troubleshooting](#). For the full config reference, see [Configuration](#).

Table of contents

[Quick start and first-run setup]

[I'm stuck what's the fastest way to get unstuck?](#)

[What's the recommended way to install and set up OpenClaw?](#)

[How do I open the dashboard after onboarding?](#)

[How do I authenticate the dashboard \(token\) on localhost vs remote?](#)

[What runtime do I need?](#)

[Does it run on Raspberry Pi?](#)

[Any tips for Raspberry Pi installs?](#)

[It is stuck on “wake up my friend” / onboarding will not hatch.](#)

[What now?](#)

[Can I migrate my setup to a new machine \(Mac mini\) without redoing onboarding?](#)

[Where do I see what is new in the latest version?](#)

[I can't access docs.openclaw.ai \(SSL error\). What now?](#)



What's the difference between stable and beta?

How do I install the beta version, and what's the difference between beta and dev?

How do I try the latest bits?

How long does install and onboarding usually take?

Installer stuck? How do I get more feedback?

Windows install says git not found or openclaw not recognized

The docs didn't answer my question - how do I get a better answer?

How do I install OpenClaw on Linux?

How do I install OpenClaw on a VPS?

Where are the cloud/VPS install guides?

Can I ask OpenClaw to update itself?

What does the onboarding wizard actually do?

Do I need a Claude or OpenAI subscription to run this?

Can I use Claude Max subscription without an API key

How does Anthropic "setup-token" auth work?

Where do I find an Anthropic setup-token?

Do you support Claude subscription auth (Claude Pro or Max)?

Why am I seeing HTTP 429: rate_limit_error from Anthropic?

Is AWS Bedrock supported?

How does Codex auth work?

Do you support OpenAI subscription auth (Codex OAuth)?

How do I set up Gemini CLI OAuth

Is a local model OK for casual chats?

How do I keep hosted model traffic in a specific region?

Do I have to buy a Mac Mini to install this?



Do I need a Mac mini for iMessage support?

If I buy a Mac mini to run OpenClaw, can I connect it to my MacBook Pro?

Can I use Bun?

Telegram: what goes in allowFrom ?

Can multiple people use one WhatsApp number with different OpenClaw instances?

Can I run a “fast chat” agent and an “Opus for coding” agent?

Does Homebrew work on Linux?

What's the difference between the hackable (git) install and npm install?

Can I switch between npm and git installs later?

Should I run the Gateway on my laptop or a VPS?

How important is it to run OpenClaw on a dedicated machine?

What are the minimum VPS requirements and recommended OS?

Can I run OpenClaw in a VM and what are the requirements

What is OpenClaw?

What is OpenClaw, in one paragraph?

What's the value proposition?

I just set it up what should I do first

What are the top five everyday use cases for OpenClaw

Can OpenClaw help with lead gen outreach ads and blogs for a SaaS

What are the advantages vs Claude Code for web development?

Skills and automation

How do I customize skills without keeping the repo dirty?

Can I load skills from a custom folder?

How can I use different models for different tasks?



The bot freezes while doing heavy work. How do I offload that?

Cron or reminders do not fire. What should I check?

How do I install skills on Linux?

Can OpenClaw run tasks on a schedule or continuously in the background?

Can I run Apple macOS-only skills from Linux?

Do you have a Notion or HeyGen integration?

How do I install the Chrome extension for browser takeover?

Sandboxing and memory

Is there a dedicated sandboxing doc?

How do I bind a host folder into the sandbox?

How does memory work?

Memory keeps forgetting things. How do I make it stick?

Does memory persist forever? What are the limits?

Does semantic memory search require an OpenAI API key?

Where things live on disk

Is all data used with OpenClaw saved locally?

Where does OpenClaw store its data?

Where should AGENTS.md / SOUL.md / USER.md / MEMORY.md live?

What's the recommended backup strategy?

How do I completely uninstall OpenClaw?

Can agents work outside the workspace?

I'm in remote mode - where is the session store?

Config basics

What format is the config? Where is it?

I set gateway.bind: "lan" (or "tailnet") and now nothing listens / the UI says unauthorized



Why do I need a token on localhost now?

Do I have to restart after changing config?

How do I enable web search (and web fetch)?

config.apply wiped my config. How do I recover and avoid this?

How do I run a central Gateway with specialized workers across devices?

Can the OpenClaw browser run headless?

How do I use Brave for browser control?

Remote gateways and nodes

How do commands propagate between Telegram, the gateway, and nodes?

How can my agent access my computer if the Gateway is hosted remotely?

Tailscale is connected but I get no replies. What now?

Can two OpenClaw instances talk to each other (local + VPS)?

Do I need separate VPSes for multiple agents

Is there a benefit to using a node on my personal laptop instead of SSH from a VPS?

Do nodes run a gateway service?

Is there an API / RPC way to apply config?

What's a minimal "sane" config for a first install?

How do I set up Tailscale on a VPS and connect from my Mac?

How do I connect a Mac node to a remote Gateway (Tailscale Serve)?

Should I install on a second laptop or just add a node?

Env vars and .env loading

How does OpenClaw load environment variables?



"I started the Gateway via the service and my env vars disappeared." What now?

I set COPILOT_GITHUB_TOKEN , but models status shows "Shell env: off." Why?

Sessions and multiple chats

How do I start a fresh conversation?

Do sessions reset automatically if I never send /new ?

Is there a way to make a team of OpenClaw instances one CEO and many agents

Why did context get truncated mid-task? How do I prevent it?

How do I completely reset OpenClaw but keep it installed?

I'm getting "context too large" errors - how do I reset or compact?

Why am I seeing "LLM request rejected:

messages.N.content.X.tool_use.input: Field required"?

Why am I getting heartbeat messages every 30 minutes?

Do I need to add a "bot account" to a WhatsApp group?

How do I get the JID of a WhatsApp group?

Why doesn't OpenClaw reply in a group?

Do groups/threads share context with DMs?

How many workspaces and agents can I create?

Can I run multiple bots or chats at the same time (Slack) , and how should I set that up?

Models: defaults, selection, aliases, switching

What is the "default model"?

What model do you recommend?

How do I switch models without wiping my config?

Can I use self-hosted models (llama.cpp, vLLM, Ollama)?



What do OpenClaw, Flawd, and Krill use for models?

How do I switch models on the fly (without restarting)?

Can I use GPT 5.2 for daily tasks and Codex 5.3 for coding

Why do I see “Model ... is not allowed” and then no reply?

Why do I see “Unknown model: minimax/MiniMax-M2.1”?

Can I use MiniMax as my default and OpenAI for complex tasks?

Are opus / sonnet / gpt built-in shortcuts?

How do I define/override model shortcuts (aliases)?

How do I add models from other providers like OpenRouter or Z.AI?

Model failover and “All models failed”

How does failover work?

What does this error mean?

Fix checklist for No credentials found for profile "anthropic:default"

Why did it also try Google Gemini and fail?

Auth profiles: what they are and how to manage them

What is an auth profile?

What are typical profile IDs?

Can I control which auth profile is tried first?

OAuth vs API key: what's the difference?

Gateway: ports, “already running”, and remote mode

What port does the Gateway use?

Why does openclaw gateway status say Runtime: running but RPC probe: failed ?

Why does openclaw gateway status show Config (cli) and Config (service) different?

What does “another gateway instance is already listening” mean?



How do I run OpenClaw in remote mode (client connects to a Gateway elsewhere)?

The Control UI says “unauthorized” (or keeps reconnecting). What now?

I set gateway.bind: "tailnet" but it can't bind / nothing listens

Can I run multiple Gateways on the same host?

What does “invalid handshake” / code 1008 mean?

Logging and debugging

Where are logs?

How do I start/stop/restart the Gateway service?

I closed my terminal on Windows - how do I restart OpenClaw?

The Gateway is up but replies never arrive. What should I check?

“Disconnected from gateway: no reason” - what now?

Telegram setMyCommands fails with network errors. What should I check?

TUI shows no output. What should I check?

How do I completely stop then start the Gateway?

ELI5: openclaw gateway restart vs openclaw gateway

What's the fastest way to get more details when something fails?

Media and attachments

My skill generated an image/PDF, but nothing was sent

Security and access control

Is it safe to expose OpenClaw to inbound DMs?

Is prompt injection only a concern for public bots?

Should my bot have its own email GitHub account or phone number

Can I give it autonomy over my text messages and is that safe

Can I use cheaper models for personal assistant tasks?



I ran /start in Telegram but didn't get a pairing code

WhatsApp: will it message my contacts? How does pairing work?

Chat commands, aborting tasks, and "it won't stop"

How do I stop internal system messages from showing in chat

How do I stop/cancel a running task?

How do I send a Discord message from Telegram? ("Cross-context messaging denied")

Why does it feel like the bot "ignores" rapid-fire messages?

First 60 seconds if something's broken

1. Quick status (first check)

```
openclaw status
```

Fast local summary: OS + update, gateway/service reachability, agents/sessions, provider config + runtime issues (when gateway is reachable).

2. Pasteable report (safe to share)

```
openclaw status --all
```

Read-only diagnosis with log tail (tokens redacted).

3. Daemon + port state

```
openclaw gateway status
```



Shows supervisor runtime vs RPC reachability, the probe target URL, and which config the service likely used.

4. Deep probes

```
openclaw status --deep
```

Runs gateway health checks + provider probes (requires a reachable gateway). See .

5. Tail the latest log

```
openclaw logs --follow
```

If RPC is down, fall back to:

```
tail -f "$(ls -t /tmp/openclaw/openclaw-*.log | head -1)"
```

File logs are separate from service logs; see and .

6. Run the doctor (repairs)

```
openclaw doctor
```

Repairs/migrates config/state + runs health checks. See .

7. Gateway snapshot

```
openclaw health --json  
openclaw health --verbose # shows the target URL + config path on errors
```



Asks the running gateway for a full snapshot (WS-only). See [Health](#).

>

Quick start and first-run setup

I'm stuck what's the fastest way to get unstuck

Use a local AI agent that can **see your machine**. That is far more effective than asking in Discord, because most “I’m stuck” cases are **local config or environment issues** that remote helpers cannot inspect.

Claude Code: <https://www.anthropic.com/clause-code/>

OpenAI Codex: <https://openai.com/codex/>

These tools can read the repo, run commands, inspect logs, and help fix your machine-level setup (PATH, services, permissions, auth files). Give them the **full source checkout** via the hackable (git) install:

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

This installs OpenClaw **from a git checkout**, so the agent can read the code + docs and reason about the exact version you are running. You can always switch back to stable later by re-running the installer without `--install-method git`.

Tip: ask the agent to **plan and supervise** the fix (step-by-step), then execute only the necessary commands. That keeps changes small and easier to audit.

If you discover a real bug or fix, please file a GitHub issue or send a PR:

Start with these commands (share outputs when asking for help):

```
openclaw status
openclaw models status
openclaw doctor
>
```

What they do:

`openclaw status` : quick snapshot of gateway/agent health + basic config.

`openclaw models status` : checks provider auth + model availability.

`openclaw doctor` : validates and repairs common config/state issues.

Other useful CLI checks: `openclaw status --all` , `openclaw logs --follow` ,
`openclaw gateway status` , `openclaw health --verbose` .

Quick debug loop: . Install
docs: , , .

What's the recommended way to install and set up OpenClaw

The repo recommends running from source and using the onboarding wizard:

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

The wizard can also build UI assets automatically. After onboarding, you typically run the Gateway on port **18789**.

From source (contributors/dev):



```
git clone https://github.com/openclaw/openclaw.git
cd openclaw
pnpm install
pnpm build
pnpm ui:build # auto-installs UI deps on first run
openclaw onboard
```

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

How do I open the dashboard after onboarding

The wizard opens your browser with a clean (non-tokenized) dashboard URL right after onboarding and also prints the link in the summary. Keep that tab open; if it didn't launch, copy/paste the printed URL on the same machine.

How do I authenticate the dashboard token on localhost vs remote

localhost (same machine):

Open `http://127.0.0.1:18789/`.

If it asks for auth, paste the token from `gateway.auth.token` (or `OPENCLAW_GATEWAY_TOKEN`) into Control UI settings.

Retrieve it from the gateway host: `openclaw config get gateway.auth.token` (or generate one: `openclaw doctor --generate-gateway-token`).

Not on localhost:

Tailscale Serve (recommended): keep bind loopback, run `openclaw gateway --tailscale serve`, open `https://<magicdns>/`. If `gateway.auth.allowTailscale` is `true`, identity headers satisfy auth (no token).

 **Tailnet bind:** run `openclaw gateway --bind tailnet --token "<token>"`, open `http://<tailscale-ip>:18789/`, paste token in dashboard settings.

SSH tunnel: `ssh -N -L 18789:127.0.0.1:18789 user@host` then open `http://127.0.0.1:18789/` and paste the token in Control UI settings.

See [**Dashboard**](#) and [**Web surfaces**](#) for bind modes and auth details.

What runtime do I need

Node ≥ 22 is required. `pnpm` is recommended. `Bun` is **not recommended** for the Gateway.

Does it run on Raspberry Pi

Yes. The Gateway is lightweight - docs list **512MB-1GB RAM**, **1 core**, and about **500MB** disk as enough for personal use, and note that a **Raspberry Pi 4** can run it.

If you want extra headroom (logs, media, other services), **2GB is recommended**, but it's not a hard minimum.

Tip: a small Pi/VPS can host the Gateway, and you can pair **nodes** on your laptop/phone for local screen/camera/canvas or command execution.

See [**Nodes**](#).

Any tips for Raspberry Pi installs

Short version: it works, but expect rough edges.

Use a **64-bit** OS and keep Node ≥ 22 .

Prefer the **hackable (git) install** so you can see logs and update fast.

Start without channels/skills, then add them one by one.

 If you hit weird binary issues, it is usually an **ARM compatibility** problem.

Docs: [Linux](#), [Install](#).

It is stuck on wake up my friend onboarding will not hatch What now

That screen depends on the Gateway being reachable and authenticated. The TUI also sends “Wake up, my friend!” automatically on first hatch. If you see that line with **no reply** and tokens stay at 0, the agent never ran.

1. Restart the Gateway:

```
openclaw gateway restart
```

2. Check status + auth:

```
openclaw status  
openclaw models status  
openclaw logs --follow
```

3. If it still hangs, run:

```
openclaw doctor
```

If the Gateway is remote, ensure the tunnel/Tailscale connection is up and that the UI is pointed at the right Gateway. See [.](#)

Can I migrate my setup to a new machine Mac mini without redoing onboarding

Yes. Copy the **state directory** and **workspace**, then run Doctor once.

This keeps your bot “exactly the same” (memory, session history, auth, and channel state) as long as you copy **both** locations:

>

1. Install OpenClaw on the new machine.
2. Copy `$OPENCLAW_STATE_DIR` (default: `~/.openclaw`) from the old machine.
3. Copy your workspace (default: `~/.openclaw/workspace`).
4. Run `openclaw doctor` and restart the Gateway service.

That preserves config, auth profiles, WhatsApp creds, sessions, and memory. If you’re in remote mode, remember the gateway host owns the session store and workspace.

Important: if you only commit/push your workspace to GitHub, you’re backing up **memory + bootstrap files**, but **not** session history or auth. Those live under `~/.openclaw/` (for example `~/.openclaw/agents/<agentId>/sessions/`).

Related: [Migrating](#), [Where things live on disk](#), [Agent workspace](#), [Doctor](#), [Remote mode](#).

Where do I see what is new in the latest version

Check the GitHub changelog:

<https://github.com/openclaw/openclaw/blob/main/CHANGELOG.md>

Newest entries are at the top. If the top section is marked **Unreleased**, the next dated section is the latest shipped version. Entries are grouped by **Highlights**, **Changes**, and **Fixes** (plus docs/other sections when needed).

I cant access docs.openclaw.ai SSL error What now

Some Comcast/Xfinity connections incorrectly block `docs.openclaw.ai` via Xfinity Advanced Security. Disable it or allowlist `docs.openclaw.ai`, then retry. More detail: [Troubleshooting](#). Please help us unblock it by reporting here: https://spa.xfinity.com/check_url_status.

If you still can't reach the site, the docs are mirrored on GitHub:
<https://github.com/openclaw/openclaw/tree/main/docs>

What's the difference between stable and beta

Stable and **beta** are `npm dist-tags`, not separate code lines:

```
latest = stable  
beta = early build for testing
```

We ship builds to **beta**, test them, and once a build is solid we **promote that same version to latest**. That's why beta and stable can point at the **same version**.

See what changed:

<https://github.com/openclaw/openclaw/blob/main/CHANGELOG.md>

How do I install the beta version and whats the difference between beta and dev

Beta is the npm dist-tag `beta` (may match `latest`). **Dev** is the moving head of `main` (git); when published, it uses the npm dist-tag `dev`.

One-liners (macOS/Linux):

```
curl -fsSL --proto '=https' --tlsv1.2 https://openclaw.ai/install.sh |
```

```
curl -fsSL --proto '=https' --tlsv1.2 https://openclaw.ai/install.sh |
```

Windows installer (PowerShell): <https://openclaw.ai/install.ps1>



More detail: [Development channels](#) and [Installer flags](#).

>

How long does install and onboarding usually take

Rough guide:

Install: 2-5 minutes

Onboarding: 5-15 minutes depending on how many channels/models you configure

If it hangs, use [Installer stuck](#) and the fast debug loop in [I'm stuck](#).

How do I try the latest bits

Two options:

1. Dev channel (git checkout):

```
openclaw update --channel dev
```

This switches to the `main` branch and updates from source.

2. Hackable install (from the installer site):

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

That gives you a local repo you can edit, then update via git.

If you prefer a clean clone manually, use:

```
git clone https://github.com/openclaw/openclaw.git  
cd openclaw  
pnpm install  
pnpm build
```

Docs:

,

,

.

Installer stuck How do I get more feedback

Re-run the installer with **verbose output**:

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

Beta install with verbose:

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

For a hackable (git) install:

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

Windows (PowerShell) equivalent:

```
# install.ps1 has no dedicated -Verbose flag yet.  
Set-PSDebug -Trace 1  
& ([scriptblock]::Create((iwr -useb https://openclaw.ai/install.ps1))) -NoOnboard  
Set-PSDebug -Trace 0
```

More options:

.

Windows install says git not found or openclaw not recognized



Two common Windows issues:

1) npm error spawn git / git not found

Install **Git for Windows** and make sure `git` is on your PATH.

Close and reopen PowerShell, then re-run the installer.

2) openclaw is not recognized after install

Your npm global bin folder is not on PATH.

Check the path:

```
npm config get prefix
```

Ensure `<prefix>\bin` is on PATH (on most systems it is `%AppData%\npm`).

Close and reopen PowerShell after updating PATH.

If you want the smoothest Windows setup, use **WSL2** instead of native Windows. Docs: .

The docs didnt answer my question how do I get a better answer

Use the **hackable (git) install** so you have the full source and docs locally, then ask your bot (or Claude/Codex) *from that folder* so it can read the repo and answer precisely.

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

More detail: and .

How do I install OpenClaw on Linux



Short answer: follow the Linux guide, then run the onboarding wizard.

Linux quick path + service install: [Linux](#).

Full walkthrough: [Getting Started](#).

Installer + updates: [Install & updates](#).

How do I install OpenClaw on a VPS

Any Linux VPS works. Install on the server, then use SSH/Tailscale to reach the Gateway.

Guides: [exe.dev](#), [Hetzner](#), [Fly.io](#). Remote access: [Gateway remote](#).

Where are the cloudVPS install guides

We keep a **hosting hub** with the common providers. Pick one and follow the guide:

[VPS hosting](#) (all providers in one place)

[Fly.io](#)

[Hetzner](#)

[exe.dev](#)

How it works in the cloud: the **Gateway runs on the server**, and you access it from your laptop/phone via the Control UI (or Tailscale/SSH). Your state + workspace live on the server, so treat the host as the source of truth and back it up.

You can pair **nodes** (Mac/iOS/Android/headless) to that cloud Gateway to access local screen/camera/canvas or run commands on your laptop while keeping the Gateway in the cloud.

Hub: [Platforms](#). Remote access: [Gateway remote](#). Nodes: [Nodes](#), [Nodes GEI](#).

>

Can I ask OpenClaw to update itself

Short answer: **possible, not recommended**. The update flow can restart the Gateway (which drops the active session), may need a clean git checkout, and can prompt for confirmation. Safer: run updates from a shell as the operator.

Use the CLI:

```
openclaw update
openclaw update status
openclaw update --channel stable|beta|dev
openclaw update --tag <dist-tag|version>
openclaw update --no-restart
```

If you must automate from an agent:

```
openclaw update --yes --no-restart
openclaw gateway restart
```

Docs: [CLI](#), [Gateway API](#), [Gateway Configuration](#)

What does the onboarding wizard actually do

`openclaw onboard` is the recommended setup path. In **local mode** it walks you through:

Model/auth setup (Anthropic **setup-token** recommended for Claude subscriptions, OpenAI Codex OAuth supported, API keys optional, LM Studio local models supported)

Workspace location + bootstrap files



Gateway settings (bind/port/auth/tailscale)

Providers (WhatsApp, Telegram, Discord, Mattermost (plugin), Signal, iMessage)

Daemon install (LaunchAgent on macOS; systemd user unit on Linux/WSL2)

Health checks and **skills** selection

It also warns if your configured model is unknown or missing auth.

Do I need a Claude or OpenAI subscription to run this

No. You can run OpenClaw with **API keys** (Anthropic/OpenAI/others) or with **local-only models** so your data stays on your device.

Subscriptions (Claude Pro/Max or OpenAI Codex) are optional ways to authenticate those providers.

Docs: [Anthropic](#), [OpenAI](#), [Local models](#), [Models](#).

Can I use Claude Max subscription without an API key

Yes. You can authenticate with a **setup-token** instead of an API key. This is the subscription path.

Claude Pro/Max subscriptions **do not include an API key**, so this is the correct approach for subscription accounts. Important: you must verify with Anthropic that this usage is allowed under their subscription policy and terms. If you want the most explicit, supported path, use an Anthropic API key.

How does Anthropic setup token auth work

`claude setup-token` generates a **token string** via the Claude Code CLI (it is not available in the web console). You can run it on **any machine**. Choose **Anthropic token (paste setup-token)** in the wizard or paste it

with `openclaw models auth paste-token --provider anthropic`. The token is stored as an auth profile for the **anthropic** provider and used like an API key (no auto-refresh). More detail: [OAuth](#).

>

Where do I find an Anthropic setuptoken

It is **not** in the Anthropic Console. The setup-token is generated by the **Claude Code CLI** on **any machine**:

```
claude setup-token
```

Copy the token it prints, then choose **Anthropic token (paste setup-token)** in the wizard. If you want to run it on the gateway host, use `openclaw models auth setup-token --provider anthropic`. If you ran `claude setup-token` elsewhere, paste it on the gateway host with `openclaw models auth paste-token --provider anthropic`. See .

Do you support Claude subscription auth (Claude Pro or Max)

Yes - via **setup-token**. OpenClaw no longer reuses Claude Code CLI OAuth tokens; use a setup-token or an Anthropic API key. Generate the token anywhere and paste it on the gateway host. See and .

Note: Claude subscription access is governed by Anthropic's terms. For production or multi-user workloads, API keys are usually the safer choice.

Why am I seeing HTTP 429 ratelimiterror from Anthropic

That means your **Anthropic quota/rate limit** is exhausted for the current window. If you use a **Claude subscription** (setup-token or Claude Code OAuth), wait for the window to reset or upgrade your plan. If you use an **Anthropic API key**, check the Anthropic Console for usage/billing and raise limits as needed.

Tip: set a **fallback model** so OpenClaw can keep replying while a provider is rate-limited. See [Models](#) and [OAuth](#).

>

Is AWS Bedrock supported

Yes - via pi-ai's [Amazon Bedrock \(Converse\)](#) provider with [manual config](#). You must supply AWS credentials/region on the gateway host and add a Bedrock provider entry in your models config. See [Amazon Bedrock](#) and [Model providers](#). If you prefer a managed key flow, an OpenAI-compatible proxy in front of Bedrock is still a valid option.

How does Codex auth work

OpenClaw supports [OpenAI Code \(Codex\)](#) via OAuth (ChatGPT sign-in). The wizard can run the OAuth flow and will set the default model to `openai-codex/gpt-5.3-codex` when appropriate. See [Model providers](#) and [Wizard](#).

Do you support OpenAI subscription auth Codex OAuth

Yes. OpenClaw fully supports [OpenAI Code \(Codex\) subscription OAuth](#). The onboarding wizard can run the OAuth flow for you.

See [OAuth](#), [Model providers](#), and [Wizard](#).

How do I set up Gemini CLI OAuth

Gemini CLI uses a [plugin auth flow](#), not a client id or secret in `openclaw.json`.

Steps:

1. Enable the plugin: `openclaw plugins enable google-gemini-cli-auth`
2. Login: `openclaw models auth login --provider google-gemini-cli --set-default`

This stores OAuth tokens in auth profiles on the gateway host.

Details: [Model providers](#).

>

Is a local model OK for casual chats

Usually no. OpenClaw needs large context + strong safety; small cards truncate and leak. If you must, run the **largest** MiniMax M2.1 build you can locally (LM Studio) and see [/gateway/local-models](#).

Smaller/quantized models increase prompt-injection risk - see [Security](#).

How do I keep hosted model traffic in a specific region

Pick region-pinned endpoints. OpenRouter exposes US-hosted options for MiniMax, Kimi, and GLM; choose the US-hosted variant to keep data in-region. You can still list Anthropic/OpenAI alongside these by using `models.mode: "merge"` so fallbacks stay available while respecting the regioned provider you select.

Do I have to buy a Mac Mini to install this

No. OpenClaw runs on macOS or Linux (Windows via WSL2). A Mac mini is optional - some people buy one as an always-on host, but a small VPS, home server, or Raspberry Pi-class box works too.

You only need a Mac **for macOS-only tools**. For iMessage, use [BlueBubbles](#) (recommended) - the BlueBubbles server runs on any Mac, and the Gateway can run on Linux or elsewhere. If you want other macOS-only tools, run the Gateway on a Mac or pair a macOS node.

Docs: [BlueBubbles](#), [Nodes](#), [Mac remote mode](#).

Do I need a Mac mini for iMessage support

You need **some macOS device** signed into Messages. It does **not** have to be a Mac mini - any Mac works. Use **BlueBubbles** (recommended) for iMessage - the BlueBubbles server runs on macOS, while the Gateway can run on Linux or elsewhere.

Common setups:

Run the Gateway on Linux/VPS, and run the BlueBubbles server on any Mac signed into Messages.

Run everything on the Mac if you want the simplest single-machine setup.

Docs: [BlueBubbles](#), [Nodes](#), [Mac remote mode](#).

If I buy a Mac mini to run OpenClaw can I connect it to my MacBook Pro

Yes. The **Mac mini can run the Gateway**, and your MacBook Pro can connect as a **node** (companion device). Nodes don't run the Gateway - they provide extra capabilities like screen/camera/canvas and `system.run` on that device.

Common pattern:

Gateway on the Mac mini (always-on).

MacBook Pro runs the macOS app or a node host and pairs to the Gateway.

Use `openclaw nodes status` / `openclaw nodes list` to see it.

Docs: [Nodes](#), [Nodes CLI](#).

Can I use Bun

Bun is **not recommended**. We see runtime bugs, especially with WhatsApp and Telegram. Use **Node** for stable gateways.

If you still want to experiment with Bun, do it on a non-production gateway without WhatsApp/Telegram.

>

Telegram what goes in allowFrom

`channels.telegram.allowFrom` is the human sender's Telegram user ID (numeric). It is not the bot username.

The onboarding wizard accepts `@username` input and resolves it to a numeric ID, but OpenClaw authorization uses numeric IDs only.

Safer (no third-party bot):

DM your bot, then run `openclaw logs --follow` and read `from.id`.

Official Bot API:

DM your bot, then call `https://api.telegram.org/bot<bot_token>/getUpdates` and read `message.from.id`.

Third-party (less private):

DM `@userinfobot` or `@getidsbot`.

See [/channels/telegram](#).

Can multiple people use one WhatsApp number with different OpenClaw instances

Yes, via **multi-agent routing**. Bind each sender's WhatsApp DM (peer `kind: "direct"`, sender E.164 like `+15551234567`) to a different `agentId`, so each person gets their own workspace and session store. Replies still come from the **same WhatsApp account**, and DM access control (`channels.whatsapp.dmPolicy` / `channels.whatsapp.allowFrom`) is global per WhatsApp account. See [Multi-Agent Routing](#) and [WhatsApp](#).

Can I run a fast chat agent and an Opus for coding agent



Yes. Use multi-agent routing: give each agent its own default model, then bind inbound routes (provider account or specific peers) to each agent. Example config lives in [Multi-Agent Routing](#). See also [Models](#) and [Configuration](#).

Does Homebrew work on Linux

Yes. Homebrew supports Linux (Linuxbrew). Quick setup:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

If you run OpenClaw via systemd, ensure the service PATH includes `/home/linuxbrew/.linuxbrew/bin` (or your brew prefix) so `brew`-installed tools resolve in non-login shells. Recent builds also prepend common user bin dirs on Linux systemd services (for example `~/.local/bin`, `~/.npm-global/bin`, `~/.local/share/pnpm`, `~/.bun/bin`) and honor `PNPM_HOME`, `NPM_CONFIG_PREFIX`, `BUN_INSTALL`, `VOLTA_HOME`, `ASDF_DATA_DIR`, `NVM_DIR`, and `FNM_DIR` when set.

What's the difference between the hackable git install and npm install

Hackable (git) install: full source checkout, editable, best for contributors. You run builds locally and can patch code/docs.

npm install: global CLI install, no repo, best for “just run it.” Updates come from npm dist-tags.

Docs:

[Hackable git install](#)

Can I switch between npm and git installs later



Yes. Install the other flavor, then run Doctor so the gateway service points at the new entrypoint. This **does not delete your data** - it only changes the OpenClaw code install. Your state (`~/.openclaw`) and workspace (`~/.openclaw/workspace`) stay untouched.

From npm → git:

```
git clone https://github.com/openclaw/openclaw.git
cd openclaw
pnpm install
pnpm build
openclaw doctor
openclaw gateway restart
```

From git → npm:

```
npm install -g openclaw@latest
openclaw doctor
openclaw gateway restart
```

Doctor detects a gateway service entrypoint mismatch and offers to rewrite the service config to match the current install (use `--repair` in automation).

Backup tips: see .

Should I run the Gateway on my laptop or a VPS

Short answer: **if you want 24/7 reliability, use a VPS.** If you want the lowest friction and you're okay with sleep/restarts, run it locally.

Laptop (local Gateway)



Pros: no server cost, direct access to local files, live browser window.

Cons: sleep/network drops = disconnects, OS updates/reboots
> interrupt, must stay awake.

VPS / cloud

Pros: always-on, stable network, no laptop sleep issues, easier to keep running.

Cons: often run headless (use screenshots), remote file access only, you must SSH for updates.

OpenClaw-specific note: WhatsApp/Telegram/Slack/Mattermost (plugin)/Discord all work fine from a VPS. The only real trade-off is **headless browser** vs a visible window. See [Browser](#).

Recommended default: VPS if you had gateway disconnects before. Local is great when you're actively using the Mac and want local file access or UI automation with a visible browser.

How important is it to run OpenClaw on a dedicated machine

Not required, but **recommended for reliability and isolation**.

Dedicated host (VPS/Mac mini/Pi): always-on, fewer sleep/reboot interruptions, cleaner permissions, easier to keep running.

Shared laptop/desktop: totally fine for testing and active use, but expect pauses when the machine sleeps or updates.

If you want the best of both worlds, keep the Gateway on a dedicated host and pair your laptop as a **node** for local screen/camera/exec tools. See [Nodes](#). For security guidance, read [Security](#).

What are the minimum VPS requirements and recommended OS

OpenClaw is lightweight. For a basic Gateway + one chat channel:



Absolute minimum: 1 vCPU, 1GB RAM, ~500MB disk.

Recommended: 1-2 vCPU, 2GB RAM or more for headroom (logs, media, multiple channels). Node tools and browser automation can be resource hungry.

OS: use **Ubuntu LTS** (or any modern Debian/Ubuntu). The Linux install path is best tested there.

Docs: [Linux](#), [VPS hosting](#).

Can I run OpenClaw in a VM and what are the requirements

Yes. Treat a VM the same as a VPS: it needs to be always on, reachable, and have enough RAM for the Gateway and any channels you enable.

Baseline guidance:

Absolute minimum: 1 vCPU, 1GB RAM.

Recommended: 2GB RAM or more if you run multiple channels, browser automation, or media tools.

OS: Ubuntu LTS or another modern Debian/Ubuntu.

If you are on Windows, **WSL2 is the easiest VM style setup** and has the best tooling compatibility. See [Windows](#), [VPS hosting](#). If you are running macOS in a VM, see [macOS VM](#).

What is OpenClaw?

What is OpenClaw in one paragraph

OpenClaw is a personal AI assistant you run on your own devices. It replies on the messaging surfaces you already use (WhatsApp, Telegram,

Slack, Mattermost (plugin), Discord, Google Chat, Signal, iMessage, WebChat) and can also do voice + a live Canvas on supported platforms. The **Gateway** is the always-on control plane; the assistant is the product.'

What's the value proposition

OpenClaw is not "just a Claude wrapper." It's a **local-first control plane** that lets you run a capable assistant on **your own hardware**, reachable from the chat apps you already use, with stateful sessions, memory, and tools - without handing control of your workflows to a hosted SaaS.

Highlights:

Your devices, your data: run the Gateway wherever you want (Mac, Linux, VPS) and keep the workspace + session history local.

Real channels, not a web sandbox:

WhatsApp/Telegram/Slack/Discord/Signal/iMessage/etc, plus mobile voice and Canvas on supported platforms.

Model-agnostic: use Anthropic, OpenAI, MiniMax, OpenRouter, etc., with per-agent routing and failover.

Local-only option: run local models so **all data can stay on your device** if you want.

Multi-agent routing: separate agents per channel, account, or task, each with its own workspace and defaults.

Open source and hackable: inspect, extend, and self-host without vendor lock-in.

Docs: [Gateway](#), [Channels](#), [Multi-agent](#), [Memory](#).

I just set it up what should I do first

Good first projects:



Build a website (WordPress, Shopify, or a simple static site).

Prototype a mobile app (outline, screens, API plan).

Organize files and folders (cleanup, naming, tagging).

Connect Gmail and automate summaries or follow ups.

It can handle large tasks, but it works best when you split them into phases and use sub agents for parallel work.

What are the top five everyday use cases for OpenClaw

Everyday wins usually look like:

Personal briefings: summaries of inbox, calendar, and news you care about.

Research and drafting: quick research, summaries, and first drafts for emails or docs.

Reminders and follow ups: cron or heartbeat driven nudges and checklists.

Browser automation: filling forms, collecting data, and repeating web tasks.

Cross device coordination: send a task from your phone, let the Gateway run it on a server, and get the result back in chat.

Can OpenClaw help with lead gen outreach ads and blogs for a SaaS

Yes for **research, qualification, and drafting**. It can scan sites, build shortlists, summarize prospects, and write outreach or ad copy drafts.

For **outreach or ad runs**, keep a human in the loop. Avoid spam, follow local laws and platform policies, and review anything before it is sent. The safest pattern is to let OpenClaw draft and you approve.

Docs: [Security.](#)



What are the advantages vs Claude Code for web development

OpenClaw is a **personal assistant** and coordination layer, not an IDE replacement. Use Claude Code or Codex for the fastest direct coding loop inside a repo. Use OpenClaw when you want durable memory, cross-device access, and tool orchestration.

Advantages:

Persistent memory + workspace across sessions

Multi-platform access (WhatsApp, Telegram, TUI, WebChat)

Tool orchestration (browser, files, scheduling, hooks)

Always-on Gateway (run on a VPS, interact from anywhere)

Nodes for local browser/screen/camera/exec

Showcase: <https://openclaw.ai/showcase>

Skills and automation

How do I customize skills without keeping the repo dirty

Use managed overrides instead of editing the repo copy. Put your changes in `~/.openclaw/skills/<name>/SKILL.md` (or add a folder via `skills.load.extraDirs` in `~/.openclaw/openclaw.json`). Precedence is `<workspace>/skills > ~/.openclaw/skills > bundled`, so managed overrides win without touching git. Only upstream-worthy edits should live in the repo and go out as PRs.

Can I load skills from a custom folder

Yes. Add extra directories via `skills.load.extraDirs` in `~/.openclaw/openclaw.json` (lowest precedence). Default precedence

remains: <workspace>/skills → ~/.openclaw/skills → bundled → skills.load.extraDirs . clawhub installs into ./skills by default, which OpenClaw treats as <workspace>/skills .
→

How can I use different models for different tasks

Today the supported patterns are:

Cron jobs: isolated jobs can set a model override per job.

Sub-agents: route tasks to separate agents with different default models.

On-demand switch: use /model to switch the current session model at any time.

See [Cron jobs](#), [Multi-Agent Routing](#), and [Slash commands](#).

The bot freezes while doing heavy work How do I offload that

Use **sub-agents** for long or parallel tasks. Sub-agents run in their own session, return a summary, and keep your main chat responsive.

Ask your bot to “spawn a sub-agent for this task” or use /subagents . Use /status in chat to see what the Gateway is doing right now (and whether it is busy) .

Token tip: long tasks and sub-agents both consume tokens. If cost is a concern, set a cheaper model for sub-agents via agents.defaults.subagents.model .

Docs: [Sub-agents](#).

Cron or reminders do not fire What should I check

Cron runs inside the Gateway process. If the Gateway is not running continuously, scheduled jobs will not run.

Checklist:



Confirm cron is enabled (`cron.enabled`) and `OPENCLAW_SKIP_CRON` is not set. >

Check the Gateway is running 24/7 (no sleep/restarts).

Verify timezone settings for the job (`--tz` vs host timezone).

Debug:

```
openclaw cron run <jobId> --force  
openclaw cron runs --id <jobId> --limit 50
```

Docs:

,

.

How do I install skills on Linux

Use **ClawHub** (CLI) or drop skills into your workspace. The macOS Skills UI isn't available on Linux. Browse skills at .

Install the ClawHub CLI (pick one package manager):

```
npm i -g clawhub
```

```
pnpm add -g clawhub
```

Can OpenClaw run tasks on a schedule or continuously in the background

Yes. Use the Gateway scheduler:

Cron jobs for scheduled or recurring tasks (persist across restarts).

Heartbeat for “main session” periodic checks.

 **Isolated jobs** for autonomous agents that post summaries or deliver to chats.

Docs: [Cron jobs](#), [Cron vs Heartbeat](#), [Heartbeat](#).

Can I run Apple macOS-only skills from Linux?

Not directly. macOS skills are gated by `metadata.openclaw.os` plus required binaries, and skills only appear in the system prompt when they are eligible on the **Gateway host**. On Linux, `darwin`-only skills (like `apple-notes`, `apple-reminders`, `things-mac`) will not load unless you override the gating.

You have three supported patterns:

Option A - run the Gateway on a Mac (simplest). Run the Gateway where the macOS binaries exist, then connect from Linux in [remote mode](#) or over Tailscale. The skills load normally because the Gateway host is macOS.

Option B - use a macOS node (no SSH). Run the Gateway on Linux, pair a macOS node (menubar app), and set **Node Run Commands** to “Always Ask” or “Always Allow” on the Mac. OpenClaw can treat macOS-only skills as eligible when the required binaries exist on the node. The agent runs those skills via the `nodes` tool. If you choose “Always Ask”, approving “Always Allow” in the prompt adds that command to the allowlist.

Option C - proxy macOS binaries over SSH (advanced). Keep the Gateway on Linux, but make the required CLI binaries resolve to SSH wrappers that run on a Mac. Then override the skill to allow Linux so it stays eligible.

1. Create an SSH wrapper for the binary (example: `memo` for Apple Notes):



```
#!/usr/bin/env bash
set -euo pipefail
exec ssh -T user@mac-host /opt/homebrew/bin/memo "$@"
```

2. Put the wrapper on PATH on the Linux host (for example `~/bin/memo`).
3. Override the skill metadata (workspace or `~/.openclaw/skills`) to allow Linux:

```
---
name: apple-notes
description: Manage Apple Notes via the memo CLI on macOS.
metadata: { "openclaw": { "os": ["darwin", "linux"], "requires": { "bins": [
---
```

4. Start a new session so the skills snapshot refreshes.

Do you have a Notion or HeyGen integration

Not built-in today.

Options:

Custom skill / plugin: best for reliable API access (Notion/HeyGen both have APIs).

Browser automation: works without code but is slower and more fragile.

If you want to keep context per client (agency workflows), a simple pattern is:

One Notion page per client (context + preferences + active work).

Ask the agent to fetch that page at the start of a session.

If you want a native integration, open a feature request or build a skill targeting those APIs.

Install skills:

```
clawhub install <skill-slug>
clawhub update --all
```

ClawHub installs into `./skills` under your current directory (or falls back to your configured OpenClaw workspace); OpenClaw treats that as `<workspace>/skills` on the next session. For shared skills across agents, place them in `~/.openclaw/skills/<name>/SKILL.md`. Some skills expect binaries installed via Homebrew; on Linux that means Linuxbrew (see the Homebrew Linux FAQ entry above). See [... and ...](#).

How do I install the Chrome extension for browser takeover

Use the built-in installer, then load the unpacked extension in Chrome:

```
openclaw browser extension install
openclaw browser extension path
```

Then Chrome → `chrome://extensions` → enable “Developer mode” → “Load unpacked” → pick that folder.

Full guide (including remote Gateway + security notes):

If the Gateway runs on the same machine as Chrome (default setup), you usually **do not** need anything extra. If the Gateway runs elsewhere, run a node host on the browser machine so the Gateway can proxy browser actions. You still need to click the extension button on the tab you want to control (it doesn't auto-attach).

Sandboxing and memory



Is there a dedicated sandboxing doc

Yes. See [Sandboxing](#). For Docker-specific setup (full gateway in Docker or sandbox images), see [Docker](#).

Docker feels limited How do I enable full features

The default image is security-first and runs as the `node` user, so it does not include system packages, Homebrew, or bundled browsers. For a fuller setup:

Persist `/home/node` with `OPENCLAW_HOME_VOLUME` so caches survive.

Bake system deps into the image with `OPENCLAW_DOCKER_APT_PACKAGES`.

Install Playwright browsers via the bundled CLI: `node /app/node_modules/playwright-core/cli.js install chromium`

Set `PLAYWRIGHT_BROWSERS_PATH` and ensure the path is persisted.

Docs: [Docker](#), [Browser](#).

Can I keep DMs personal but make groups public sandboxed with one agent

Yes - if your private traffic is **DMs** and your public traffic is **groups**.

Use `agents.defaults.sandbox.mode: "non-main"` so group/channel sessions (non-main keys) run in Docker, while the main DM session stays on-host. Then restrict what tools are available in sandboxed sessions via `tools.sandbox.tools`.

Setup walkthrough + example config: [Groups: personal DMs + public groups](#)

Key config reference: [Gateway configuration](#)

How do I bind a host folder into the sandbox



Set `agents.defaults.sandbox.docker.binds` to `["host:path:mode"]` (e.g., `"/home/user/src:/src:ro"`). Global + per-agent binds merge; per-agent binds are ignored when `scope: "shared"`. Use `:ro` for anything sensitive and remember binds bypass the sandbox filesystem walls. See [Sandboxing](#) and [Sandbox vs Tool Policy vs Elevated](#) for examples and safety notes.

How does memory work

OpenClaw memory is just Markdown files in the agent workspace:

Daily notes in `memory/YYYY-MM-DD.md`

Curated long-term notes in `MEMORY.md` (main/private sessions only)

OpenClaw also runs a **silent pre-compaction memory flush** to remind the model to write durable notes before auto-compaction. This only runs when the workspace is writable (read-only sandboxes skip it). See [Memory](#).

Memory keeps forgetting things How do I make it stick

Ask the bot to `write the fact to memory`. Long-term notes belong in `MEMORY.md`, short-term context goes into `memory/YYYY-MM-DD.md`.

This is still an area we are improving. It helps to remind the model to store memories; it will know what to do. If it keeps forgetting, verify the Gateway is using the same workspace on every run.

Docs: [Memory](#), [Agent workspace](#).

Does semantic memory search require an OpenAI API key

Only if you use **OpenAI embeddings**. Codex OAuth covers chat/completions and does **not** grant embeddings access, so **signing in with Codex (OAuth or the Codex CLI login)** does not help for semantic memory search.

OpenAI embeddings still need a real API key (`OPENAI_API_KEY` or `models.providers.openai.apiKey`).

If you don't set a provider explicitly, OpenClaw auto-selects a provider when it can resolve an API key (auth profiles, `models.providers.*.apiKey`, or env vars). It prefers OpenAI if an OpenAI key resolves, otherwise Gemini if a Gemini key resolves. If neither key is available, memory search stays disabled until you configure it. If you have a local model path configured and present, OpenClaw prefers `local`.

If you'd rather stay local, set `memorySearch.provider = "local"` (and optionally `memorySearch.fallback = "none"`). If you want Gemini embeddings, set `memorySearch.provider = "gemini"` and provide `GEMINI_API_KEY` (or `memorySearch.remote.apiKey`). We support **OpenAI**, **Gemini**, or **local** embedding models - see [**Memory**](#) for the setup details.

Does memory persist forever What are the limits

Memory files live on disk and persist until you delete them. The limit is your storage, not the model. The **session context** is still limited by the model context window, so long conversations can compact or truncate. That is why memory search exists - it pulls only the relevant parts back into context.

Docs: [**Memory**](#), [**Context**](#).

Where things live on disk

Is all data used with OpenClaw saved locally

No - OpenClaw's state is local, but external services still see what you send them.

Local by default: sessions, memory files, config, and workspace live on the Gateway host (`~/.openclaw + your workspace directory`).



Remote by necessity: messages you send to model providers (Anthropic/OpenAI/etc.) go to their APIs, and chat platforms (WhatsApp/Telegram/Slack/etc.) store message data on their servers.

➤ **You control the footprint:** using local models keeps prompts on your machine, but channel traffic still goes through the channel's servers.

Related: [Agent workspace](#), [Memory](#).

Where does OpenClaw store its data

Everything lives under `$OPENCLAW_STATE_DIR` (default: `~/.openclaw`):

| Path | Purpose |
|---|---|
| <code>\$OPENCLAW_STATE_DIR/openclaw.json</code> | Main config (JSON5) |
| <code>\$OPENCLAW_STATE_DIR/credentials/oauth.json</code> | Legacy OAuth import (copied into auth profiles on first use) |
| <code>\$OPENCLAW_STATE_DIR/agents/<agentId>/agent/auth-profiles.json</code> | Auth profiles (OAuth + API keys) |
| <code>\$OPENCLAW_STATE_DIR/agents/<agentId>/agent/auth.json</code> | Runtime auth cache (managed automatically) |
| <code>\$OPENCLAW_STATE_DIR/credentials/</code> | Provider state (e.g. <code>whatsapp/<accountId>/creds.json</code>) |
| <code>\$OPENCLAW_STATE_DIR/agents/</code> | Per-agent state (agentDir + sessions) |
| <code>\$OPENCLAW_STATE_DIR/agents/<agentId>/sessions/</code> | Conversation history & state (per agent) |
| <code>\$OPENCLAW_STATE_DIR/agents/<agentId>/sessions/sessions.json</code> | Session metadata (per agent) |

Legacy single-agent path: `~/.openclaw/agent/*` (migrated by `openclaw doctor`).

Your **workspace** (AGENTS.md, memory files, skills, etc.) is separate and configured via `agents.defaults.workspace` (default: `~/.openclaw/workspace`).

>

Where should AGENTSmd SOULmd USERmd MEMORYmd live

These files live in the **agent workspace**, not `~/.openclaw`.

Workspace (per agent): AGENTS.md , SOUL.md , IDENTITY.md , USER.md , MEMORY.md (or memory.md) , memory/YYYY-MM-DD.md , optional HEARTBEAT.md .

State dir (`~/.openclaw`): config, credentials, auth profiles, sessions, logs, and shared skills (`~/.openclaw/skills`).

Default workspace is `~/.openclaw/workspace` , configurable via:

```
{  
  agents: { defaults: { workspace: "~/.openclaw/workspace" } },  
}
```

If the bot “forgets” after a restart, confirm the Gateway is using the same workspace on every launch (and remember: remote mode uses the **gateway host's** workspace, not your local laptop).

Tip: if you want a durable behavior or preference, ask the bot to **write it into AGENTS.md or MEMORY.md** rather than relying on chat history.

See [AGENTS.md](#) and [MEMORY.md](#).

What's the recommended backup strategy

Put your **agent workspace** in a **private** git repo and back it up somewhere private (for example GitHub private). This captures memory + AGENTS/SOUL/USER files, and lets you restore the assistant's “mind” later.

Do **not** commit anything under `~/.openclaw` (credentials, sessions, tokens). If you need a full restore, back up both the workspace and the state directory separately (see the migration question above).

>

Docs: [Agent workspace](#).

How do I completely uninstall OpenClaw

See the dedicated guide: [Uninstall](#).

Can agents work outside the workspace

Yes. The workspace is the `default cwd` and memory anchor, not a hard sandbox. Relative paths resolve inside the workspace, but absolute paths can access other host locations unless sandboxing is enabled. If you need isolation, use [agents.defaults.sandbox](#) or per-agent sandbox settings. If you want a repo to be the default working directory, point that agent's `workspace` to the repo root. The OpenClaw repo is just source code; keep the workspace separate unless you intentionally want the agent to work inside it.

Example (repo as default cwd):

```
{  
  agents: {  
    defaults: {  
      workspace: "~/Projects/my-repo",  
    },  
  },  
}
```

I'm in remote mode where is the session store

Session state is owned by the `gateway host`. If you're in remote mode, the session store you care about is on the remote machine, not your

local laptop. See [Session management](#).

Config basics

What format is the config Where is it

OpenClaw reads an optional **JSON5** config from `$OPENCLAW_CONFIG_PATH` (default: `~/.openclaw/openclaw.json`):

```
$OPENCLAW_CONFIG_PATH
```

If the file is missing, it uses safe-ish defaults (including a default workspace of `~/.openclaw/workspace`).

I set gateway.bind lan or tailnet and now nothing listens the UI says unauthorized

Non-loopback binds **require auth**. Configure `gateway.auth.mode` + `gateway.auth.token` (or use `OPENCLAW_GATEWAY_TOKEN`).

```
{
  gateway: {
    bind: "lan",
    auth: {
      mode: "token",
      token: "replace-me",
    },
  },
}
```

Notes:

`gateway.remote.token` is for **remote CLI calls** only; it does not enable local gateway auth.

The Control UI authenticates via `connect.params.auth.token` (stored in app/UI settings). Avoid putting tokens in URLs.

>

Why do I need a token on localhost now

The wizard generates a gateway token by default (even on loopback) so **local WS clients must authenticate**. This blocks other local processes from calling the Gateway. Paste the token into the Control UI settings (or your client config) to connect.

If you **really** want open loopback, remove `gateway.auth` from your config. Doctor can generate a token for you any time: `openclaw doctor --generate-gateway-token`.

Do I have to restart after changing config

The Gateway watches the config and supports hot-reload:

```
gateway.reload.mode: "hybrid" (default): hot-apply safe changes,  
restart for critical ones  
  
hot , restart , off are also supported
```

How do I enable web search and web fetch

`web_fetch` works without an API key. `web_search` requires a Brave Search API key. **Recommended:** run `openclaw configure --section web` to store it in `tools.web.search.apiKey`. Environment alternative: set `BRAVE_API_KEY` for the Gateway process.



```
tools: {
  web: {
    search: {
      enabled: true,
      apiKey: "BRAVE_API_KEY_HERE",
      maxResults: 5,
    },
    fetch: {
      enabled: true,
    },
  },
},
```

Notes:

If you use allowlists, add `web_search` / `web_fetch` or `group:web`.
`web_fetch` is enabled by default (unless explicitly disabled).
Daemons read env vars from `~/.openclaw/.env` (or the service environment).

Docs:

How do I run a central Gateway with specialized workers across devices

The common pattern is **one Gateway** (e.g. Raspberry Pi) plus **nodes** and **agents**:

Gateway (central): owns channels (Signal/WhatsApp), routing, and sessions.

Nodes (devices): Macs/iOS/Android connect as peripherals and expose local tools (`system.run`, `canvas`, `camera`).

Agents (workers): separate brains/workspaces for special roles (e.g. “Hetzner ops”, “Personal data”).

 **Sub-agents:** spawn background work from a main agent when you want parallelism.

TUI: connect to the Gateway and switch agents/sessions.

>

Docs: [Nodes](#), [Remote access](#), [Multi-Agent Routing](#), [Sub-agents](#), [TUI](#).

Can the OpenClaw browser run headless

Yes. It's a config option:

```
{  
  browser: { headless: true },  
  agents: {  
    defaults: {  
      sandbox: { browser: { headless: true } },  
    },  
  },  
}
```

Default is `false` (headful). Headless is more likely to trigger anti-bot checks on some sites. See [here](#).

Headless uses the **same Chromium engine** and works for most automation (forms, clicks, scraping, logins). The main differences:

No visible browser window (use screenshots if you need visuals).

Some sites are stricter about automation in headless mode (CAPTCHAs, anti-bot). For example, X/Twitter often blocks headless sessions.

How do I use Brave for browser control

Set `browser.executablePath` to your Brave binary (or any Chromium-based browser) and restart the Gateway. See the full config examples in

Remote gateways and nodes



How do commands propagate between Telegram the gateway and nodes

Telegram messages are handled by the **gateway**. The gateway runs the agent and only then calls nodes over the **Gateway WebSocket** when a node tool is needed:

Telegram → Gateway → Agent → `node.*` → Node → Gateway → Telegram

Nodes don't see inbound provider traffic; they only receive node RPC calls.

How can my agent access my computer if the Gateway is hosted remotely

Short answer: **pair your computer as a node**. The Gateway runs elsewhere, but it can call `node.*` tools (screen, camera, system) on your local machine over the Gateway WebSocket.

Typical setup:

1. Run the Gateway on the always-on host (VPS/home server).
2. Put the Gateway host + your computer on the same tailnet.
3. Ensure the Gateway WS is reachable (tailnet bind or SSH tunnel).
4. Open the macOS app locally and connect in **Remote over SSH** mode (or direct tailnet) so it can register as a node.
5. Approve the node on the Gateway:

```
openclaw nodes pending  
openclaw nodes approve <requestId>
```

No separate TCP bridge is required; nodes connect over the Gateway WebSocket.

Security reminder: pairing a macOS node allows `system.run` on that machine. Only pair devices you trust, and review [Security](#).

Docs: [Nodes](#), [Gateway protocol](#), [macOS remote mode](#), [Security](#).

Tailscale is connected but I get no replies What now

Check the basics:

Gateway is running: `openclaw gateway status`

Gateway health: `openclaw status`

Channel health: `openclaw channels status`

Then verify auth and routing:

If you use Tailscale Serve, make sure `gateway.auth.allowTailscale` is set correctly.

If you connect via SSH tunnel, confirm the local tunnel is up and points at the right port.

Confirm your allowlists (DM or group) include your account.

Docs: [Tailscale](#), [Remote access](#), [Channels](#).

Can two OpenClaw instances talk to each other local VPS

Yes. There is no built-in “bot-to-bot” bridge, but you can wire it up in a few reliable ways:

Simplest: use a normal chat channel both bots can access (Telegram/Slack/WhatsApp). Have Bot A send a message to Bot B, then let Bot B reply as usual.

CLI bridge (generic): run a script that calls the other Gateway with `openclaw agent --message ... --deliver`, targeting a chat where the other

bot listens. If one bot is on a remote VPS, point your CLI at that Remote Gateway via SSH/Tailscale (see [Remote access](#)).

Example pattern (run from a machine that can reach the target Gateway):

```
openclaw agent --message "Hello from local bot" --deliver --channel te
```

Tip: add a guardrail so the two bots do not loop endlessly (mention-only, channel allowlists, or a “do not reply to bot messages” rule).

Docs:

[CLI](#), [Gateway](#), [Nodes](#), [Webhooks](#)

Do I need separate VPSes for multiple agents

No. One Gateway can host multiple agents, each with its own workspace, model defaults, and routing. That is the normal setup and it is much cheaper and simpler than running one VPS per agent.

Use separate VPSes only when you need hard isolation (security boundaries) or very different configs that you do not want to share. Otherwise, keep one Gateway and use multiple agents or sub-agents.

Is there a benefit to using a node on my personal laptop instead of SSH from a VPS

Yes - nodes are the first-class way to reach your laptop from a remote Gateway, and they unlock more than shell access. The Gateway runs on macOS/Linux (Windows via WSL2) and is lightweight (a small VPS or Raspberry Pi-class box is fine; 4 GB RAM is plenty), so a common setup is an always-on host plus your laptop as a node.

No inbound SSH required. Nodes connect out to the Gateway WebSocket and use device pairing.

 **Safer execution controls.** `system.run` is gated by node allowlists/approvals on that laptop.

More device tools. Nodes expose `canvas`, `camera`, and `screen` in addition to `system.run`.

Local browser automation. Keep the Gateway on a VPS, but run Chrome locally and relay control with the Chrome extension + a node host on the laptop.

SSH is fine for ad-hoc shell access, but nodes are simpler for ongoing agent workflows and device automation.

Docs: [Nodes](#), [Nodes CLI](#), [Chrome extension](#).

Should I install on a second laptop or just add a node

If you only need **local tools** (`screen/camera/exec`) on the second laptop, add it as a **node**. That keeps a single Gateway and avoids duplicated config. Local node tools are currently macOS-only, but we plan to extend them to other OSes.

Install a second Gateway only when you need **hard isolation** or two fully separate bots.

Docs: [Nodes](#), [Nodes CLI](#), [Multiple gateways](#).

Do nodes run a gateway service

No. Only **one gateway** should run per host unless you intentionally run isolated profiles (see [Multiple gateways](#)). Nodes are peripherals that connect to the gateway (iOS/Android nodes, or macOS “node mode” in the menubar app). For headless node hosts and CLI control, see [Node host CLI](#).

A full restart is required for `gateway`, `discovery`, and `canvasHost` changes.

Is there an API RPC way to apply config



Yes. `config.apply` validates + writes the full config and restarts the Gateway as part of the operation.

configapply wiped my config How do I recover and avoid this

`config.apply` replaces the **entire config**. If you send a partial object, everything else is removed.

Recover:

Restore from backup (git or a copied `~/.openclaw/openclaw.json`).

If you have no backup, re-run `openclaw doctor` and reconfigure channels/models.

If this was unexpected, file a bug and include your last known config or any backup.

A local coding agent can often reconstruct a working config from logs or history.

Avoid it:

Use `openclaw config set` for small changes.

Use `openclaw configure` for interactive edits.

Docs: [Config](#), [Configure](#), [Doctor](#).

What's a minimal sane config for a first install

```
{  
  agents: { defaults: { workspace: "~/.openclaw/workspace" } },  
  channels: { whatsapp: { allowFrom: ["+15555550123"] } },  
}
```

This sets your workspace and restricts who can trigger the bot.



How do I set up Tailscale on a VPS and connect from my Mac

Minimal steps:

1. Install + login on the VPS

```
curl -fsSL https://tailscale.com/install.sh | sh  
sudo tailscale up
```

2. Install + login on your Mac

Use the Tailscale app and sign in to the same tailnet.

3. Enable MagicDNS (recommended)

In the Tailscale admin console, enable MagicDNS so the VPS has a stable name.

4. Use the tailnet hostname

SSH: ssh user@your-vps.tailnet-xxxx.ts.net

Gateway WS: ws://your-vps.tailnet-xxxx.ts.net:18789

If you want the Control UI without SSH, use Tailscale Serve on the VPS:

```
openclaw gateway --tailscale serve
```

This keeps the gateway bound to loopback and exposes HTTPS via Tailscale. See [.](#)

How do I connect a Mac node to a remote Gateway Tailscale Serve

Serve exposes the **Gateway Control UI + WS**. Nodes connect over the same Gateway WS endpoint.

Recommended setup:

1. Make sure the VPS + Mac are on the same tailnet.
2. Use the macOS app in Remote mode (SSH target can be the tailnet hostname). The app will tunnel the Gateway port and connect as a node.
3. Approve the node on the gateway:

```
openclaw nodes pending  
openclaw nodes approve <requestId>
```

Docs: , , .

Env vars and .env loading

How does OpenClaw load environment variables

OpenClaw reads env vars from the parent process (shell, launchd/systemd, CI, etc.) and additionally loads:

```
.env from the current working directory  
a global fallback .env from ~/.openclaw/.env (aka  
$OPENCLAW_STATE_DIR/.env )
```

Neither .env file overrides existing env vars.

You can also define inline env vars in config (applied only if missing from the process env):



```
env: {
  OPENROUTER_API_KEY: "sk-or-....",
  vars: { GROQ_API_KEY: "gsk-...." },
},
}
```

See [full precedence and sources](#).

I started the Gateway via the service and my env vars disappeared What now

Two common fixes:

1. Put the missing keys in `~/.openclaw/.env` so they're picked up even when the service doesn't inherit your shell env.
2. Enable shell import (opt-in convenience):

```
{
  env: {
    shellEnv: {
      enabled: true,
      timeoutMs: 15000,
    },
  },
}
```

This runs your login shell and imports only missing expected keys (never overrides). Env var equivalents: `OPENCLAW_LOAD_SHELL_ENV=1` , `OPENCLAW_SHELL_ENV_TIMEOUT_MS=15000` .

I set COPILOTGITHUBTOKEN but models status shows Shell env off Why

`openclaw models status` reports whether **shell env import** is enabled.

Shell env: off" does **not** mean your env vars are missing - it just means OpenClaw won't load your login shell automatically.

>

If the Gateway runs as a service (launchd/systemd), it won't inherit your shell environment. Fix by doing one of these:

1. Put the token in `~/.openclaw/.env` :

```
COPilot_GITHUB_TOKEN=...
```

2. Or enable shell import (`env.shellEnv.enabled: true`) .

3. Or add it to your config `env` block (applies only if missing) .

Then restart the gateway and recheck:

```
openclaw models status
```

Copilot tokens are read from `COPilot_GITHUB_TOKEN` (also `GH_TOKEN` / `GITHUB_TOKEN`). See and .

Sessions and multiple chats

How do I start a fresh conversation

Send `/new` or `/reset` as a standalone message. See .

Do sessions reset automatically if I never send new

Yes. Sessions expire after `session.idleMinutes` (default **60**). The **next** message starts a fresh session id for that chat key. This does not delete transcripts - it just starts a new session.



```
session: {  
    idleMinutes: 240,  
},  
}
```

Is there a way to make a team of OpenClaw instances one CEO and many agents

Yes, via **multi-agent routing** and **sub-agents**. You can create one coordinator agent and several worker agents with their own workspaces and models.

That said, this is best seen as a **fun experiment**. It is token heavy and often less efficient than using one bot with separate sessions. The typical model we envision is one bot you talk to, with different sessions for parallel work. That bot can also spawn sub-agents when needed.

Docs:

[Multi-Agent Routing](#) , [Sub-agents](#) , [Parallel Workspaces](#)

Why did context get truncated midtask How do I prevent it

Session context is limited by the model window. Long chats, large tool outputs, or many files can trigger compaction or truncation.

What helps:

Ask the bot to summarize the current state and write it to a file.

Use `/compact` before long tasks, and `/new` when switching topics.

Keep important context in the workspace and ask the bot to read it back.

Use sub-agents for long or parallel work so the main chat stays smaller.



Pick a model with a larger context window if this happens often.

How do I completely reset OpenClaw but keep it installed

Use the reset command:

```
openclaw reset
```

Non-interactive full reset:

```
openclaw reset --scope full --yes --non-interactive
```

Then re-run onboarding:

```
openclaw onboard --install-daemon
```

Notes:

The onboarding wizard also offers **Reset** if it sees an existing config. See [here](#).

If you used profiles (`--profile / OPENCLAW_PROFILE`), reset each state dir (defaults are `~/.openclaw-<profile>`).

Dev reset: `openclaw gateway --dev --reset` (dev-only; wipes dev config + credentials + sessions + workspace).

I'm getting context too large errors how do I reset or compact

Use one of these:

Compact (keeps the conversation but summarizes older turns):



/compact

or `/compact <instructions>` to guide the summary.

Reset (fresh session ID for the same chat key):

/new

/reset

If it keeps happening:

Enable or tune **session pruning** (`agents.defaults.contextPruning`) to trim old tool output.

Use a model with a larger context window.

Docs:

[Session Pruning](#) , [Tool Use](#) , [Input](#) .

Why am I seeing LLM request rejected messages? content X tool use input Field required

This is a provider validation error: the model emitted a `tool_use` block without the required `input`. It usually means the session history is stale or corrupted (often after long threads or a tool/schema change).

Fix: start a fresh session with `/new` (standalone message).

Why am I getting heartbeat messages every 30 minutes?

Heartbeats run every **30m** by default. Tune or disable them:



```

agents: {
  defaults: {
    heartbeat: {
      every: "2h", // or "0m" to disable
    },
  },
},
}

```

If `HEARTBEAT.md` exists but is effectively empty (only blank lines and markdown headers like `# Heading`), OpenClaw skips the heartbeat run to save API calls. If the file is missing, the heartbeat still runs and the model decides what to do.

Per-agent overrides use `agents.list[].heartbeat`. Docs: [...](#).

Do I need to add a bot account to a WhatsApp group

No. OpenClaw runs on **your own account**, so if you're in the group, OpenClaw can see it. By default, group replies are blocked until you allow senders (`groupPolicy: "allowlist"`).

If you want only **you** to be able to trigger group replies:

```
{
  channels: {
    whatsapp: {
      groupPolicy: "allowlist",
      groupAllowFrom: ["+15551234567"],
    },
  },
}
```

How do I get the JID of a WhatsApp group

Option 1 (fastest): tail logs and send a test message in the group:



```
openclaw logs --follow --json
```

>

Look for `chatId` (or `from`) ending in `@g.us`, like: `1234567890-1234567890@g.us`.

Option 2 (if already configured/allowlisted): list groups from config:

```
openclaw directory groups list --channel whatsapp
```

Docs:

[,](#) [,](#) [.](#)

Why doesn't OpenClaw reply in a group

Two common causes:

Mention gating is on (default). You must @mention the bot (or match `mentionPatterns`).

You configured `channels.whatsapp.groups` without "*" and the group isn't allowlisted.

See [See](#) and [and](#).

Do group threads share context with DMs

Direct chats collapse to the main session by default. Groups/channels have their own session keys, and Telegram topics / Discord threads are separate sessions. See [See](#) and [and](#).

How many workspaces and agents can I create

No hard limits. Dozens (even hundreds) are fine, but watch for:



Disk growth: sessions + transcripts live under

`~/.openclaw/agents/<agentId>/sessions/` .

Token cost: more agents means more concurrent model usage.

>

Ops overhead: per-agent auth profiles, workspaces, and channel routing.

Tips:

Keep one **active** workspace per agent (`agents.defaults.workspace`) .

Prune old sessions (delete JSONL or store entries) if disk grows.

Use `openclaw doctor` to spot stray workspaces and profile mismatches.

Can I run multiple bots or chats at the same time Slack and how should I set that up

Yes. Use **Multi-Agent Routing** to run multiple isolated agents and route inbound messages by channel/account/peer. Slack is supported as a channel and can be bound to specific agents.

Browser access is powerful but not “do anything a human can” - anti-bot, CAPTCHAs, and MFA can still block automation. For the most reliable browser control, use the Chrome extension relay on the machine that runs the browser (and keep the Gateway anywhere).

Best-practice setup:

Always-on Gateway host (VPS/Mac mini).

One agent per role (bindings).

Slack channel(s) bound to those agents.

Local browser via extension relay (or a node) when needed.

Docs: [Multi-Agent Routing](#), [Slack](#), [Browser](#), [Chrome extension](#), [Nodes](#).

Models: defaults, selection, aliases, switching



What is the default model

OpenClaw's default model is whatever you set as:

```
agents.defaults.model.primary
```

Models are referenced as `provider/model` (example: `anthropic/clause-opus-4-6`). If you omit the provider, OpenClaw currently assumes `anthropic` as a temporary deprecation fallback - but you should still **explicitly** set `provider/model`.

What model do you recommend

Recommended default: `anthropic/clause-opus-4-6`. **Good alternative:** `anthropic/clause-sonnet-4-5`. **Reliable (less character):** `openai/gpt-5.2` - nearly as good as Opus, just less personality. **Budget:** `zai/glm-4.7`.

MiniMax M2.1 has its own docs: [here](#) and [here](#).

Rule of thumb: use the **best model you can afford** for high-stakes work, and a cheaper model for routine chat or summaries. You can route models per agent and use sub-agents to parallelize long tasks (each sub-agent consumes tokens). See [here](#) and [here](#).

Strong warning: weaker/over-quantized models are more vulnerable to prompt injection and unsafe behavior. See [here](#).

More context: [here](#).

Can I use selfhosted models llamacpp vLLM Ollama

Yes. If your local server exposes an OpenAI-compatible API, you can point a custom provider at it. Ollama is supported directly and is the

easiest path.



Security note: smaller or heavily quantized models are more vulnerable to prompt injection. We strongly recommend **large models** for any bot that can use tools. If you still want small models, enable sandboxing and strict tool allowlists.

Docs: [Ollama](#), [Local models](#), [Model providers](#), [Security](#), [Sandboxing](#).

How do I switch models without wiping my config

Use **model commands** or edit only the **model** fields. Avoid full config replaces.

Safe options:

```
/model in chat (quick, per-session)  
openclaw models set ... (updates just model config)  
openclaw configure --section model (interactive)  
edit agents.defaults.model in ~/.openclaw/openclaw.json
```

Avoid `config.apply` with a partial object unless you intend to replace the whole config. If you did overwrite config, restore from backup or re-run `openclaw doctor` to repair.

Docs: [Models](#), [Configure](#), [Config](#), [Doctor](#).

What do OpenClaw, Flawd, and Krill use for models

OpenClaw + Flawd: Anthropic Opus (`anthropic/clause-opus-4-6`) - see [Anthropic](#).

Krill: MiniMax M2.1 (`minimax/MiniMax-M2.1`) - see [MiniMax](#).

How do I switch models on the fly without restarting

Use the `/model` command as a standalone message:



```
/model sonnet  
/model haiku  
/model opus  
/model gpt  
/model gpt-mini  
/model gemini  
/model gemini-flash
```

You can list available models with `/model`, `/model list`, or `/model status`.

`/model` (and `/model list`) shows a compact, numbered picker. Select by number:

```
/model 3
```

You can also force a specific auth profile for the provider (per session):

```
/model opus@anthropic:default  
/model opus@anthropic:work
```

Tip: `/model status` shows which agent is active, which `auth-profiles.json` file is being used, and which auth profile will be tried next. It also shows the configured provider endpoint (`baseUrl`) and API mode (`api`) when available.

How do I unpin a profile I set with profile

Re-run `/model` without the `@profile` suffix:

 /model anthropic/clause-opus-4-6

If you want to return to the default, pick it from `/model` (or send `/model <default provider/model>`). Use `/model status` to confirm which auth profile is active.

Can I use GPT 5.2 for daily tasks and Codex 5.3 for coding

Yes. Set one as default and switch as needed:

Quick switch (per session): `/model gpt-5.2` for daily tasks, `/model gpt-5.3-codex` for coding.

Default + switch: set `agents.defaults.model.primary` to `openai/gpt-5.2`, then switch to `openai-codex/gpt-5.3-codex` when coding (or the other way around).

Sub-agents: route coding tasks to sub-agents with a different default model.

See [agents.defaults.model](#) and [agents.defaults.models](#).

Why do I see Model is not allowed and then no reply

If `agents.defaults.models` is set, it becomes the **allowlist** for `/model` and any session overrides. Choosing a model that isn't in that list returns:

Model "provider/model" is not allowed. Use `/model` to list available models.

That error is returned **instead of** a normal reply. Fix: add the model to `agents.defaults.models`, remove the allowlist, or pick a model from `/model list`.

Why do I see Unknown model minimaxMiniMaxM21



This means the **provider isn't configured** (no MiniMax provider config or auth profile was found), so the model can't be resolved. A fix for this detection is in **2026.1.12** (unreleased at the time of writing).

Fix checklist:

1. Upgrade to **2026.1.12** (or run from source `main`), then restart the gateway.
2. Make sure MiniMax is configured (wizard or JSON), or that a MiniMax API key exists in env/auth profiles so the provider can be injected.
3. Use the exact model id (case-sensitive): `minimax/MiniMax-M2.1` or `minimax/MiniMax-M2.1-lightning`.
4. Run:

```
openclaw models list
```

and pick from the list (or `/model list` in chat).

See [Minimax](#) and [OpenAI](#).

Can I use MiniMax as my default and OpenAI for complex tasks

Yes. Use **MiniMax as the default** and switch models **per session** when needed. Fallbacks are for **errors**, not "hard tasks," so use `/model` or a separate agent.

Option A: switch per session



```

env: { MINIMAX_API_KEY: "sk-...", OPENAI_API_KEY: "sk-..." },
agents: {
  defaults: {
    model: { primary: "minimax/MiniMax-M2.1" },
    models: {
      "minimax/MiniMax-M2.1": { alias: "minimax" },
      "openai/gpt-5.2": { alias: "gpt" },
    },
  },
},
}

```

Then:

```
/model gpt
```

Option B: separate agents

Agent A default: MiniMax

Agent B default: OpenAI

Route by agent or use /agent to switch

Docs: [https://docs.openclaw.ai/help/faq#agent-routing](#), [https://docs.openclaw.ai/help/faq#agent-routing-with-models](#), [https://docs.openclaw.ai/help/faq#agent-routing-with-priority](#).

Are opus sonnet gpt builtin shortcuts

Yes. OpenClaw ships a few default shorthands (only applied when the model exists in `agents.defaults.models`):

`opus` → `anthropic/clause-opus-4-6`

`sonnet` → `anthropic/clause-sonnet-4-5`

`gpt` → `openai/gpt-5.2`

`gpt-mini` → `openai/gpt-5-mini`



```
gemini → google/gemini-3-pro-preview  
gemini-flash → google/gemini-3-flash-preview
```

If you set your own alias with the same name, your value wins.

How do I define/override model shortcuts/aliases

Aliases come from `agents.defaults.models.<modelId>.alias`. Example:

```
{  
  agents: {  
    defaults: {  
      model: { primary: "anthropic/clause-opus-4-6" },  
      models: {  
        "anthropic/clause-opus-4-6": { alias: "opus" },  
        "anthropic/clause-sonnet-4-5": { alias: "sonnet" },  
        "anthropic/clause-haiku-4-5": { alias: "haiku" },  
      },  
    },  
  },  
}
```

Then `/model sonnet` (or `/<alias>` when supported) resolves to that model ID.

How do I add models from other providers like OpenRouter or ZAI

OpenRouter (pay-per-token; many models):



```

agents: {
  defaults: {
    model: { primary: "openrouter/anthropic/clause-sonnet-4-5" },
    models: { "openrouter/anthropic/clause-sonnet-4-5": {} },
  },
},
env: { OPENROUTER_API_KEY: "sk-or-..." },
}

```

Z.AI (GLM models):

```

{
agents: {
  defaults: {
    model: { primary: "zai/glm-4.7" },
    models: { "zai/glm-4.7": {} },
  },
},
env: { ZAI_API_KEY: "..." },
}

```

If you reference a provider/model but the required provider key is missing, you'll get a runtime auth error (e.g. No API key found for provider "zai").

No API key found for provider after adding a new agent

This usually means the **new agent** has an empty auth store. Auth is per-agent and stored in:

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

Fix options:



Run `openclaw agents add <id>` and configure auth during the wizard.

Or copy `auth-profiles.json` from the main agent's `agentDir` into the new agent's `agentDir`.

Do **not** reuse `agentDir` across agents; it causes auth/session collisions.

Model failover and “All models failed”

How does failover work

Failover happens in two stages:

1. **Auth profile rotation** within the same provider.
2. **Model fallback** to the next model in `agents.defaults.model.fallbacks`.

Coldowns apply to failing profiles (exponential backoff), so OpenClaw can keep responding even when a provider is rate-limited or temporarily failing.

What does this error mean

No credentials found for profile "anthropic:default"

It means the system attempted to use the auth profile ID `anthropic:default`, but could not find credentials for it in the expected auth store.

Fix checklist for No credentials found for profile anthropicdefault

Confirm where auth profiles live (new vs legacy paths)

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

Legacy: `~/.openclaw/agent/*` (migrated by `openclaw doctor`)



Confirm your env var is loaded by the Gateway

If you set `ANTHROPIC_API_KEY` in your shell but run the Gateway via `systemd/launchd`, it may not inherit it. Put it in `~/.openclaw/.env` or enable `env.shellEnv`.

Make sure you're editing the correct agent

Multi-agent setups mean there can be multiple `auth-profiles.json` files.

Sanity-check model/auth status

Use `openclaw models status` to see configured models and whether providers are authenticated.

Fix checklist for No credentials found for profile anthropic

This means the run is pinned to an Anthropic auth profile, but the Gateway can't find it in its auth store.

Use a setup-token

Run `claude setup-token`, then paste it with `openclaw models auth setup-token --provider anthropic`.

If the token was created on another machine, use `openclaw models auth paste-token --provider anthropic`.

If you want to use an API key instead

Put `ANTHROPIC_API_KEY` in `~/.openclaw/.env` on the **gateway host**.

Clear any pinned order that forces a missing profile:

```
openclaw models auth order clear --provider anthropic
```

Confirm you're running commands on the gateway host

In remote mode, auth profiles live on the gateway machine, not your laptop.

Why did it also try Google Gemini and fail



If your model config includes Google Gemini as a fallback (or you switched to a Gemini shorthand), OpenClaw will try it during model fallback. If you haven't configured Google credentials, you'll see `No API key found for provider "google"`.

Fix: either provide Google auth, or remove/avoid Google models in `agents.defaults.model.fallbacks` / aliases so fallback doesn't route there.

LLM request rejected message thinking signature required google antigravity

Cause: the session history contains **thinking blocks without signatures** (often from an aborted/partial stream). Google Antigravity requires signatures for thinking blocks.

Fix: OpenClaw now strips unsigned thinking blocks for Google Antigravity Claude. If it still appears, start a **new session** or set `/thinking off` for that agent.

Auth profiles: what they are and how to manage them

Related: [/concepts/oauth](#) (OAuth flows, token storage, multi-account patterns)

What is an auth profile

An auth profile is a named credential record (OAuth or API key) tied to a provider. Profiles live in:

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

What are typical profile IDs



OpenClaw uses provider-prefixed IDs like:

```
> anthropic:default (common when no email identity exists)  
anthropic:<email> for OAuth identities  
custom IDs you choose (e.g. anthropic:work )
```

Can I control which auth profile is tried first

Yes. Config supports optional metadata for profiles and an ordering per provider (`auth.order.<provider>`). This does **not** store secrets; it maps IDs to provider/mode and sets rotation order.

OpenClaw may temporarily skip a profile if it's in a short **cooldown** (rate limits/timeouts/auth failures) or a longer **disabled** state (billing/insufficient credits). To inspect this, run `openclaw models status --json` and check `auth.unusableProfiles`. Tuning: `auth.cooldowns.billingBackoffHours*`.

You can also set a **per-agent** order override (stored in that agent's `auth-profiles.json`) via the CLI:

```
# Defaults to the configured default agent (omit --agent)  
openclaw models auth order get --provider anthropic  
  
# Lock rotation to a single profile (only try this one)  
openclaw models auth order set --provider anthropic anthropic:default  
  
# Or set an explicit order (fallback within provider)  
openclaw models auth order set --provider anthropic anthropic:work anthropic:default  
  
# Clear override (fall back to config auth.order / round-robin)  
openclaw models auth order clear --provider anthropic
```

To target a specific agent:

```
openclaw models auth order set --provider anthropic --agent main anthr
```

>

OAuth vs API key what's the difference

OpenClaw supports both:

OAuth often leverages subscription access (where applicable).

API keys use pay-per-token billing.

The wizard explicitly supports Anthropic setup-token and OpenAI Codex OAuth and can store API keys for you.

Gateway: ports, “already running”, and remote mode

What port does the Gateway use

`gateway.port` controls the single multiplexed port for WebSocket + HTTP (Control UI, hooks, etc.).

Precedence:

```
--port > OPENCLAW_GATEWAY_PORT > gateway.port > default 18789
```

Why does `openclaw gateway status` say Runtime running but RPC probe failed

Because “running” is the **supervisor’s** view (launchd/systemd/schtasks). The RPC probe is the CLI actually connecting to the gateway WebSocket and calling `status`.

Use `openclaw gateway status` and trust these lines:

Probe target: (the URL the probe actually used)



Listening: (what's actually bound on the port)

Last gateway error: (common root cause when the process is alive but the port isn't listening)

>

Why does openclaw gateway status show Config cli and Config service different

You're editing one config file while the service is running another (often a `--profile` / `OPENCLAW_STATE_DIR` mismatch).

Fix:

```
openclaw gateway install --force
```

Run that from the same `--profile` / environment you want the service to use.

What does another gateway instance is already listening mean

OpenClaw enforces a runtime lock by binding the WebSocket listener immediately on startup (default `ws://127.0.0.1:18789`). If the bind fails with `EADDRINUSE`, it throws `GatewayLockError` indicating another instance is already listening.

Fix: stop the other instance, free the port, or run with `openclaw gateway --port <port>`.

How do I run OpenClaw in remote mode client connects to a Gateway elsewhere

Set `gateway.mode: "remote"` and point to a remote WebSocket URL, optionally with a token/password:



```
gateway: {
  mode: "remote",
  remote: {
    url: "ws://gateway.tailnet:18789",
    token: "your-token",
    password: "your-password",
  },
},
}
```

Notes:

`openclaw gateway` only starts when `gateway.mode` is `local` (or you pass the override flag).

The macOS app watches the config file and switches modes live when these values change.

The Control UI says unauthorized or keeps reconnecting What now

Your gateway is running with auth enabled (`gateway.auth.*`), but the UI is not sending the matching token/password.

Facts (from code):

The Control UI stores the token in browser localStorage key `openclaw.control.settings.v1`.

Fix:

Fastest: `openclaw dashboard` (prints + copies the dashboard URL, tries to open; shows SSH hint if headless).

If you don't have a token yet: `openclaw doctor --generate-gateway-token`.

If remote, tunnel first: `ssh -N -L 18789:127.0.0.1:18789 user@host` then open `http://127.0.0.1:18789/`.



Set `gateway.auth.token` (or `OPENCLAW_GATEWAY_TOKEN`) on the gateway host.

In the Control UI settings, paste the same token.

>

Still stuck? Run `openclaw status --all` and follow [Troubleshooting](#).

See [Dashboard](#) for auth details.

I set `gatewaybind tailnet` but it cant bind nothing listens

`tailnet` bind picks a Tailscale IP from your network interfaces (100.64.0.0/10). If the machine isn't on Tailscale (or the interface is down), there's nothing to bind to.

Fix:

Start Tailscale on that host (so it has a 100.x address), or

Switch to `gateway.bind: "loopback" / "lan"`.

Note: `tailnet` is explicit. `auto` prefers loopback; use `gateway.bind: "tailnet"` when you want a tailnet-only bind.

Can I run multiple Gateways on the same host

Usually no - one Gateway can run multiple messaging channels and agents. Use multiple Gateways only when you need redundancy (ex: rescue bot) or hard isolation.

Yes, but you must isolate:

`OPENCLAW_CONFIG_PATH` (per-instance config)

`OPENCLAW_STATE_DIR` (per-instance state)

`agents.defaults.workspace` (workspace isolation)

`gateway.port` (unique ports)

Quick setup (recommended):

 Use `openclaw --profile <name> ...` per instance (auto-creates `~/.openclaw-<name>`).

Set a unique `gateway.port` in each profile config (or pass `--port` for manual runs).

Install a per-profile service: `openclaw --profile <name> gateway install`.

Profiles also suffix service names (`bot.molt.<profile>` ; legacy `com.openclaw.*` , `openclaw-gateway-<profile>.service` , `OpenClaw Gateway (<profile>)`). Full guide: [Multiple gateways](#).

What does invalid handshake code 1008 mean

The Gateway is a **WebSocket server**, and it expects the very first message to be a `connect` frame. If it receives anything else, it closes the connection with **code 1008** (policy violation).

Common causes:

You opened the **HTTP** URL in a browser (`http://...`) instead of a WS client.

You used the wrong port or path.

A proxy or tunnel stripped auth headers or sent a non-Gateway request.

Quick fixes:

1. Use the WS URL: `ws://<host>:18789` (or `wss://... if HTTPS`).
2. Don't open the WS port in a normal browser tab.
3. If auth is on, include the token/password in the `connect` frame.

If you're using the CLI or TUI, the URL should look like:

```
openclaw tui --url ws://<host>:18789 --token <token>
```

Protocol details: [Gateway protocol.](#)



Logging and debugging

Where are logs

File logs (structured):

```
/tmp/openclaw/openclaw-YYYY-MM-DD.log
```

You can set a stable path via `logging.file`. File log level is controlled by `logging.level`. Console verbosity is controlled by `--verbose` and `logging.consoleLevel`.

Fastest log tail:

```
openclaw logs --follow
```

Service/supervisor logs (when the gateway runs via launchd/systemd):

macOS: `$OPENCLAW_STATE_DIR/logs/gateway.log` and `gateway.err.log`
(default: `~/.openclaw/logs/...` ; profiles use `~/.openclaw-<profile>/logs/...`)

Linux: `journalctl --user -u openclaw-gateway[-<profile>].service -n 200 --no-pager`

Windows: `schtasks /Query /TN "OpenClaw Gateway (<profile>)" /V /FO LIST`

See [FAQ](#) for more.

How do I start/stop/restart the Gateway service

Use the gateway helpers:

```
openclaw gateway status  
openclaw gateway restart
```

>

If you run the gateway manually, `openclaw gateway --force` can reclaim the port. See .

I closed my terminal on Windows how do I restart OpenClaw

There are **two Windows install modes**:

- 1) **WSL2 (recommended)**: the Gateway runs inside Linux.

Open PowerShell, enter WSL, then restart:

```
wsl  
openclaw gateway status  
openclaw gateway restart
```

If you never installed the service, start it in the foreground:

```
openclaw gateway run
```

- 2) **Native Windows (not recommended)**: the Gateway runs directly in Windows .

Open PowerShell and run:

```
openclaw gateway status  
openclaw gateway restart
```

If you run it manually (no service), use:

 openclaw gateway runDocs: [Gateway](#) [Status](#) [Logs](#)

The Gateway is up but replies never arrive What should I check

Start with a quick health sweep:

```
openclaw status  
openclaw models status  
openclaw channels status  
openclaw logs --follow
```

Common causes:

Model auth not loaded on the **gateway host** (check `models status`).

Channel pairing/allowlist blocking replies (check channel config + logs).

WebChat/Dashboard is open without the right token.

If you are remote, confirm the tunnel/Tailscale connection is up and that the Gateway WebSocket is reachable.

Docs: [Gateway](#) [Status](#) [Logs](#)

Disconnected from gateway no reason what now

This usually means the UI lost the WebSocket connection. Check:

1. Is the Gateway running? `openclaw gateway status`
2. Is the Gateway healthy? `openclaw status`
3. Does the UI have the right token? `openclaw dashboard`
4. If remote, is the tunnel/Tailscale link up?

Then tail logs:



```
openclaw logs --follow
```

Docs:

[,](#) [,](#) [.](#)

Telegram setMyCommands fails with network errors What should I check

Start with logs and channel status:

```
openclaw channels status
openclaw channels logs --channel telegram
```

If you are on a VPS or behind a proxy, confirm outbound HTTPS is allowed and DNS works. If the Gateway is remote, make sure you are looking at logs on the Gateway host.

Docs:

[,](#) [.](#)

TUI shows no output What should I check

First confirm the Gateway is reachable and the agent can run:

```
openclaw status
openclaw models status
openclaw logs --follow
```

In the TUI, use `/status` to see the current state. If you expect replies in a chat channel, make sure delivery is enabled (`/deliver on`).

Docs:

[,](#) [.](#)

How do I completely stop then start the Gateway



If you installed the service:

```
>  
openclaw gateway stop  
openclaw gateway start
```

This stops/starts the **supervised service** (launchd on macOS, systemd on Linux). Use this when the Gateway runs in the background as a daemon.

If you're running in the foreground, stop with Ctrl-C, then:

```
openclaw gateway run
```

Docs: [.](#)

ELI5 openclaw gateway restart vs openclaw gateway

`openclaw gateway restart` : restarts the **background service** (launchd/systemd).

`openclaw gateway` : runs the gateway **in the foreground** for this terminal session.

If you installed the service, use the gateway commands. Use `openclaw gateway` when you want a one-off, foreground run.

What's the fastest way to get more details when something fails

Start the Gateway with `--verbose` to get more console detail. Then inspect the log file for channel auth, model routing, and RPC errors.

Media and attachments

My skill generated an imagePDF but nothing was sent



Outbound attachments from the agent must include a `MEDIA:<path-or-url>` line (on its own line). See [OpenClaw assistant setup](#) and [Agent send](#).

CLI sending:

```
openclaw message send --target +15555550123 --message "Here you go" --
```

Also check:

The target channel supports outbound media and isn't blocked by allowlists.

The file is within the provider's size limits (images are resized to max 2048px).

See

.

Security and access control

Is it safe to expose OpenClaw to inbound DMs

Treat inbound DMs as untrusted input. Defaults are designed to reduce risk:

Default behavior on DM-capable channels is **pairing**:

Unknown senders receive a pairing code; the bot does not process their message.

Approve with: `openclaw pairing approve <channel> <code>`

Pending requests are capped at **3 per channel**; check `openclaw pairing list <channel>` if a code didn't arrive.

Opening DMs publicly requires explicit opt-in (`dmPolicy: "open"` and `allowlist: "*"`).

Run `openclaw doctor` to surface risky DM policies.

Is prompt injection only a concern for public bots

No. Prompt injection is about **untrusted content**, not just who can DM the bot. If your assistant reads external content (web search/fetch, browser pages, emails, docs, attachments, pasted logs), that content can include instructions that try to hijack the model. This can happen even if **you are the only sender**.

The biggest risk is when tools are enabled: the model can be tricked into exfiltrating context or calling tools on your behalf. Reduce the blast radius by:

- using a read-only or tool-disabled “reader” agent to summarize untrusted content

- keeping `web_search` / `web_fetch` / `browser` off for tool-enabled agents

- sandboxing and strict tool allowlists

Details: [**Security**](#).

Should my bot have its own email GitHub account or phone number

Yes, for most setups. Isolating the bot with separate accounts and phone numbers reduces the blast radius if something goes wrong. This also makes it easier to rotate credentials or revoke access without impacting your personal accounts.

Start small. Give access only to the tools and accounts you actually need, and expand later if required.

Docs: [**Security**](#), [**Pairing**](#).

Can I give it autonomy over my text messages and is that safe



We do **not** recommend full autonomy over your personal messages. The safest pattern is:

Keep DMs in **pairing mode** or a tight allowlist.

Use a **separate number or account** if you want it to message on your behalf.

Let it draft, then **approve before sending**.

If you want to experiment, do it on a dedicated account and keep it isolated. See [Security](#).

Can I use cheaper models for personal assistant tasks

Yes, **if** the agent is chat-only and the input is trusted. Smaller tiers are more susceptible to instruction hijacking, so avoid them for tool-enabled agents or when reading untrusted content. If you must use a smaller model, lock down tools and run inside a sandbox. See [Security](#).

I ran start in Telegram but didn't get a pairing code

Pairing codes are sent **only** when an unknown sender messages the bot and `dmPolicy: "pairing"` is enabled. `/start` by itself doesn't generate a code.

Check pending requests:

```
openclaw pairing list telegram
```

If you want immediate access, allowlist your sender id or set `dmPolicy: "open"` for that account.

WhatsApp will it message my contacts How does pairing work

No. Default WhatsApp DM policy is **pairing**. Unknown senders only get a pairing code and their message is **not processed**. OpenClaw only replies to chats it receives or to explicit sends you trigger.

>

Approve pairing with:

```
openclaw pairing approve whatsapp <code>
```

List pending requests:

```
openclaw pairing list whatsapp
```

Wizard phone number prompt: it's used to set your **allowlist/owner** so your own DMs are permitted. It's not used for auto-sending. If you run on your personal WhatsApp number, use that number and enable `channels.whatsapp.selfChatMode`.

Chat commands, aborting tasks, and “it won’t stop”

How do I stop internal system messages from showing in chat

Most internal or tool messages only appear when **verbose** or **reasoning** is enabled for that session.

Fix in the chat where you see it:

```
/verbose off  
/reasoning off
```

If it is still noisy, check the session settings in the Control UI and set verbose to **inherit**. Also confirm you are not using a bot profile with `verboseDefault` set to `on` in config.

Docs: [Thinking and verbose](#), [Security](#).



How do I stop/cancel a running task

Send any of these **as a standalone message** (no slash):

```
stop  
abort  
esc  
wait  
exit  
interrupt
```

These are abort triggers (not slash commands).

For background processes (from the exec tool), you can ask the agent to run:

```
process action:kill sessionId:XXX
```

Slash commands overview: see [.](#)

Most commands must be sent as a **standalone** message that starts with `/`, but a few shortcuts (like `/status`) also work inline for allowlisted senders.

How do I send a Discord message from Telegram Crosscontext messaging denied

OpenClaw blocks **cross-provider** messaging by default. If a tool call is bound to Telegram, it won't send to Discord unless you explicitly allow it.

Enable cross-provider messaging for the agent:



```
agents: {
    defaults: {
        tools: {
            message: {
                crossContext: {
                    allowAcrossProviders: true,
                    marker: { enabled: true, prefix: "[from {channel}] " },
                },
            },
        },
    },
}
```

Restart the gateway after editing config. If you only want this for a single agent, set it under `agents.list[].tools.message` instead.

Why does it feel like the bot ignores rapidfire messages

Queue mode controls how new messages interact with an in-flight run. Use `/queue` to change modes:

- `steer` - new messages redirect the current task
- `followup` - run messages one at a time
- `collect` - batch messages and reply once (default)
- `steer-backlog` - steer now, then process backlog
- `interrupt` - abort current run and start fresh

You can add options like `debounce:2s cap:25 drop:summarize` for followup modes.

Answer the exact question from the screenshot/chat log

Q: "What's the default model for Anthropic with an API key?"



A: In OpenClaw, credentials and model selection are separate. Setting `ANTHROPIC_API_KEY` (or storing an Anthropic API key in auth profiles) enables authentication, but the actual default model is whatever you configure in `agents.defaults.model.primary` (for example, `anthropic/claudesonnet-4-5` or `anthropic/claudie-opus-4-6`). If you see `No credentials found for profile "anthropic:default"`, it means the Gateway couldn't find Anthropic credentials in the expected `auth-profiles.json` for the agent that's running.

Still stuck? Ask in [Discord](#) or open a [GitHub discussion](#).

◀ [Troubleshooting](#)

[OpenClaw Lore](#) ▶

Powered by [mintlify](#)