



---

≡ Configuration and operations > **Doctor**

---

## Configuration and operations

# Doctor

`openclaw doctor` is the repair + migration tool for OpenClaw. It fixes stale config/state, checks health, and provides actionable repair steps.

## Quick start

```
openclaw doctor
```

## Headless / automation

```
openclaw doctor --yes
```

Accept defaults without prompting (including restart/service/sandbox repair steps when applicable).

```
openclaw doctor --repair
```

Apply recommended repairs without prompting (repairs + restarts where safe).

```
openclaw doctor --repair --force
```

Apply aggressive repairs too (overwrites custom supervisor configs).



```
openclaw doctor --non-interactive  
>
```

Run without prompts and only apply safe migrations (config normalization + on-disk state moves). Skips restart/service/sandbox actions that require human confirmation. Legacy state migrations run automatically when detected.

```
openclaw doctor --deep
```

Scan system services for extra gateway installs (launchd/systemd/schtasks).

If you want to review changes before writing, open the config file first:

```
cat ~/.openclaw/openclaw.json
```

## What it does (summary)

Optional pre-flight update for git installs (interactive only).

UI protocol freshness check (rebuilds Control UI when the protocol schema is newer).

Health check + restart prompt.

Skills status summary (eligible/missing/blocked).

Config normalization for legacy values.

OpenCode Zen provider override warnings ( `models.providers.opencode` ).

Legacy on-disk state migration (sessions/agent dir/WhatsApp auth).



State integrity and permissions checks (sessions, transcripts, state dir).

Config file permission checks (chmod 600) when running locally.

Model auth health: checks OAuth expiry, can refresh expiring tokens, and reports auth-profile cooldown/disabled states.

Extra workspace dir detection ( ~/openclaw ).

Sandbox image repair when sandboxing is enabled.

Legacy service migration and extra gateway detection.

Gateway runtime checks (service installed but not running; cached launchd label).

Channel status warnings (probed from the running gateway).

Supervisor config audit (launchd/systemd/schtasks) with optional repair.

Gateway runtime best-practice checks (Node vs Bun, version-manager paths).

Gateway port collision diagnostics (default 18789 ).

Security warnings for open DM policies.

Gateway auth warnings when no gateway.auth.token is set (local mode; offers token generation).

systemd linger check on Linux.

Source install checks (pnpm workspace mismatch, missing UI assets, missing tsx binary).

Writes updated config + wizard metadata.

## Detailed behavior and rationale

### 0) Optional update (git installs)

If this is a git checkout and doctor is running interactively, it offers to update (fetch/rebase/build) before running doctor.

## 1) Config normalization



If the config contains legacy value shapes (for example `messages.ackReaction` without a channel-specific override), doctor normalizes them into the current schema.

## 2) Legacy config key migrations

When the config contains deprecated keys, other commands refuse to run and ask you to run `openclaw doctor`.

Doctor will:

Explain which legacy keys were found.

Show the migration it applied.

Rewrite `~/.openclaw/openclaw.json` with the updated schema.

The Gateway also auto-runs doctor migrations on startup when it detects a legacy config format, so stale configs are repaired without manual intervention.

Current migrations:

`routing.allowFrom` → `channels.whatsapp.allowFrom`

`routing.groupChat.requireMention` →  
`channels.whatsapp/telegram/imessage.groups.*.requireMention`

`routing.groupChat.historyLimit` → `messages.groupChat.historyLimit`

`routing.groupChat.mentionPatterns` → `messages.groupChat.mentionPatterns`

`routing.queue` → `messages.queue`

`routing.bindings` → top-level bindings

`routing.agents` / `routing.defaultAgentId` → `agents.list` +  
`agents.list[].default`

`routing.agentToAgent` → `tools.agentToAgent`

`routing.transcribeAudio` → `tools.media.audio.models`



```
bindings[].match.accountID → bindings[].match.accountId
```

```
identity → agents.list[].identity
```

```
agent.* → agents.defaults , + tools.*
```

```
(tools/elevated/exec/sandbox/subagents)
```

```
agent.model / allowedModels / modelAliases / modelFallbacks / imageModelFallbacks  
→ agents.defaults.models + agents.defaults.model.primary/fallbacks +  
agents.defaults.imageModel.primary/fallbacks
```

## 2b) OpenCode Zen provider overrides

If you've added `models.providers.opencode` (or `opencode-zen`) manually, it overrides the built-in OpenCode Zen catalog from `@mariozechner/pi-ai`. That can force every model onto a single API or zero out costs. Doctor warns so you can remove the override and restore per-model API routing + costs.

## 3) Legacy state migrations (disk layout)

Doctor can migrate older on-disk layouts into the current structure:

Sessions store + transcripts:

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

Agent dir:

```
from ~/.openclaw/agent/ to ~/.openclaw/agents/<agentId>/agent/
```

WhatsApp auth state (Baileys):

```
from legacy ~/.openclaw/credentials/*.json (except oauth.json)  
to ~/.openclaw/credentials/whatsapp/<accountId>/... (default account  
id: default)
```

These migrations are best-effort and idempotent; doctor will emit warnings when it leaves any legacy folders behind as backups. The Gateway/CLI also auto-migrates the legacy sessions + agent dir on

startup so history/auth/models land in the per-agent path without a manual doctor run. WhatsApp auth is intentionally only migrated via `openclaw doctor` .

&gt;

## 4) State integrity checks (session persistence, routing, and safety)

The state directory is the operational brainstem. If it vanishes, you lose sessions, credentials, logs, and config (unless you have backups elsewhere).

Doctor checks:

**State dir missing:** warns about catastrophic state loss, prompts to recreate the directory, and reminds you that it cannot recover missing data.

**State dir permissions:** verifies writability; offers to repair permissions (and emits a `chown` hint when owner/group mismatch is detected).

**Session dirs missing:** `sessions/` and the session store directory are required to persist history and avoid `ENOENT` crashes.

**Transcript mismatch:** warns when recent session entries have missing transcript files.

**Main session “1-line JSONL”:** flags when the main transcript has only one line (history is not accumulating).

**Multiple state dirs:** warns when multiple `~/.openclaw` folders exist across home directories or when `OPENCLAW_STATE_DIR` points elsewhere (history can split between installs).

**Remote mode reminder:** if `gateway.mode=remote` , doctor reminds you to run it on the remote host (the state lives there).

**Config file permissions:** warns if `~/.openclaw/openclaw.json` is group/world readable and offers to tighten to `600` .

## 5) Model auth health (OAuth expiry)



Doctor inspects OAuth profiles in the auth store, warns when tokens are expiring/expired, and can refresh them when safe. If the Anthropic Claude Code profile is stale, it suggests running `claude setup-token` (or pasting a setup-token). Refresh prompts only appear when running interactively (TTY); `--non-interactive` skips refresh attempts.

Doctor also reports auth profiles that are temporarily unusable due to:

- short cooldowns (rate limits/timeouts/auth failures)
- longer disables (billing/credit failures)

## 6) Hooks model validation

If `hooks.gmail.model` is set, doctor validates the model reference against the catalog and allowlist and warns when it won't resolve or is disallowed.

## 7) Sandbox image repair

When sandboxing is enabled, doctor checks Docker images and offers to build or switch to legacy names if the current image is missing.

## 8) Gateway service migrations and cleanup hints

Doctor detects legacy gateway services (launchd/systemd/schtasks) and offers to remove them and install the OpenClaw service using the current gateway port. It can also scan for extra gateway-like services and print cleanup hints. Profile-named OpenClaw gateway services are considered first-class and are not flagged as “extra.”

## 9) Security warnings

Doctor emits warnings when a provider is open to DMs without an allowlist, or when a policy is configured in a dangerous way.

---

&gt;

## 10) systemd linger (Linux)

If running as a systemd user service, doctor ensures lingering is enabled so the gateway stays alive after logout.

## 11) Skills status

Doctor prints a quick summary of eligible/missing/blocked skills for the current workspace.

## 12) Gateway auth checks (local token)

Doctor warns when `gateway.auth` is missing on a local gateway and offers to generate a token. Use `openclaw doctor --generate-gateway-token` to force token creation in automation.

## 13) Gateway health check + restart

Doctor runs a health check and offers to restart the gateway when it looks unhealthy.

## 14) Channel status warnings

If the gateway is healthy, doctor runs a channel status probe and reports warnings with suggested fixes.

## 15) Supervisor config audit + repair

Doctor checks the installed supervisor config (launchd/systemd/schtasks) for missing or outdated defaults (e.g., systemd network-online dependencies and restart delay). When it finds



a mismatch, it recommends an update and can rewrite the service file/task to the current defaults.

Notes:

>

`openclaw doctor` prompts before rewriting supervisor config.

`openclaw doctor --yes` accepts the default repair prompts.

`openclaw doctor --repair` applies recommended fixes without prompts.

`openclaw doctor --repair --force` overwrites custom supervisor configs.

You can always force a full rewrite via `openclaw gateway install --force`.

## 16) Gateway runtime + port diagnostics

Doctor inspects the service runtime (PID, last exit status) and warns when the service is installed but not actually running. It also checks for port collisions on the gateway port (default 18789) and reports likely causes (gateway already running, SSH tunnel).

## 17) Gateway runtime best practices

Doctor warns when the gateway service runs on Bun or a version-managed Node path ( `nvm` , `fnm` , `volta` , `asdf` , etc.). WhatsApp + Telegram channels require Node, and version-manager paths can break after upgrades because the service does not load your shell init. Doctor offers to migrate to a system Node install when available (Homebrew/apt/choco).

## 18) Config write + wizard metadata

Doctor persists any config changes and stamps wizard metadata to record the doctor run.

## 19) Workspace tips (backup + memory system)



Doctor suggests a workspace memory system when missing and prints a backup tip if the workspace is not already under git.

See [/concepts/agent-workspace](#) for a full guide to workspace structure and git backup (recommended private GitHub or GitLab).

[< Heartbeat](#)

[Logging >](#)

Powered by [mintlify](#)