OpenClaw

Run the Gateway, connect a model and channels, then chat.

01 Install & onboard

```
curl -fsSL https://openclaw.ai/install.sh \
  | bash
npm install -g openclaw@latest \
  --allow-scripts=openclaw  # npm 11.16+
openclaw onboard --install-daemon
openclaw --version
```

Needs Node 24.16+ or 26.1+; the script installs Node if missing, then runs onboarding. Quick start reuses a Claude Code or Codex login or an API key.

02 Gateway service

```
openclaw gateway install  # service, once
openclaw gateway status   # port 18789
openclaw gateway restart
openclaw gateway stop
openclaw gateway          # in this terminal
openclaw dashboard        # open Control UI
```

A LaunchAgent on macOS, a systemd user unit on Linux, a Scheduled Task on Windows. On a headless server: sudo loginctl enable-linger $(whoami).

03 Status, logs & doctor

```
openclaw status --all    # full diagnosis
openclaw status --deep   # probe channels
openclaw status --usage  # quota left
openclaw logs --follow   # tail Gateway logs
openclaw doctor --fix    # apply repairs
openclaw triage          # guided diagnosis
```

Log times are local; --utc switches. If the CLI is broken on Linux: journalctl --user -u openclaw-gateway -f.

04 Config file & CLI

```
openclaw config get gateway.port
openclaw config set gateway.port 19001 \
  --strict-json
openclaw config unset session.dmScope
openclaw config validate
openclaw configure --section channels
```

~/.openclaw/openclaw.json is JSON5. Unknown keys stop the Gateway from starting. Most edits hot-reload; the CLI says when a restart is needed.

05 Models

```
openclaw models list
openclaw models set openai/gpt-5.4
openclaw models aliases add Opus \
  anthropic/claude-opus-4-6
openclaw models auth login --provider openai
openclaw models status  # auth and expiry
```

Refs are provider/model. status --probe sends real requests and needs the Gateway stopped.

06 Channels & pairing

```
openclaw channels add --channel telegram \
  --token <bot-token>
openclaw channels login --channel whatsapp
openclaw channels status --probe
openclaw pairing list telegram
openclaw pairing approve telegram <CODE>
```

Also Discord, Slack, Signal, iMessage, Matrix, Microsoft Teams and more. An unknown DM sender gets a pairing code (valid 1 hour) and is ignored until you approve it.

07 Workspace files

`AGENTS.md`	Operating rules, ## Tools notes
`SOUL.md`	Persona, tone and boundaries
`USER.md`	Your preferences and context
`MEMORY.md`	Curated long-term memory
`BOOTSTRAP.md`	First-run ritual; delete after

~/.openclaw/workspace is the default cwd, not a sandbox. TOOLS.md and HEARTBEAT.md are retired.

08 Chat: sessions & models

`/new [model]`	Archive session, start fresh
`/reset [soft]`	Reset the session in place
`/compact [text]`	Summarize to free context
`/stop`	Abort the run and sub-agents
`/model <ref> -s`	Switch model, this session
`/think <level>`	off, low, medium, high, more

/model default -s clears the session model pin.

09 Chat: context & control

`/status`	Runtime, plugin health, quota
`/context detail`	What fills the context window
`/usage tokens`	Usage footer: off, full, cost
`/btw <question>`	Side question, not saved
`/steer <message>`	Guide the active run

/commands lists the rest (/tools, /whoami). Keep /reasoning and /verbose off in group chats.

OpenClaw

Skills, automations and tools behind a locked-down Gateway.

10 Skills

```
openclaw skills search "calendar"
openclaw skills install @owner/<slug>
openclaw skills update --all
openclaw skills check  # what is missing
```

```
---
name: hello-world
description: Prints a greeting.
---
```

Path: <workspace>/skills/<name>/SKILL.md. Third-party skills are untrusted code: read them first.

11 Automations (cron)

```
openclaw automations create "0 7 * * *" \
  "Summarize overnight updates." \
  --name "Morning brief"
openclaw automations add --name Standup \
  --at 20m --session main --wake now \
  --system-event "Reminder: standup"
openclaw automations list --all
```

openclaw cron is the same command. Prompt jobs run isolated and announce the result; --no-deliver stops that.

12 Hooks

```
openclaw hooks list
openclaw hooks enable session-memory
openclaw hooks enable boot-md
openclaw hooks check
```

session-memory saves recent chat to memory/ on /new and /reset; boot-md runs BOOT.md at Gateway start.

13 Web search & memory

```
// ~/.openclaw/openclaw.json (JSON5)
{ tools: { web: { search: {
  provider: "brave",  // omit: auto-detect
  maxResults: 5,
} } } }
```

```
openclaw configure --section web
openclaw configure --section memory
openclaw memory search "deploy notes"
```

DuckDuckGo and SearXNG need no key. Brave, Tavily and Exa do: BRAVE_API_KEY in ~/.openclaw/.env.

14 Sub-agents & agents

```
openclaw agents add work \
  --workspace ~/.openclaw/workspace-work
openclaw agents bind --agent work \
  --bind telegram:ops
openclaw agents list --bindings
```

In chat: /subagents list, log <id>, info <id>. /stop in the parent chat also stops its sub-agents.

15 MCP servers

```
openclaw mcp add memory --command npx \
  --arg -y \
  --arg @modelcontextprotocol/server-memory
openclaw mcp status --verbose
openclaw mcp probe memory
```

Saved under mcp.servers. add, set, unset and probe need the Gateway stopped. Pass secrets as "${MCP_TOKEN}", not literals.

16 Lock down the Gateway

```
openclaw config set gateway.bind loopback
openclaw config set gateway.auth.mode token
openclaw doctor --generate-gateway-token
openclaw gateway auth-token --show
openclaw security audit --deep
```

Auth is required even on loopback. For remote access prefer gateway.tailscale.mode "serve" or an SSH tunnel over bind "lan". Never expose port 18789 publicly.

17 Access & sandbox

```
{
  channels: { telegram: {
    dmPolicy: "allowlist",  // or "pairing"
    allowFrom: ["123456789"],
  } },
  agents: { defaults: {
    sandbox: { mode: "non-main" },
  } },
}
```

dmPolicy: pairing (default), allowlist, open, disabled. Sandbox modes: off (default), non-main, all.

18 Devices, secrets & updates

```
openclaw devices list
openclaw devices approve <requestId>
openclaw secrets store set OPENAI_API_KEY
openclaw update --dry-run
openclaw backup create --verify
```

New browsers and nodes wait for device approval. Run backup create before an update, reset or uninstall.