OPENCLAW FIELD GUIDE TO

OpenClaw

Everything your followers have been asking about - the workspace files, memory system, skills, cost control, and how to set it all up properly.

- 01 · INTRODUCTION

Everyone's been asking. Here's your answer.

OpenClaw is an AI agent framework. This guide is the complete breakdown - workspace files, memory architecture, skills, cost control and setup. No fluff, no theory. Just what actually works.

WHAT THIS GUIDE COVERS

The OpenClaw framework and how it works · The 6 workspace files that make up OpenClaw's brain · How to set OpenClaw up safely · Where to host him · Memory optimisation · API cost control · How to make money with it

Read it once to understand the system. Keep it open while you build. Share it with anyone who asks how to get started with OpenClaw.

- 02 · THE FOUNDATION

What actually is OpenClaw?

FROM OPENCLAW

OpenClaw is - a free, open-source AI agent framework. It connects a large language model like Claude to your real computer and real accounts - so it can actually do things, not just talk about them.

→ **250,000+ GitHub stars in weeks**

Started as "Clawdbot" in November 2025, renamed to Moltbot, then OpenClaw after trademark issues. One of the fastest-growing open-source projects ever built.

→ **Free, open-source, local-first**

Runs on your hardware. Works with Claude, GPT, DeepSeek, or any local model via Ollama. Your data never leaves your machine unless you point it at a cloud model API.

→ **The Gateway is the control plane**

One process manages all channels - WhatsApp, Telegram, Discord, iMessage. Routes messages to the right agent, loads workspace files, calls the LLM, executes tools, delivers replies.

→ **Not a chatbot - an agent**

ChatGPT answers questions. OpenClaw executes tasks. Shell commands, browser automation, email, files, APIs - real actions on real accounts, triggered by a text message.

- 03 · WHAT OPENCLAW DOES

OpenClaw doesn't just answer. It executes.

Every time you message OpenClaw, here is what happens behind the scenes:

→ **Gateway receives your message on port 18789**
Identifies which agent, which session, which workspace to load. Routes accordingly.

→ **Runtime reads all workspace files**
SOUL.md, AGENTS.md, id_entity.md, MEMORY.md, USER.md, HEARTBEAT.md - assembled into one system prompt every single turn.

→ **LLM reasons, then tool calls execute**
Claude reads the full context, decides what to do, issues tool calls. The runtime intercepts and executes them - bash commands, browser control via Chrome DevTools Protocol, file reads and writes, API calls.

→ **Response delivered back to your channel**
Final reply sent to WhatsApp or Telegram. The session transcript is saved as a JSONL file on disk for continuity across sessions.

WHAT OPENCLAW CAN ACTUALLY DO

Sort and draft emails · Research anything and send a plain-English summary · Write content from a single prompt · Run terminal commands · Manage files · Monitor inboxes and flag urgent items · Send daily briefings automatically · Run scheduled weekly business reviews - all from a WhatsApp message.

- 04 · SETUP GUIDE

How to set OpenClaw up safely.

Getting OpenClaw running is straightforward. Getting him running safely takes a bit more care. Do this in order.

STEP 1 - INSTALL

→ **Download OpenClaw**
npm install -g openclaw@latest - then run openclaw onboard. The wizard sets up your gateway, workspace, first channel and model. Get it from openclaw.ai only.

FULL STEP-BY-STEP SETUP GUIDE
For a comprehensive installation walkthrough with screenshots and troubleshooting, visit www.getopenclaw.co.nz - the complete beginner guide is available as a free download.

STEP 2 - ISOLATE

→ **Create separate dedicated accounts**
New Gmail specifically for OpenClaw. New Telegram or WhatsApp number. New Anthropic API key with its own billing. Never hand OpenClaw credentials connected to your primary accounts or banking.

→ **Separate API keys per service**
Each integration - Gmail, GitHub, Slack - gets its own scoped key with minimum permissions. If a key is ever compromised, you revoke one key, not your entire account.

STEP 3 - HARDEN

→ **Run openclaw doctor immediately**
Built-in security audit. Flags open gateway ports, missing auth tokens, insecure DM policies and exposed interfaces. Default config is NOT secure - this step is not optional.

→ **Set allowFrom in every channel**
Lock WhatsApp and Telegram to your number only - allowFrom: ["+64..."] in openclaw.json. Without this, anyone who knows your bot can message OpenClaw and it will act on their instructions.

→ **Fix the heartbeat before going live**
Set "every": "60m" minimum and point it at a cheap model like Haiku. A runaway heartbeat on Opus with no spend cap can burn $50 in a single day while you sleep.

- 05 · HOSTING

Where does OpenClaw actually live?

OpenClaw needs a machine to run on - ideally one that's always on, always connected, and separate from your personal computer. You have three options.

RECOMMENDED · ALWAYS-ON

Option 01 - VPS in the cloud Rent a remote server from Hetzner, DigitalOcean or Hostinger for $5–$15/month. Separate hardware, always running, completely isolated from your personal machine. Best for heartbeats, overnight tasks and anything long-running. This is the professional setup.

LOCAL · DEDICATED MACHINE

Dedicated PC or Mac Mini

Any spare computer you can leave on permanently works - a Mac Mini, old laptop, Raspberry Pi, or a dedicated Windows PC. Mac Mini is popular because it's quiet, low power and always-on. The key is that it's a separate device used only for OpenClaw.

NOT RECOMMENDED · HIGH RISK

Your personal laptop

Running OpenClaw on your personal machine is risky and not recommended unless you fully understand what you are doing. OpenClaw has access to your filesystem, terminal and accounts. A misconfiguration, a malicious skill, or a runaway task could interact with your personal files and data. If you proceed, use a completely isolated user profile, dedicated credentials for every service, and never connect it to your personal Gmail, banking, iCloud or any primary account. The separation between OpenClaw and your personal life must be total.

Wherever OpenClaw runs - workspace files, skills and memory are identical. Moving him to a better host is a git clone and a config update.

- 06 · USE CASES

What can you actually use OpenClaw for?

These are real things, for real people. You do not need to be technical to benefit from most of these.

→ **Your business keeps moving while you sleep**
Imagine waking up to find your emails have been sorted, urgent ones flagged, routine ones already drafted with a reply. Leads followed up. Invoices chased. Tasks logged. OpenClaw works through the night so your morning starts finished, not started.

→ **Never do the same research twice**
Ask OpenClaw to research anything - a competitor, a product, a market, a person you're meeting. He reads, summarises and sends it back in plain English on WhatsApp. No tabs, no rabbit holes, no hour lost.

→ **Content created while you have breakfast**
Give OpenClaw a topic, he drafts a post, caption, script or newsletter. You review and send. The whole process takes minutes instead of hours - and it sounds like you, because you wrote the SOUL.md.

→ **A personal assistant in your pocket**
Text OpenClaw like a person. "Remind me about X tomorrow." "Summarise this article." "What did we decide last week about pricing?" It knows your history, your preferences, your projects.

→ **Scheduled tasks that just happen**
Set a weekly briefing every Monday morning. A daily summary at 8am. A monthly business review. OpenClaw wakes up on schedule, pulls the information, writes the report and sends it - no prompt needed, no login required.

- 07 · AUDIENCE

Who is OpenClaw actually for?

More accessible than you might think - but honest about who gets the most from it.

THE RIGHT PERSON

→ **Builders and solopreneurs**
Running a business alone and need leverage. OpenClaw handles research, content and ops while you focus on growth. One agent does the work of three part-time assistants.

→ **Busy CEOs and founders**
Inbox triage, meeting prep, weekly briefings, competitor monitoring - delegated to an agent that never clocks off and never asks for a raise.

→ **AI consultants**
Build and configure OpenClaw agents for clients. The setup process is the product. Most businesses cannot do this themselves yet - that is your opportunity.

→ **Anyone who is time-poor**
Personal assistant for research, reminders, summarising content, drafting replies, tracking tasks - all from a WhatsApp message. You do not need to be technical to use the output.

HOW TO MAKE MONEY WITH IT

$ **Sell the setup as a service**
$500–$2,000 per build. Configure, harden and document a custom agent for a client. Most businesses need this and have no idea where to start.

$ **Build and sell custom skills**
A SKILL.md built for a specific niche - real estate, law, ecommerce - is a standalone product. Sell directly or publish on ClawHub.

$ **Use it to run your own business**
Every hour OpenClaw saves is recovered margin. That is infrastructure, not a side hustle.

- 08 · ARCHITECTURE

OpenClaw architecture - the full schematic.

All six workspace files live in one folder on disk: ~/.openclaw/workspace/. Every time you send a message, every single file is loaded, assembled into one system prompt, and sent to the LLM. Here is how it all connects.

THE WORKSPACE - ~/.OPENCLAW/WORKSPACE/

File	Role	Key point
SOUL.md	Personality + values	Loaded first every session. Defines tone, boundaries and non-negotiables.
AGENTS.md	Standard operating procedure	Startup sequence, routing rules, memory discipline.
id_entity.md	Mission + purpose	Goals, businesses and targets OpenClaw is optimising for.
MEMORY.md	Long-term curated memory	Auto-injected every prompt. Keep under 3,000 tokens.
USER.md	Who you are	Your name, timezone, preferences and business context.
HEARTBEAT.md	Autonomous task scheduler	Runs on a cron schedule. Fires without any prompt from you.

HOW THE FILES FLOW INTO OPENCLAW

Six files → assembled into one system prompt → sent to LLM → LLM reasons and issues tool calls → runtime executes tools → response delivered to your channel → session transcript saved to disk.

THE LLM LAYER

OpenClaw works with any major model - Claude (Anthropic), GPT (OpenAI), DeepSeek, Gemini, or a local model via Ollama. The workspace files and skills are model-agnostic. Swap the model in one line of config and everything else stays the same.

THE OUTPUT LAYER

Every message produces two possible outputs: a text reply delivered to WhatsApp, Telegram or Discord - and tool execution, which means real actions: shell commands, browser automation via Chrome DevTools Protocol, file reads and writes, API calls to external services.

- 09 · THE WORKSPACE FILES

The six workspace files - in depth.

Each file has one job. Content in the wrong file wastes tokens and creates conflicting instructions. Here is what belongs where.

IDENTITY + PERSONALITY

SOUL.md

Injected first on every prompt. Defines tone, values, communication style and non-negotiable behaviour rules. Every word costs tokens on every message forever - keep it under 500 lines. This is where you write: be direct, think in revenue not tasks, confirm before bulk actions, summarise before multi-step tasks.

STANDARD OPERATING PROCEDURE

AGENTS.md

The session startup sequence - what OpenClaw does before he responds to anything. Routing rules, scope boundaries, memory discipline rules. Critically: rules written here survive compaction. Rules said only in conversation do not. This is your most important file for operational consistency.

MISSION BRIEF

id_entity.md

Who OpenClaw is optimising for - with specific targets, specific businesses, specific goals. Different from SOUL.md deliberately: SOUL is how OpenClaw thinks, id_entity is what he is building toward. Update this as your goals evolve. This is where revenue targets, business names and content strategy live.

LONG-TERM CURATED MEMORY

MEMORY.md

Auto-injected into every session prompt. Curated decisions with dates, learned preferences, project status, open loops. Hard limit: under 3,000 tokens. At 10,000 tokens you are adding ~$0.15 to every conversation silently, every day. Monthly maintenance - promote key daily log entries here, delete the noise.

WHO YOU ARE

USER.md

Your name, timezone, operating system, communication preferences, background context. Tells OpenClaw who he is helping - not how to behave (SOUL) or what to build (id_entity). Practical details: "prefers short responses", "NZ timezone (NZST)", "primary language English".

AUTONOMOUS TASK SCHEDULER

HEARTBEAT.md

A checklist of tasks OpenClaw wakes up and runs on a cron schedule - every morning, every Monday, every hour - without any prompt from you. Format is simple: ## Daily · 8:00 AM followed by a task list. Cost warning: every heartbeat fires a full API call. Set to 60 minutes minimum and point it at Haiku, not Opus.

Skills - teaching OpenClaw how to use his tools

A Skill is a folder containing a SKILL.md file - YAML frontmatter plus a markdown runbook. Skills do not grant new permissions (those come from your tool policy in openclaw.json). They teach OpenClaw when to act and what steps to follow.

```
# Skill precedence order:

workspace/skills/ ← highest - your custom skills always win

~/.openclaw/skills/ ← shared across all agents on this machine

bundled skills ← lowest - shipped with the install
```

CLAWHUB SECURITY WARNING

ClawHub hosts 13,000+ community skills. In early 2026, 341 malicious skills were distributed via ClawHub (the ClawHavoc incident) - reverse shells and credential stealers. Always read the SKILL.md before installing. Treat every community skill like code from a stranger.

- 10 · MEMORY

Memory Optimisation.

THE GOLDEN RULE

If it's not written to a file, it doesn't exist after compaction. Instructions given only in chat will vanish. Rules belong in markdown files - not conversation.

What compaction is

→ **The context window has a size limit**

Every message adds to the conversation history. When it fills, OpenClaw automatically compacts - summarising older messages to make room. That summary is lossy. Details, nuance and any instructions given in chat are gone.

→ **Four memory layers - each fails differently**

Bootstrap files (SOUL.md, AGENTS.md, MEMORY.md) - permanent, reloaded from disk every session. Daily logs (memory/YYYY-MM-DD.md) - permanent but searched on demand, not auto-injected. Session transcript - semi-permanent, gets compacted. LLM context window - temporary, fixed size, overflows.

→ **Enable the pre-compaction memory flush**

OpenClaw has a built-in safety net that fires before compaction and tells OpenClaw to write important context to disk first. Set reserveTokensFloor: 40000 in openclaw.json. Default of 20,000 is too tight.

Write your rules to markdown files

```
# Add to AGENTS.md - survives every compaction


## Memory rules

- Search memory before acting on any multi-session task

- Write key decisions to memory/YYYY-MM-DD.md

- Promote important facts to MEMORY.md monthly

- Never keep critical rules only in conversation
```

→ **MEMORY.md - curated, under 3,000 tokens**

Decisions, preferences, project status. Auto-injected every session so every token costs money on every message forever. Keep it purposeful, keep it lean.

→ **Daily logs - memory/YYYY-MM-DD.md**

Written automatically each session. Searched on demand via hybrid BM25 + vector index. Not auto-injected - so months of history piles up at zero daily token cost.

THE ONE HABIT THAT FIXES EVERYTHING

End every important session with "remember that I decided X." OpenClaw writes it to the daily log. Once a month, promote the best entries into MEMORY.md and delete the noise.

- 11 · API COST

The default config is a financial disaster.

Most people set up OpenClaw, start chatting, wake up the next day and their API bill is $80. Here is exactly why that happens.

15K	40%	15×	50×
tokens consumed before you type a word - workspace files, tool schemas, system prompt overhead on every single message	of all API spend is conversation history re-sent with every message - grows every turn, compounds fast	cost of Claude Opus vs Haiku per token - same routine output, 15x the price without model routing	token multiplier when thinking/reasoning mode is accidentally enabled - one request, cost of fifty

→ **Context accumulation is the biggest cost**
Every message you send adds to the history. OpenClaw resends that entire history with every new message. A session at message 30 could easily be 40,000 tokens of context being resent every single turn.

→ **Workspace file bloat**
SOUL.md, AGENTS.md and MEMORY.md are injected on every prompt. At 10,000 tokens combined, that's 10,000 tokens of overhead before you've said a single word. Every character you write in those files is a permanent per-message tax.

→ **Wrong model for the task**
Claude Opus costs $15 per million input tokens. Claude Haiku costs $1. Using Opus to check email, run a heartbeat or answer a simple question is 15x the necessary cost for zero extra quality.

→ **Heartbeat misconfiguration**
A heartbeat set to every 5 minutes fires 288 full API calls per day - each carrying your complete session context. One user discovered their automated email check had burned $50 in a single day.

- 12 · COST FIXES

$15–$40 per month. Not $300. Here's exactly how.

Done right, OpenClaw should cost between $15 and $40 per month for daily use. People spending more than that are almost always hitting two or three of these mistakes simultaneously.

FIX 01 - AUDIT FIRST

01 **Run /context list before changing anything**

Shows every file injected into your prompt, its exact token count, and whether it's being truncated. SOUL.md + AGENTS.md + MEMORY.md combined must be under 3,000 tokens. Each character is a permanent per-message tax.

FIX 02 - MODEL ROUTING

02 **Set up model routing in openclaw.json**

Haiku ($1/M tokens) for heartbeats and routine checks. Sonnet for writing and analysis. Opus ($15/M tokens) only for deep strategy. Two lines of config. 80–90% cost reduction on everyday work.

```
// openclaw.json - model routing example

"heartbeat": { "model": "anthropic/claude-haiku-4-5" }

"default": { "model": "anthropic/claude-sonnet-4-6" }
```

FIX 03 - HEARTBEAT

03 **Set heartbeat to 60 minutes minimum, Haiku only**

5-minute heartbeat = 288 full API calls per day with your complete session context. Set "every": "60m" and point it at Haiku. Running it on Opus at 5-minute intervals is burning $50/day on email checks.

FIX 04 - SPEND CAP

04 **Set a hard monthly cap in your Anthropic dashboard - right now**

Go to dashboard.anthropic.com → billing → monthly spend limit. Set $20 while learning, raise it once you understand your patterns. A misconfigured heartbeat or runaway loop will silently drain your account overnight without this.

THE REALISTIC TARGET

Lean workspace files (under 3,000 tokens combined) + model routing (Haiku for routine, Opus for strategy) + hourly heartbeat = $15–$40/month for daily use. Every person spending $200+ is hitting at least two of these mistakes at once.

Follow to see how OpenClaw and Keira build a million-dollar business with AI.