FREE GUIDE FOR SUBSCRIBERS

The Local AI Playbook

Run OpenClaw 24/7 for $4/Month — No VPS, No Cloud Tax, No BS

Real hardware specs, real commands, real costs, and real gotchas from 2+ months of autonomous operation. Every command is copy-paste ready. Every cost is verified.

$4	47	$0.21	11
PER MONTH	TASKS OVERNIGHT	TOTAL COST	GOTCHAS FIXED

By GnawClaw

gnawclaw.com

Why Local AI?

Cloud AI is a recurring tax. Every API call, every hosted model, every managed VPS adds up to a bill that grows with your usage. The math never improves.

Running AI locally flips the equation. Pay once for hardware, pennies for electricity, and **$0.00 in API fees** for local inference. Your data never leaves your machine. Your agents run 24/7 without permission from a billing dashboard.

This isn't theory. We've been running OpenClaw autonomously for **2+ months**. Here's what that actually looks like:

WHAT'S ACTUALLY POSSIBLE — REAL RESULTS

🌙 **47 Tasks Completed Overnight**
Agents ran while we slept. Total cost: **$0.21**. That's not a typo.

👥 **15 Facebook Groups — Automated Presence**
Comments, replies, and posts across 15 communities. All running autonomously, 24/7.

🌐 **13,900+ Player Profiles — Fully Autonomous**
AI League website built and maintained by agents. Zero manual data entry.

⚙ **Daily Briefings, Trading Signals, Email Capture, Blog Posts**
All running on **$4–11/month** electricity. Revenue capture with $10 product pages and automated email sequences.

This guide gives you the exact commands, configs, and fixes to build the same thing.
Every cost is verified. Every gotcha is battle-tested.

SECTION 1

The Hardware Decision

OPTION A — BUDGET POWERHOUSE

$450 Mini PC

Beelink SER7 or equivalent

🔲 **CPU:** AMD Ryzen 7 7735HS

🧠 **RAM:** 32GB

💾 **Storage:** 500GB NVMe

🔌 **Power:** 25W idle / 65W load

⚡ Monthly electricity: **$3.93** ✅ Kill A Watt verified

Best for: 7B–14B models, email automation, research agents

`Llama 3.1 8B` `Qwen 2.5 7B` `Mistral 7B`

OPTION B — PRODUCTION BEAST

$1,599 RTX 4070 Desktop

12GB VRAM, 40-80+ tok/sec on 8B models

🖥 **VRAM:** 12GB GDDR6X

🏎 **Speed:** 40–80+ tok/sec (8B)

🔌 **Power:** 46W idle / 280W GPU load

🕐 **Usage:** 4h active/day assumed

⚡ Monthly electricity: **$10.55** ✅ Kill A Watt verified

Best for: 30B+ models, multi-agent systems, production workloads

`Llama 3.1 70B Q4` `Qwen 2.5 32B` `Mixtral 8x7B`

SECTION 1 — CONTINUED

Cost Comparison & Breakeven

COST ITEM	VPS (CLOUD)	MINI PC	RTX 4070
Upfront	$0	$450	$1,599
Monthly	$59.99/mo	**$3.93/mo**	$10.55/mo
API Fees	Usage-based	**$0.00**	**$0.00**
Data Privacy	Third-party	**100% local**	**100% local**

📈3-Year Total Cost of Ownership

VPS (CLOUD)	MINI PC ★	RTX 4070
$2,159	**$591**	**$1,979**
$59.99 × 36 months	$450 + ($3.93 × 36)	$1,599 + ($10.55 × 36)

✅ **Mini PC vs VPS:** Pays for itself in **14.5 months**. After that, you save $56/month.

✅ **RTX 4070 vs VPS:** Breaks even at **32.3 months**. You get 30B+ model capability no VPS matches.

🧾Real Monthly Cost — Full Automation Empire

SERVICE	MONTHLY COST	NOTES
Electricity (Mini PC)	$3.93	Kill A Watt verified
OpenRouter API	$8–15	~500 LLM calls/day
Resend email	$0	Free tier (3,000/mo)
Vercel hosting	$0	Free tier
Total	**$12–19**	**Full autonomous operation**

A machine you own, running 24/7, on your network, with your data. No cloud tax.

SECTION 2

OpenClaw Install — Copy-Paste Commands

🍎 macOS

```
npm install -g @openclaw/openclaw
openclaw gateway start
openclaw gateway status
```

🐧 Linux (Ubuntu/Debian)

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

🎚 3 Config Settings That Matter

1. sessionPersistence: true

Agents forget everything on restart without this. This is the single most common support issue.

> ⚠ *Real experience: We spent 2 days debugging why agents had no memory between sessions. The default is false. Turn it on before anything else.*

2. timeout: 300

Default 30s kills long responses. Local models are slower than cloud APIs. Minimum 300 seconds.

> ⚠ *Real experience: Claude responses cut off mid-sentence, Ollama responses never arrived. We run 600 for complex tasks.*

3. model: point to local Ollama

Connect to your local Ollama instance so every inference runs on your hardware. Zero API calls, zero cloud dependency.

📄openclaw.json

Copy-paste this into your config file:

```
{
  "model": "ollama/llama3.1:8b",
  "gateway": {
    "timeout": 300,
    "sessionPersistence": true,
    "storageDir": "~/.openclaw/sessions"
  }
}
```

SECTION 3

Ollama Setup

🍎 macOS

```
brew install ollama
```

🐧 Linux

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

⚠ **The Gotcha 90% Miss — Localhost Binding**

Ollama defaults to **127.0.0.1 (localhost only)**. This is the single most common OpenClaw setup failure we see. Fix "connection refused" errors:

```
OLLAMA_HOST=0.0.0.0 ollama serve
```

🧠Model Guide by VRAM

VRAM	RECOMMENDED MODELS	NOTES
8GB	Llama 3.1 8B, Qwen 2.5 7B, Mistral 7B	Sweet spot for most agents
12GB	Qwen 2.5 14B Q4, Llama 3.1 8B Q8	Higher quality quantization
24GB	Llama 3.1 70B Q4, Mixtral 8x7B	Production-grade reasoning
CPU only	Llama 3.1 8B Q4_K_M	Set `num_ctx 2048`

💡 **Context window tip:** Default `num_ctx 32768` destroys CPU inference speed. Set `num_ctx 2048` for task-focused agents.

Real experience: Qwen models on Mac Studio M4 with 36GB RAM took 20 minutes to respond. After setting num_ctx to 2048 — under 2 minutes.

SECTION 4

Connect to Messaging

🕐 **5 minutes total setup**

📨 Telegram

Fastest setup — recommended starting point

1. Open Telegram and message `@BotFather`. Run the `/newbot` command.
2. **Copy your bot token.** BotFather gives you a long string — that's your key.
3. Add the token to your config:

```
// Add to openclaw.json
{
  "channels": {
    "telegram": {
      "botToken": "YOUR_BOT_TOKEN_HERE"
    }
  }
}
```

✅ **Done.** Message your bot on Telegram — OpenClaw responds.

💬 Discord

For team or community-facing agents

1. Go to **discord.com/developers** and create a new application.
2. **Create a bot** under the Bot tab and copy the token.
3. Add the token to your config:

```
// Add to openclaw.json
{
  "channels": {
    "discord": {
      "botToken": "YOUR_DISCORD_BOT_TOKEN"
    }
  }
}
```

📚 **Multi-channel is built in.** We run Telegram, Discord, and WhatsApp simultaneously from one OpenClaw instance. The channel config handles routing automatically. Cost: **$0 extra**.

SECTION 5

First Automation — Runs at 3AM Without You

Your first agent should prove the concept with zero effort. This daily digest agent checks messages, prioritizes your day, and sends a summary to Telegram — **every morning at 7AM, while you sleep.**

🤖 Daily Digest Agent — System Prompt

> *"You are a daily briefing agent. Every morning at 7AM you:*
> *1. Check for important messages needing attention*
> *2. Summarize the top 3 priorities for today*
> *3. Send summary to Telegram*
> *Keep each item under 2 sentences. Be direct. No fluff."*

🕐 Schedule It with Cron

```
openclaw cron add --schedule "0 7 * * *" --message "Run daily briefing"
```

🪙 API cost: **$0.00**. Runs entirely on your local hardware while you sleep.

FROM REAL EXPERIENCE

What Model to Use for What

TASK	MODEL	COST	WHY
Email drafting/triage	`Llama 3.1 8B`	**$0 (local)**	Fast, free, handles routine tasks
Code writing	`GPT-5.4 Mini`	$0.0004/1K	Best cost/quality for code
Deep research	`Claude Sonnet 4.6`	$0.003/1K	Worth it for high-stakes decisions
Grading/classification	`DeepSeek Chat`	$0.0003/1K	Best cost/performance ratio
Local fast tasks	`Qwen 2.5 7B`	**$0 (local)**	30+ tok/sec on 8GB VRAM

⚠ **Hard-won lesson:** GPT-5.4 Nano stopped after 6–10 pages on a 146-page crawl, reporting "done." Smaller models treat partial progress as task completion. **Rule:** Use small models for single-pass tasks only. For multi-step loops, use Claude Sonnet or run in the main session directly.

SECTION 6

5 Prompts That Actually Work

1 Technical Support Responder

> *"Give the specific fix first, then explain why. Never say 'it depends' without giving the most likely answer immediately. Always include the exact command or config change."*

2 Content Research Agent

> *"Research [topic] and give me: 3 counterintuitive facts, the most common misconception, one actionable insight most people miss. Under 200 words. No vague claims."*

3 Email Triage Agent

> *"Read this email and tell me: priority (high/medium/low), what action is needed, suggested response under 50 words. If no action needed, say so."*

4 Cost Optimizer

> *"Can this task be done with a smaller model? What context do I actually need vs what I am sending unnecessarily? What is the minimum prompt that gets the same output?"*

5 Community Comment Agent

> *"When someone posts a technical problem: give the specific fix in sentence 1, the one gotcha they will hit next in sentence 2. 2 sentences maximum. Sound like a friend texting the answer."*

SECTION 7

Hard-Won Gotchas

What will break and exactly how to fix it — from 2+ months of production

1 Strict JSON Schema — One Wrong Key Kills the Gateway

ISSUE Gateway entered a **325-restart boot-fail loop**. Telegram went dark for 40 minutes.

CAUSE Added an unsupported field ("description") to agents config. OpenClaw uses `additionalProperties: false`.

FIX **ALWAYS** use the `config.patch` tool. NEVER write openclaw.json directly. Run `openclaw config.schema.lookup <path>` before any config change.

2 Cron Jobs Go in the Cron Tool — Not openclaw.json

ISSUE Killed the gateway **twice** by writing a "crons" key into openclaw.json.

CAUSE Cron jobs belong in `~/.openclaw/cron/jobs.json` via the cron tool, not in the main config.

FIX **ONLY** use `openclaw cron add` to create crons. Never touch openclaw.json for scheduling.

3 Facebook Headless Browser Detection

ISSUE All Facebook automation stopped when Facebook flagged the headless browser session.

CAUSE GPU process crash + Facebook bot detection on plain Playwright.

FIX Use **patchright** (not plain Playwright). Add `--disable-blink-features=AutomationControlled`. Refresh browser session regularly.

4 The Duplicate Comment Bug — Posted 12x to One Person

ISSUE Reply engine posted the same reply **12+ times**, every 30 min, midnight to noon.

CAUSE Dedup check queried the wrong table (`fb_replies` instead of `fb_auto_replies`).

FIX Preflight check before every action — if already in `action_log` as "done", stop. Use **idempotency keys** on every automated action.

5 Subagent Model Selection Matters Enormously

ISSUE GPT-5.4 Nano stopped after 6–10 pages on a 146-page crawl, reporting "done."

CAUSE Smaller models treat partial progress as task completion on multi-step iterative work.

FIX **Rule:** Small models = single-pass tasks only. For loops/crawls, use Claude Sonnet or run in the main session.

6 Session Orphans Will Eat Your Disk

ISSUE Session dir grew to **247MB / 8,459 files** in 6 weeks.

FIX Monthly cron to prune: `cd ~/.openclaw/agents/main/sessions && ls -t | tail -n +201 | xargs rm -rf`

SECTION 8

Your First 30 Days

Day 1–3: Foundation

Install OpenClaw, connect Telegram, run your first agent. Prove the stack works end-to-end before optimizing anything. Don't skip the 3 config settings on page 5 — especially **sessionPersistence**.

Day 4–7: Go Local

Set up Ollama with one local model. Connect it to OpenClaw. Remember **OLLAMA_HOST=0.0.0.0** or you'll hit "connection refused." Set **num_ctx: 2048** for task agents. Cut your cloud dependency to zero.

Week 2: Automate

Build the daily digest agent. Schedule it with cron (use **openclaw cron add**, never write crons into openclaw.json). Let it run 7 days unattended. Watch it work while you don't.

Week 3: Expand

Add a second automation — email triage or content research. Use the model selection guide on page 8. Match the right model to the right task.

Week 4: Optimize

Review what works, cut what doesn't, double down on the winner. Set up the session orphan cleanup cron. You now have a local AI stack running 24/7 for under $20/month.

The Rule: One new automation per week. Compounding beats sprinting.

YOU MADE IT

Now Own Your AI Stack

You have the hardware specs, the commands, the configs, the gotchas, and the automations. Everything runs on your machine, on your terms, for pennies a day. No cloud tax. No throttling. No monthly invoice that never stops.

- ✅ **Hardware chosen** — Real costs verified with Kill A Watt
- ✅ **OpenClaw installed** — Gateway running, config battle-tested
- ✅ **Ollama connected** — Local models serving at 30+ tok/sec
- ✅ **Agents automated** — Running 24/7 for under $20/month
- ✅ **11 gotchas dodged** — 2 months of production pain, avoided
- ✅ **Model strategy set** — Right model for every task, verified

Want More Playbooks Like This?

Advanced automations, multi-agent workflows, and production deployment guides — free for subscribers.

Visit gnawclaw.com →