🦞 COMPLETE BEGINNER'S GUIDE

The OpenClaw Beginner's Guide

Your AI Assistant, From Install to Production

Everything you need to set up, configure, and get productive with OpenClaw — your self-hosted AI gateway that connects WhatsApp, Telegram, Discord, and more to a personal AI agent running on your own hardware. Real commands. Real configs. Ready to copy and paste.

Amir Brooks

amirbrooks.com.au · lavonglobal

2026 Edition

CONTENTS

What's Inside

01

SECTION ONE

What Is OpenClaw?

Meet your self-hosted AI gateway — the bridge between your favourite chat apps and an intelligent, always-on personal assistant running on your own hardware.

- Core Concept & Architecture
- Key Features
- The Two-Phone Setup
- 5-Minute Quick Start

What Is OpenClaw?

OpenClaw is a **self-hosted gateway** that connects WhatsApp, Telegram, Discord, iMessage, and more to AI agents. Think of it as the infrastructure layer that turns a chat message into an AI-powered action — running on your own machine, on your own terms.

> 💡 **The Core Idea**
> You get a dedicated phone number (or bot) that behaves like an always-on AI assistant. Message it on WhatsApp or Telegram and it runs commands, reads files, searches the web, and responds — all from your own hardware.

Why OpenClaw?

When you put an AI agent in a position to receive your messages, it can:

- Run commands on your machine (depending on your tool policy)
- Read and write files in your workspace
- Send messages back out via WhatsApp, Telegram, Discord, or Mattermost
- Search the web, browse pages, and summarise documents
- Manage calendars, emails, notes, and reminders via skills
- Spawn sub-agents to run parallel tasks in the background

🔒

Self-Hosted

Your data stays on your hardware. No vendor cloud storing your conversations or workspace files.

🌐

Multi-Channel

WhatsApp, Telegram, Discord, iMessage, Signal, Mattermost, Matrix, and more — one agent, many surfaces.

🤖

Any Model

35+ providers including Anthropic, OpenAI, Google, Ollama (local), and any OpenAI-compatible endpoint.

⚡

Always On

Runs as a daemon (macOS LaunchAgent or Linux systemd) — survives reboots, works while you sleep.

> 💚 **Pro Tip: One-Line Definition**
> OpenClaw = chat interface + AI agent + your tools, all connected and running on hardware you control.

Key Features

Channels

- WhatsApp, Telegram, Discord, iMessage (built-in)
- Mattermost, Matrix, MS Teams, Nostr, Signal (plugins)
- Group chat support with mention-based activation
- DM safety with allowlists and pairing codes

Agent Runtime

- Embedded agent with tool streaming
- Multi-agent routing with isolated sessions
- Sessions per workspace or per sender
- Streaming and chunking for long responses

Auth & Providers

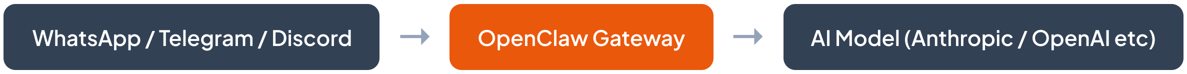
- 35+ model providers supported
- Anthropic, OpenAI, Google, xAI, and more
- Subscription auth via OAuth (e.g. Codex)
- Self-hosted support: Ollama, vLLM, SGLang

Media Support

- Images, audio, video, documents in/out
- Voice note transcription (Whisper)
- Text-to-speech (multiple providers)
- Canvas UI for node displays

Architecture Overview

OpenClaw sits between your chat channels and your AI model:


WhatsApp / Telegram / Discord → OpenClaw Gateway → AI Model (Anthropic / OpenAI etc)

↕ Tools: files, exec, browser, web, skills

Session Model

Each conversation gets its own isolated session. Direct chats collapse into a shared `main` session. Groups are isolated by default.

SESSION KEY FORMAT

```
# DM sessions
agent:main:telegram:direct:123456789
agent:main:discord:channel:987654321

# Group sessions
agent:main:telegram:group:-1001234567890

# Sub-agent sessions
agent:main:subagent:<uuid>

# Reset a session at any time
/new   # or /reset
```

🚀 **Pro Tip: Session Scoping**

Set `session.dmScope: "per-channel-peer"` to prevent cross-user context leakage when multiple people can DM your bot.

The Two-Phone Setup

The recommended way to use OpenClaw with WhatsApp is with a **dedicated second phone number** for your AI assistant. This keeps your personal messages separate from your agent's inbox.

> ⚠ **Important: Don't Use Your Personal Number**
>
> If you link your personal WhatsApp to OpenClaw, *every message sent to you* becomes agent input. That's rarely what you want. Use a dedicated SIM or eSIM for the assistant.

TWO-PHONE ARCHITECTURE

```
Your Phone (personal)         Second Phone (assistant)      Your Mac
+1-555-YOU (WhatsApp)  →→→  +1-555-ASSIST (WhatsApp)  →→→  OpenClaw Gateway
                                                              ↓
                                                         AI Agent (Claude, GPT, etc)
                                                              ↓
                                                         Response sent back to you
```

Minimal Config for Personal Assistant

~/.OPENCLAW/OPENCLAW.JSON

```
{
  logging: { level: "info" },
  agent: {
    model: "anthropic/claude-opus-4-6",
    workspace: "~/.openclaw/workspace",
    thinkingDefault: "high",
    timeoutSeconds: 1800,
    heartbeat: { every: "0m" }, // Start with 0; enable later
  },
  channels: {
    whatsapp: {
      allowFrom: ["+15555550123"],
      groups: { "*": { requireMention: true } },
    },
  },
  session: {
    scope: "per-sender",
    resetTriggers: ["/new", "/reset"],
    reset: {
      mode: "daily",
      atHour: 4,
      idleMinutes: 10080,
    },
  },
}
```

Session Commands

- `/new` or `/reset` — starts a fresh session for that chat
- `/compact [instructions]` — compacts session context to save tokens

5-Minute Quick Start

1. **Pair WhatsApp Web**
   Run `openclaw channels login` — scan the QR code with your assistant phone.
2. **Start the Gateway**
   Run `openclaw gateway --port 18789` and leave it running.
3. **Add Minimal Config**
   Put your config in `~/.openclaw/openclaw.json` with your `allowFrom` number.
4. **Send a Test Message**
   Message the assistant number from your phone: "Hello!" — the agent should reply.

Heartbeats (Proactive Mode)

Heartbeats let the agent take initiative on a schedule. Every 30 minutes (default), OpenClaw runs the agent with a built-in prompt asking it to check `HEARTBEAT.md` for tasks.

HEARTBEAT CONFIG

```
{
  agent: {
    heartbeat: { every: "30m" }, // "0m" to disable
  },
}
```

Pro Tip: Start with Heartbeats Off

Set `heartbeat.every: "0m"` while you're getting started. Enable it once your workspace files are set up and you know what you want the agent to do proactively. If the agent replies HEARTBEAT_OK, OpenClaw suppresses delivery automatically.

SECTION 1 QUICK REFERENCE

`openclaw status`	Local status (creds, sessions, queued events)	`openclaw status --all`	Full diagnosis (pasteable to support)
`openclaw status --deep`	Adds gateway health probes	`openclaw health --json`	Gateway health snapshot (JSON)
`/new`	Start a fresh session	`/compact`	Compact session context

02

SECTION TWO

Installation

Install OpenClaw on any platform — macOS, Linux, Windows (WSL2), or a Raspberry Pi. Real commands, real options, nothing skipped.

- Installer Script
- macOS App
- Linux & WSL2
- Raspberry Pi

Installing OpenClaw

The fastest way to install OpenClaw is the official installer script. It handles Node.js, the CLI, and daemon setup automatically.

Recommended: Installer Script

MACOS / LINUX / WSL2

```
# Standard install with onboarding wizard
curl -fsSL https://openclaw.ai/install.sh | bash

# Skip onboarding (manual setup)
curl -fsSL https://openclaw.ai/install.sh | bash -s -- --no-onboard
```

WINDOWS POWERSHELL

```
# Standard install
iwr -useb https://openclaw.ai/install.ps1 | iex

# Skip onboarding
& ([scriptblock]::Create((iwr -useb https://openclaw.ai/install.ps1))) -NoOnboard
```

System Requirements

Requirement	Details
Node.js	Node 24 recommended · Node 22.16+ minimum · Installer handles this
OS	macOS, Linux, Windows (WSL2 preferred), Raspberry Pi OS
API Key	From any supported provider (Anthropic, OpenAI, Google, etc.)
RAM	512MB minimum · 2GB+ recommended for smooth operation

Alternative Install Methods

NPM / PNPM / GITHUB

```
# npm global
npm install -g openclaw@latest

# pnpm global
pnpm add -g openclaw@latest && pnpm approve-builds -g

# From GitHub main branch
npm install -g github:openclaw/openclaw#main

# Build from source
git clone https://github.com/openclaw/openclaw.git
cd openclaw && pnpm install && pnpm ui:build && pnpm build
pnpm link --global
```

Installer Flags

Flag	Description
`--install-method npm|git`	Choose install method (default: npm)
`--version <version>`	npm version, dist-tag, or spec
`--no-onboard`	Skip onboarding wizard
`--dry-run`	Print actions without applying
`--verbose`	Enable debug output

Onboarding Wizard

After installing, run the onboarding wizard. It walks you through everything — model setup, workspace, gateway, channels, and daemon installation.

RUN ONBOARDING

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

Onboarding Steps

1. **Model/Auth**
   Choose your AI provider, set your API key, and pick a default model (e.g. `anthropic/claude-opus-4-6`).
2. **Workspace**
   Set the location for agent files. Default: `~/.openclaw/workspace`
3. **Gateway**
   Set port (default `18789`), bind address, auth mode, and Tailscale exposure if needed.
4. **Channels**
   Choose: WhatsApp, Telegram, Discord, Google Chat, Mattermost, Signal, BlueBubbles, or iMessage.
5. **Daemon**
   Installs LaunchAgent (macOS) or systemd user unit (Linux/WSL2) for auto-start on reboot.
6. **Health Check**
   Starts the Gateway and verifies it's running. Then installs recommended skills.

> 💚 **Pro Tip: QuickStart vs Advanced Mode**
> **QuickStart** uses sensible defaults: local gateway, loopback bind, port 18789, token auth, tools profile "coding".
> **Advanced** gives you full control over every step. Start with QuickStart unless you have specific requirements.

Verify & Open Dashboard

VERIFICATION COMMANDS

```
openclaw --version       # confirm CLI is installed
openclaw doctor           # check for config issues
openclaw gateway status   # verify Gateway is running
openclaw dashboard        # open web UI in browser
```

macOS Setup

The macOS app is a **menu-bar companion** that shows native notifications, owns TCC permission prompts, and manages the Gateway. The CLI is separate.

What the App Does

- Shows native notifications and status in menu bar
- Owns TCC prompts (Notifications, Accessibility, Screen Recording, Microphone)
- Runs or connects to the local/remote Gateway
- Exposes macOS-only tools (Canvas, Camera, `system.run`)

Launchd Control

BASH

```bash
# Restart gateway
launchctl kickstart -k \
  gui/$UID/ai.openclaw.gateway

# Stop gateway
launchctl bootout \
  gui/$UID/ai.openclaw.gateway
```

⚠ **Avoid iCloud Sync on State Dir**

Do NOT put the OpenClaw state directory in iCloud or any cloud-synced folder. Keep it at: `OPENCLAW_STATE_DIR=~/.openclaw`

Windows Setup (WSL2 Recommended)

WSL2 is more stable than native Windows for OpenClaw. The core CLI works natively on Windows, but WSL2 is the preferred path.

WINDOWS → WSL2 → OPENCLAW

```
# Step 1: Install WSL2 (PowerShell as Admin)
wsl --install
# Or pick a specific distro:
wsl --install -d Ubuntu-24.04

# Step 2: Enable systemd (inside WSL)
sudo tee /etc/wsl.conf >/dev/null <<'EOF'
[boot]
systemd=true
EOF

# Step 3: Restart WSL (from PowerShell)
wsl --shutdown

# Step 4: Install OpenClaw inside WSL
curl -fsSL https://openclaw.ai/install.sh | bash
```

Auto-Start Before Windows Login

AUTO-START SETUP

```
# Inside WSL:
sudo loginctl enable-linger "$(whoami)"
openclaw gateway install

# PowerShell as Administrator:
schtasks /create /tn "WSL Boot" /tr "wsl.exe -d Ubuntu --exec /bin/true" /sc onstart /ru SYSTEM
```

Linux / VPS Setup

Running OpenClaw on a VPS gives you a persistent, always-on gateway without any hardware to manage. Setup takes about 5 minutes.

⚠ **Use Node, Not Bun**

Node.js is the recommended runtime on Linux. **Bun is NOT recommended for the Gateway.**

VPS Quick Path

LINUX VPS SETUP (UBUNTU/DEBIAN)

```
# 1. Install Node 24
curl -fsSL https://deb.nodesource.com/setup_24.x | sudo -E bash -
sudo apt install -y nodejs

# 2. Install OpenClaw
npm i -g openclaw@latest

# 3. Run Onboarding
openclaw onboard --install-daemon

# 4. SSH tunnel from your laptop to the VPS
ssh -N -L 18789:127.0.0.1:18789 <user>@<host>

# 5. Open dashboard at http://127.0.0.1:18789/
```

systemd User Service

~/.CONFIG/SYSTEMD/USER/OPENCLAW-GATEWAY.SERVICE

```
[Unit]
Description=OpenClaw Gateway
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/usr/local/bin/openclaw gateway --port 18789
Restart=always
RestartSec=5

[Install]
WantedBy=default.target
```

ENABLE & START

```
systemctl --user enable --now openclaw-gateway.service
```

Environment Variables

Variable	Purpose
`OPENCLAW_HOME`	Home directory for internal path resolution
`OPENCLAW_STATE_DIR`	Override the state directory (default: `~/.openclaw`)
`OPENCLAW_CONFIG_PATH`	Override the config file path

💚 **Pro Tip: VPS vs Pi Cost**

DigitalOcean costs ~$6/mo, Hetzner ~$4/mo. A Raspberry Pi 4 (4GB) costs ~$55 one-time. Break-even is ~6–12 months, then the Pi is free forever.

Raspberry Pi Setup

A Pi gives you a persistent, always-on gateway for **~$35–80 one-time** cost — no monthly fees. Ideal for home labs and personal setups.

Hardware Guide

Model	RAM	Status	Notes
Pi 5	4GB/8GB	✅ Best	Fastest, recommended
Pi 4	4GB	✅ Good	Sweet spot
Pi 4	2GB	✅ OK	Works, add swap
Pi 4	1GB	⚠️ Tight	Possible with swap + minimal config
Pi Zero 2 W	512MB	❌ No	Not recommended

Full Setup Script

RASPBERRY PI OS (ARM64)

```
# 1. Update system
sudo apt update && sudo apt upgrade -y
sudo apt install -y git curl build-essential

# 2. Set timezone
sudo timedatectl set-timezone America/Chicago

# 3. Install Node.js 24 (ARM64)
curl -fsSL https://deb.nodesource.com/setup_24.x | sudo -E bash -
sudo apt install -y nodejs
node --version  # Should show v24.x.x

# 4. Add 2GB Swap (important for 2GB or less RAM)
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile && sudo mkswap /swapfile && sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
echo 'vm.swappiness=10' | sudo tee -a /etc/sysctl.conf && sudo sysctl -p

# 5. Install OpenClaw
curl -fsSL https://openclaw.ai/install.sh | bash

# 6. Run onboarding
openclaw onboard --install-daemon
```

Performance Optimisation

~/.BASHRC ADDITIONS

```
grep -q 'NODE_COMPILE_CACHE' ~/.bashrc || cat >> ~/.bashrc <<'EOF'
export NODE_COMPILE_CACHE=/var/tmp/openclaw-compile-cache
mkdir -p /var/tmp/openclaw-compile-cache
export OPENCLAW_NO_RESPAWN=1
EOF
source ~/.bashrc
```

🚫 **Don't Run Local LLMs on Pi**

Even small local models are too slow on Pi hardware. Use cloud APIs (Anthropic Claude Sonnet or OpenAI GPT-4o-mini are great choices).

Node.js Setup

Required: Node 22.16+ minimum, Node 24 recommended. The installer handles this automatically.

INSTALL NODE.JS BY PLATFORM

```
# Check version
node -v

# macOS (Homebrew)
brew install node

# Ubuntu/Debian
curl -fsSL https://deb.nodesource.com/setup_24.x | sudo -E bash -
sudo apt-get install -y nodejs

# Windows (winget)
winget install OpenJS.NodeJS.LTS
```

Common Fixes

openclaw: command not found

FIX PATH

```
npm prefix -g          # Find global npm prefix
echo "$PATH"            # Check if it's on PATH

# Add to ~/.zshrc or ~/.bashrc:
export PATH="$(npm prefix -g)/bin:$PATH"
```

Permission Errors on Linux

FIX NPM PERMISSIONS

```
mkdir -p "$HOME/.npm-global"
npm config set prefix "$HOME/.npm-global"
export PATH="$HOME/.npm-global/bin:$PATH"
```

💚 **Pro Tip: Pi Model Selection**

For Raspberry Pi, use `anthropic/claude-sonnet-4-20250514` as primary with `openai/gpt-4o-mini` as fallback. Cost-effective and fast enough for most tasks.

⚡ SECTION 2 QUICK REFERENCE

`openclaw onboard --install-daemon`	Full setup wizard + daemon install	`openclaw --version`	Confirm CLI is installed
`openclaw doctor`	Check for config issues	`openclaw gateway status`	Verify Gateway is running
`openclaw dashboard`	Open web UI in browser	`openclaw update`	Update to latest version

🦞

Want the Full Guide?

This free preview covers What is OpenClaw and Installation. The complete guide includes Workspace Files, Channels, Security, Tools & Skills, Multi-Agent setups, and Business Workflows with copy-paste configs.

Get the Complete Guide

amirbrooks.com.au/openclaw-guide