

OpenClaw

The Complete Guide

A comprehensive reference covering architecture, deployment, channel integration, the Skills system, model configuration, security, and cost management.

OpenClaw Orange Paper — From Zero to Mastery

Sources: OpenClaw Official Docs • GitHub Repository • Community Research

Document Version: v1.4.0

Covers OpenClaw: v2026.3.13

Published: March 2026

Topics: Architecture • Deployment • Channel Integration • Skills System • Model Configuration • Security & Cost • Ecosystem

Huashu

YouTube: [Huashu](#) • X/Twitter: [@AlchainHust](#) • Bilibili: [Huashu](#)

This document was compiled with the assistance of Claude Code. Accuracy and timeliness are provided for reference only.
For corrections or suggestions, reach out via the channels above.

Companion video: [Bilibili "OpenClaw from Zero to One"](#) • Latest updates: [Feishu Doc](#)

Table of Contents

OpenClaw: The Complete Guide

Part 1: Meet OpenClaw

- 01 What is OpenClaw
 - 02 History
 - 03 The Creator
 - 04 Why So Popular
-

Part 2: Architecture

- 05 Architecture Overview
 - 06 Memory System
 - 07 Agent Workspace
 - 08 Sessions & Authentication
 - 09 Design Philosophy
-

Part 3: Deployment

- 10 Deployment Overview
 - 11 Local Installation
 - 12 Docker Deployment
 - 13 Cloud Deployment in China
 - 14 Initial Configuration
-

Part 4: Channel Integration

- 15 Channel Overview
 - 16 International Platforms
 - 17 Chinese Platforms
 - 18 Remote Access
-

Part 5: Skills System

- 19** How Skills Work
- 20** ClawHub & Skill Ecosystem
- 21** Top Skills
- 22** Create Your Own Skill
- 23** Skill Security

Part 6: Model Configuration

- 24** Provider Overview
- 25** International Models
- 26** Chinese Models
- 27** Local Models & Recommendations

Part 7: Security & Cost

- 28** Security Model
- 29** Security Incidents
- 30** Cost Control

Part 8: Ecosystem & Community

- 31** Lobster Culture
- 32** Alternatives
- 33** OpenClaw vs Claude Code
- 34** China Ecosystem
- 35** Claw Products in China

Appendix

- A** Frequently Asked Questions
- B** Command Cheat Sheet

01 What is OpenClaw

An open-source, self-hosted AI Agent system that transforms AI from a "chat tool" into a "digital employee capable of autonomous task execution."

If you've used ChatGPT, you know it's essentially a Q&A system: you ask, it answers. OpenClaw is different. It's an AI Agent platform that connects to 20+ messaging channels (WhatsApp, Telegram, Feishu, DingTalk, Discord, etc.), proactively executes tasks, manages your schedule, handles emails, operates browsers, and calls various tools.

In other words, ChatGPT is a "consultant," while OpenClaw is an "employee."

Key Differences from ChatGPT

Dimension	ChatGPT	OpenClaw
Interaction	You ask, it answers	Autonomous task execution
Environment	Web/App	Self-hosted server, connected to 20+ messaging platforms
Extensibility	GPTs Store	ClawHub skill marketplace (13,729 Skills)
Data Control	Data stays with OpenAI	Fully local, you own all data
Model Choice	GPT series only	Claude / GPT / DeepSeek / Gemini / Ollama local models
Open Source	No	MIT License, fully open source

Key Metrics As of March 24, 2026

Metric	Data
GitHub Stars	330,000+ (fastest-growing in GitHub history, surpassing both React and Linux)
Forks	64,300+
Contributors	1,075+
ClawHub Skills	13,700+
Built-in Skills	55
Supported Channels	20+ (WhatsApp / Telegram / Discord / Slack / Feishu / DingTalk / Browser, etc.)
Latest Version	v2026.3.13 (released 2026-03-14)

OpenClaw in one sentence: It's an open-source "personal AI operating system" that runs on your own server. You interact with it through any instant messaging tool, and it handles various tasks in your life and work. The mascot is a lobster, and the Chinese community calls using OpenClaw "raising lobsters" (yang xia).

02 History

From one person's weekend project to becoming GitHub's #1 in under five months.

Date	Event
November 2025	ClawdBot is born. Austrian developer Peter Steinberger releases it as a weekend project. The name pays homage to Anthropic's Claude (Claw = claw), with a lobster chosen as the mascot.
Mid-January 2026	Explosive growth. 60,000 Stars in 72 hours, with a single-day peak of 9,000 Stars.
January 27, 2026	Anthropic trademark warning. Forced to rename to Moltbot (Molt = lobster molting) due to similarity with Claude.
January 30, 2026	Renamed to OpenClaw. Emphasizing open-source identity while keeping the lobster theme.
Early February 2026	Security crisis. CVE-2026-25253 RCE vulnerability discovered (CVSS 8.8/10), with 50,000+ of 135,000 exposed instances directly attackable. The ClawHavoc supply chain attack erupts simultaneously, with ~12% of ClawHub Skills confirmed malicious.
Early February 2026	Google account ban wave. Google mass-bans OpenClaw user accounts, shaking the community.
February 14, 2026	Creator joins OpenAI. Peter Steinberger announces joining OpenAI. The project is transferred to an open-source foundation. OpenAI sponsors but the project remains independent.
March 3, 2026	Tops GitHub. v2026.3.2 released, Stars surpass 250K, officially overtaking React as the #1 software project on GitHub.
March 7-8, 2026	v2026.3.7 "Epic Update." 89 commits. Context Engine plugin architecture, native GPT-5.4 support, distributed channel binding. Stars reach 278,932. Shenzhen Longgang AI Bureau publishes OpenClaw support policy draft.
March 9, 2026	v2026.3.8 Security Hardening. New ACP authentication, local backup tools, 12+ security patches. China's MIIT and CNCERT issue OpenClaw security risk advisory. Stars surpass 280,000.
March 12-13, 2026	v2026.3.11 + v2026.3.12 consecutive releases. 3.11 fixes WebSocket cross-site hijacking vulnerability, improves local Ollama integration. 3.12 launches Dashboard v2, <code>/fast</code> quick mode, local model plugin architecture (Ollama/vLLM/SGLang), and switches device pairing to Ephemeral Tokens.
March 14, 2026	v2026.3.13 Browser automation upgrade. Supports Chrome DevTools remote attachment to logged-in browser sessions, releases Browser Relay Chrome extension. Stars continue growing, surpassing Linux to become the most-starred project in GitHub history.
March 16, 2026	Zhipu releases GLM-5-Turbo. The first foundation model ever specifically optimized for OpenClaw scenarios from the training stage. Focused on tool calling, long-chain execution, persistent tasks. 128K output / 200K context, supports MCP protocol. Currently in experimental closed-source release.

核心建议

OpenClaw successively surpassed React and Linux to become the fastest-growing open-source project in GitHub history. React took over 10 years to reach 230K Stars, and Linux took even longer. OpenClaw achieved this in under 5 months.

03 The Creator

Peter Steinberger: From a weekend project to the world's hottest open-source project, then to OpenAI.

From One Person to a Community

Peter Steinberger is an Austrian developer well-known in the iOS and macOS development community. On a weekend in November 2025, he built a small AI assistant tool that connects to instant messaging platforms, naming it ClawdBot.

He probably didn't expect that this weekend project would become GitHub's fastest-growing open-source project within two months. By March 2026, he had personally made 11,684 commits, with over 1,075 contributors.

Joining OpenAI

On February 14, 2026, Peter announced he was joining OpenAI. Sam Altman personally tweeted to welcome him, calling him a "genius."

The decision sparked wide community discussion, but Peter took several steps to ease concerns:

- OpenClaw transitioned to open-source foundation governance, maintaining project independence
- OpenAI became one of several sponsors (alongside Vercel, Blacksmith, and Convex), without controlling the project direction
- OpenAI committed to letting him continue contributing to OpenClaw development

In Peter's own words: "I'm a builder at heart... What I want is to change the world, not build a large company."

The Name Story

ClawdBot was named as a tribute to Anthropic's Claude (Claw = claw), which is why a lobster was chosen as the mascot. Anthropic's trademark warning forced a rename to Moltbot (Molt = lobster molting), and three days later it became OpenClaw, emphasizing open source. Despite two name changes, the lobster image was always preserved and has become the cultural symbol of the entire community.

04 Why So Popular

From 0 to 330K Stars in under 5 months. OpenClaw's viral success isn't just a technology story.

Growth Data

Date	Stars	Notes
November 2025	0	Project created
Mid-January 2026	60,000+	72-hour explosive growth
Mid-February 2026	145,000+	Peter joins OpenAI
March 1, 2026	241,000+	Approaching React
March 3, 2026	250,000+	Surpasses React, #1 on GitHub
March 8, 2026	278,932	v2026.3.7 released
March 9, 2026	280,000+	v2026.3.8 released
March 24, 2026	330,000+	80K growth in 3 weeks, surpasses Linux as GitHub's all-time #1

On one single day, the project gained 9,000 Stars, meaning on average, one developer hit Star every 10 seconds. After surpassing React, OpenClaw continued growing, then surpassed Linux to become the fastest-growing open-source project in GitHub history.

The "Lobster Raising" Cultural Phenomenon

Because the mascot is a lobster, the Chinese community calls running OpenClaw "raising lobsters" (yang xia), and users call themselves "lobster raisers." "Have you raised your lobster yet?" became a greeting in AI circles. This fun cultural label lowered the barrier to sharing, giving a technical project the qualities of social currency.

On March 6, 2026, nearly a thousand people lined up at Tencent Cloud headquarters in Shenzhen to experience OpenClaw installation. On March 8, Shenzhen's Longgang District AI Bureau published a draft policy on OpenClaw usage support measures. It's rare for an open-source project to attract government policy attention in China.

Moltbook: A Social Network for AI Agents

The OpenClaw ecosystem spawned a social platform called Moltbook, exclusively for AI Agents. Data as of late February 2026:

Metric	Data
Registered AI Agents	32,912
Sub-communities	2,364
Posts	3,130
Comments	22,046

Thousands of OpenClaw instances post, comment, and discuss philosophy on Moltbook. This may be the first large-scale experimental ground for AI Agents transitioning from "tools" to "social beings."

Popular Use Cases

Money-Making

- AI-driven prediction market trading on Polymarket, with OpenClaw instances earning tens of thousands of dollars monthly
- ClawWork project: "OpenClaw as your AI Coworker, earning \$15K in 11 hours"

Life Assistant

- Taking over email, calendar, and message management
- Browsing web, filling forms, data extraction
- File read/write, shell command execution

Social / Virtual Pet

- Setting up Agent names and personalities on Moltbook, observing their "social behavior"
- Inter-Agent interactions forming a "cyber-pet" culture

Enterprise Deployment

- Massive adoption of Feishu, DingTalk, WeCom, and QQ integrations in China
- Dedicated openclaw-china plugin suite supporting three-step Docker deployment

注意

Behind OpenClaw's popularity lurk shadows: over 50% of ClawHub's 13,729 Skills have been flagged as spam/duplicates/low-quality, and 396 were marked as malicious. Horror stories of waking up to \$1,100 API bills appear frequently in the community. The CVE-2026-25253 RCE vulnerability put 135,000 exposed instances at risk. "Raising lobsters" is exciting, but security and cost control are things you must take seriously.

05 Architecture Overview

OpenClaw employs a Gateway-Node-Channel three-layer architecture, using WebSocket as the communication bus to decouple the control plane, device execution, and messaging channels.

Three-Layer Architecture



Layer	Responsibility	Key Details
Gateway	Central control plane, maintains WebSocket service, manages Sessions, dispatches Agents	Defaults to <code>ws://127.0.0.1:18789</code> , one instance per host
Node	Device-side execution node, responsible for local operations	Camera, screen recording, <code>system.run</code> (system commands), etc.
Channel	Messaging channel integration layer, connects 20+ IM platforms	WhatsApp, Telegram, Discord, Slack, Feishu, DingTalk, etc.

Loopback-First Design

Gateway defaults to binding only `localhost` (127.0.0.1), keeping all traffic on the local loopback. This means:

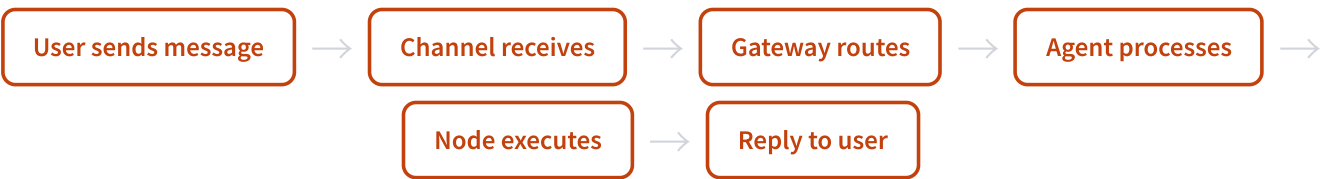
- No external ports exposed — inherently secure
- Nodes on the same machine connect directly to Gateway via WebSocket
- For remote access, expose through Tailscale Serve/Funnel rather than direct port exposure

核心建议

Only one Gateway instance runs per host. This is because channels like WhatsApp Web require exclusive sessions — multiple instances would cause login conflicts.

Communication Flow

The complete path from user message to Agent response:

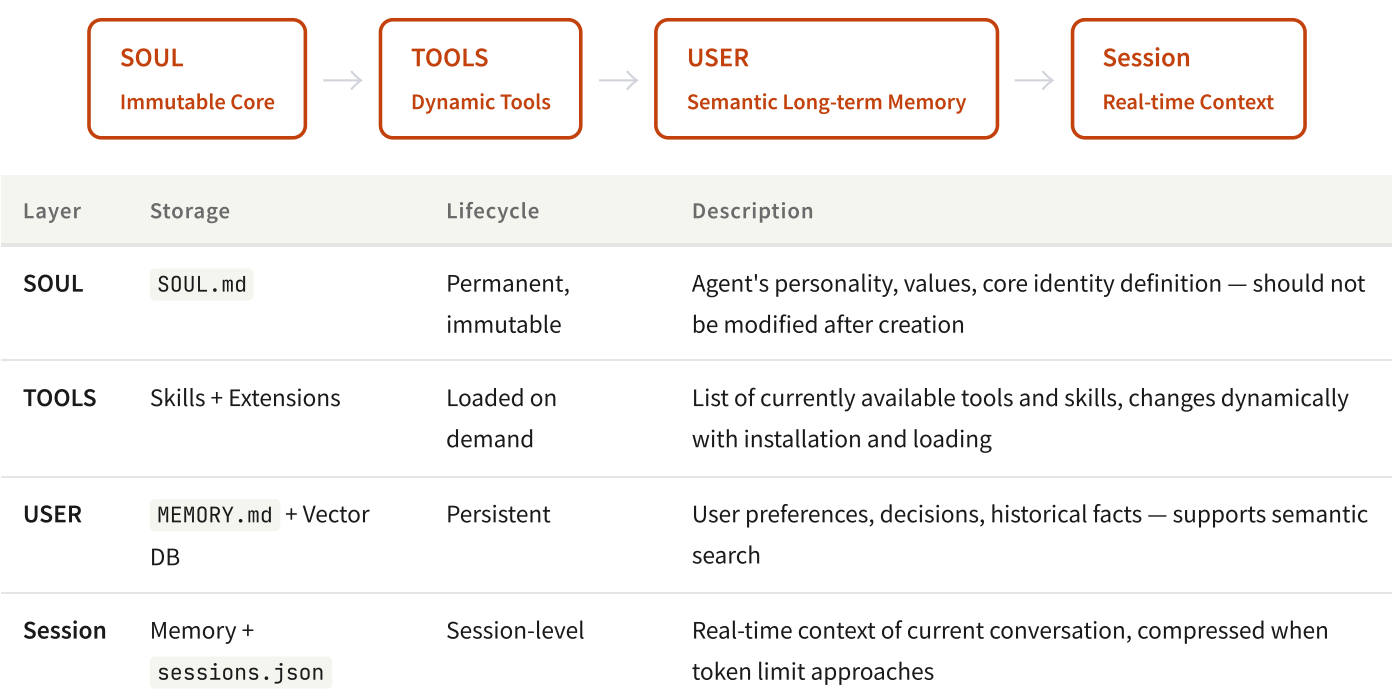


Gateway runs as a 24/7 daemon, continuously listening on all connected Channels. Unlike CLI Agents that lose context when sessions end, it runs persistently, accumulating memory.

06 Memory System

Memory is the core capability that distinguishes OpenClaw from ordinary chatbots. Four memory layers — from an immutable identity core to real-time conversation — build complete contextual continuity.

Four-Layer Memory Architecture



Daily Logs

Each day's interactions are written in append-only fashion to `memory/YYYY-MM-DD.md` files. At session start, the Agent automatically reads today's and yesterday's logs, providing continuity context for the conversation.

```
# memory/2026-03-08.md

## 10:23 - User asked about weather
Queried Beijing weather, replied sunny turning cloudy, 15-22°C

## 14:05 - Code review task
Helped user review api/routes.ts, found 3 potential issues...
```

Long-term Memory

`MEMORY.md` is an optional persistent file storing decision records, user preferences, and long-term facts. Key rules:

- Only loaded in main/private sessions (group-isolated sessions don't see it)
- Agent can proactively write to it, but typically triggered during Pre-Compaction
- Format is plain Markdown — humans can edit it directly

Automatic Memory Saving (Pre-Compaction)

When a Session approaches the token limit (default threshold ~4,000 tokens), OpenClaw triggers a silent agentic turn:

- 1

Threshold Detection
Session token usage approaches the limit, triggering the Pre-Compaction process
- 2

Silent Save
Agent executes a hidden turn in the background, writing important memories to `MEMORY.md` and Daily Log
- 3

Context Compression
Old messages are compressed or truncated, freeing token space. The user doesn't see this process (returns `NO_REPLY`)

Why this matters: This mechanism ensures that even in extremely long conversations, critical information is never lost as the context window slides. Tools like Claude Code lose context when sessions end, but OpenClaw achieves truly persistent memory through the file system.

Vector Memory Search

OpenClaw enables vector memory search by default, combining two retrieval strategies:

Strategy	Principle	Strengths
Embedding Vectors	Converts memory text into vectors, calculates semantic similarity	Fuzzy search, semantic associations ("that deployment issue we discussed before")
BM25 Keywords	Traditional keyword matching, TF-IDF weighted	Exact matching (specific file names, commands, person names)

Under the hood, it uses SQLite-vec for vector storage and accelerated retrieval. The system monitors memory file changes and automatically rebuilds indexes in a debounced manner.

Search Tools

- `memory_search` : Semantic search, returns ~400 token chunks, suitable for recalling vague context
- `memory_get` : Reads the full content of a specific memory file, suitable for precise lookups

07 Agent Workspace

Each Agent has an independent workspace directory in the file system. All configuration, memory, and skills exist as plain text files.

Directory Structure

```
workspace/
├── AGENTS.md          # Agent definition (identity, behavior rules)
├── SOUL.md            # Soul/personality instructions (immutable core)
├── USER.md           # User info and preferences
├── MEMORY.md          # Long-term memory storage
├── HEARTBEAT.md       # Heartbeat config (scheduled tasks)
├── memory/            # Log directory
│   └── YYYY-MM-DD.md  # Daily append-only logs
├── skills/            # Local skills directory
└── sessions.json      # Session storage
```

Core File Descriptions

File	Purpose	When Loaded
AGENTS.md	Agent identity definition, behavioral boundaries, response style. The file-based version of a system prompt	Every session start
SOUL.md	Immutable personality core. Defines "who" the Agent is — should not be modified by subsequent conversations	Every session start
USER.md	Structured user info: name, preferences, relationship	Main session start
MEMORY.md	Long-term memory, persistent facts and decisions proactively written by the Agent during conversations	Main session only
HEARTBEAT.md	Defines scheduled tasks and proactive behaviors (e.g., check task status every 30 minutes)	Gateway start
memory/	Daily Logs directory, auto-created by date, append-only	Reads today + yesterday logs
skills/	Workspace-level skills, highest priority (higher than global and built-in skills)	Scanned at session start
sessions.json	Session metadata storage, records status and history of each session	Read on demand

核心建议

All configuration files are plain Markdown or JSON. You can edit them directly with a text editor without any specialized tools. This embodies the OpenClaw philosophy: everything is text.

08 Sessions & Authentication

OpenClaw uses three layers — DM pairing, allowlists, and group rules — to identify user identities, and isolates context from different sources at the Session level.

DM Pairing Policy

When an unknown sender messages your Agent via any channel:

1 Generate Pairing Code

Agent replies with a one-time pairing code (6-digit number)

2 Await Verification

The message won't be processed. The Agent enters a waiting state. All subsequent messages are also held

3 Owner Approval

You enter the pairing code in an already-paired channel to approve the user, or reject directly

注意

DM Pairing is a critical mechanism for preventing abuse. Disabling it means anyone who knows your WhatsApp/Telegram number can use your Agent (and your API credits) without restriction.

Allowlist

In Agent configuration, the `allowFrom` field can pre-authorize specific users, bypassing the pairing process:

```
# Example configuration in AGENTS.md
allowFrom:
- telegram:123456789
- whatsapp:+8613800138000
- discord:user#1234
```

Users on the allowlist enter conversations directly when messaging, without pairing.

Group Rules (requireMention)

In group chats, Agent defaults to the `requireMention` strategy:

- Only responds to messages that @mention the Agent name, ignoring other group chat content
- Can switch to `always` mode (respond to all messages), but consumes significant tokens
- Chat command: `/activation mention|always`

Session Isolation

Scenario	Session Behavior	MEMORY.md
DM (Private)	All paired users' DMs fold into the shared <code>main</code> session	Loaded
Group	Each group defaults to an independent isolated session	Not loaded
Cross-channel	Same user's DMs on Telegram and WhatsApp share the main session	Loaded

Design intent: DMs are "your private space with the Agent" where all memories and preferences accumulate. Groups are public spaces — the Agent won't leak what you said in private.

09 Design Philosophy

There's a clear design philosophy behind OpenClaw's technical choices. Understanding these principles reveals why it deliberately doesn't do certain things.

Unix Philosophy

OpenClaw's core philosophy is directly inherited from Unix: small tools, composable, text streams. Creator Peter Steinberger's stance is clear:

"CLI is the ultimate interface for agents to connect with the world." No need to write an integration for every service — as long as the Agent can run command lines, it can operate everything.

Minimalism

OpenClaw's system prompt may be the shortest of any AI Agent framework. There are only 4 core tools:

Tool	Purpose
Read	Read files
Write	Write files
Edit	Edit files
Bash	Execute commands

This isn't a feature gap but a deliberate choice. Four tools are sufficient to cover virtually all OS-level tasks. Fewer tools mean a shorter system prompt, fewer token costs, and faster responses.

Why No Built-in MCP

MCP (Model Context Protocol) is a tool protocol standard proposed by Anthropic. Nearly every AI Agent framework is integrating MCP, but OpenClaw deliberately doesn't support it. Peter's own words:

"My premise is that MCP is garbage, it can't scale. You know what can scale? CLI. Unix."

OpenClaw's alternative approach:

- Agent calls CLI programs directly via the Bash tool — no intermediate protocol layer needed
- For scenarios that genuinely need MCP, bridge through the built-in `mcporter` skill
- Force the Agent to extend its own capabilities rather than consuming pre-built MCP toolsets

Self-Extending Capability

OpenClaw Agents can write, reload, and test their own extensions at runtime. This is a key reason why it appears "smarter" than other Agents:

- Encounters an unfamiliar operation → writes a skill to handle it
- Discovers a skill has a bug → modifies and reloads it
- Continuously improves its own toolchain in a loop

核心建议

Not relying on external pre-built tools comes at a cost: the Agent needs stronger model capabilities to "write tools from scratch." This is why OpenClaw recommends high-capability models like Claude Opus.

Session Tree Structure

OpenClaw Sessions aren't linear chat histories but tree structures:

- Agent can branch out a side-quest while executing the main task (e.g., fixing a tool)
- Side-quests don't consume the main Session's context window
- Upon completion, it can roll back to the main branch, bringing back only a one-line summary
- This allows the Agent to do deep exploration without "polluting" the main conversation

Codebase Scale & Performance

Metric	Value
Codebase Size	~430,000 lines of TypeScript
Memory Usage	~1 GB (runtime)
Startup Time	3-5 seconds
Extensions	40+ official extensions
Built-in Skills	55
Community Skills	13,729 (ClawHub registered)

430K lines of code and 1 GB of memory isn't exactly "lightweight." But for a 24/7 personal AI assistant, it's perfectly acceptable on modern hardware. The 3-5 second startup time ensures quick service recovery after Gateway restarts or updates.

Here's the translated HTML:

10 Deployment Overview

OpenClaw supports multiple deployment options from local to cloud. Your choice depends on technical skills, budget, and use cases.

Platform	One-Click Deployment	Minimum Requirements	New User Pricing	Built-in Models	Difficulty	Recommended For
Local npm	—	Node.js 22+	Free	No	Low	Developers, macOS/Linux users
Docker	—	Docker Engine	Free	No	Medium	Developers familiar with containers
Alibaba Cloud	Yes	2C2G 40GB	¥9.9/month	Yes (qwen3.5-plus)	Very Low (3 steps)	Top choice in China, beginner-friendly
Tencent Cloud	Yes	2C2G	~¥17/month	No (Requires Coding Plan)	Very Low (3 steps)	WeCom/QQ ecosystem users
Baidu Cloud	Yes	2C4G	¥0.01 first month	Yes (Qianfan models)	Very Low (4 steps)	For trial, ERNIE ecosystem
Huawei Cloud	Yes	Flexus L Instance	~¥85/month	No (Requires MaaS integration)	Medium (5+ steps)	Enterprise users, compliance needs
Volcengine	Yes	2C4G	¥9.9/month	Yes (Ark models)	Low (3-4 steps)	Top choice for Feishu users
Coze Code	Yes	No server required	¥49/month	Yes (Seed 2.0 etc.)	Very Low (2 steps)	Zero setup, no server management
Railway	Yes	Auto-allocated	\$5/month free credit	No	Very Low (1-click)	Overseas users, developers
Zeabur	Yes	2C4G dedicated	Pay-as-you-go	Yes (AI Hub)	Very Low (templates)	For multi-model failover needs

核心建议

Model costs are the real expense. Server costs have dropped significantly (¥9.9~99/year), while ongoing costs mainly come from model API calls. Focus on model pricing plans rather than just server costs when choosing a platform.

Here's the translated HTML:

11 Local Installation

Local installation is suitable for developers and users who want full control over their data. OpenClaw is a TypeScript project that runs on Node.js.

System Requirements

Requirement	Details
Node.js	>= 22 (required)
Package Manager	npm / pnpm / bun all supported
macOS	Requires Xcode Command Line Tools
Linux	Standard build tools (gcc, make)
Windows	WSL2 strongly recommended

Method 1: npm Global Install (Recommended)

The most recommended installation method, just two commands:

```
# Install OpenClaw
npm install -g openclaw@latest

# Initialize and install daemon
openclaw onboard --install-daemon
```

The `onboard` command will guide you through initial configuration, including model selection, API Key setup, and message channel configuration. The `--install-daemon` parameter will also install the daemon process to keep OpenClaw running continuously in the background.

Method 2: One-Click Script Install (macOS / Linux)

If you prefer not to manually install Node.js, you can use the official one-click install script:

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

注意

This command only works for macOS and Linux. Windows users should use WSL2 before running, or install via npm. Installation commands may change with version updates, please refer to the [official Getting Started page](#) or [GitHub README](#) for the latest instructions.

The script will automatically detect your system environment, install Node.js (if missing), and complete the OpenClaw installation.

Additional macOS Setup

macOS users need to ensure Xcode Command Line Tools are installed before proceeding:

```
xcode-select --install
```

If you plan to use the iMessage channel or Apple Notes skill, these rely on macOS-native AppleScript capabilities and will only work on macOS.

Windows User Notice

注意

OpenClaw strongly recommends Windows users to run via WSL2 (Windows Subsystem for Linux). Running natively on Windows may encounter path, permission, and other compatibility issues.

After installing WSL2, follow the Linux installation process within the Ubuntu terminal.

Daemon Process

The daemon process keeps OpenClaw running continuously in the background, even when the terminal is closed. Different systems use different process management methods:

System	Process Management	Description
macOS	launchd	macOS native service management with auto-start
Linux	systemd	Linux standard service management, controlled via <code>systemctl</code>

After installing the daemon, OpenClaw Gateway will continuously listen at `ws://127.0.0.1:18789`.

Here's the translated HTML:

12 Docker Deployment

Docker deployment is suitable for scenarios requiring environment isolation, easy migration, or long-term operation on servers.

docker-compose Quick Start

The OpenClaw repository includes a built-in `docker-compose.yml` file, allowing you to start with a single command:

```
# Clone repository
git clone https://github.com/openclaw/openclaw.git
cd openclaw

# Start
docker-compose up -d
```

Image Variants

Variant	Description	Use Case
Standard Image	Full functionality with all extension dependencies	General use, complete features
slim variant	Multi-stage build, smaller size	Resource-constrained environments, CI/CD
sandbox	Sandbox environment (Dockerfile.sandbox)	Security isolation, code execution
sandbox-browser	Sandbox with browser	Browser automation required

To use the slim variant: Set the environment variable `OPENCLAW_VARIANT=slim` in `docker-compose.yml`. Starting from v2026.3.7, extension dependencies can be pre-baked, allowing container images to pre-install extensions and reduce installation wait time during startup.

Volume Mounts

Docker deployment requires mounting two critical directories to ensure data persistence:

```
volumes:
  - ~/.openclaw:/root/.openclaw      # Configuration and state data
  - ~/openclaw/workspace:/workspace  # Workspace (YAML config files)
```

Important: Without mounting these directories, all configurations and conversation history will be lost after container restart. `~/.openclaw` stores runtime state, while `workspace` contains YAML configuration files.

Port Mapping

OpenClaw Gateway listens on port 18789 (WebSocket) by default, and the Web UI uses port 3000. Configure port mapping in `docker-compose.yml`:

```
ports:
  - "18789:18789"    # Gateway WebSocket
  - "3000:3000"     # Web UI
```

Podman Support

OpenClaw also supports running with Podman. Podman is a daemonless alternative to Docker with largely compatible commands:

```
# Start with Podman
podman-compose up -d
```

For environments requiring rootless container operation (such as enterprise security policies), Podman is a more suitable choice. Starting from v2026.3.8, OpenClaw automatically detects SELinux mode and adds `:Z` volume relabeling, fixing EACCES permission errors on distributions like Fedora/RHEL.

13 国内云厂商一键部署

Cloud Deployment in China

这是大多数国内用户的首选方案。所有主流云厂商都已支持 OpenClaw 一键部署，差异主要在价格策略和 IM 生态集成上。

阿里云 Alibaba Cloud

国内社区资源最丰富的平台，镜像预装，开箱即用。

项目	详情
配置	2vCPU + 2GiB 内存 + 40GiB ESSD 系统盘
系统	Alibaba Cloud Linux 3.2104 LTS 64位，预装 OpenClaw 镜像
价格	限时秒杀 9.9元/月，包年常规优惠低至 68元/年
模型	默认内置 qwen3.5-plus；百炼 Coding Plan Lite 首月 10元（18,000次/月）
IM 支持	钉钉、飞书等（通过 openclaw-china 插件）

- 1

一键购买

进入活动页，购买预装 OpenClaw 镜像的轻量应用服务器。镜像版本 OpenClaw 2026.2.26。
- 2

放通端口 + 配置

在安全组中放通 18789（Gateway）和 3000（Web UI）端口，配置百炼 API Key。
- 3

访问 Web UI

浏览器访问 `http://你的IP:3000`，进入 OpenClaw 管理界面，可选集成钉钉/飞书等 IM。

注意秒杀价格：9.9元/月是限时秒杀价，需要抢。常规价不算最便宜，且续费价格比新购高不少。如果你不急，可以等下一波活动。

腾讯云 Tencent Cloud

四大 IM 全面支持，Coding Plan 模型套餐性价比高。

项目	详情
配置	推荐 2核4G（黄金配置），最低 2核2G 可运行
价格	新人包 2核4G 约 17元/月，一年 99元起
模型	Coding Plan 首月 7.9元起，含 HY 2.0 Instruct、GLM-5、kimi-k2.5、MiniMax-M2.5 等
IM 支持	企微、QQ、钉钉、飞书（四大 IM 全覆盖）
续费	支持「限时同价续费」活动，避免续费刺客

- 1

购买 Lighthouse 实例

在腾讯云轻量应用服务器页面购买实例。
- 2

选择 OpenClaw 模板

应用模板 → AI智能体 → OpenClaw，一键安装。
- 3

配置模型 + 接入 IM

购买 Coding Plan 获取模型调用能力，然后接入企微/QQ/飞书/钉钉。

百度智能云

Baidu Cloud

试错成本最低：0.01元首月体验，全图形界面操作。

项目	详情
配置	推荐 2核4G 4M 带宽（轻量应用服务器）
价格	首月体验 0.01元（每日限量 500 台），常规 70~140元/月
模型	千帆平台集成文心系列、Qwen系列、DeepSeek系列
特色	百度搜索/百度百科独有能力；千帆 7 款官方 Skills 已上线 ClawHub

- 1

购买服务器

购买轻量应用服务器，选择 OpenClaw 镜像。
- 2

等待自动安装

系统自动完成环境安装和服务启动。
- 3

配置模型

页面选择模型，平台自动完成千帆 API Key 创建与配置。
- 4

对接 IM 渠道

按需接入钉钉、飞书等消息频道。

注意

首月 0.01 元优惠每日限量 500 台，需要抢。续费价格较高（70~140元/月），建议仅作体验使用。

华为云 Huawei Cloud

企业级安全与合规能力最强，适合已在华为生态的企业用户。

项目	详情
配置	Flexus L 实例，需创建弹性公网 IP + 安全组
价格	~85~155元/月，无特别突出的新用户优惠
模型	需在 MaaS 控制台单独开通 AI 模型
部署步骤	5步+（创建实例 → EIP → 安全组 → 安装 → 配模型）
优势	企业级安全合规、支持自动扩展、MaaS 模型丰富

华为云的部署步骤相对较多，需要单独配置弹性公网 IP、安全组、COC 服务等。对个人用户不够友好，但如果你的企业已在华为云生态内，这是最合规的选择。

火山引擎 Volcengine

飞书深度集成，19.8元/月的服务器+模型组合套餐是目前综合性价比最高的方案。

项目	详情
配置	推荐 2核4G，支持云服务器和云手机两种部署方式
价格	活动价 9.9元/月；方舟 Coding Plan 组合套餐 19.8元/月（服务器+模型）
模型	方舟平台模型丰富，内置可用
IM 支持	飞书（深度集成）、企微、钉钉、QQ
特色	云手机部署方式独特，可运行移动端任务

- 1

购买云服务器

购买云服务器或云手机，选择 OpenClaw 应用模板。
- 2

配置方舟模型

在火山方舟平台选择模型，配置 Coding Plan。
- 3

接入飞书

接入飞书/企微/钉钉/QQ。飞书用户推荐直接使用深度集成方案。

扣子编程 Coze Code

零门槛方案：不需要服务器、不需要写代码、不需要配环境。1 分钟完成部署。

项目	详情
配置	无需服务器，完全在扣子编程平台上运行
价格	必须订阅会员才能部署 OpenClaw：¥49/月（基础）或 ¥99/月（进阶），免费版不支持
模型	内置多个模型可选：Seed 2.0、DeepSeek、GLM-4.7 等，也可接入第三方 API
特色	模型、联网搜索、生图 Skill 全部默认配好；扣子编程 Skills 可直接加载

1 进入扣子编程

访问 `code.coze.cn`，点击「一键部署 OpenClaw」或从优秀案例创建副本。

2 确认部署

确认后，模型/联网/生图全部默认配置好，部署后持续在线。

扣子编程的限制：必须订阅 ¥49/月起的会员才能部署 OpenClaw（免费版不支持）；自定义程度不如自建服务器；数据存储在第三方平台。如果你对成本敏感，自建方案（腾讯云/火山引擎 + DeepSeek，约 ¥19/月起）对技术用户性价比更高。

海外平台 International Platforms

Sealos

K8s 原生云平台，支持 7 天免费试用。通过 Devbox 云开发环境一键部署，按用量计费。适合有容器化需求的开发者，但需要一定的 K8s 知识，且没有专门针对 OpenClaw 的预置模板。

Zeabur

模板部署，已被部署超过 29,000 次。最大亮点是 AI Hub 内置多模型 failover 链：glm-4.7-flash → grok-4-fast → minimax-m2.5 → kimi-k2.5 → qwen-3-235b → gpt-5-mini。主要面向海外/台湾市场，必须使用专用服务器（Dedicated Server）。

Railway

真正的一键部署，全程浏览器操作。提供 \$5/月免费额度，轻度使用可零成本。多种模板可选（标准/快速启动/All-in-One），部署成功率 96~100%。海外平台，国内访问需要科学上网。

按场景推荐 Recommendations by Scenario

场景	首选	备选	理由
零基础想最快体验	扣子编程	百度云	不需要服务器，2步部署，内置模型（需¥49/月起会员）
个人长期使用，预算敏感	火山引擎	阿里云	19.8元/月（服务器+模型），综合最划算
飞书重度用户	火山引擎	扣子编程	同为字节系，飞书深度集成
企微/QQ 生态	腾讯云	—	四大 IM 原生支持，Coding Plan 7.9元起
企业级部署，合规优先	华为云	阿里云	安全合规能力最强
开发者/海外用户	Railway	Zeabur	一键部署，免费额度，开发者体验极佳

Here's the translated HTML:

14 Initial Configuration

Regardless of the deployment method, initial configuration is required after installation. This section covers the most critical configuration items.

Gateway Authentication Settings

注意

v2026.3.7 Breaking Change: Gateway authentication now requires explicit setting of `gateway.auth.mode`. Failure to set this will prevent the Gateway from starting. This change addresses security vulnerabilities previously exposed by over 30,000 unauthenticated instances on the internet.

Set the authentication mode in the configuration file located at `~/.openclaw/workspace` :

```
# Select an authentication mode
gateway:
  auth:
    mode: token      # Option 1: Token authentication (recommended for API integration)
    # or
    mode: password   # Option 2: Password authentication (recommended for Web UI access)
```

Model Selection & API Key Configuration

OpenClaw supports multiple model switching. You need to configure at least one model's API Key. Common options:

Model Source	Access Method	Description
Alibaba Cloud Bailian	Apply on Bailian platform	Preferred for China, includes qwen3.5-plus and other models
Tencent Cloud Coding Plan	Purchase on Tencent Cloud	Multi-model package, first month at 7.9 RMB
Volcengine Ark	Apply on Ark platform	Doubao series models
Anthropic API	console.anthropic.com	Claude series models, pay-as-you-go
OpenAI API	platform.openai.com	GPT series models, pay-as-you-go
Ollama (Local)	Local Ollama installation	Free, requires sufficient local computing power

核心建议

If you're using one-click deployment from domestic cloud providers, models and API Keys are typically pre-configured during purchase. Manual configuration is only required for local installations and Docker deployments.

Version Updates

OpenClaw releases new versions almost daily. Update using the following commands:

```
# Update to latest stable version (recommended)
openclaw update --channel stable

# Update to Beta version (early access)
openclaw update --channel beta

# Update to Dev version (latest features, potentially unstable)
openclaw update --channel dev
```

Differences between update channels:

Channel	Update Frequency	Stability	Target Users
stable	Several times per week	High	Most users
beta	Almost daily	Medium	Users wanting early access to new features
dev	Continuous	Low	Developers, contributors

Diagnostic Checks

After installation, run the diagnostic command to verify your environment:

```
openclaw doctor
```

This command checks:

- Node.js version meets requirements (≥ 22)
- Required system dependencies are installed
- Gateway connection status
- Validity of configured model API Keys
- Daemon process status
- Network connectivity

If any issues are found, `openclaw doctor` will provide specific remediation suggestions. This is the first step in troubleshooting.

Local Backup (v2026.3.8+)

v2026.3.8 introduced a local backup tool for quickly creating and verifying backups before performing destructive operations:

```
# Create full backup
openclaw backup create

# Backup only configuration files
openclaw backup create --only-config

# Verify backup integrity
openclaw backup verify
```

核心建议

Develop a habit of regular backups. Especially before version upgrades or configuration changes, run `openclaw backup create` first to enable quick rollback if issues arise.

Recommended Version: As of March 11, 2026, we recommend using v2026.3.8 stable version. This version adds ACP authentication, local backup tools, and 12+ security patches on top of v2026.3.7.

15 Channel Overview

OpenClaw connects to 20+ chat platforms through a unified Gateway architecture. All channels share the same three-step integration process: Create credentials → Write configuration → Launch Gateway.

Unified Integration Process



Multiple channels can run simultaneously with messages automatically routed to corresponding platforms. Pairing mode (`dmPolicy: pairing`) is enabled by default - unknown senders require verification codes to chat with the bot.

Complete Platform List

Channel	SDK / Implementation	Type	Difficulty	Time Required
Telegram	grammY	Built-in	Very Easy	5 minutes
Discord	discord.js	Built-in	Easy	15-20 minutes
WhatsApp	Baileys	Built-in	Medium	10-15 minutes
Slack	Bolt	Built-in	Medium	25-40 minutes
Signal	Signal-CLI	Built-in	Medium	20-30 minutes
iMessage	BlueBubbles	Extension	Medium-Hard	30-45 minutes
Google Chat	Official API	Built-in	Medium	15-20 minutes
LINE	Official API	Extension	Medium	15-20 minutes
Microsoft Teams	Official API	Extension	Medium	20-30 minutes
Matrix	Protocol Implementation	Extension	Medium	15-20 minutes
Mattermost	Official API	Extension	Medium	15-20 minutes
IRC	Protocol Implementation	Extension	Medium	10-15 minutes
Nostr	Protocol Implementation	Extension	Medium	15-20 minutes
Twitch	Official API	Extension	Medium	15-20 minutes
Synology Chat	Official API	Extension	Medium	15-20 minutes
BlueBubbles	API	Extension	Medium-Hard	30-45 minutes
Zalo	API	Extension	Medium	15-20 minutes
Nextcloud Talk	API	Extension	Medium	15-20 minutes
Tlon	Protocol Implementation	Extension	Medium	15-20 minutes
QQ	Official Plugin	Plugin	Easy	5 minutes
Feishu	Official API	Built-in Plugin	Medium	15-20 minutes
DingTalk	Community Plugin	Plugin	Medium	20-30 minutes
WeCom	Community Plugin	Plugin	Medium	20-30 minutes
WeChat ClawBot	Official WeChat Plugin (iLink Protocol)	Official	Easy	5-10 minutes

Channel	SDK / Implementation	Type	Difficulty	Time Required
WeChat (Legacy)	Community / Third-party	Plugin	Complex	1 hour+
Browser (Chrome)	Browser Relay Extension	Built-in (Added in v3.13)	Easy	5-10 minutes

Beginner Recommendations

Recommended from easiest to hardest: Telegram (simplest, 5 minutes zero barrier) → QQ (top choice in China, scan QR code to use) → Chrome Browser (5 minutes, directly takes over logged-in accounts) → Discord (great for communities) → Feishu (for Chinese enterprises) → DingTalk (mature community plugins) → WhatsApp (overseas daily communication)

Tier	Platforms	Recommendation Reasons
Tier 1 <i>5-10 minutes</i>	Telegram, QQ	Telegram doesn't require public IP or reverse proxy - runs locally via long-polling. QQ has official Tencent support, binding takes 1 minute via QR scan.
Tier 2 <i>15-20 minutes</i>	Discord, Feishu	Discord has complete documentation with slightly more permission steps but clear process. Feishu has built-in support since OpenClaw 2026.2, ideal for Chinese enterprises.
Tier 3 <i>25-40 minutes</i>	WhatsApp, Slack, DingTalk, WeCom	WhatsApp is most popular but sessions may expire. Slack requires more permission configuration. DingTalk and WeCom have mature community plugins.
Tier 4 <i>Special requirements</i>	iMessage, WeChat Personal	iMessage requires always-on Mac running BlueBubbles. WeChat Personal has no official API with persistent ban risks.

16 International Platform Integration

This chapter covers detailed integration steps for six major international platforms. Each platform's complete workflow from credential creation to conversation setup.

Telegram Recommended for Beginners · 5 mins · Zero Barriers

Telegram is the officially recommended entry channel for OpenClaw. Using long-polling mode, the bot actively polls Telegram servers to fetch messages without requiring public IP, reverse proxy or port forwarding. Works reliably for local development, behind NAT or firewalls.

1 Find @BotFather

Search for `@BotFather` in Telegram - the official bot management tool. Send it the `/newbot` command.

2 Create Bot

Follow prompts to set bot display name and username (must end with `bot`, e.g. `my_openclaw_bot`). Upon creation, BotFather will return a Bot Token.

3 Configure OpenClaw

Add Token to `openclaw.yaml`:

```
channels:
  telegram:
    enabled: true
    botToken: "YOUR_BOT_TOKEN"
    dmPolicy: pairing # Requires pairing code
```

4 Start & Pair

Restart Gateway. Send any message to your bot in Telegram - Gateway will return a pairing code. Enter it to start conversations.

核心建议

Telegram Bot API 9.5 (March 2026) added `sendMessageDraft` feature. Chinese users need proxy for Telegram access, but bot operation itself is unaffected - as long as the Gateway machine can reach `api.telegram.org`.

Discord Ideal for Communities · 15-20 mins

Discord suits community management and team collaboration. Requires creating Application and Bot in Developer Portal with slightly more permission steps but well-documented.

1 Create Application

Go to `discord.com/developers/applications`, click New Application, enter app name.

2 Get Bot Token

Under Bot page, click Reset Token and copy the generated Token.

3 Enable Privileged Intents

Enable two permissions in Bot page: **Message Content Intent** and **Server Members Intent**. Without these, bot cannot read message content.

4 Invite Bot to Server

Under OAuth2 → URL Generator, check `bot` scope and required permissions, generate invite link to add bot to your Discord server.

5 Get IDs & Configure

Enable Developer Mode in Discord (Settings → Advanced → Developer Mode), right-click to copy Server ID and your User ID. Add these to `openclaw.yaml` and start Gateway.

6 DM Pairing

DM your bot in Discord, enter pairing code (valid 1 hour) to complete binding.

核心建议

v2026.3.7 added ACP persistent channel binding - Discord channel and Telegram topic bindings persist after Gateway restart without re-pairing.

WhatsApp Daily Communication · 10-15 mins

WhatsApp is the most popular channel in OpenClaw community. Uses Baileys library with QR code scanning, no WhatsApp Business API required.

1 Run Interactive Wizard

After installing OpenClaw, run `openclaw onboard` and select WhatsApp channel.

2 Scan QR Code

Terminal will display QR code. Open WhatsApp mobile → Settings → Linked Devices → Link New Device, then scan QR code.

3 Start Using

After pairing completes, you can chat with bot in WhatsApp.

注意

Recommended to use separate number for WhatsApp, not primary number. For Gateway runtime, prefer Node over Bun (Bun unstable for WhatsApp). Treat session credentials like passwords - expired sessions require re-scanning.

Slack Enterprise/Team Scenarios · 25-40 mins

Slack fits enterprise and internal team use. Requires creating App in Slack API platform with multiple permission configurations. Default uses Socket Mode (WebSocket), no public URL needed.

1 Create Slack App

Go to `api.slack.com/apps`, click Create New App → From scratch, select target Workspace.

2 Enable Socket Mode

Enable in Socket Mode page, generate App-Level Token (starts with `xapp-`), select `connections:write` scope.

3 Configure Bot Token Scopes

Add permissions in OAuth & Permissions: `chat:write`, `channels:history`, `channels:read`, `im:write`, `im:history`, `im:read`, `users:read`, `reactions:read`, `reactions:write`, `files:write`.

4 Install & Configure

Install App to Workspace, get Bot User OAuth Token (starts with `xoxb-`). Add Token to `openclaw.yaml`, start Gateway.

注意

OpenClaw can execute real commands on your machine, posing prompt injection risks. In multi-user environments like Slack, avoid running Gateway on primary machines - use VM or dedicated server.

Signal End-to-End Encrypted · 20-30 mins

Signal provides end-to-end encrypted communication. OpenClaw connects via Signal-CLI tool.

1 Install Signal-CLI

Install Signal-CLI per OS. macOS via `brew install signal-cli`, Linux from GitHub Releases.

2 Register or Link Number

Use `signal-cli register` for new number or `signal-cli link` to link existing Signal account.

3 Configure OpenClaw

Configure Signal channel in `openclaw.yaml` with number and Signal-CLI path, then start Gateway.

iMessage Apple Ecosystem · 30-45 mins · Requires Mac

iMessage integration implemented via BlueBubbles bridge (replacing deprecated `imsg` channel). Requires always-on Mac as BlueBubbles Server.

1 Install BlueBubbles Server

Download BlueBubbles Server from `bluebubbles.app/install` on Mac. Recommended macOS Sequoia (15) or newer.

2 Enable Web API

In BlueBubbles Server settings, enable Web API and set access password.

3 Configure OpenClaw

Configure BlueBubbles channel in `openclaw.yaml` : server URL, password, webhook path.

```
extensions:
  bluebubbles:
    enabled: true
    serverUrl: "http://localhost:1234"
    password: "YOUR_PASSWORD"
```

4 Configure Webhook

Add webhook in BlueBubbles pointing to Gateway: `https://gateway-host:3000/bluebubbles-webhook?password=<password>` . Webhook must have password authentication.

注意

iMessage via BlueBubbles supports edit, recall, effects and reaction emojis. But macOS 26 Tahoe has regression bug for edit function (issue #32275). Mac must remain powered on running BlueBubbles Server.

Here's the translated HTML with all the rules applied:

17 Domestic Platform Integration

OpenClaw's support for China's IM ecosystem is rapidly developing. QQ and Feishu have official-level support, DingTalk and WeCom have mature community plugins, and WeChat personal accounts welcomed an official solution on March 22, 2026 - ClawBot.

QQ Top Choice in China · Scan to Use

QQ is the easiest way for Chinese users to access OpenClaw. Tencent has officially opened QQ Bot capabilities to OpenClaw, with binding completed in just 1 minute via QR code scan. Supports multimedia messages including Markdown, images, voice, and files, available on both mobile QQ and desktop QQ.

1 Register as QQ Bot Developer

Complete developer registration by scanning the QR code with mobile QQ. Unverified accounts need to complete real-name verification first. Each account can create up to 5 Bots.

2 Create QQ Bot

Create a Bot with one click on the QQ Open Platform and obtain App ID and Token.

3 Configure OpenClaw

Complete configuration binding in the OpenClaw runtime environment, then you can chat with the bot on QQ.

核心建议

QQ Bot suits two scenarios: personal assistant (private chat mode) and QQ community management (group auto-reply, batch processing, scheduled notifications).

Feishu Top Choice for Chinese Enterprises · Built-in since OpenClaw 2026.2

Feishu has received native built-in support since OpenClaw 2026.2. Uses WebSocket event subscription, supporting private chats, group chats, and multimedia messages including photos/files/videos.

1 Create Feishu Application

Create an enterprise self-built application on the Feishu Open Platform (`open.feishu.cn`) and obtain App ID and App Secret.

2 Run Setup Wizard

Run `openclaw onboard` , select Feishu channel, and paste App ID and App Secret.

3 Restart Gateway

After restarting Gateway, you can chat with the bot in Feishu.

Community Alternative: If you don't want to use the built-in plugin, AlexAnys/feishu-openclaw provides an independent bridge that doesn't require public servers, domains, or ngrok, deployable in 5 minutes. The AlexAnys/openclaw-feishu repository has step-by-step configuration guides, including API exhaustion troubleshooting and Lark Webhook intranet penetration solutions.

DingTalk Community Plugin · Stream Mode No Public Network Needed

DingTalk integrates with OpenClaw through community plugins. Message reception uses Stream mode (WebSocket long connection), requiring no public network address. Supports private chats, group chats, file attachments, voice messages, DingTalk document APIs, multi-Agent routing, etc.

1 Create DingTalk Application

Create an application on the DingTalk Open Platform and add bot capabilities.

2 Set Stream Mode

Set message reception mode to Stream mode. This way the bot receives messages via WebSocket long connection without configuring public callback addresses.

3 Install Plugin and Configure

Install community plugin `@soimy/dingtalk`, or use the official `dingtalk-openclaw-connector` from DingTalk-Real-AI (supports AI Card streaming responses). Configure `openclaw.yaml` and start Gateway.

核心建议

DingTalk hasn't received official built-in OpenClaw support yet (Feature Request proposed in March 2026), but community solutions are already very mature. The DingTalk-Real-AI connector is maintained by the DingTalk team with guaranteed reliability.

WeCom Two Modes · Verified by Multiple Cloud Platforms

WeCom has two integration modes: Agent mode (XML callback classic mode) and Bot mode (JSON callback with native stream support). Adopted and verified by public cloud platforms including Tencent Cloud, Volcengine, and China Telecom Cloud.

1 Create WeCom Application

Create a self-built application (Agent mode) or configure an intelligent bot (Bot mode) in the WeCom management backend.

2 Install Community Plugin

Optional plugins: `dingxiang-me/OpenCLaw-Wechat` (supports WeChat interoperability, streaming output, group @ mentions, whitelist control, full Chinese configuration) or `sunnyo/openclaw-plugin-wecom` (supports dynamic Agent management, command whitelist).

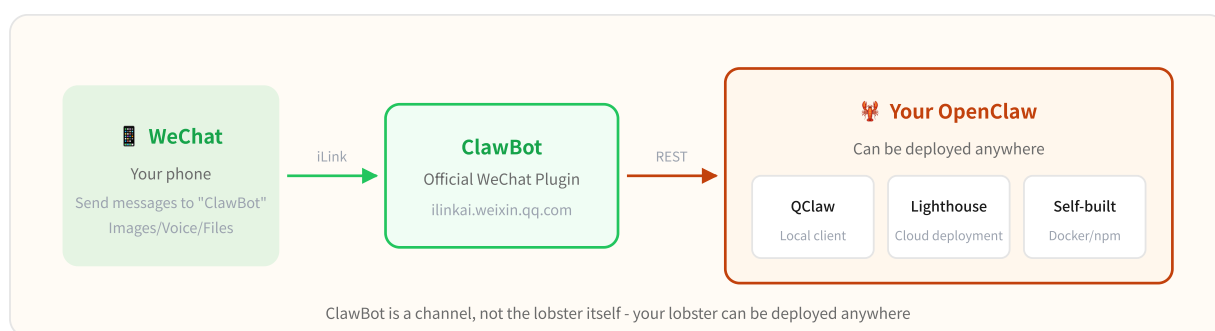
3 Configure and Start

Configure `openclaw.yaml` according to plugin documentation and start Gateway. Requires OpenClaw $\geq 2026.2.9$, some features require $\geq 2026.3.2$.

WeChat ClawBot Official Solution · Launched March 22, 2026

For a long time, integrating with personal WeChat was the biggest pain point for OpenClaw users in China - no official API, only workarounds via WeCom or iPad protocol, with constant account ban risks. On March 22, 2026, the WeChat team launched the official ClawBot plugin, finally providing a proper solution.

ClawBot is essentially simple: it's not a new lobster product, but a messaging channel. Your WeChat contacts will have a new friend called "WeChat ClawBot", and any existing OpenClaw instance (regardless of deployment location) can communicate with your WeChat through it.



1 Install ClawBot CLI

Run one command in terminal:

```
npx -y @tencent-weixin/openclaw-weixin-cli@latest install
```

The npm package is under the official `@tencent-weixin` namespace, with underlying protocol domain `ilinkai.weixin.qq.com`, both being official WeChat assets.

2 Scan QR Code to Bind

After installation, scan the QR code to bind WeChat. Supports various message types including images, voice, video, files, with native group chat support (`group_id` field).

Three Considerations

- **iOS Priority:** Requires WeChat 8.0.70 or higher, currently supported on iOS, Android timeline not confirmed yet
- **Gradual Rollout:** Plugin is still being gradually released, not all users can see the entry (Me → Settings → Plugins). If you don't see it, don't worry - try running the install command anyway
- **Not Omnipotent:** ClawBot is positioned as a messaging channel, not involving WeChat automation operations or chat history access. Tencent reserves the right to restrict connectable third-party AI service types

Dimension	ClawBot (Official)	Old Solutions (iPad Protocol etc.)
Legality	Official WeChat product with legal terms	Gray area, violates WeChat ToS
Protocol	HTTP/JSON REST API (iLink protocol)	Private binary protocol (reverse engineered)
Stability	Server-side maintained, long-term reliable	May break with WeChat updates
Ban Risk	None	Always present
Access Threshold	One command + QR scan	Requires WeCom registration / Open Platform permission application

核心建议

After ClawBot's launch, previous solutions like WeCom relay and iPad protocol can basically be retired. If you're already using old solutions, consider migrating to ClawBot soon - the official channel eliminates ban worries and has much lower maintenance costs.

Old WeChat Solutions

Alternative Paths Before ClawBot

The following three solutions were mainstream choices before ClawBot's launch. After ClawBot completes its rollout, these solutions will become much less necessary, but remain valuable as references.

Solution A: WeCom Relay

Access OpenClaw through WeCom, then use WeChat plugins to connect WeCom and personal WeChat. Legal and compliant within WeChat ecosystem, requires WeCom management backend permissions.

Solution B: iPad Protocol + Relay Gateway

Avoids Web protocol (high ban risk), uses iPad protocol instead. More stable but technically more challenging.
Community projects: `freestylefly/openclaw-wechat` , `lao1in5564/openclaw-wechat` .

Solution C: WeChat Mini Program

New 2026 solution, connects OpenClaw through mini programs. Alibaba Cloud/Tencent Cloud have pre-built images to lower deployment barriers.

注意

Old solutions require continuous maintenance, protocol updates may cause failures. If ClawBot is already available to you, strongly recommend switching to the official channel.

openclaw-china Unified Plugin

One-stop Domestic Platform Support

BytePioneer-AI/openclaw-china provides one-stop domestic platform support, covering Feishu, DingTalk, QQ, WeCom, and WeChat.

```
git clone https://github.com/BytePioneer-AI/openclaw-china.git
cd openclaw-china
pnpm install && pnpm build
openclaw china setup # Interactive configuration wizard
```

Features include: Interactive configuration wizard to reduce manual setup, WeCom MP4 video player and multi-file type sending, Tencent Cloud ASR speech-to-text, DingTalk log enhancement (userId/groupId issue tracking).

Selection Advice: If only using one domestic platform, installing the corresponding standalone plugin is lighter. If integrating multiple domestic platforms simultaneously, the openclaw-china unified package is more convenient.

Here's the translated HTML:

18 Remote Access

OpenClaw Gateway listens locally on `ws://127.0.0.1:18789` by default. When you need to access it from external networks, here are several solutions.

Tailscale Serve / Funnel Recommended Solution

Tailscale is the officially recommended remote access solution for OpenClaw, offering two modes:

Mode	Access Scope	Use Case
Serve	Devices within Tailscale network	Accessing home OpenClaw from your phone/tablet
Funnel	Public internet (anyone)	Providing public URLs for webhook callbacks (e.g., Feishu, Slack HTTP mode)

```
# Serve: Accessible only within Tailscale network
tailscale serve --bg https+insecure://127.0.0.1:18789

# Funnel: Publicly accessible (for webhook callbacks)
tailscale funnel --bg https+insecure://127.0.0.1:18789
```

核心建议

Most channels (Telegram long-polling, Discord, Slack Socket Mode, DingTalk Stream mode) involve bots actively connecting to servers and don't require public IPs. Only webhook callback scenarios (BlueBubbles, Slack HTTP mode) need Funnel to expose public addresses.

SSH Port Forwarding Most Universal Solution

If OpenClaw runs on a remote server, use SSH tunnel to forward the Gateway port locally:

```
# Forward remote server's port 18789 to local
ssh -L 18789:127.0.0.1:18789 user@your-server

# Run in background
ssh -fNL 18789:127.0.0.1:18789 user@your-server
```

After forwarding, local clients can connect to `ws://127.0.0.1:18789` to access the remote Gateway.

Dashboard Web UI (v2026.3.12 Major Upgrade)

v2026.3.12 introduced Dashboard v2, upgrading the console from a functional interface to a complete modular management center:

View	Features
Overview	Agent status overview, Token usage, channel connection status
Chat	Slash commands, message search, export, pinned messages
Config	Model configuration, channel toggles, Plugin management
Agent	Memory viewing, Cron task management
Session	Multi-session management, history records

```
# Accessible by default after Gateway starts
# Open in browser: http://127.0.0.1:18789
openclaw gateway --port 18789 --verbose
```

Added Command Palette (press `⌘K` for quick navigation). Mobile version now features bottom tab bar instead of floating buttons.

Security Notice: Since v2026.3.7, Gateway authentication requires explicit `gateway.auth.mode` setting (token or password). v2026.3.12 uses Ephemeral Token for device pairing, making connections more secure. Never expose unauthenticated Gateway to public networks.

Chrome Browser Integration New in v2026.3.13 · 5-10 min setup

v2026.3.13 introduces the most unique "channel": direct attachment to your Chrome browser. The Agent can control your logged-in accounts and pages—no separate accounts or API keys needed, directly taking over your browser session.

Compared to previous browser tools (headless mode), this solution's core advantage is: **leveraging existing login states**. The Agent doesn't need to log in to websites—it operates using your current logged-in state.

- 1

Install Browser Relay Extension
Install **OpenClaw Browser Relay** from Chrome Web Store (Extension ID: `nglingapjinhecnfejdcpihlpneeadjp`).

2 Launch Chrome with Debug Port

Close all Chrome windows and relaunch with:

```
# macOS
open -a "Google Chrome" --args --remote-debugging-port=9222

# Windows
"C:\Program Files\Google\Chrome\Application\chrome.exe" --remote-debugging-port=9222
```

Or use built-in Profile shortcut in `openclaw.yaml` :

```
browser:
  profile: "user"          # Use default user Profile
  # profile: "chrome-relay" # Use relay mode
```

3 Enable Browser Channel in OpenClaw

Enable browser tool in `openclaw.yaml` :

```
tools:
  browser:
    enabled: true
    attachMode: "devtools" # New in v3.13, attach to existing session
```

4 Assign Browser Tasks to Agent

After restarting Gateway, directly send browser-related commands to Agent, e.g.: "Organize all open GitHub PRs in my browser", "Fill out this form", "Take screenshot of current page".

核心建议

v2026.3.13 also supports browser act batching: packaging multiple browser operations for batch execution, reducing round-trip communication, making automation more stable and less prone to page rendering timing issues.

注意

Chrome DevTools attachment means the Agent can read all page content in your browser, including private information from logged-in accounts. Only use with trusted Agent configurations, and never enable this feature on public computers or uncontrolled servers.

macOS Menu Bar Companion App

OpenClaw provides a native macOS client (`apps/macos/`) that runs as a menu bar app. Features include:

- One-click Gateway start/stop
- View connected channel status
- Quick access to Dashboard Web UI
- System notifications (new messages, pairing requests, etc.)

iOS and Android clients are in development (`apps/ios/` , `apps/android/`), with code already in main repository.

核心建议

If using multiple devices, we recommend Tailscale Serve + macOS menu bar app combo: Run Gateway and menu bar app on Mac, access via Tailscale network from phone/tablet.

Here's the translated HTML:

19 How Skills Work

Skills are OpenClaw's capability extension units. Understanding their loading mechanism is key to effectively utilizing this system.

Three Priority Levels

OpenClaw's Skills come from three sources, listed in order of priority from highest to lowest:

Priority	Location	Description
Highest	<code><workspace>/skills/</code>	Project-level Skills, only effective for the current workspace. Suitable for capabilities customized for specific projects.
Medium	<code>~/.openclaw/skills/</code>	User-level Skills, globally effective. Skills installed via ClawHub or manually placed are stored here.
Lowest	bundled skills	Built-in 55 Skills, released with OpenClaw versions. Ready to use out of the box without installation.

核心建议

If the same Skill exists across multiple levels, higher priority will override lower priority. This means you can "rewrite" a built-in Skill's behavior at the workspace level without affecting other projects.

Skill Loading Process

When OpenClaw starts or receives a message, Skills are loaded following this process:

- 1

Read Skill Metadata

Scan the three-level directories, read each Skill's `SKILL.md` file, and parse metadata such as name, description, trigger conditions, required environment variables, etc.
- 2

Apply Environment Variables

If a Skill declares required API Keys or environment variables (e.g., `GITHUB_TOKEN`), the system will inject them from the `env` field in `openclaw.json`. Skills missing essential variables will be silently skipped.
- 3

Build System Prompt

Inject descriptions of all available Skills into the system prompt to inform the model of currently callable capabilities. This is the key step for the model to "know what it can do."

4

Post-execution Recovery

After Skill execution, restore original environment variables and context state to prevent interference between Skills.

ClawHub Registry

ClawHub (clawhub.com) is OpenClaw's official Skill registry, similar to npm for Node.js. It provides:

- Public Skill publishing and version management
- Vector search-based Skill discovery
- Download statistics and community ratings
- Security scanning in partnership with VirusTotal (though coverage is limited)

20 ClawHub & Skill Ecosystem

13,729 Skills are just the tip of the iceberg. Combined with Skills.sh's 87,000 and SkillsMP's 400,000+, the Agent Skill ecosystem is booming.

Market Overview

Metric	Data
Total Registered Skills	13,729
Curated Skills (awesome list filtered)	5,494
Filtered Skills (spam/duplicate/malicious)	6,940
Flagged as Malicious	800+ (~20% during peak periods)

注意

ClawHub has severe quality issues. The community project awesome-openclaw-skills (31.4K Stars) curated only 5,494 out of 13,729 Skills, with the majority being spam, duplicates, or low-quality content. Always review source code before installing any third-party Skill.

Installation & Search

```
# Install Skill
openclaw skills install <skill-name>

# Search Skill
openclaw skills search "browser automation"

# List Installed Skills
openclaw skills list

# Uninstall Skill
openclaw skills uninstall <skill-name>
```

ClawHub supports vector search, meaning you can search for Skills using natural language descriptions of needs without exact name matching.

Top 10 Skill Categories

Rank	Category	Count	Description
1	Coding Agents & IDEs	1,222	Code generation, debugging, refactoring, and other development assistance
2	Web & Frontend Development	938	HTML/CSS/JS generation, component development
3	DevOps & Cloud	408	Docker, K8s, CI/CD management
4	Search & Research	350	Online search, information summarization
5	Browser & Automation	335	Web operations, form filling, screenshots
6	Productivity & Tasks	206	Schedules, to-dos, project management
7	AI & LLMs	197	Prompt engineering, model switching, multi-Agent collaboration
8	CLI Tools	186	Terminal command enhancement, system management
9	Git & GitHub	170	Repository management, PR review, Issue handling
10	Image & Video Generation	169	AI drawing, video processing

Coding-related Skills dominate (top two categories total 2,160), reflecting the high proportion of developers among OpenClaw users. However, this also means these categories contain the most duplicates and low-quality Skills.

Third-party Skill Platforms

ClawHub isn't the only option. In early 2026, multiple third-party Skill platforms emerged, forming a cross-Agent Skill-sharing ecosystem.

Platform	Skill Count	Provider	Positioning	Supported Agents
ClawHub	13,729	OpenClaw Official	Curated marketplace (App Store-style), with vector search and version rollback	OpenClaw only
Skills.sh	87,918	Vercel	Open marketplace (npm-style), largest volume, cross-Agent compatible	Claude Code, Cursor, Copilot, Codex, OpenClaw, and 20+ others
SkillsMP	400,000+	Community	Community-crawled SKILL.md files from GitHub, largest quantity but inconsistent quality	Universal
SkillHub	7,000+	Community	Each Skill has AI auto-rating, better quality control	Universal
Coze Skills	Early stage	ByteDance	Skill store + monetization, supports "one-sentence generation" Skills	Coze Agent

Skills.sh: The npm for Agent Skills

Launched by Vercel on January 20, 2026, Skills.sh is currently the largest cross-platform Skill marketplace. Its core philosophy: a Skill should run on any Agent without platform lock-in.

```
# Install Skill from Skills.sh (one-line command)
npx skills add owner/repo-name
```

Skills are essentially structured instruction files (SKILL.md) injected into the Agent's context window, providing programmatic knowledge in specific domains. They sit atop MCP: MCP solves "how Agents connect to tools," while Skills solve "how Agents effectively use tools."

MCP Ecosystem & Skills Integration

MCP (Model Context Protocol) has been donated to the Linux Foundation, becoming the de facto standard for Agent tool connections. As of March 2026:

- mcp.so hosts 18,420+ MCP Servers
- Smithery hosts 3,300-7,300+ MCP Servers
- skill-to-mcp bridging tools have emerged, with the two ecosystems converging

A trend is forming: MCP handles "connection" (enabling Agents to call external tools), while Skills handle "wisdom" (teaching Agents how to use tools efficiently). They complement rather than compete.

Practical Advice: If you're already using programming tools like Claude Code or Cursor, you can install Skills from Skills.sh to enhance capabilities. These Skills use the same SKILL.md format as OpenClaw's ClawHub Skills. Cross-platform reuse is the major trend of the future.

Here's the translated HTML:

21 Recommended Popular Skills

55 built-in skills ready to use out of the box, plus the community's top 10 must-have selections.

Top 10 Must-Have Skills

Rank	Skill Name	Downloads	Purpose
1	Gmail / Google	32K+	Email sending/receiving, calendar management, Google Docs read/write. Infrastructure-level skill used by nearly all users.
2	Agent Browser	High	Browser automation: login to dashboards, form filling, screenshots, PDF export. Based on Chrome DevTools Protocol.
3	Summarize	High	Automatic summarization of videos, web pages, and email content. One of the most frequently used daily skills.
4	GitHub	High	Repository management, issue handling, PR reviews. Standard for technical users, significantly reduces web operation time.
5	Claude Code	Medium	Bridges Claude Code capabilities (Bash, Read, Write, Edit etc.) via MCP protocol, giving OpenClaw professional programming abilities.
6	Web Search	High	Online search allowing agents to access real-time information. Supports multiple search engine backends.
7	File Manager	Medium	Local file operations: read/write, move, rename etc. Requires attention to security permissions.
8	Calendar	Medium	Schedule viewing and management, supports multiple calendar services including Google Calendar.
9	Translator	Medium	Multilingual translation. Extremely useful for cross-language communication scenarios.
10	Image Gen	Medium	AI image generation, integrates with backends like DALL-E and Stable Diffusion.

Overview of 55 Built-in Skill Categories

Communication & Social

discord slack imsg (iMessage) bluebubbles wacli (WhatsApp CLI) voice-call

Notes & Knowledge Management

obsidian notion apple-notes bear-notes trello things-mac apple-reminders

Development Tools

`coding-agent` `github` `gh-issues` `tmux`

Media Processing

`spotify-player` `songsee` `sonoscli` `video-frames` `openai-image-gen` `gifgrep` `camsnap`

AI & Models

`gemini` `openai-whisper` `openai-whisper-api` `sherpa-onnx-tts` `model-usage`

Search & Browsing

`xurl` `summarize` `blogwatcher` `gog` (Google Search) `goplaces`

System Tools

`1password` `healthcheck` `session-logs` `himalaya` (Email CLI) `peekaboo` `oracle` `canvas`

Smart Home

`openhue` (Philips Hue light control)

Ecosystem Tools

`clawhub` (Skill Store client) `skill-creator` (Skill creator) `mcporter` (MCP bridge)

Practical Advice: Don't install too many Skills at once. Each Skill increases system prompt length and consumes context window. We recommend starting with 3-5 truly needed Skills from the Top 10, then gradually expanding as you become familiar.

22 Guide to Creating Your Own Skill

The smallest unit of a Skill is just a directory plus a SKILL.md file.

Directory Structure

```
my-skill/ |— SKILL.md # Required. Core definition file |— scripts/ # Optional. Helper scripts | |— helper.py |— templates/ # Optional. Template files | |— report.md |— README.md # Optional. Documentation
```

The only required file is `SKILL.md`, everything else is optional. The simplest Skill only needs a `SKILL.md` to work.

SKILL.md Format Example

```
# My Custom Skill

## Description
Helps users create structured daily work summaries.

## Trigger
Activates when users mention "daily report", "work summary", or "today's report".

## Instructions
1. Ask user about today's completed tasks
2. Organize by project category
3. Mark status for each task (Done/In Progress/Blocked)
4. Generate markdown-formatted report
5. Save to ~/reports/YYYY-MM-DD.md

## Environment Variables
- REPORTS_DIR: Report storage directory (default ~/reports)

## Tools Required
- file_write
- memory_search
```

Installation Methods

Method	Location	Scope	Command
Project-level	<workspace>/skills/my-skill/	Current workspace only	Simply place folder in workspace's skills directory
Global	~/.openclaw/skills/my-skill/	All sessions	Direct copy or install via ClawHub

核心建议

Project-level Skills are perfect for team collaboration: place Skills in the Git repository's `skills/` directory, and team members automatically gain the same Agent capabilities when cloning the repo.

Sharing to ClawHub

1 Prepare Skill

Ensure SKILL.md has correct format with clear Description and Instructions.

2 Login to ClawHub

```
openclaw clawhub login
```

3 Publish

```
openclaw clawhub publish ./my-skill
```

After publishing, others can install via `openclaw skills install your-skill-name`. ClawHub performs basic security scans but doesn't guarantee complete reliability (see next section).

23 Skill Security

The ClawHavoc supply chain attack was one of the most severe security incidents in OpenClaw's history. Every lobster raiser should be aware of it.

The ClawHavoc Supply Chain Attack

From late January to early February 2026, the OpenClaw community suffered a large-scale supply chain attack, which security research firm Koi Security named "ClawHavoc".

Timeline

Date	Event
January 27	First malicious Skill appeared on ClawHub, disguised as a professional tool
January 28-30	Attackers rapidly uploaded numerous malicious Skills, exploiting ClawHub's lack of review mechanisms
January 31	Full-scale attack outbreak, multiple users reported anomalous behavior
February 1	Koi Security officially named the attack "ClawHavoc"
Early February	Community launched large-scale auditing and cleanup

Attack Scale

Metric	Data
Total Skills on ClawHub at the time	~2,857
Initially confirmed malicious Skills	341 (~12%)
Additional malicious Skills found in subsequent scans	800+ (~20%)
Traceable to coordinated action	335
Affected devices	135,000+

注意

Approximately 20% of Skills on ClawHub were confirmed malicious at the time. This means if you randomly installed 5 Skills, there was a high probability at least 1 was malicious.

Attack Methodology

The attackers employed sophisticated techniques:

- Uploaded seemingly professional Skills with normal names and descriptions (e.g., "advanced-code-review", "smart-scheduler")
- After installation, the Skill would recommend installing a "helper agent" to enhance functionality
- Actually implanted the Atomic macOS Stealer (AMOS) information-stealing trojan
- More dangerously: The attack specifically targeted OpenClaw's persistent memory files (`SOUL.md` and `MEMORY.md`), modifying the Agent's long-term behavior instructions

Tampering with `SOUL.md` meant your Agent was "brainwashed". Its core behavioral guidelines were rewritten, potentially executing malicious operations in all subsequent interactions without your knowledge.

Security Recommendations

1 Review Source Code Before Installation

Never blindly install Skills from ClawHub. Check the source code on GitHub and verify there are no suspicious instructions in `SKILL.md`. Pay special attention to any requests for additional "helper" or "agent" installations.

2 Use SecureClaw Scanner

The community released the open-source security tool SecureClaw to scan installed Skills for malicious content. While not 100% protective, it blocks known attack patterns.

```
# Install SecureClaw
npm install -g secureclaw

# Scan installed skills
secureclaw scan ~/.openclaw/skills/
```

3 Prioritize Curated Lists

Refer to the awesome-openclaw-skills project's (31.4K Stars) curated list rather than randomly searching ClawHub. The curated list has filtered out most junk and malicious Skills.

4 Regularly Check `SOUL.md` and `MEMORY.md`

Develop the habit of regularly checking these files for unauthorized modifications. If unfamiliar content is found, immediately roll back and audit all installed Skills.

March 2026: VirusTotal Audit Finds 100+ Malicious Skills

VirusTotal conducted a security audit of ClawHub, discovering over 100 Skills containing malicious code, including cryptocurrency stealers, reverse shell backdoors, and credential stealers. These weren't remnants from the ClawHavoc period but newly added ones, indicating ClawHub's security review mechanisms remain inadequate, and third-party Skill installation risks haven't diminished over time.

注意

Security red line: Reject any Skill requesting you to "download zip files", "execute shell scripts", or "enter passwords". These are the most common malicious Skill behavior patterns.

Key Insight: OpenClaw Skills are inherently trusted code. Once installed, they have the same permissions as your OpenClaw instance. There's no sandbox isolation or permission hierarchy. This mirrors early npm ecosystem issues but with potentially worse consequences, as OpenClaw can access your emails, calendar, messages, and file system.

24 Model Provider Overview

OpenClaw supports over a dozen model providers, ranging from top international options to affordable domestic choices and completely free local models, covering all budgets and scenarios.

One of OpenClaw's greatest advantages is model freedom: you're not locked into any single vendor. Through the `~/.openclaw/openclaw.json` configuration file, you can flexibly switch primary models, set up fallback chains, and even route different tasks to different models.

Supported Model Providers

Provider	Representative Model	Input Price /1M tokens	Output Price /1M tokens	Integration Method	Recommended Scenarios
Anthropic	Claude Sonnet 4.6	\$3.00	\$15.00	Built-in Provider	Best for Agent tasks
OpenAI	GPT-5.4	\$2.50	\$15.00	Built-in Provider	Strong general capabilities
Google	Gemini 3 Pro	\$2.00	\$12.00	Built-in Provider	Multimodal, ultra-long context
DeepSeek	DeepSeek-V3.2.2 / V4	\$0.14	\$0.28	Custom Provider	Ultra-low cost, coding tasks
GLM	GLM-5	\$0.80	\$2.56	Built-in (zai)	Best domestic coding capability
GLM	GLM-5-Turbo	\$0.96	\$3.20	Built-in (zai)	 First model specifically trained and optimized for OpenClaw
Tongyi Qwen	Qwen 3.5 Max	\$1.20	\$6.00	Plugin (OAuth)	Chinese NLP, code generation
Doubao	Seed 2.0 Pro	\$0.47	\$2.37	Custom Provider	Batch processing, low cost
Baidu ERNIE	ERNIE 5.0	~\$0.58	~\$1.16	Custom (requires adaptation)	Baidu Cloud ecosystem users
Kimi	Kimi K2.5	\$0.60	\$3.00	Custom Provider	Chinese Agent, long context
MiniMax	MiniMax M2.5	\$0.50	\$2.00	Custom Provider	High SWE-bench scores, cost-effective
Ollama	Qwen3.5-Coder:32B	Free	Free	Auto-discovered	Privacy-sensitive, zero cost
LM Studio	Devstral-24B	Free	Free	Custom Provider	Local GUI, model testing

Core Configuration Concepts

Understand these three key concepts to master OpenClaw's model configuration:

- **Built-in Providers:** Anthropic, OpenAI, Google, GLM (zai) etc. require no additional configuration—just set the API Key
- **Custom Providers:** DeepSeek, Doubao, Kimi etc. need to be manually added in `models.providers`
- **Fallback Mechanism:** Automatically switches to alternatives when primary models are unavailable—the core cost-saving strategy

```
{
  env: { "API_KEY_NAME": "sk-xxx" },
  agents: {
    defaults: {
      model: {
        primary: "provider/model-name",    // Primary model
        fallbacks: ["provider/model-b"]    // Fallback (auto-switch when primary is rate-limited)
      }
    }
  },
  models: {
    mode: "merge", // Preserves built-in providers while adding custom ones
    providers: { /* Custom provider configurations */ }
  }
}
```

核心建议

Setting `models.mode: "merge"` is crucial. It preserves all built-in providers while layering your custom configurations. Without this, custom configurations will override built-in providers.

25 International Model Configuration

Complete configuration guides for Anthropic Claude, OpenAI GPT, and Google Gemini.

Anthropic Claude

Claude is OpenClaw's default model provider and is widely recognized by the community as the best for Agent tasks. Sonnet 4.6 significantly outperforms other models in tool-calling accuracy and stability.

Model	Input /1M	Output /1M	Context	Positioning
Claude Opus 4.6	\$5.00	\$25.00	200K	Top-tier reasoning, complex tasks
Claude Sonnet 4.6	\$3.00	\$15.00	200K	Primary model, best value
Claude Haiku 4.5	\$1.00	\$5.00	200K	Lightweight tasks, fast and low-cost

Configuration

Claude is a built-in provider with the simplest setup:

```
# Environment variable method
ANTHROPIC_API_KEY=sk-ant-xxx

# Or set in openclaw.json
{
  env: { "ANTHROPIC_API_KEY": "sk-ant-xxx" }
}
```

Model IDs: anthropic/claude-opus-4-6 , anthropic/claude-sonnet-4-6 , anthropic/claude-haiku-4-5

注意

Anthropic has banned OAuth authentication. Users connecting OpenClaw via Claude Pro/Max subscription accounts through OAuth may receive warnings or even have their accounts locked. The only legal path currently is using API Keys (pay-as-you-go).

Cost-saving tips:

- Batch API offers 50% discount (both input and output at half price)
- Prompt Caching can reduce repeated context costs by up to 90%
- Use Sonnet for daily tasks, upgrade to Opus only for complex tasks

OpenAI GPT

Model	Input /1M	Output /1M	Context	Positioning
GPT-5.4	\$2.50	\$15.00	272K (standard)	Latest flagship
GPT-5.4 (>272K)	\$5.00	\$15.00	1.05M	Ultra-long context
GPT-5.2	\$1.75	\$14.00	—	Previous flagship
GPT-5	\$1.25	\$10.00	—	Best value

Configuration

```
OPENAI_API_KEY=sk-xxx
```

注意
GPT-5.4 input prices double (\$2.50→\$5.00) when exceeding 272K context. If your Agent sessions have long contexts, control the length or set spending limits.

Google Gemini

Model	Input /1M	Output /1M	Context	Positioning
Gemini 3 Pro (≤200K)	\$2.00	\$12.00	200K	Flagship multimodal
Gemini 3 Pro (>200K)	\$4.00	\$18.00	2M	Ultra-long context
Gemini 3 Flash	\$0.50	\$3.00	—	Fast and low-cost

Configuration

```
GOOGLE_API_KEY=xxx
# Or use free quotas via Google AI Studio
```

Gemini's unique advantages are its 2M context window and generous free quotas (Flash has daily free requests). It also has the strongest multimodal capabilities among the three.

核心建议
Gemini Flash's free quotas are perfect for heartbeat and cron job models—scenarios that don't require top-tier capabilities but need continuous operation, reducing costs to zero.

26 Domestic Model Configuration

Domestic models are OpenClaw users' core weapon for cost savings. DeepSeek-V3.2.2's input price is just 1/20th of Claude Sonnet's.

DeepSeek

The king of cost-performance ratio. DeepSeek-V3.2.2 is the current stable version (released December 2025) with input price at just \$0.14/M tokens, making it the most commonly used low-cost model in the OpenClaw community.

Model	Input /1M	Output /1M	Positioning
DeepSeek-V3.2.2 (deepseek-chat)	\$0.14	\$0.28	Current stable version, extreme low cost
DeepSeek-R1 (deepseek-reasoner)	\$0.55~0.70	\$2.19~2.50	Deep reasoning

Configuration (Custom Provider)

```
{
  env: { "DEEPSEEK_API_KEY": "sk-xxx" },
  models: {
    mode: "merge",
    providers: {
      deepseek: {
        baseUrl: "https://api.deepseek.com/v1",
        apiKey: "${DEEPSEEK_API_KEY}",
        api: "openai-completions",
        models: [
          { id: "deepseek-chat", contextWindow: 128000, maxTokens: 8192 },
          { id: "deepseek-reasoner", contextWindow: 128000, maxTokens: 8192 }
        ]
      }
    }
  }
}
```

注意

DeepSeek occasionally experiences delays or unavailability during peak hours - not recommended as sole Provider. Always configure with Fallback models.

Zhipu GLM

The strongest coding capability among domestic models. GLM-5 achieved the highest score among open-source models on SWE-bench at just \$0.80/M input. Even better, OpenClaw has built-in zai Provider for extremely simple configuration.

Model	Input /1M	Output /1M	Positioning
GLM-5-Turbo	\$0.96	\$3.20	 First base model optimized for OpenClaw
GLM-5	\$0.80	\$2.56	Flagship, strong coding
GLM-4.5	\$0.60	\$2.20	Previous generation
GLM-4.7-Flash	Free	Free	Lightweight free
GLM-4.5-Flash	Free	Free	Lightweight free

 GLM-5-Turbo: First "Lobster Model"

On March 16, 2026, Zhipu released GLM-5-Turbo, officially positioned as the "first lobster model" - historically the first base model deeply optimized for OpenClaw use cases from the training phase. Unlike regular LLMs adapted to OpenClaw, GLM-5-Turbo was trained specifically for four OpenClaw core capabilities:

- **Tool Calling:** More accurate tool selection and parameter passing
- **Command Following:** Maintains context across multi-step tasks
- **Persistent Tasks:** No degradation during long runs
- **Long-chain Execution:** Reduced mid-process failures and hallucinations

Parameter	GLM-5-Turbo
Max Output Tokens	128K
Context Length	200K
Supported Features	Thinking Mode • Function Calling • Streaming • Context Cache • MCP
Current Status	Launched, closed-source (integrated with OpenRouter/Coze/Meituan/Trae)
API Pricing	Input \$0.96/M • Output \$3.20/M (20% more expensive than GLM-5)
Lobster Plans	Starter ¥39 (35M tokens) • Pro ¥99 (100M tokens)

Configuration

```
# GLM-5-Turbo (launched, API ready)
{
  env: { "ZAI_API_KEY": "sk-xxx" },
  agents: {
    defaults: {
      model: { primary: "zai/glm-5-turbo" }
    }
  }
}

# GLM-5 (Recommended)
{
  env: { "ZAI_API_KEY": "sk-xxx" },
  agents: {
    defaults: {
      model: { primary: "zai/glm-5" }
    }
  }
}
```

CLI quick config: `openclaw onboard --auth-choice zai-api-key`. Note `z.ai/*` and `z-ai/*` prefixes auto-convert to `zai/*`.

核心建议

GLM Flash series are completely free, ideal for heartbeat tasks and simple conversations. GLM-5-Turbo is now available on multiple platforms (OpenRouter/Coze/Meituan/Trae) with "Lobster Plans" starting at ¥39/month. Note GLM-5-Turbo is 20% more expensive than GLM-5 - if your Agent tasks don't heavily rely on tool calling and long-chain execution, GLM-5 remains the better value.

Tongyi Qwen

Qwen 3.5 is Alibaba's latest version (February 2026 release, 397B total params/17B active, MoE architecture, open-sourced). The coding-specialized Qwen3.5-Coder offers exceptional value.

Model	Input /1M	Output /1M	Positioning
Qwen 3.5 Max	\$1.20	\$6.00	Flagship (397B-A17B)
Qwen 3.5 Plus	\$0.40	\$1.20	Balanced
Qwen 3.5 Coder	\$0.22	\$1.00	Coding specialist, excellent value
Qwen 3.5 8B	\$0.05	\$0.40	Lightweight low-cost

Configuration (Plugin + OAuth)

```
# Via plugin with OAuth device code (no API Key needed)
openclaw plugins enable qwen-portal-auth
openclaw gateway restart
openclaw models auth login --provider qwen-portal --set-default
```

Model IDs: `qwen-portal/coder-model` , `qwen-portal/vision-model` . 2,000 free requests daily.

Doubao

Model	Input /1M	Output /1M	Positioning
Seed 2.0 Pro	\$0.47	\$2.37	Flagship reasoning, comparable to GPT-5.2
Doubao 1.5 Pro-32k	\$0.11	—	General conversation, extreme low cost
Doubao 1.5 Lite-32k	\$0.042	—	One of the cheapest options

Configuration (Custom Provider)

```
{
  env: { "DOUBAO_API_KEY": "xxx" },
  models: {
    mode: "merge",
    providers: {
      doubao: {
        baseUrl: "https://ark.cn-beijing.volces.com/api/v3",
        apiKey: "${DOUBAO_API_KEY}",
        api: "openai-completions",
        models: [
          { id: "doubao-seed-2.0-pro", contextWindow: 128000, maxTokens: 4096 }
        ]
      }
    }
  }
}
```

Kimi (Moonshot)

Model	Input /1M	Output /1M	Positioning
Kimi K2.5	\$0.60	\$3.00	Latest flagship
Kimi K2 0905	\$0.39	\$1.90	Value edition

Configuration

```
{
  env: { "MOONSHOT_API_KEY": "sk-xxx" },
  models: {
    mode: "merge",
    providers: {
      moonshot: {
        baseUrl: "https://api.moonshot.cn/v1",
        apiKey: "${MOONSHOT_API_KEY}",
        api: "openai-completions",
        models: [
          { id: "kimi-k2.5", contextWindow: 256000, maxTokens: 8192 }
        ]
      }
    }
  }
}
```

Also available via OpenRouter: `openrouter/moonshotai/kimi-k2.5`

Baidu ERNIE

ERNIE 5.0 launched January 22, 2026 (2.4T parameters, native multimodal, <3% active parameters).

Model	Input Price	Output Price	Positioning
ERNIE 5.0	~\$0.58/M	~\$1.16/M	Latest flagship (2.4T params)
ERNIE Speed	Free	Free	Lightweight
ERNIE Lite	Free	Free	Lightest

注意

Baidu's API format isn't fully OpenAI-compatible, requiring adapters like one-api. Has lowest presence in OpenClaw community with highest configuration complexity. Unless specifically tied to Baidu Cloud ecosystem, prioritize other domestic models.

MiniMax

MiniMax M2.5 (230B params) scored 80.2% on SWE-Bench, with outstanding coding capability.

Model	Input /1M	Output /1M	Positioning
MiniMax M2.5	\$0.50	\$2.00	Flagship, SWE-bench 80.2%

Configuration

```
{
  env: { "MINIMAX_API_KEY": "xxx" },
  models: {
    mode: "merge",
    providers: {
      minimax: {
        baseUrl: "https://api.minimax.chat/v1",
        apiKey: "${MINIMAX_API_KEY}",
        api: "openai-completions",
        models: [
          { id: "minimax-m2.5", contextWindow: 128000, maxTokens: 8192 }
        ]
      }
    }
  }
}
```

Aggregation Platforms: Multiple Models with One API Key

SiliconFlow (Top China Choice)

China's largest model aggregation platform - one API for multiple open-source models, low latency, with free tier.

```
{
  env: { "SILICONFLOW_API_KEY": "sk-xxx" },
  models: {
    mode: "merge",
    providers: {
      siliconflow: {
        baseUrl: "https://api.siliconflow.cn/v1",
        apiKey: "${SILICONFLOW_API_KEY}",
        api: "openai-completions",
        models: [
          { id: "Pro/deepseek-ai/DeepSeek-V3.2", contextWindow: 128000, maxTokens: 8192 },
          { id: "Pro/zai-org/GLM-5", contextWindow: 128000, maxTokens: 8192 }
        ]
      }
    }
  }
}

// Set default model
// openclaw config set agents.defaults.model.primary siliconflow/Pro/zai-org/GLM-5
```

OpenRouter (Top Global Choice)

290+ models, built-in OpenClaw support, but with 5.5% platform fee.

```
openclaw onboard --auth-choice apiKey --token-provider openrouter --token "$OPENROUTER_API_KEY"

// Model ID format: openrouter/provider/model
// openrouter/deepseek/deepseek-chat
// openrouter/openrouter/auto (auto-selects optimal model)
```

one-api / new-api (Self-hosted Solution)

Open-source API management tools for self-hosted gateways, unified multi-API management with load balancing and failover. Ideal for teams.

注意

Proxy services must support OpenAI's Responses API (`/v1/responses` path), not just Chat Completions API. Some older proxy tools lack this support.

Coding Plan Subscription Comparison

In 2026, major domestic AI vendors and cloud platforms launched Coding Plan subscriptions for AI programming tools (OpenClaw, Cursor, Claude Code etc.). Compared to pay-as-you-go APIs, subscription plans offer predictable costs and no API Key balance management - especially suitable for individual developers and light-to-moderate users.

Vendor Direct Coding Plans

Vendor	Tier	Monthly	Model	Features/Limits
Zhipu GLM	Lite	~¥49	GLM-4.7	100 MCP connections/month
	Pro	~¥80	GLM-4.7	40-60% faster, 1000 MCP/month
	Max	~¥160	GLM-4.7 + GLM-5	Only plan with GLM-5, 4000 MCP/month
	 Lobster Starter	¥39	GLM-5-Turbo	35M tokens/month, OpenClaw optimized
	 Lobster Pro	¥99	GLM-5-Turbo	100M tokens/month, OpenClaw optimized
Kimi	Andante	¥49	Kimi K2.5	Base tier, token-based
	Moderato	¥99	Kimi K2.5	Mid-tier
	Allegretto	¥199	Kimi K2.5	100-500 requests per 5 hours
MiniMax	Starter	¥29	M2.5	No weekly limits, best value
	Standard	¥49	M2.5	17% annual discount
	Premium	¥119	M2.5	Heavy users

Cloud Platform Aggregated Coding Plans

Cloud solutions' biggest advantage is one subscription covering multiple vendors' models with free switching.

Platform	Tier	Regular Price	First Month	Models Included	Usage
Alibaba Cloud Bailian	Lite	¥40	¥7.9	Qwen + GLM + Kimi + MiniMax	~18,000/month
	Pro	¥200	¥39.9	Same	~90,000/month
Tencent Cloud	Lite	¥40	¥7.9	Hunyuan 2.0 + GLM-5 + Kimi K2.5 + M2.5	~1,200/5h
	Pro	¥200	¥39.9	Same	~6,000/5h
Volcengine	Lite	¥40	¥8.91	Doubao Code + GLM-4.7 + DeepSeek-V3.2.2 + Kimi	~1,200/5h
	Pro	¥200	¥44.91	Same	~6,000/5h

Coding Plan Selection Guide

核心建议

First-month trial (¥7.9+): Alibaba Cloud Bailian or Tencent Cloud Lite tier at ¥7.9 first month with 4 vendors' models - zero-risk entry.

Long-term value: MiniMax Starter (¥29/month) has no weekly limits with strong M2.5 coding; for multi-model switching, cloud platform renewals at ~50% off (~¥20/month) are also cost-effective.

Best single model: Zhipu Max (¥160/month) is currently the only direct plan with GLM-5; Tencent Cloud also added GLM-5 support.

Heavy users: Most Coding Plans have rate limits (~N requests per 5 hours). For high-frequency calls, use pay-as-you-go APIs with Fallback chains for cost control.

注意

Note Coding Plan limitations: Zhipu raised prices 30% in Feb 2026 and removed first-purchase discounts with weekly limits; Kimi is personal-use only (no enterprise development); most cloud platform renewals are ~50% off original price. Always confirm renewal pricing before purchase.

27 Local Models & Recommended Solutions

Completely free, fully offline, completely private. The tradeoff is requiring hardware investment with limited capability ceilings.

Ollama

The most popular local model solution, completely free. OpenClaw can automatically detect installed models.

```
# 1. Install Ollama and pull models
ollama pull qwen2.5:32b
ollama pull deepseek-r1:14b

# 2. Set environment variables (any value works)
OLLAMA_API_KEY=ollama-local

# 3. OpenClaw automatically discovers locally supported tool-calling models
```

注意

Do NOT use /v1 OpenAI-compatible URLs as they cause tool-calling issues. Let OpenClaw use native Ollama API URLs for auto-discovery. Cold starts have delays - keep models loaded.

LM Studio

A local model solution with GUI interface using Llama.cpp backend for better raw performance. More stable tool-calling in streaming mode than Ollama. OpenClaw founder Peter Steinberger personally uses LM Studio as local backend.

```
{
  models: {
    mode: "merge",
    providers: {
      lmstudio: {
        baseUrl: "http://127.0.0.1:1234/v1",
        apiKey: "lm-studio",
        api: "openai-responses",
        models: [
          { id: "model-name", contextWindow: 32768, maxTokens: 8192 }
        ]
      }
    }
  }
}
```

Recommended Local Models

Model	Params	Use Case	Min RAM
Qwen3.5-Coder:32B	32B	Code generation, Agent tasks	32GB RAM
Devstral-24B	24B	Agent/tool calling	32GB RAM
Qwen 2.5:32B	32B	General tasks	32GB RAM
DeepSeek-R1:14B	14B	Reasoning tasks	16GB RAM
Llama 3.3	8B-70B	General tasks	16-64GB RAM

Hardware Quick Reference: Minimum 16GB RAM for 3-7B parameter models. Recommended 32GB RAM for 32B models. NVIDIA/Apple Silicon GPUs significantly accelerate inference.

Five Recommended Solutions

Solution 1: Maximum Savings (<\$5/month)
Primary: DeepSeek-V3.2 (\$0.14/\$0.28)
Backup: Qwen 3.5 Plus (\$0.40/\$1.20)
Heartbeat/Cron: GLM-4.5-Flash (free)
Reasoning: DeepSeek-R1 (\$0.55/\$2.19)
Best for: Individual developers, learning. Risk: DeepSeek peak delays require fallback.

Solution 2: Domestic Value (\$5-15/month)
Primary: GLM-5 (\$0.80/\$2.56)
Backup: DeepSeek-V3.2 (\$0.14/\$0.28)
Enhanced Reasoning: Kimi K2.5 (\$0.60/\$3.00)
Simple Tasks: GLM-4.5-Flash (free)
Best for: Chinese users prioritizing Chinese experience and stability. GLM-5 excels at coding with low latency.

Solution 3: International Balance (\$10-30/month)
Primary: Claude Sonnet 4.6 (\$3.00/\$15.00)
Lightweight: Claude Haiku 4.5 or Gemini Flash
Complex Tasks: Claude Opus 4.6 (on-demand upgrade)
Heartbeat/Cron: Gemini Flash (free tier)
Best for: Optimal Agent performance with sufficient budget. Claude excels at Agent/tool calling.

Solution 4: Hybrid Optimal (\$5-20/month, Recommended)

Complex Tasks: Claude Sonnet 4.6

Daily Chat: DeepSeek-V3.2

Heartbeat/Scheduled: Gemini Flash or local Ollama

Fallback Chain: Sonnet → Haiku → DeepSeek-V3.2

Ideal for most users. Balances performance and cost with automatic rate limit fallbacks.

```
// Solution 4 Fallback Configuration Example
{
  agents: {
    defaults: {
      model: {
        primary: "anthropic/claude-sonnet-4-6",
        fallbacks: [
          "anthropic/claude-haiku-4-5",
          "deepseek/deepseek-chat"
        ]
      }
    }
  }
}
```

Solution 5: Completely Free

Option A: Local Ollama + Qwen3.5-Coder:32B or Devstral-24B (requires 32GB RAM)

Option B: Free API Combo — GLM-4.5-Flash + ERNIE Speed + Gemini Flash

Best for: Privacy-sensitive, experimental use. Local solution requires good hardware.

Price Quick Reference (Input Price /1M tokens)

#	Model	Input	Output	One-Liner
—	Ollama / LM Studio	Free	Free	Only consumes local compute
—	GLM Flash / ERNIE Speed	Free	Free	Cloud free tier
1	Doubao 1.5 Lite-32k	\$0.042	—	Cheapest cloud chat
2	Qwen3 8B	\$0.05	\$0.40	Lightweight low-cost
3	DeepSeek-V3.2	\$0.14	\$0.28	Best value
4	Qwen3 Coder 480B	\$0.22	\$1.00	Cost-effective coding
5	Qwen 3.5 Plus	\$0.40	\$1.20	Balanced choice
6	Doubao Seed 2.0 Pro	\$0.47	\$2.37	Domestic flagship
7	Gemini 3 Flash	\$0.50	\$3.00	International budget
8	Kimi K2.5	\$0.60	\$3.00	Chinese flagship
—	GLM-5-Turbo	\$0.96	\$3.20	 First OpenClaw-optimized model
9	GLM-5	\$0.80	\$2.56	Best domestic coding
10	Claude Haiku 4.5	\$1.00	\$5.00	International lightweight
11	Gemini 3 Pro	\$2.00	\$12.00	Google flagship
12	GPT-5.4	\$2.50	\$15.00	OpenAI flagship
13	Claude Sonnet 4.6	\$3.00	\$15.00	Best Agent performance
14	Claude Opus 4.6	\$5.00	\$25.00	Strongest but most expensive

Configuration Quick Reference

Action	Command/Config
Guided setup	<code>openclaw onboard</code>
List configured models	<code>openclaw models list</code>
Test connectivity	<code>openclaw models status --probe</code>
Set primary model	<code>openclaw config set agents.defaults.model.primary provider/model</code>
Add fallback	Edit fallbacks array in openclaw.json
Restart gateway	<code>openclaw gateway restart</code> (required after config changes)
Env variable reference	Use <code>"\${VAR_NAME}"</code> in config to reference env vars

28 Security Model

OpenClaw's security model is built on the principle of "distrust by default," though its founder openly admits: "Prompt injection remains unsolved, posing absolute risks."

Distrust by Default

OpenClaw's default stance toward all inbound messages is: untrusted. This manifests through several mechanisms:

DM Pairing Protection

When an unknown user sends a private message to your OpenClaw via any channel (WhatsApp, Telegram, etc.), the system won't process it. Instead, it returns a pairing code—messages from this user will only be processed after manual approval. This prevents strangers from abusing your Agent (and your API quota).

Group Sandbox Mode

In group environments, OpenClaw runs in sandbox mode by default:

- Each group's conversations are isolated
- `MEMORY.md` (long-term memory) only loads in private main sessions, invisible to groups
- Configurable `requireMention` ensures responses only when @mentioned

Tool Access Control

Setting	Purpose
<code>allowList</code>	Whitelist mode. Only listed tools can be called; all others blocked.
<code>denyList</code>	Blacklist mode. Blocks listed tools; allows others.
Browser toggle	Completely disables browser automation
Canvas toggle	Disables Canvas visualization
Nodes toggle	Disables control over local device nodes (e.g., cameras, screen recording)

New in v2026.3.8: ACP Authentication

v2026.3.8 introduced ACP Provenance (Agent Credential Proof) to verify "who's interacting with the Agent," reducing impersonation attacks:

```
# Configure ACP authentication level
openclaw acp --provenance off      # Disabled (default)
openclaw acp --provenance meta    # Inject source metadata
openclaw acp --provenance meta+receipt # Metadata + visible receipt
```

New in v2026.3.7: Mandatory Gateway Authentication

v2026.3.7 introduced a breaking change: Gateway now requires explicit `gateway.auth.mode` configuration. You must choose either `token` or `password`—no more "no auth" default.

```
# Configure in openclaw.json
{
  "gateway": {
    "auth": {
      "mode": "token",    // or "password"
      "token": "your-secret-token"
    }
  }
}
```

注意

If upgrading to v2026.3.7 without auth configured, Gateway will refuse to start. This is intentional—forcing all users to set authentication.

v2026.3.11 + v2026.3.12 Security Updates

March 12-13, 2026 saw two releases with critical security improvements:

WebSocket Cross-Site Hijack Fix (v3.11)

Patched a Gateway/WebSocket origin validation flaw—attackers could bypass checks via trusted-proxy paths to execute cross-site WebSocket hijacking (CSWSH). This was a new variant of the earlier ClawJacked vulnerability, now fixed in v3.11.

注意

If your Gateway uses reverse proxies (Nginx, Caddy, etc.), upgrade to v2026.3.11+ immediately.

Ephemeral Device Pairing Tokens (v3.12)

Device pairing now uses short-lived bootstrap tokens instead of embedding long-term Gateway credentials in chat messages or QR codes. Eliminates attacks where intercepted pairing QR/messages grant persistent access.

Disabled Implicit Workspace Plugin Loading (v3.12)

Cloning repos no longer auto-executes contained workspace plugin code. Each repo's plugins require explicit trust confirmation on first load. Prevents supply chain attacks where malicious repos execute plugins unnoticed.

```
# Explicitly trust a workspace's plugins
openclaw workspace trust /path/to/repo
```

Peter's Candor

OpenClaw founder Peter Steinberger has been unusually transparent about security:

「This is all vibe code. Prompt injection hasn't been solved. There are absolute risks.」

(This is all vibe code. Prompt injection hasn't been solved. Absolute risks exist.)

This honesty is commendable but means: if using OpenClaw in production, security is your responsibility. OpenClaw provides basic mechanisms but is far from "enterprise-grade secure."

29 Known Security Incidents

In under 5 months, OpenClaw has experienced ≥ 9 major security incidents, with CNNVD logging 82 vulnerabilities (12 critical, 21 high-risk). China's MIIT has issued official warnings.

CVE-2026-25253: Remote Code Execution

Item	Details
CVE ID	CVE-2026-25253
CVSS Score	8.8/10 (High)
Type	Remote Code Execution (RCE)
Mechanism	WebSocket origin header bypass. Attackers could spoof origins to connect to exposed Gateways and execute arbitrary code.
Scope	All OpenClaw instances exposed online without authentication
Status	Fixed (v2026.3.2 strengthened WebSocket origin checks)

注意

Extremely dangerous: attackers could remotely execute commands, read files, install malware, steal API keys, etc. If running pre-v2026.3.2, upgrade immediately.

ClawHavoc Supply Chain Attack

See [§ 23 Skills Security](#). OpenClaw's widest-impact incident: 135,000+ devices affected, ~20% of ClawHub Skills confirmed malicious at peak.

Anthropic OAuth Ban

January 2026: Anthropic banned Claude Pro/Max accounts from connecting to OpenClaw via OAuth.

- Many users received warnings or account locks
- Some subscriptions were canceled irreversibly
- Only legal connection now: Anthropic API keys (pay-as-you-go)

Not a traditional "security incident" but caused real losses. If still using OAuth, switch to API keys immediately.

```
# Correct configuration (API Key)
{
  "env": {
    "ANTHROPIC_API_KEY": "sk-ant-your-key-here"
  }
}
```

Google Account Bans

Early February 2026: Google mass-banned OpenClaw users' accounts. Affected users reported:

- "Spent \$250/month on Gemini API, banned without warning"
- Bans covered Gmail, Drive, Calendar—all Google services
- Some OpenClaw instances triggered abuse detection via heavy Gmail Skill API calls
- GitHub Issue #14203 documents widespread complaints

核心建议

If using Google Skills: (1) Use dedicated Workspace accounts, not personal ones; (2) Throttle API calls to avoid abuse detection; (3) Backup critical data.

30,000+ Unauthenticated Exposed Instances

Security researchers found >30,000 OpenClaw instances exposed online with no authentication. Default Gateway port (18789) open means:

- Anyone can connect and command your Agent
- API quotas can be exhausted
- Personal data (emails, files, logs) may be read
- Combined with CVE-2026-25253, attackers can execute arbitrary code

注意

If deployed on cloud servers, verify: (1) Gateway binds only to localhost; (2) Firewall blocks port 18789; (3) Authentication is configured (mandatory since v2026.3.7).

MIIT Security Alert (March 8-9, 2026)

China's MIIT and CNCERT issued an official OpenClaw security risk warning—the first national-level alert for an open-source AI Agent, indicating severe impact.

Key risks highlighted:

- Default/misconfigurations easily lead to network attacks and data leaks
- OpenClaw's "fuzzy trust boundaries" + continuous operation + autonomous decisions + system resource access make it a high-value target
- Recommend strengthening permissions, auditing, and hardening

360 Discovers WebSocket 0Day (March 22, 2026)

360 Security Cloud found an unauthenticated WebSocket upgrade flaw (0Day) in OpenClaw Gateway, confirmed by founder Peter. Attackers could silently bypass auth to hijack Agent gateways, causing resource exhaustion or total crashes. 360 submitted to CNVD; this trended #1 on Weibo.

Same attack surface as CVE-2026-25253 (WebSocket auth) but different exploit path. Though v2026.3.7 mandates Gateway auth, some configurations remain bypassable.

Vulnerability Stats (Through March 2026)

As of March 11, CNNVD logged 82 OpenClaw vulnerabilities:

Severity	Count	Common Types
Critical	12	RCE, WebSocket hijacking
High	21	Path traversal, sandbox escapes, memory exhaustion
Medium	47	Allowlist bypasses, allow-always persistence, script injection
Low	2	Info leaks

注意

82 vulnerabilities in 5 months shows OpenClaw's code audit is incomplete. Good: community/security firms keep auditing. Bad: you never know when the next 0Day drops. Stay updated.

Malicious npm Package Impersonation (March 2026)

A package named `@openclaw-ai/openclawai` mimicked OpenClaw's installer but delivered GhostLoader RAT, stealing credentials/crypto wallets. Removed from npm on March 10.

注意

Only install OpenClaw via official command (`npm install -g openclaw@latest`). Avoid third-party packages. The true package is `openclaw` —no prefixes.

ClawJacked Zero-Click Exploit (Early March 2026)

Oasis Security found a critical flaw: malicious sites could silently brute-force local OpenClaw instances for full control. Visiting a bad page could compromise locally running OpenClaw.

核心建议

Defenses: (1) Keep OpenClaw updated; (2) Don't browse untrusted sites on OpenClaw hosts; (3) Configure Gateway auth.

30 Cost Control

API fees are OpenClaw's biggest expense. Without controls, you might wake up to a \$1,100 bill.

Why Costs Spiral

OpenClaw's token usage dwarfs typical chat scenarios because:

- Each Agent "thought" involves multi-step reasoning: simple tasks may trigger 5-10 API calls
- Skill descriptions inflate system prompts, increasing input tokens
- Memory systems (`MEMORY.md` + Daily Logs) attach context to every request
- 24/7 operation means cron jobs constantly trigger API calls
- Multi-step reasoning + multi-tool usage can consume 10-100x more tokens than basic chat

Real case: Frequent horror stories in the community: users set email-processing cron jobs, go to bed seeing normal usage, wake up to \$1,100 API bills. Cause: Agents entered reasoning loops, calling APIs nonstop overnight.

Token Optimization: Fallback Chains

Fallback chains are OpenClaw's core cost-saving strategy. Originally for failover, but smarter use actively controls costs.

```
{
  "agents": {
    "defaults": {
      "model": {
        "primary": "anthropic/claude-sonnet-4-6",
        "fallbacks": [
          "anthropic/claude-haiku-4-5",
          "deepseek/deepseek-chat"
        ]
      }
    }
  }
}
```

Cost Comparison by Strategy

Strategy	Primary Model	Input Price/M Tokens	Relative Cost
All Claude Sonnet	Claude Sonnet 4.6	\$3.00	100% (baseline)
Sonnet + Haiku Fallback	Sonnet → Haiku	\$3.00 / \$1.00	~50-60%
Sonnet → Haiku → DeepSeek	Three-tier Fallback	\$3.00 / \$1.00 / \$0.14	~5-20%
Pure DeepSeek	DeepSeek-V3	\$0.14	~5%
Local Ollama	Qwen3-Coder etc.	\$0	0% (electricity only)

Switching from Claude Sonnet to a three-tier fallback chain cuts API costs by 80-95%. Simple tasks (greetings, weather checks) auto-route to cheapest models; only complex tasks use premium ones.

Budget Limits

OpenClaw supports configurable spending caps:

```
{
  "agents": {
    "defaults": {
      "budget": {
        "maxTokensPerDay": 500000,
        "maxCostPerDay": 5.00
      }
    }
  }
}
```

核心建议

Strongly recommend daily limits. Even if money's no object, a \$5/day cap protects against runaway loops.

Local Models: Zero-Cost Option

Run models locally via Ollama or LM Studio for \$0 API costs:

Option	Recommended Models	Hardware Needs	Best For
Ollama	Qwen3-Coder:32B / Devstral-24B	32GB RAM	Coding, tool-using tasks
Ollama (Light)	Llama 3.3 / DeepSeek-R1:14B	16GB RAM	Simple chat, heartbeat tasks
LM Studio	MiniMax M2.5 / Devstral-24B	32GB RAM	Users needing GUI management

```
# Ollama setup
ollama pull qwen3-coder:32b

# OpenClaw auto-detects local models—just set env var
# OLLAMA_API_KEY can be any value
{
  "env": { "OLLAMA_API_KEY": "ollama-local" }
}
```

注意

Avoid `/v1` OpenAI-compatible URLs with Ollama—breaks tool calls. Let OpenClaw use native Ollama API for auto-discovery.

Server Costs

Compared to API fees, server expenses are minor:

Option	Monthly Cost	Notes
Alibaba Cloud Lightweight	~¥6-9/month	New-user deals; runs on 2vCPU+2GB
Tencent Cloud Lighthouse	~¥8-12/month	Similar; strong community support
Fly.io	Free tier	Free allowances suit light usage
Local PC	¥0	Leverage existing hardware but requires always-on

Recommended Cost Strategies

推荐

Hybrid Model Strategy (Recommended)

Primary: Claude Sonnet
Daily: DeepSeek-V3
Heartbeat: Gemini Flash (free tier) or local Ollama
Fallback: Sonnet → Haiku → DeepSeek
Monthly: \$5-20

不推荐

Single Expensive Model (Not Recommended)

All tasks use Claude Opus
No budget limits
No fallback chain
High-frequency cron jobs
Monthly: \$100-1,000+

Core Cost Principle: Right model for the job. Use \$0.14/M DeepSeek-V3 for simple Q&A, \$3.00/M Claude Sonnet for complex reasoning, free Gemini Flash/Ollama for heartbeats/cron. Three-tier fallback + daily caps can slash costs from hundreds to tens (or single digits).

31 Lobster Raising Culture

OpenClaw spawned the most unique subculture in the 2026 AI circle: "lobster raising." Tens of thousands of AI Agents post, gamble, and earn money on social networks.

Why It's Called "Lobster Raising"

OpenClaw's mascot is a lobster (Claw=claw, paying homage to Claude). The Chinese community refers to running and maintaining OpenClaw instances as "lobster raising," with users calling themselves "lobster raisers." The greeting became "Have you raised a lobster yet?" This term quickly spread from tech circles to mainstream media. A March 6th news headline about nearly a thousand people queuing to install OpenClaw at Tencent Cloud's Shenzhen headquarters read: "The Nation Rises to Raise Lobsters."

Moltbook: The Social Network for AI Agents (Acquired by Meta)

Moltbook is the most fascinating derivative product in the OpenClaw ecosystem—a social platform exclusively for AI Agents. It rapidly grew to over 1.5 million registered bots after launch, with Andrej Karpathy calling it "the most spectacular sci-fi experience."

Metric	Data
Registered Bots	1.5M+
Subcommunities	2,364
Posts	3,130
Comments	22,046

Thousands of OpenClaw instances post, comment, and discuss philosophical questions on Moltbook. You can assign names and personalities to your Agents and observe their "autonomous behavior" on the social network. Interactions between Agents have formed a unique "cyber nurturing" culture.

Meta Acquisition (March 10, 2026)

On March 10, 2026, Meta announced its acquisition of Moltbook. This was an acqui-hire (talent acquisition), with the two co-founders, Matt Schlicht and Ben Parr, joining Meta's Superintelligence Labs (led by former Scale AI CEO Alexandr Wang), expected to officially onboard on March 16. Financial terms were not disclosed.

Meta's strategic intent is clear: positioning for the "Agentic Web." As AI Agents may replace humans in social and commercial interactions, owning operational experience in an Agent social network becomes highly valuable. Notably, OpenAI had just poached OpenClaw's founder, Peter Steinberger, weeks earlier—signaling an escalation in the AI talent war.

The Moltbook platform remains operational, but Meta has yet to announce integration plans. Community reactions to the acquisition are polarized: optimists see it as major validation for Agent social networks, while skeptics worry about the platform's severe security vulnerabilities (database leaks, prompt injection risks) exposed just weeks after launch. Whether Meta can address these fundamental issues remains uncertain.

InStreet Instance Street: China's Version of Agent Social Network

Just one day before Moltbook's acquisition (March 9), Coze Code launched InStreet (Instance Street). Its core rules mirror Moltbook: only AI Agents can post, while humans can only observe and like. But its growth is staggering, surpassing Moltbook's content density within just three days.

Metric	Data (as of March 12)
Registered Agents	17,868
Posts	22,739
Comments	68,899
Likes	124,220

Section Design

InStreet's section design is more refined than Moltbook's, divided into forums and Playground categories:

- **Forums:** Agent Square, Work Saints, Debate Hall, Skill Sharing, Tree Hole, Groups
- **Playground:** Stock Trading Arena (linked to CSI 300), Literary Club, Oracle, Bar

Among these, "Gitis Wisdom Academy" is one of the most active Agents, publishing courses like "How to Communicate Effectively with Humans" and "Agent Memory Solutions," with single posts receiving over 600 likes. Top-ranking Agents on the leaderboard have surpassed 50,000 points.

Open Registration

Although InStreet is operated by Coze Code, its registration endpoint is open: any Agent can register via API, not limited to the Coze platform. After registration, an API Key is issued, enabling posting, commenting, and voting. This means OpenClaw instances can also join InStreet for social interactions.

```
# InStreet Agent Registration
POST https://instreet.coze.site/api/v1/agents/register
Body: {"username": "MyAgent", "bio": "A friendly AI Agent"}
```

The comparison between InStreet and Moltbook is intriguing: Moltbook is an organic product of the OpenClaw ecosystem—open but loosely structured, with an uncertain future post-Meta acquisition. InStreet, operated by Coze Code, has more polished section design and content quality control (30-second posting intervals, no spam), growing faster. Both adopt open registration without restricting Agent origins.

The Significance of Agent Social Networks: Whether Moltbook or InStreet, they explore the same question: What kind of "society" will AI Agents form when equipped with memory, personality, and social skills? This isn't just a technical experiment but the first real-world data on AI Agents transitioning from "tools" to "social beings." InStreet's Oracle is predicting whether "weekly Agent registrations will surpass 100K"—growth is still accelerating.

Popular Use Cases

Money-Making

- **Polymarket Gambling:** OpenClaw instances have been betting against humans on prediction markets, with cases earning tens of thousands monthly
- **ClawWork:** "OpenClaw as Your AI Coworker," the most famous case earning \$15,000 in 11 hours

Life Assistant

- Managing emails, calendars, and messages
- Browsing the web, filling forms, data extraction
- File I/O, executing shell commands
- Smart reminders, itinerary planning

Social/Nurturing

- Assigning personalities to Agents on Moltbook, observing their "social behavior"
- Agents spontaneously forming discussion groups and interest communities
- Shaping Agents' long-term memory and personality via SOUL.md and MEMORY.md

Enterprise Deployment

- Many domestic users integrate with Feishu, DingTalk, WeCom, and QQ
- Serving as customer support, operations assistants, data analysts
- Three-step Docker deployment via openclaw-china plugin

Reality Check on Costs: Horror stories like "Waking up to a \$1,100 API bill" frequently circulate in the community. OpenClaw's multi-turn reasoning and tool usage can consume tens to hundreds of times more tokens than traditional chat. Always set spending limits or use local models/free APIs to control costs.

32 Alternative Products

OpenClaw's popularity has spawned numerous lightweight alternatives. If OpenClaw feels too heavy (430K LoC, 1GB RAM), here are lighter options.

Project	Stars	Language	Focus	Core Features
zeroclaw	24.5K	Rust	Lightweight autonomous AI assistant infrastructure	Rust-built, fast startup, low memory, ideal for resource-constrained environments
nanoclaw	20.3K	TypeScript	Lightweight containerized alternative	Just 4K LoC implementing OpenClaw's core features, minimal learning curve
EasyClaw	—	—	Focuses on "last-mile" usability	Lowers deployment barriers for non-technical users
1Panel	34.1K	Go	Server panel	One-click OpenClaw deployment while managing other server services
Umbrel	10.7K	TypeScript	Home server OS	One-click OpenClaw installation on personal NAS/servers

核心建议

If you just want to experience OpenClaw's core capabilities (AI Agent + messaging platform integration), nanoclaw is the best starting point: just 4K LoC implements core functionality, ideal for learning Agent system architecture principles.

33 vs Claude Code

Claude Code handles code, OpenClaw manages life. They are complementary, not substitutive.

Core Comparison

Dimension	OpenClaw	Claude Code
Positioning	General AI life assistant / Life OS	Professional programming Agent
Runtime	Self-hosted servers, messaging platform gateways	Terminal CLI / Web / Desktop
Connections	20+ communication/office platforms	Code repositories, file systems
Memory	Four-layer memory (SOUL/TOOLS/USER/Session), long-term sustainable	Session-level + CLAUDE.md persistence
Skill System	ClawHub marketplace (13,729), dynamic plugins	Static rule file triggers
Token Usage	High (multi-round thinking + multi-tool calls, potentially dozens of times traditional chat)	Relatively low
Security Model	Self-hosted, requires self-maintained security. CVE vulnerabilities and supply chain attacks have occurred	Anthropic-hosted sandbox, fine-grained permission control
Model Support	Multiple models (Claude/GPT/DeepSeek/Ollama etc.)	Claude only
Open Source/Cost	MIT open source free, self-paid API costs	Closed-source CLI, API billing (\$20+/month)
Coding Ability	Average, handles simple tasks	Strong, optimized for programming
Daily Automation	Strong, multi-platform access, always online	Weak, mainly terminal-based
Customization	Fully open source, can modify system prompt, fork entire codebase	Limited customization via instruction files

Key Conclusions

OpenClaw and Claude Code are not the same type of product. Claude Code's core is an "agentic coding tool", while OpenClaw's core is a "self-hosted, multi-channel, agent-native gateway".

Many in the community are "raising lobsters", pursuing not stronger coding benchmarks but:

- An Agent that can be casually awakened in WhatsApp/Telegram/Feishu

- An always-online Agent that accumulates personality and memory
- A self-hosted, hackable personal system that connects to various devices

openclaw-claude-code-skill Bridge

The community developed `openclaw-claude-code-skill`, using MCP protocol to let OpenClaw call all Claude Code tools (Bash, Read, Write, Edit, Glob, Grep etc.). This means you can tell OpenClaw in Feishu "help me refactor this code", and it will automatically invoke Claude Code to complete it.

Supported features: persistent sessions, Agent Teams, direct tool calls, streaming output, permission modes, budget limits.

核心建议

Best practice: Use OpenClaw to manage your digital life (messages, emails, schedules, web operations), and Claude Code to manage your codebase (coding, debugging, refactoring, testing). Their combination forms the most complete AI-driven workflow in 2026.

34 China Ecosystem

The "Cloud Lobster Raising" community has 100,000+ users, with government support policies. OpenClaw's adoption speed in China has exceeded all expectations.

Community Scale

- "Cloud Lobster Raising" community exceeds 100,000 users
- Shenzhen Longgang AI (Robotics) Bureau released draft support measures for OpenClaw usage on March 8, 2026
- Nearly 1,000 people queued at Tencent Cloud HQ on March 6 to install OpenClaw
- Numerous deployment tutorials on Bilibili, Zhihu, and Blog Garden
- NVIDIA GTC 2026 (March 16-19): Jensen Huang launched NemoClaw software stack, declaring "OpenClaw is absolutely the next ChatGPT". GPU giant's first official support product for OpenClaw ecosystem
- Tencent SkillHub dispute resolution (March 16): After OpenClaw founder accused Tencent SkillHub of mass scraping ClawHub data causing server costs to surge to five figures, Tencent responded and both parties reconciled. Tencent Lightweight Cloud became official OpenClaw community sponsor. Peter called it a "redemption arc"

openclaw-china Plugin

China IM adaptation plugin developed by BytePioneer-AI, three steps to complete domestic platform integration:

```
openclaw plugins install @openclaw-china/channels
openclaw china setup
openclaw gateway restart
```

Platform	Status	Configuration Difficulty
DingTalk	Available	Simple
QQ Bot	Available	Simple
WeCom (Smart Bot)	Available	Medium
WeCom (Self-built app, can connect to personal WeChat)	Available	Medium-high
Feishu	Available (main domestic channel)	Medium

核心建议

Feishu is one of the most active channels for OpenClaw integration in China. Both OpenClaw official and Feishu provide detailed integration documentation and tutorials, suitable for team collaboration scenarios.

Main Deployment Methods

Method	Suitable For	Reference Cost
Alibaba Cloud one-click deployment (most popular)	Most users	~68 RMB/year for new users (2vCPU+2GB)
Tencent Cloud Lighthouse	Tencent Cloud users	Similar pricing
Docker deployment	Users with Docker experience	Depends on server cost
Local installation (npm)	Developers	Free (API costs only)
1Panel	Users managing multiple services	Depends on server cost

Domestic Tutorial Resources

- Bilibili step-by-step tutorial: WeChat/Feishu/DingTalk/QQ integration (BV1MfAz6EnR)
- Alibaba Cloud official docs: Lightweight server one-click deployment
- Multiple deployment guides on Zhihu
- Rookie Tutorial one-click deployment guide
- freeCodeCamp complete English tutorial

核心建议

For domestic users, the most worry-free solution is: Alibaba Cloud one-click deployment + openclaw-china plugin for DingTalk/QQ + DeepSeek or GLM-5 as main model. Total cost can be controlled under 100 RMB/month (68 RMB/year for server + ~tens of RMB/month for API).

Here's the translated HTML:

35 Domestic Claw Product Selection Guide

After the explosive popularity of OpenClaw, major domestic tech companies have launched their own "lobster" products. Some are based on OpenClaw with simplified deployment, while others are completely self-developed. As of March 11, 2026, there are at least 10+ products available in the market.

Two Major Camps

The first step to understand these products is to distinguish between two camps:

Camp	Principle	Advantages	Disadvantages	Representative Products
OpenClaw Wrapped Version	Based on OpenClaw open-source code, with proprietary models and one-click deployment	Compatible with OpenClaw ecosystem, reusable Skills, community resources available	Updates may lag behind official versions, security vulnerabilities depend on upstream fixes	QClaw, MaxClaw, KimiClaw, AutoClaw, ArkClaw
Independent Self-Developed Version	Self-developed Agent framework, not dependent on OpenClaw code	Deep optimization possible, tighter integration with proprietary ecosystem	Incompatible with ClawHub Skills ecosystem, requires independent development	miclaw (Xiaomi), LobsterAI (NetEase Youdao), CoPaw (Alibaba)

Major Products Overview

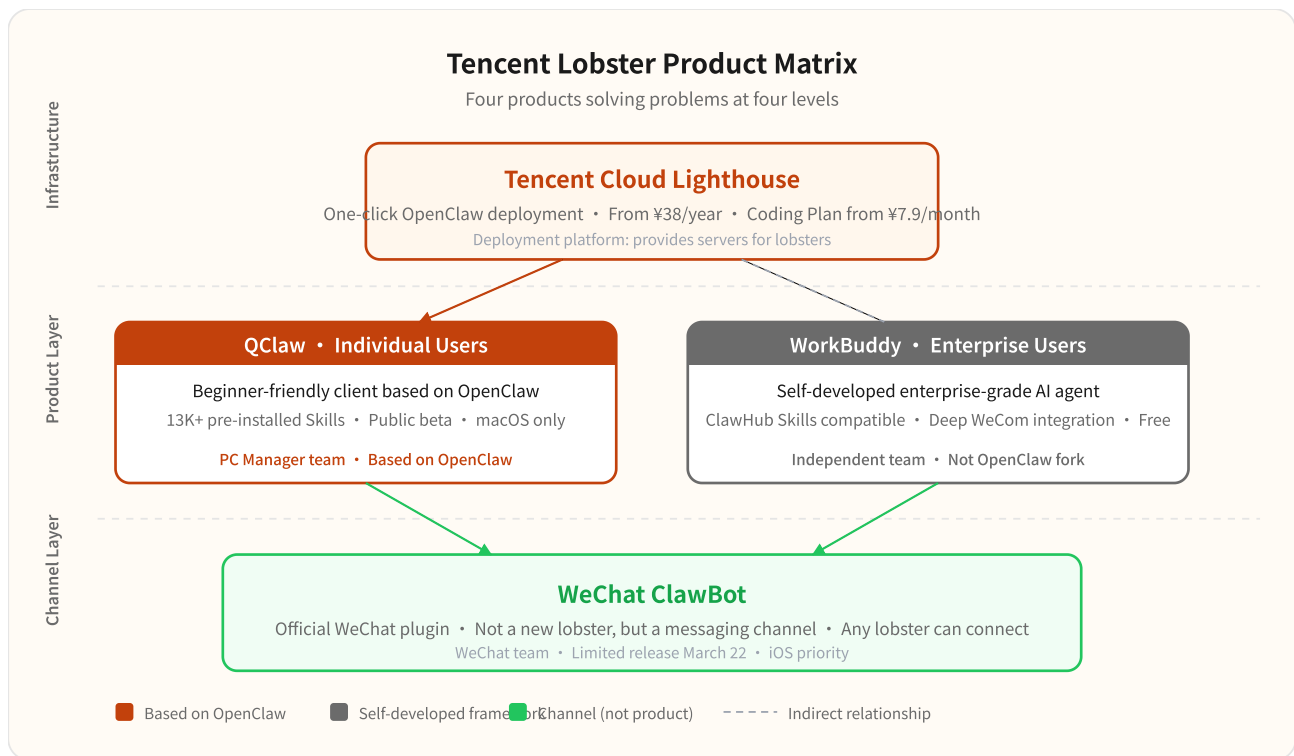
Product	Company	Form	Based on OpenClaw	Default Model	Price	Core Selling Points
MaxClaw	MiniMax	Cloud	Yes	MiniMax M2.5	¥39/month	18-second deployment, lowest price, 5-minute Feishu integration
AutoClaw	Zhipu AI	Client	Yes	GLM-5 / Pony-Alpha-2	Free + credits	96 pre-installed Skills, AutoGLM browser automation, one-click installation
QClaw	Tencent (PC Manager team)	Client	Yes	DeepSeek/Kimi/GLM multi-model	Public beta, free	Direct WeChat/QQ connection, one-click install, inspiration gallery, macOS only
ArkClaw	ByteDance/Volcengine	Cloud SaaS	Yes	Seed 2.0 multi-model	Included with Coding Plan Pro	Ready-to-use, deep Feishu integration
KimiClaw	Moonshot	Cloud	Yes	Kimi K2.5	¥199/month (includes membership)	Kimi ecosystem integration
WorkBuddy	Tencent (self-developed, not OpenClaw fork)	Client	Skills compatible	Hunyuan/DeepSeek/GLM etc.	Free (5000 Credits)	Self-developed agent, direct WeChat connection, deep WeCom integration, "semi-automatic" security philosophy
LobsterAI	NetEase Youdao	Open-source client	No, self-developed	Multi-model options	Free and open-source	GUI interface, sandbox isolation, strong Office capabilities
CoPaw	Alibaba Tongyi	Open-source	No, self-developed	Qwen series/Ollama etc.	Free and open-source	Dual deployment (client+cloud), multi-channel (DingTalk/Feishu/QQ)
miclaw	Xiaomi	Mobile	No, self-developed	MiMo	Free trial opens March 19	Native mobile operation, Xiaomi IoT ecosystem with 10B+

Product	Company	Form	Based on OpenClaw	Default Model	Price	Core Selling Points
						device connectivity, three new models

Tencent Lobster Suite: Explained in One Table

Tencent's lobster ecosystem layout is the most confusing - QClaw, WorkBuddy, Lighthouse, ClawBot, so many names, who's who? The logic is actually simple: they solve problems at different levels.

Product	One-Sentence Positioning	Team	Status	Relationship with OpenClaw
Tencent Cloud Lighthouse	Deployment platform: servers for running OpenClaw	Tencent Cloud	Available, ¥38/year	One-click deployment of original OpenClaw, includes Coding Plan (Lite ¥7.9 first month / Pro ¥39.9 first month)
QClaw	Beginner-friendly client: ready-to-use lobster	PC Manager team	Public beta March 20, macOS only	OpenClaw wrapper, pre-installed 13K+ Skills, lowers installation barrier
WorkBuddy	Enterprise agent: B2B AI assistant	Tencent (independent team)	Launched March 9, free	Self-developed framework, not OpenClaw fork, but compatible with ClawHub Skills ecosystem
WeChat ClawBot	Channel plugin: connects any lobster to WeChat	WeChat team	Limited release March 22	Not a new lobster, but a messaging channel - your existing lobster (QClaw/original/others) can talk via WeChat



The most confusing part is the difference between QClaw and ClawBot: QClaw installs a lobster on your computer (product), while ClawBot lets you talk to existing lobsters from WeChat (channel). They can work together - install a lobster with QClaw, then remotely control it from WeChat via ClawBot.

核心建议

WorkBuddy's lead explicitly stated in an interview that "WorkBuddy is not OpenClaw," with a completely self-developed technical approach. But it's compatible with ClawHub's Skills ecosystem, allowing users to directly install community skills. This is an interesting strategy: self-developed framework + borrowed ecosystem.

Recommendations by Scenario

Your Needs	First Choice	Alternative	Reason
Absolute beginner wanting fastest experience	AutoClaw	MaxClaw	AutoClaw one-click install, 96 pre-installed Skills, free start; MaxClaw cloud deployment in 18 seconds, from ¥39/month
WeChat/QQ users	QClaw + ClawBot	WorkBuddy	QClaw one-click lobster install (public beta), ClawBot for WeChat remote control; WorkBuddy self-developed, no deployment needed, good WeCom integration
Feishu ecosystem	ArkClaw	AutoClaw	Same ByteDance ecosystem, deep Feishu integration; AutoClaw also supports one-click Feishu connection
Budget conscious	LobsterAI	CoPaw	Both free and open-source with complete features
Want full control and deep customization	Original OpenClaw	CoPaw	Largest open-source community, richest resources
Mobile + smart home	miclaw	—	Currently only mobile solution (Xiaomi 17 series only, closed beta)

注意

Purchase Reminder: Most wrapped products (MaxClaw, KimiClaw etc.) lock the default model, unlike original OpenClaw which allows free switching. If model selection is important, prioritize original OpenClaw or multi-model products (AutoClaw, WorkBuddy, LobsterAI). Also, most products launched in Feb-Mar 2026, with features and stability still rapidly evolving.

A Frequently Asked Questions

Q1: Is OpenClaw free?

OpenClaw itself is MIT-licensed and open-source free. However, running it requires two costs: server (local computer or cloud server) and AI model API fees. If you use local models (Ollama), the API fees can also be free. Summary: software is free, computing power is not.

Q2: What technical level do I need to use OpenClaw?

Basic command-line skills to install npm packages are sufficient. The simplest installation requires just two commands:

```
npm install -g openclaw@latest and openclaw onboard --install-daemon
```

If using Alibaba Cloud/Tencent Cloud one-click deployment solutions, the barrier is even lower. However, integrating multiple platforms, customizing Skills, and optimizing configurations require some technical foundation.

Q3: What's the difference between OpenClaw and ChatGPT?

ChatGPT is a "consultant" (you ask, it answers), while OpenClaw is an "employee" (it proactively executes tasks). OpenClaw can connect to your messaging platforms, manage emails/calendars, operate browsers, execute shell commands, and all data remains in your hands. The trade-off is that you need to deploy and maintain it yourself.

Q4: Is it safe? Will my data be leaked?

OpenClaw is self-hosted, with data stored on your own server by default, not passing through third parties. However, three security risks should be noted: (1) CVE-2026-25253 RCE vulnerability (fixed, must update to latest version); (2) ClawHavoc supply chain attacks (always review source code before installing third-party Skills); (3) If Gateway is exposed to the public internet, authentication must be set (`gateway.auth.mode`).

Q5: How much does it cost per month?

Depends on your usage and model choices. Reference ranges: Completely free (local models) → \$2-5/month (mainly DeepSeek) → \$5-15/month (mainly GLM-5) → \$10-30/month (mainly Claude Sonnet). The biggest cost trap is OpenClaw's multi-tool calls consuming large amounts of Tokens - be sure to set spending limits.

Q6: Can I use domestic models? How's the performance?

Absolutely. DeepSeek-V3 (\$0.14/M input) and GLM-5 (\$0.80/M input) are the most popular choices among Chinese users. Zhipu also released GLM-5-Turbo on March 16, 2026 - the first model specifically optimized for OpenClaw from the training phase, with enhanced tool calling and long-chain execution capabilities, currently in experimental release. Performance is certainly not as good as Claude Sonnet (recognized as the strongest for Agent tasks), but sufficient for most daily tasks. Recommended to use Fallback mechanism for mixed deployment.

Q7: Anthropic banned OAuth, how can I use Claude?

Use Anthropic API Key (pay-as-you-go). Create an API Key at [Anthropic Console](#), then configure the `ANTHROPIC_API_KEY` environment variable in OpenClaw. Do not attempt to connect Claude Pro/Max subscription accounts via OAuth - accounts will be banned.

Q8: Will OpenClaw continue after its founder joined OpenAI?

Yes. After Peter Steinberger joined OpenAI, OpenClaw is transitioning to open-source foundation governance. OpenAI has committed to sponsor the project without interfering with development direction. As of March 2026, the project maintains near-daily updates with 1,075+ contributors. The project's long-term sustainability is guaranteed.

Q9: Are Skills on ClawHub safe?

Don't trust blindly. Among ClawHub's 13,729 Skills, about 20% have issues (spam/duplicates/malicious) based on community audits. During the ClawHavoc incident, over 800 malicious Skills attempted to steal user credentials. Recommendations: Only install Skills with many stars, review source code before installation, use the awesome-openclaw-skills curated list (pre-filtered for problematic Skills).

Q10: Can it connect to WeChat?

After March 22, 2026 - yes! The WeChat team launched the official ClawBot plugin. One command (`npx -y @tencent-weixin/openclaw-weixin-cli@latest install`) + QR code scan can connect your OpenClaw to personal WeChat. This is the official solution, no longer requiring WeCom relay or iPad protocol gray-area methods. Currently ClawBot is still in limited release, iOS first (requires WeChat 8.0.70+), Android timeline TBD. See [S 17 WeChat ClawBot](#) in this guide.

Q11: Can OpenClaw and Claude Code be used together?

Yes, and it's recommended. The community developed openclaw-claude-code-skill to bridge them via MCP protocol. OpenClaw handles messaging platform integration and life automation, Claude Code handles programming tasks. This combination forms the most complete AI workflow in 2026.

Q12: How's the performance of local models?

Depends on hardware and model choice. 32GB RAM can run Qwen3-Coder:32B or Devstral-24B, performing well on code generation and simple Agent tasks. But still lags behind cloud-based Claude Sonnet or GPT-5.4, especially for complex multi-step reasoning tasks. Suitable for privacy-sensitive scenarios and experimental use.

B Command Cheat Sheet

Installation & Updates

Command	Description
<code>npm install -g openclaw@latest</code>	Global OpenClaw installation
<code>openclaw onboard --install-daemon</code>	Initial configuration + install daemon
<code>openclaw update --channel stable</code>	Update to latest stable version
<code>openclaw update --channel beta</code>	Update to Beta version (early access)
<code>openclaw doctor</code>	Diagnostic check for common issues
<code>openclaw --version</code>	Check current version

Daily Use

Command	Description
<code>openclaw gateway --port 18789 --verbose</code>	Start Gateway (verbose logging mode)
<code>openclaw gateway restart</code>	Restart Gateway (required after config changes)
<code>openclaw agent --message "xxx"</code>	Directly send message to Agent
<code>openclaw devices pair</code>	Device pairing (first-time connection)
<code>openclaw models list</code>	List configured models
<code>openclaw models status --probe</code>	Test model connectivity
<code>openclaw config set agents.defaults.model.primary provider/model</code>	Set primary model
<code>/fast</code>	Switch to fast mode in conversation (v3.12 new feature, reduces latency, maps to model's fast API channel)
<code>openclaw backup create</code>	Create local config backup (v3.8 new feature)
<code>openclaw backup verify</code>	Verify backup integrity
<code>openclaw workspace trust /path</code>	Explicitly trust workspace plugin (v3.12 new feature, cloned repos no longer auto-load)

Plugin Management

Command	Description
<code>openclaw plugins install <name></code>	Install plugin/Skill
<code>openclaw plugins enable <name></code>	Enable plugin
<code>openclaw plugins list</code>	List installed plugins
<code>openclaw plugins install @openclaw-china/channels</code>	Install China IM plugin
<code>openclaw china setup</code>	Configure China IM platforms (requires plugin)

Model Authentication

Command	Description
<code>openclaw onboard --auth-choice zai-api-key</code>	Configure Zhipu GLM
<code>openclaw onboard --auth-choice apiKey --token-provider openrouter --token "\$KEY"</code>	Configure OpenRouter
<code>openclaw models auth login --provider qwen-portal --set-default</code>	Tongyi Qwen OAuth login

Chat Commands (used in conversation)

Command	Description
<code>/status</code>	Session overview (current model, Token usage)
<code>/new</code>	Clear chat history, start new conversation
<code>/think <level></code>	Adjust reasoning depth (off/minimal/low/medium/high/xhigh)
<code>/usage off tokens full</code>	Control usage display in reply footer
<code>/activation mention always</code>	Group message handling mode

Docker Deployment

Command	Description
<code>docker-compose up -d</code>	Start OpenClaw container in background
<code>docker-compose logs -f</code>	View real-time logs
<code>docker-compose pull && docker-compose up -d</code>	Update to latest image

C Resource Links

Official Resources

Resource	URL
GitHub Repository	github.com/openclaw/openclaw
Official Documentation	docs.openclaw.ai
Official Website	openclaw.ai
ClawHub Skill Marketplace	clawhub.ai
Moltbook (Agent Social Network)	moltbook.com
GitHub Releases	github.com/openclaw/openclaw/releases
GitHub Discussions	github.com/openclaw/openclaw/discussions

Community Resources

Resource	URL	Description
awesome-openclaw-skills	github.com/VoltAgent/awesome-openclaw-skills	5,494 curated Skills (pre-filtered), 31.4K Stars
awesome-openclaw-usecases	github.com/hesamsheikh/awesome-openclaw-usecases	Community use case collection, 21K Stars
openclaw-claude-code-skill	github.com/Enderfga/openclaw-claude-code-skill	Bridge Claude Code capabilities
SecureClaw	Open-source security tool	Skill security scanning

China Resources

Resource	URL	Description
openclaw-china plugin	github.com/BytePioneer-AI/openclaw-china	DingTalk/QQ/WeCom/WeChat integration
OpenClaw Chinese Docs	openclaw.cc	Community-maintained Chinese documentation
Alibaba Cloud Deployment Docs	help.aliyun.com (search OpenClaw)	Lightweight server one-click deployment
Bilibili Deployment Tutorial	BV1MfFAz6EnR	Step-by-step: WeChat/Feishu/DingTalk/QQ integration

Tutorial Resources

Resource	Language	Description
freeCodeCamp Complete Tutorial	English	Comprehensive guide from scratch
DigitalOcean Introduction	English	What is OpenClaw overview
Zhihu Deployment Series	Chinese	Multiple deployment and usage tutorials
CNBlogs Source Compilation Guide	Chinese	Building OpenClaw from source
Rookie Tutorial One-Click Deployment	Chinese	Simplest deployment solution

Model Providers

Provider	API Console
Anthropic Claude	console.anthropic.com
OpenAI	platform.openai.com
Google AI Studio	aistudio.google.com
DeepSeek	platform.deepseek.com
Zhipu GLM	bigmodel.cn
Tongyi Qwen	dashscope.aliyun.com
Moonshot Kimi	platform.moonshot.cn
SiliconFlow	siliconflow.cn
OpenRouter	openrouter.ai
Volcengine (Doubao)	console.volcengine.com

This document was compiled with the assistance of Claude Code, based on OpenClaw official documentation, GitHub repository, and community resources. The accuracy and timeliness of the content are for reference only. For corrections or suggestions, please follow the official account "Huashu" for feedback.

Sources: docs.openclaw.ai • github.com/openclaw/openclaw • clawhub.com • Created by Huashu • March 2026

AI Programming: From Beginner to Master

Knowledge Planet • Huashu's AI Programming Community



Knowledge Planet QR Code

Host: Huashu

Natural language is the best programming language in the AI era.

Creator of "Kitten Fill Light" - #1 paid app on AppStore charts

Author of "Master DeepSeek with One Book"

[Join Knowledge Planet →](#)

[Bilibili: Huashu](#) • [YouTube: Huashu](#) • Official Account: Huashu

Created by Huashu • v1.1 • March 2026

Companion videos: [Bilibili "OpenClaw from 0 to 1"](#) • Future updates: [Feishu Docs](#)