The Complete OpenClaw Deployment Guide

Updated Edition (v1.1)

By ZeroSignal

Deploy your AI. Own your stack. This guide covers reliable OpenClaw installs across macOS, Windows, Docker, Linux, and VPS environments—plus hardening, persistence, and troubleshooting.

Release date: 2026-02-23

What’s New in v1.1

• Added an install decision tree (native vs Docker vs WSL2) to reduce ambiguity.

• Added verification steps after each major phase (install, setup, gateway, UI).

• Fixed Windows Docker bind-mount example and explained host-path vs container-path rules.

• Clarified ports and components (Gateway vs Web UI) and how they map in each install mode.

• Added security + ethics notes for Kali / offensive tooling usage.

• Expanded troubleshooting: common errors, where logs live, and fastest diagnostic commands.

Table of Contents

1. Overview and Core Concepts

What is OpenClaw? OpenClaw is a self-hosted AI gateway that runs on your hardware and routes requests to model providers (OpenAI, OpenRouter, etc.), can integrate with messaging channels, and can run automations and tools.

Key components:

• **CLI**: the *openclaw* command used for setup, config, and management.

• **Gateway**: the long-running service process that handles conversations, tools, and integrations.

• **Web UI**: an optional local browser UI used to interact with your agent.

• **Workspace**: the on-disk directory where configuration, skills, and memory are stored.

Ports (common defaults):

• Web UI: 3000 (browser access) in many non-compose setups.

• Gateway service: often 18789 (and a bridge port like 18790) in containerized gateway deployments.

• If a port is already in use, OpenClaw will fail to start; troubleshooting includes fast port checks.

2. Install Decision Tree

Choose the install path that matches your goals and constraints:

• **macOS native**: simplest developer workflow on a Mac; use Homebrew + npm.

• **Windows + WSL2**: recommended; fewer edge cases and better parity with Linux instructions.

• **Windows native**: everything in PowerShell; handle execution policy and PATH issues.

• **Docker**: isolation and easy upgrades; recommended for servers and dedicated boxes.

• **Kali**: specialized Linux path for authorized security labs.

Decision shortcuts:

• If you want easiest cross-platform reproducibility: **Docker Compose**.

• If you want best Windows experience: **WSL2 + Ubuntu**.

• If you need full host access and tooling: **native install**.

3. macOS Installation (Homebrew + npm)

Prerequisites: Node.js 18+ (recommended Node 20 LTS), npm, and a terminal (zsh).

Install Homebrew (if needed):

```
/bin/bash -c "$(curl -fsSL \
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)\"
```

Install Node.js:

```
brew install node
node --version
npm --version
```

Install OpenClaw and run setup:

```
npm install -g openclaw
openclaw setup
```

Start the gateway and open the Web UI:

```
openclaw gateway start
openclaw web
```

Verify:

• CLI works: **openclaw --version**

• Gateway is running: **openclaw gateway status** (or check logs below)

• Web UI loads in browser (default: http://localhost:3000)

Common macOS issues:

• Port already in use: **lsof -i :3000** then stop the conflicting process.

• Global npm permission errors: use a user-local npm prefix instead of sudo.

```
# Fix npm global permissions (recommended)
mkdir -p ~/.npm-global
npm config set prefix "~/.npm-global"
echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.zprofile
source ~/.zprofile
npm install -g openclaw
```

4. Windows Installation (WSL2 and Native)

Recommended: WSL2 because it reduces PATH/permissions edge cases and matches Linux instructions.

Path A: WSL2 (Recommended)

```
# Run PowerShell as Administrator
wsl --install
```

Inside Ubuntu (WSL2):

```
sudo apt update && sudo apt upgrade -y
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
node --version
npm --version
sudo npm install -g openclaw
openclaw setup
openclaw gateway start
```

Path B: Native Windows (PowerShell)

• Install Node.js LTS from nodejs.org and ensure 'Add to PATH' is checked.

• If PowerShell blocks npm scripts, set execution policy for current user.

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

npm install -g openclaw
openclaw setup
openclaw gateway start
```

Windows firewall (only if accessing from other devices on your LAN):

```
New-NetFirewallRule -DisplayName "OpenClaw Web UI" -Direction Inbound -Protocol TCP
-LocalPort 3000 -Action Allow
# Optional: restrict to local subnet
New-NetFirewallRule -DisplayName "OpenClaw Web UI Local" -Direction Inbound -Protocol
TCP -LocalPort 3000 -RemoteAddress 192.168.1.0/24 -Action Allow
```

5. Docker Installation (Single container and Compose)

Docker is recommended for isolation and repeatable deployments. There are two common approaches: a single container quick start and a Compose stack.

Quick start (single container)

```
docker run -d \
  --name openclaw \
  --restart unless-stopped \
  -p 3000:3000 \
  -v openclaw-workspace:/workspace \
  -e OPENROUTER_API_KEY=your_key_here \
  openclaw/openclaw:latest
```

Compose (recommended for most users)

Key rule: **host paths must be bound to container paths**. A bare host path (no ':target') is treated as a volume target and will fail on Windows (errors like 'invalid mount path').

Example: Windows bind mount (correct)

```
services:
  openclaw-gateway:
    volumes:
      - type: bind
        source: D:/OpenClaw_Shared/openclaw_share
        target: /shared
```

Bring the stack up and check status:

```
docker compose up -d
docker compose ps
docker compose logs -f openclaw-gateway
```

Persistence locations (typical)

• /workspace/SOUL.md (agent personality)

• /workspace/MEMORY.md (long-term memory, if enabled)

• /workspace/skills/ (installed skills)

• /workspace/memory/ (session logs, if enabled)

6. Kali Linux / Security Lab Setup (Authorized use only)

This chapter is for **authorized** security testing only (your own systems or systems you have explicit permission to test).

Install Node.js 20 and OpenClaw:

```
sudo apt update && sudo apt upgrade -y
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
node --version
sudo npm install -g openclaw
openclaw setup
openclaw gateway start
```

Operational guidance:

• Keep work in a dedicated lab workspace; do not mix personal tokens on shared machines.

• Log outputs; keep evidence and scope notes for every engagement.

• Rate-limit and scope all scans; be explicit about targets and permissions.

7. VPS Deployment (Headless) + Reverse Proxy

For 24/7 operation, you can deploy OpenClaw to an Ubuntu/Debian VPS. Prefer Docker for upgrades and isolation.

Native install (Ubuntu/Debian)

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
sudo npm install -g openclaw
mkdir -p /opt/openclaw-workspace
cd /opt/openclaw-workspace
openclaw setup
openclaw gateway start
```

Reverse proxy + TLS outline

• Terminate TLS at nginx and proxy to localhost-only OpenClaw port.

• Use strong security headers and restrict access (VPN or IP allowlist) where possible.

8. Advanced Usage (Sub-agents, Cron, Skills)

OpenClaw can spawn specialized sub-agents for isolated work and schedule tasks via a cron interface.

Sub-agent delegation pattern (example)

```
Main agent receives task
→ Spawns Worker (lower-cost model) for bulk work
→ Spawns Verifier (strict model) for tests / checks
→ Merges results and reports back
```

Multi-channel setup (example)

```
openclaw config set telegram.botToken BOT_TOKEN_1
openclaw config set discord.token DISCORD_BOT_TOKEN
openclaw config set discord.guildId YOUR_SERVER_ID
openclaw gateway restart
```

API key management best practices

• Store keys in environment variables or .env files; never hardcode.

• Add .env to .gitignore; commit .env.example instead.

• Rotate keys regularly and immediately after suspected exposure.

• Use least-privilege scopes where providers support it.

9. Troubleshooting and Diagnostics

Start with these commands; they solve most first-run issues quickly.

Fast diagnostics

```
# Version checks
node --version
npm --version
openclaw --version

# Gateway logs / status
openclaw gateway logs
openclaw gateway status

# Port checks (macOS/Linux)
lsof -i :3000

# Port checks (Windows)
netstat -ano | findstr ":3000"
```

Common failure patterns

• **Gateway won’t start**: wrong Node version, port conflict, or invalid config file.

• **API calls failing**: missing API key, wrong provider selected, network/DNS issues.

• **Memory not persisting**: workspace path not writable or volume not mounted (Docker).

• **Docker container restarting**: check logs, env vars, and mount syntax; ensure host path is bound to a container target.

• **WSL2 browser can’t reach localhost**: use the WSL IP address or ensure port forwarding is active.

Appendix A: Command Cheat Sheet

Command	Purpose
openclaw setup	Run interactive setup wizard
openclaw gateway start	Start the gateway service
openclaw gateway restart	Restart gateway service
openclaw gateway logs	Show gateway logs
openclaw config get	Show current configuration
openclaw config set KEY VALUE	Update configuration value
openclaw skill install NAME	Install a skill
openclaw web	Open Web UI in browser
openclaw --version	Show version

Appendix B: Recommended Versions and Ports

Component	Minimum	Recommended
Node.js	18.x	20.x LTS
npm	9.x	10.x
Docker	24.x	Latest stable
Docker Compose	2.x	Latest stable
Ubuntu/Kali	22.04	24.04 LTS (Ubuntu) / latest Kali
macOS	Ventura (13)	Sonoma (14) or newer

Common ports:

• Web UI: 3000

• Gateway: 18789

• Bridge (if enabled): 18790