

OpenClaw

Complete Update Guide

THE SPRINT Community — February 2026 | Based on v2026.1.29 (Latest Stable)

OpenClaw moves fast — new releases drop almost daily. Staying current means you get bug fixes, security patches, new features, and better performance. This guide covers everything you need to update safely, from a simple one-liner to advanced rollback procedures.

TL;DR — Quick Update: Run: curl -fsSL https://openclaw.ai/install.sh | bash — then run: openclaw doctor

What's New in v2026.1.29 (Latest Release)

Version 2026.1.29 is the biggest release yet — it officially rebrands the project from Clawdbot to OpenClaw and ships a massive list of improvements across security, performance, channels, and tooling.

Official Rename: Clawdbot -> OpenClaw

The npm package and CLI are now officially named openclaw. A clawdbot compatibility shim is included so your existing scripts won't break, but you should migrate your commands over time.

- Package renamed to openclaw on npm
- CLI command is now: openclaw (not clawdbot)
- Extensions moved to the @openclaw/* scope
- macOS app bundle identifiers updated to bot.molt

BREAKING CHANGE — Gateway Auth Required

[BREAKING] The gateway "auth mode: none" option has been removed. Your gateway now requires a token or password. Tailscale Serve identity auth is still allowed. If you had auth mode set to none, you must configure a token before updating.

Performance Highlights

- Faster CLI startup — Node's module compile cache is now used
- Session key regexes are precompiled for better routing speed
- Config env vars now apply before variable substitution
- Gateway session metadata now prefers newest entries when combining stores

Security Improvements

- Gateway now warns when exposed without auth
- Hook tokens via query params now trigger a warning — headers preferred
- Tailscale Serve auth hardened — validates via local tailscaled before trusting headers
- Twilio webhook signature verification enforced for ngrok URLs
- npm tar dependency pinned to 7.5.4 to address security advisories
- Windows ACL audit for config includes improved

Channel Updates

Telegram received the most attention in this release:

- Sticker send/receive with vision caching
- Support for editing sent messages
- Quote replies for message tool
- Optional silent send flag (no notifications)
- Better multi-agent routing in multi-account setups
- Hardened polling and retry for transient network errors

Discord, Slack, and Matrix also received fixes and improvements.

Agent & Memory Improvements

- Dropped messages during compaction are now summarized instead of silently removed
- memory.md is now included when bootstrapping memory context
- Cooled-down providers are skipped during model failover
- Session locks are now released on process termination
- Extra paths for memory indexing can now be configured
- Image error retries prevented — size limits surfaced clearly

Web UI & macOS

- Sub-agent replies now stay visible in WebChat
- Chat compose textarea auto-expands while typing
- macOS: fixed crash when rendering code blocks (Textual 0.3.1)
- macOS: auto-scroll to bottom when sending a new message
- macOS: SSH usernames preserved in remote targets

Before You Update — Checklist

Do This First: Spend 60 seconds on this checklist before every update. It will save you hours of debugging if something goes wrong.

Know Your Install Type

Everything in this guide branches based on how you originally installed OpenClaw. Check by running:

```
openclaw update status
```

This tells you your current version, channel (stable/beta/dev), and whether you're on a git checkout or npm install.

- Global npm/pnpm install — most common. Updates via npm.
- Source install (git clone) — updates via git pull + build.
- Docker/Railway/Render/Fly.io — update by redeploying your container.

Know Your Gateway Mode

- Foreground terminal — you run openclaw gateway manually
- Supervised service — launchd (macOS) or systemd (Linux/Windows WSL2)

Run openclaw gateway status to check.

Backup Your Config

Before every major update, quickly snapshot these three locations:

```
~/.openclaw/openclaw.json      # your main config  
~/.openclaw/credentials/       # your API keys and tokens  
~/.openclaw/workspace          # your agent workspace files
```

Tip: You don't need to copy these files — just knowing where they are is enough. If an update breaks something, these files let you restore to a known-good state.

How to Update — All Methods

Method 1: Re-run the Installer (Recommended)

This is the officially recommended approach. The installer detects your existing setup, upgrades in place, and runs openclaw doctor automatically.

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

Options you might want to add:

- Add --no-onboard to skip the setup wizard (since you're already configured)
- For source installs, use: curl -fsSL https://openclaw.ai/install.sh | bash -s -- --install-method git --no-onboard

Method 2: The openclaw update Command

If you're on a source (git) install, this is your best friend. It handles everything — clean worktree check, channel switching, deps, build, doctor, and gateway restart.

```
openclaw update
```

Common variations:

```
openclaw update          # update to latest on current channel
openclaw update --channel beta    # switch to beta channel
openclaw update --channel stable # switch back to stable
openclaw update --no-restart      # update without restarting gateway
openclaw update wizard         # interactive guided update flow
openclaw update status          # check current version + availability
```

What openclaw update does automatically: Checks for a clean worktree -> switches channel -> fetches upstream -> installs deps -> builds -> builds Control UI -> runs openclaw doctor -> restarts gateway

Method 3: Global npm/pnpm Install

If you installed via npm or pnpm (no git checkout), use this:

```
npm i -g openclaw@latest
```

Or with pnpm:

```
pnpm add -g openclaw@latest
```

Then always follow up with:

```
openclaw doctor
openclaw gateway restart
openclaw health
```

Note: Bun is NOT recommended for the Gateway runtime — it causes bugs with WhatsApp and Telegram. Stick with npm or pnpm.

Method 4: Manual Source Update

If you need full control or are debugging, you can update manually from your git checkout:

```
git pull  
pnpm install  
pnpm build  
pnpm ui:build  
openclaw doctor  
openclaw health
```

Method 5: Control UI (Browser-Based Update)

If you prefer a GUI, open your Control UI and look for Update & Restart. This runs the same source update flow as openclaw update, shows you the output, and automatically pings your last active session with a report when done.

Note: This only works with source (git) installs, not npm installs.

After Every Update — Verification Steps

Never skip this. Takes 30 seconds and catches 90% of problems before they bite you.

1. **Run openclaw doctor** — this migrates deprecated config keys, checks gateway health, audits DM policies, and will tell you if anything needs fixing. Read the output.

```
openclaw doctor
```

2. **Restart the gateway**

```
openclaw gateway restart
```

3. **Verify health**

```
openclaw health
```

4. **Check logs for errors**

```
openclaw logs --follow
```

Why openclaw doctor matters: Doctor does config migrations automatically. If you skip it after an update, deprecated config keys may cause silent failures or unexpected behavior. It's intentionally boring — repair, migrate, warn. Run it every time.

Gateway Management Commands

The gateway is the core process that runs your AI agents. Here are the essential commands:

```
openclaw gateway status          # check if gateway is running
openclaw gateway restart         # restart the gateway
openclaw gateway stop            # stop the gateway
openclaw gateway install         # install as a background service
openclaw logs --follow          # stream live logs
```

If your gateway is supervised as a system service:

- macOS (launchd): launchctl kickstart -k gui/\$UID/bot.molt.gateway
- Linux / Windows WSL2 (systemd): systemctl --user restart openclaw-gateway.service

Switching Update Channels

OpenClaw has three channels. Most of you should stay on stable, but here's what each means:

Channel	Best For	Notes
stable	Production use	Tested releases. Recommended for most users and all SaaS projects.
beta	Early adopters	Pre-release builds. Good for testing upcoming features.
dev	Contributors	Tracks main branch. Requires git checkout. May have breaking changes.

To switch channels:

```
openclaw update --channel stable
openclaw update --channel beta
openclaw update --channel dev
```

Rollback — When Something Breaks

Updates occasionally break things. Here's how to get back to a working state fast.

Rollback — npm/pnpm Install

Find the last known-good version number (check the GitHub releases page), then pin to it:

```
npm i -g openclaw@2026.1.24
openclaw doctor
openclaw gateway restart
```

To see what's currently published:

```
npm view openclaw version
```

Rollback — Source Install (by date)

Find a commit from before things broke:

```
git fetch origin
git checkout "$(git rev-list -n 1 --before=\"2026-01-24\" origin/main)"
pnpm install
pnpm build
openclaw gateway restart
```

When you're ready to return to latest:

```
git checkout main && git pull
```

Troubleshooting Common Issues

Gateway won't start after update

- Run openclaw doctor — it will often tell you exactly what's wrong
- Check openclaw logs --follow for errors
- If you had auth mode: none in your config, this is now removed — add a token
- Try openclaw gateway install if the service isn't found

Update fails with dirty worktree (source installs)

The update command requires a clean git worktree. If you have uncommitted local changes:

```
git stash      # stash changes
openclaw update # now update
git stash pop   # restore your changes
```

Wrong version showing after update

- Run openclaw update status to check actual version
- If npm install, try: npm list -g openclaw
- Sometimes PATH issues cause the old binary to run — restart your terminal

Still stuck?

- Run openclaw doctor again and read the entire output carefully
- Check: <https://docs.openclaw.ai/help/troubleshooting>
- Ask in the THE SPRINT community or OpenClaw Discord: <https://discord.gg/clawd>

Quick Reference Cheat Sheet

Print this or keep it handy. These are the commands you'll use 95% of the time:

```
# Check current version
openclaw update status

# Recommended update (re-runs installer)
curl -fsSL https://openclaw.ai/install.sh | bash

# Or: update via CLI (source installs)
openclaw update

# Or: update via npm (global installs)
npm i -g openclaw@latest

# Always run after every update:
openclaw doctor
openclaw gateway restart
openclaw health
```

```
# If something breaks - rollback npm:  
npm i -g openclaw@<version>  
  
# Gateway management:  
openclaw gateway status  
openclaw gateway restart  
openclaw logs --follow
```