

2.0.0 (Major)

Architectural changes

Go

- Backend rewritten from the ground up in Go, replacing Node.js and Next.js. The stack is more scalable, uses less RAM, and is significantly more performant.
- [sqlc](#) is used for type-safe SQL against PostgreSQL (compile-time checked queries instead of ad-hoc ORM access patterns).
- [golang-migrate](#) is used for database migrations (replacing Prisma ORM).
- Structured logging with the standard library `log/slog` for cleaner, machine-parseable logs in production.

Background jobs and automation

- Background work is handled by [asyng](#) on Redis instead of BullMQ. PatchMon no longer ships the embedded **Bull Board** stack; queue visibility and triggers live in the existing **Automation** UI, which reduces attack surface, image size, and operational complexity.

Docker

- Docker is the officially supported deployment method going forward; bare-metal installs are discontinued. A migration document describes the upgrade path.
- Hardened base images are used. They ship with near-zero CVEs and a smaller footprint.
- No separate frontend container: static React build artifacts are embedded in the Go binary. The container runs that binary (by default on port 3000) with [chi](#): `/api/*` is handled by the server, so nginx inside PatchMon is no longer required. You still use nginx or another reverse proxy in front for TLS termination and public access as usual.
- A Guacamole (guacd) sidecar is included for Windows RDP. It is separate for now; RDP/VNC for Windows is an area we intend to improve.

API documentation

- **OpenAPI 3** spec is served at `/api/v1/openapi.json`, with **Swagger UI** under `/api/v1/api-docs` (authenticated) for exploring integration endpoints.

New features

- **Linux patching**: Deploy updates per host or in bulk, on demand or on a schedule. **Policies** support host/group assignments and exclusions; runs support **approval**, **stop**, **retry validation**, and **live log streaming** over WebSocket.
- **Microsoft Windows agent** (beta) and **FreeBSD** agent support.
- **Windows Updates** (beta Windows agent): server APIs for update results, reboot state, superseded cleanup, and approved-guid sync, aligned with the new Windows agent.
- **Advanced monitoring & alerting**: richer alert lifecycle (including assignment and bulk actions), optional **advanced alert configuration** for tuning and cleanup where your edition includes it.
- **Notifications**: first-class **destinations** (SMTP, webhooks, ntfy), **routes**, **delivery log**, and **scheduled reports** so operational signals leave PatchMon reliably.
- **Environment variables in the GUI**: many settings that were previously only in process environment can be **viewed and edited from the Settings UI** (per-key updates, with sensible validation), so you change less by hand in compose or shell env for day-to-day tuning.
- **OIDC / SSO**: configure OpenID Connect from the same Settings area, including **import from environment** when you are migrating from a file-based or container env setup.

Other improvements

- **Compliance / OpenSCAP**: SSG and CIS benchmarking content is **bundled in the server binary** at build time. Agents no longer pull scanning content from GitHub; everyone shares one versioned source of truth and less outbound traffic from agents.
- **SSO**: improved sign-in flows and **Entra ID** integration compared to 1.4.x OIDC edge cases (e.g. redirect loops with auto SSO).
- **Dashboard**: additional cards and data surfaces; dashboard layout preferences carry forward in the new UI.
- **Host integration config**: **apply pending config** from the server so integration changes are applied to agents in a controlled, observable way.
- **Settings reliability**: server URL and related configuration are reimplemented on the Go stack with database-backed resolution, addressing classes of “settings did not persist” issues from the Node era.
- **Reverse proxy awareness**: continued correct use of forwarded headers for HTTPS/WSS behind proxies (without the Bull Board-specific HTTP quirks from 1.4.x).
- **Optional admin pprof**: when enabled, CPU/memory profiling endpoints are available to administrators for performance investigation.

Packaging and editions

- Features are grouped into **capability modules** (e.g. patching policies, advanced alerts, custom branding, Docker inventory, compliance depth, AI assist, remote access). Core workflows stay simple; larger deployments can enable more surface area where their **subscription or license** allows. See in-app **Context** / billing documentation for your tenant.

Known issues

- **Remote Desktop (RDP)**: there is a known bug with the RDP connection flow in this release. A fix is planned for the next release.

Migrations

This covers migration for Docker, Proxmox community scripts, and legacy `setup.sh` installs:

[Migrating from 1.4.2 to 2.0.0](#)

Revision #3

Created 2026-04-24 17:35:25 UTC by lby

Updated 2026-04-24 17:37:12 UTC by lby