Depreciated - Installing PatchMon Server on Ubuntu 24

This is legacy and not applicable for PatchMon V2

Overview

The PatchMon setup script automates the full installation on Ubuntu/Debian servers. It installs all dependencies, configures services, generates credentials, and starts PatchMon - ready to use in minutes.

It supports both **fresh installations** and **updating existing instances**.

Requirements

Requirement	Minimum
OS	Ubuntu 20.04+ / Debian 11+
CPU	2 vCPU
RAM	2 GB
Disk	15 GB
Access	Root (sudo)
Network	Internet access (to pull packages and clone the repo)

Fresh Installation

1. Prepare the server

```
apt-get update -y && apt-get upgrade -y
apt install curl jq bc -y
```

2. Run the setup script

```
curl -fsSL -o setup.sh
https://raw.githubusercontent.com/PatchMon/PatchMon/refs/heads/main/setup.sh \
  && chmod +x setup.sh \
  && sudo bash setup.sh
```

3. Follow the interactive prompts

The script will ask you four things:

Prompt	Description	Default
Domain/IP	The public DNS or local IP users will access PatchMon from	`patchmon.internal`
SSL/HTTPS	Enable Let's Encrypt SSL. Use `y` for public servers, `n` for internal networks	`n`
Email	Only asked if SSL is enabled - used for Let's Encrypt certificate notifications	-
Release / Branch	Lists the latest 3 release tags plus `main`. Pick the latest release unless you need a specific version	Latest tag

After confirming your choices, the script runs fully unattended.

What the Script Does

The script performs these steps automatically:

1. **Checks timezone** - confirms (or lets you change) the server timezone
2. **Installs prerequisites** - `curl`, `jq`, `git`, `wget`, `netcat-openbsd`, `sudo`
3. **Installs Node.js 20.x** - via NodeSource
4. **Installs PostgreSQL** - creates an isolated database and user for this instance

5. **Installs Redis** - configures ACL-based authentication with a dedicated Redis user and database
6. **Installs Nginx** - sets up a reverse proxy with security headers
7. **Installs Certbot** (if SSL enabled) - obtains and configures a Let's Encrypt certificate
8. **Creates a dedicated system user** - PatchMon runs as a non-login, locked-down user
9. **Clones the repository** to `/opt/<your-domain>/`
10. **Installs npm dependencies** in an isolated environment
11. **Creates `.env` files** - generates secrets and writes `backend/.env` and `frontend/.env`
12. **Runs database migrations** - with self-healing for failed migrations
13. **Creates a systemd service** - with `NoNewPrivileges`, `PrivateTmp`, and `ProtectSystem=strict`
14. **Configures Nginx** - reverse proxy with HTTP/2, WebSocket support, and security headers
15. **Populates server settings** in the database (server URL, protocol, port)
16. **Writes `deployment-info.txt`** - all credentials and commands in one file

---

After Installation

1. Visit `http(s)://<your-domain>` in your browser
2. Complete the first-time admin setup (create your admin account)
3. All credentials and useful commands are saved to:

```
/opt/<your-domain>/deployment-info.txt
```

Directory Structure

After installation, PatchMon lives at `/opt/<your-domain>/`:

```
/opt/<your-domain>/
  backend/
    .env              # Backend environment variables
    src/
    prisma/
  frontend/
    .env              # Frontend environment variables (baked at build)
    dist/             # Built frontend (served by Nginx)
  deployment-info.txt # Credentials, ports, and diagnostic commands
  patchmon-install.log
```

Environment Variables

The setup script generates a `backend/.env` with sensible defaults. You can customise it after installation.

File location: `/opt/<your-domain>/backend/.env`

Variables set by the script

Variable	What the script sets
`DATABASE_URL`	Full connection string with generated password
`JWT_SECRET`	Auto-generated 50-character secret
`CORS_ORIGIN`	`<protocol>://<your-domain>`
`PORT`	Random port between 3001-3999
`REDIS_HOST`	`localhost`
`REDIS_PORT`	`6379`
`REDIS_USER`	Instance-specific Redis ACL user
`REDIS_PASSWORD`	Auto-generated password
`REDIS_DB`	Auto-detected available Redis database

Adding optional variables

To enable OIDC, adjust rate limits, configure TFA, or change other settings, add the relevant variables to `backend/.env` and restart the service.

For example, to enable OIDC SSO:

```
# Edit the .env file
sudo nano /opt/<your-domain>/backend/.env
```

Add at the bottom:

```
# OIDC / SSO
OIDC_ENABLED=true
OIDC_ISSUER_URL=https://auth.example.com
OIDC_CLIENT_ID=patchmon
OIDC_CLIENT_SECRET=your-client-secret
OIDC_REDIRECT_URI=https://patchmon.example.com/api/v1/auth/oidc/callback
```

```
OIDC_SCOPES=openid email profile groups
OIDC_AUTO_CREATE_USERS=true
```

Then restart:

```
sudo systemctl restart <your-domain>
```

Full list of optional variables

All optional environment variables are documented in the Docker `env.example` file and on the **Environment Variables** page. The same variables work for both Docker and native installations. Key categories include:

- **Authentication** - `JWT_EXPIRES_IN`, `JWT_REFRESH_EXPIRES_IN`, `SESSION_INACTIVITY_TIMEOUT_MINUTES`
- **Password policy** - `PASSWORD_MIN_LENGTH`, `PASSWORD_REQUIRE_UPPERCASE`, etc.
- **Account lockout** - `MAX_LOGIN_ATTEMPTS`, `LOCKOUT_DURATION_MINUTES`
- **Two-factor authentication** - `MAX_TFA_ATTEMPTS`, `TFA_REMEMBER_ME_EXPIRES_IN`, etc.
- **OIDC / SSO** - `OIDC_ENABLED`, `OIDC_ISSUER_URL`, `OIDC_CLIENT_ID`, etc.
- **Rate limiting** - `RATE_LIMIT_WINDOW_MS`, `RATE_LIMIT_MAX`, `AUTH_RATE_LIMIT_*`, `AGENT_RATE_LIMIT_*`
- **Database pool** - `DB_CONNECTION_LIMIT`, `DB_POOL_TIMEOUT`, `DB_IDLE_TIMEOUT`, etc.
- **Logging** - `LOG_LEVEL`, `ENABLE_LOGGING`
- **Network** - `ENABLE_HSTS`, `TRUST_PROXY`, `CORS_ORIGINS`
- **Encryption** - `AI_ENCRYPTION_KEY`, `SESSION_SECRET`
- **Timezone** - `TZ`
- **Body limits** - `JSON_BODY_LIMIT`, `AGENT_UPDATE_BODY_LIMIT`

> After any `.env` change, restart the service: `sudo systemctl restart <your-domain>`

Updating an Existing Installation

To update PatchMon to the latest version, re-run the setup script with `--update`:

```
sudo bash setup.sh --update
```

The update process:

1. Detects all existing PatchMon installations under `/opt/`

2. Lets you select which instance to update
3. **Backs up** the current code and database before making changes
4. Pulls the latest code from the selected branch/tag
5. Installs updated dependencies and rebuilds the frontend
6. Runs any new database migrations (with self-healing)
7. **Adds any missing environment variables** to `backend/.env` (preserves your existing values)
8. Updates the Nginx configuration with latest security improvements
9. Restarts the service

If the update fails, the script prints rollback instructions with the exact commands to restore from the backup.

Managing the Service

Replace `<your-domain>` with the domain/IP you used during installation (e.g. `patchmon.internal`).

Service commands

```
# Check status
systemctl status <your-domain>

# Restart
sudo systemctl restart <your-domain>

# Stop
sudo systemctl stop <your-domain>

# View logs (live)
journalctl -u <your-domain> -f

# View recent logs
journalctl -u <your-domain> --since "1 hour ago"
```

Other useful commands

```
# Test Nginx config
nginx -t && sudo systemctl reload nginx
```

```
# Check database connection
sudo -u <db-user> psql -d <db-name> -c "SELECT 1;"

# Check which port PatchMon is listening on
netstat -tlnp | grep <backend-port>

# View deployment info (credentials, ports, etc.)
cat /opt/<your-domain>/deployment-info.txt
```

Troubleshooting

Issue	Solution
Script fails with permission error	Run with `sudo bash setup.sh`
Service won't start	Check logs: `journalctl -u <your-domain> -n 50`
Redis authentication error	Verify `REDIS_USER` and `REDIS_PASSWORD` in `backend/.env` match Redis ACL. Run `redis-cli ACL LIST` to check
Database connection refused	Check PostgreSQL is running: `systemctl status postgresql`
SSL certificate issues	Run `certbot certificates` to check status. Renew with `certbot renew`
Nginx 502 Bad Gateway	Backend may not be running. Check `systemctl status <your-domain>` and the backend port
Migration failures	Check status: `cd /opt/<your-domain>/backend && npx prisma migrate status`
Port already in use	The script picks a random port (3001-3999). Edit `PORT` in `backend/.env` and update the Nginx config

For more help, see the **Troubleshooting** page or check the installation log:

```
cat /opt/<your-domain>/patchmon-install.log
```

Please note: This script was built to automate the deployment of PatchMon however our preferred method of installation is via Docker. This method is hard to support due to various parameters and changes within the OS such as versions of Nginx causing issues on the installer.

Anyway, do enjoy and I understand if you're like me ... want to see the files in plain sight that is being served as the app ;)

Revision #5
Created 2025-10-04 17:40:01 UTC by Iby
Updated 2026-04-23 21:47:49 UTC by Iby