Neuphlo

TECHNICAL WHITEPAPER

Run Neuphlo on your own infrastructure.

A privacy-first productivity platform you can self-host in minutes — paired to your workspace, billed by your plan, and able to run fully air-gapped with no data ever leaving your network.

Data sovereignty | Air-gapped ready | One-command install

01 — THE CASE FOR SELF-HOSTING

§ Why self-hosting matters

Most SaaS asks you to trust a vendor with your data. Self-hosting flips that relationship: the software runs where you decide, on hardware you control, behind your own firewall — and the vendor never sees a byte of your content.

For a growing number of organisations that trade-off is no longer optional. A hospital handling patient records, a law firm under client-confidentiality obligations, a defence contractor on an isolated network, or simply a European company that wants its data to stay on European soil — all of them face the same question: **can we use this tool without handing our data to someone else's cloud?**

Neuphlo's answer is yes. The exact same platform that runs at `app.neuphlo.com` can be deployed as a self-contained stack on your own server. It is not a stripped-down "community edition" or a separate codebase — it is the full product, including real-time sync, voice and video calling, file storage and even on-device AI, packaged to run anywhere Docker runs.

1	0	100%
command to install	data sent to Neuphlo	feature parity

What you get

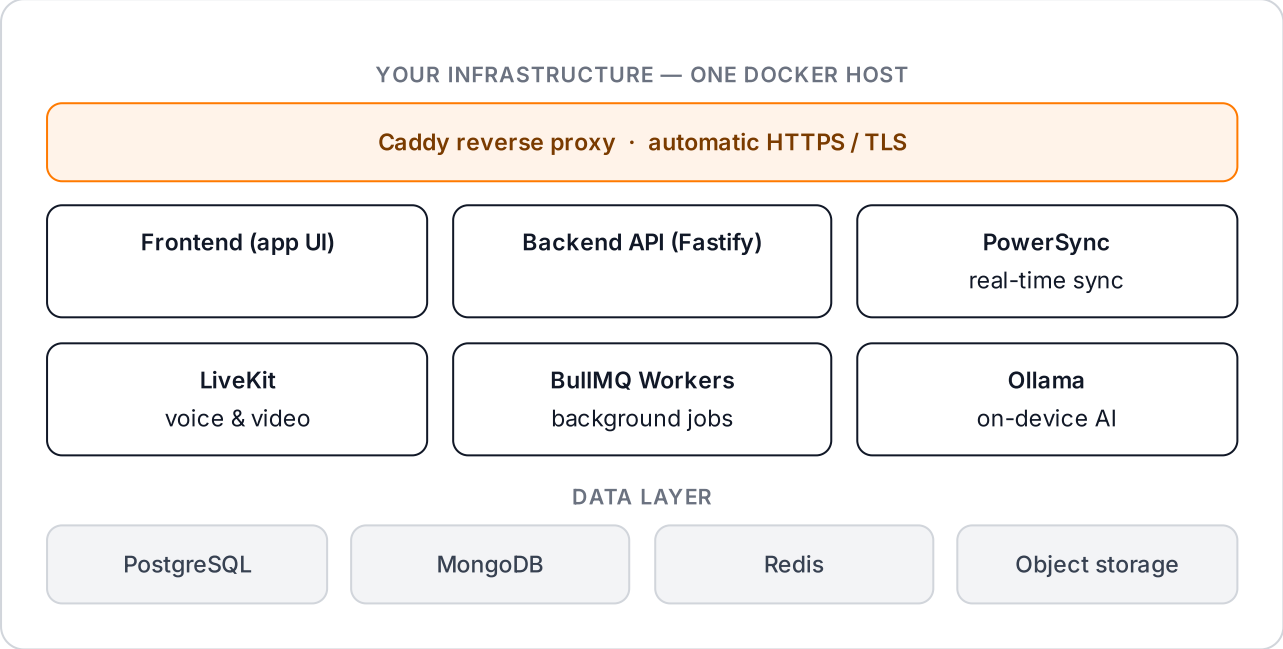
- **Complete data residency.** Your database, file uploads, message history and call media all live inside your own deployment. Choose the jurisdiction; choose the disk.
- **Air-gapped operation.** An offline licensing path lets an instance run with no outbound network access at all — verified locally, never phoning home.
- **Plan continuity.** A self-hosted instance is paired to a Neuphlo workspace and inherits that workspace's plan and seat count, so billing and entitlements stay in one place.
- **Owner-controlled lifecycle.** Pair, monitor and revoke deployments from a single panel inside your workspace.

> **This document** walks through the architecture of a self-hosted Neuphlo deployment, the two ways to license an instance (online and offline / air-gapped), the security model, and what it takes to get a stack running. The screenshots throughout are taken directly from the product.

02 — HOW IT FITS TOGETHER

§ Architecture at a glance

A self-hosted Neuphlo deployment is a single Docker Compose stack. Everything the platform needs to run is included — there are no hidden cloud dependencies for day-to-day operation.



The proxy layer (Caddy) terminates TLS and issues certificates automatically. The application layer serves the UI, the REST API, the real-time sync engine (PowerSync), the WebRTC media server for calls (LiveKit) and the background job workers. Beneath that sits the data layer: PostgreSQL for relational data, MongoDB for sync bucket storage, Redis for queues and caching, and S3-compatible object storage for file uploads.

Notably, the stack bundles **Ollama** — meaning AI features can run entirely on your own hardware, with prompts and documents never leaving the deployment. For privacy-sensitive environments this closes the last common "but the AI sees everything" gap.

Self-contained by design

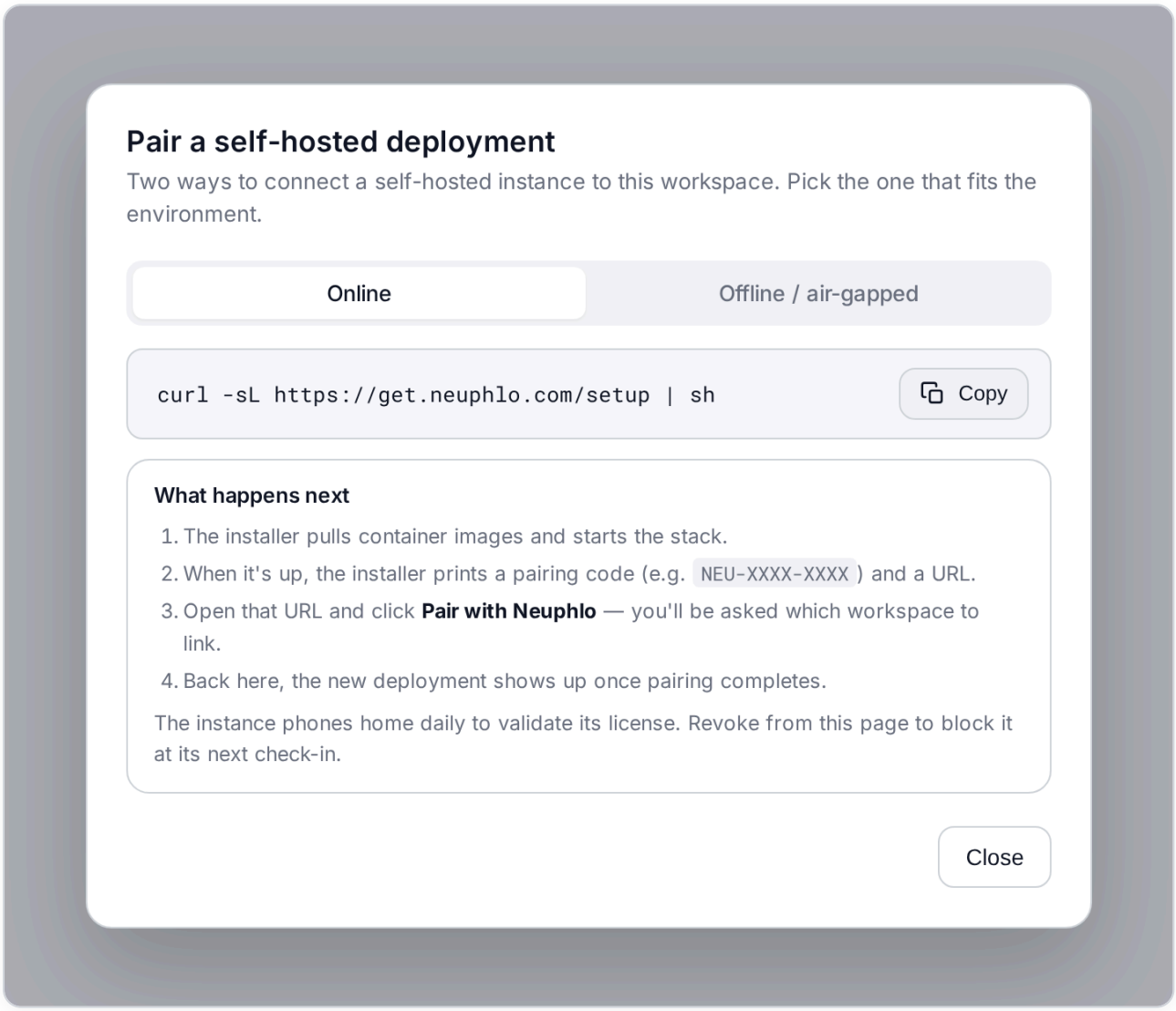
Because the stack ships email (a bundled mail service for local testing), AI, sync and media in the box, a fresh instance is functional the moment it boots — even before it is paired. The only thing it needs from Neuphlo is a license confirming which plan it runs under, and even that can be delivered offline.

03 — CONNECTING AN INSTANCE

§ Pairing: the online path

A self-hosted instance is linked to a Neuphlo workspace through **pairing**. The recommended path is a device-style flow that takes about a minute and never requires copying secrets by hand.

It starts with one command on your server. From the **Self-Host** panel in your workspace, the owner copies the installer command and runs it on the target machine:

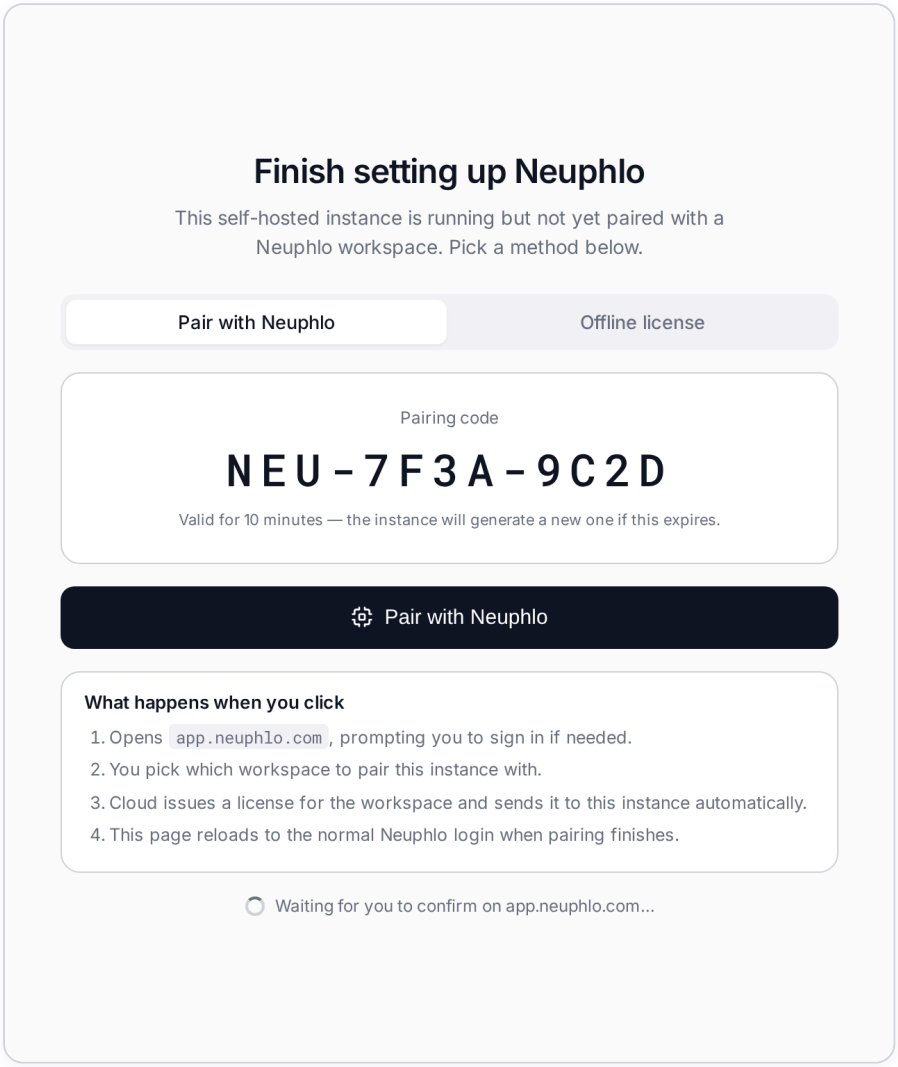


Preferences › Self-Host › Pair deployment. The online tab provides the one-line installer and explains the device-pairing flow.

The installer pulls the container images and starts the stack. Because the instance is not yet paired, it boots into its own setup screen — and serves nothing else until a license is in place.

§ The instance shows a code

On first boot the instance presents a setup page that displays a short-lived pairing code and a **Pair with Neuphlo** button. This is the only page the instance serves while unpaired; every other route returns a 503 until a license arrives.

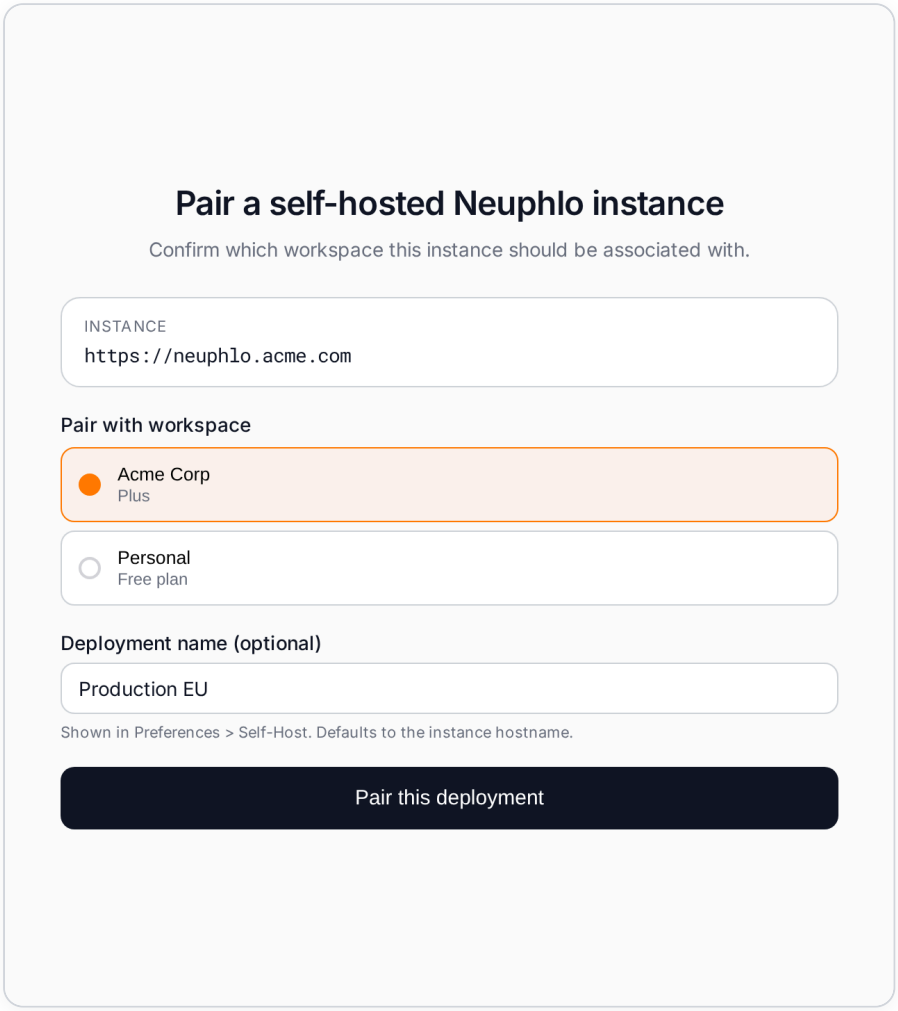


The instance's own /setup screen. The pairing code is valid for ten minutes; the instance generates a fresh one automatically if it expires.

Clicking the button sends the owner to `app.neuphlo.com` to finish linking — the next step. Meanwhile the instance quietly polls, waiting to receive its license.

§ Confirming on the cloud side

On app.neuphlo.com the workspace owner confirms which workspace the instance should belong to. Neuphlo then issues a license for that workspace and pushes it to the waiting instance automatically.



The cloud-side confirmation. Only workspace owners can pair, and the instance inherits the chosen workspace's plan and seats.

Within a few seconds the instance polls, receives its license, and flips into full operation — the setup page reloads to the normal Neuphlo login, hosted on your own domain. No tokens are emailed, pasted or stored in shell history; the secret material is exchanged machine-to-machine.

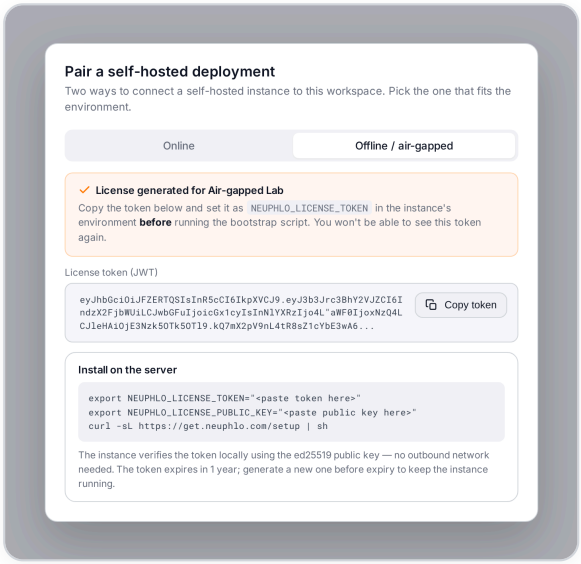
> **Why a code-and-confirm flow?** The single-use, ten-minute code plus owner authentication on the cloud means possession of the installer command alone is never enough to attach an instance to someone's workspace — the device shows a code, and a trusted human approves it elsewhere.

04 — FOR ISOLATED NETWORKS

§ Offline & air-gapped licensing

Some environments cannot reach the public internet at all — and must stay that way. For these, Neuphlo issues a signed license token the instance verifies on its own.

From the same Self-Host panel, an owner switches to the **Offline / air-gapped** tab, names the deployment, and generates a license — Neuphlo returns a signed token plus the public key needed to verify it:



Generating an offline license. The token is shown once; it is set as an environment variable on the instance before first boot.

The token is a JWT signed with an **ed25519** key. The instance carries Neuphlo's public key and verifies the signature locally, so it confirms it holds a valid plan *without any outbound connection* — no check-in, no telemetry, nothing to block on a firewall. Tokens are valid for one year; a fresh one is generated before expiry.

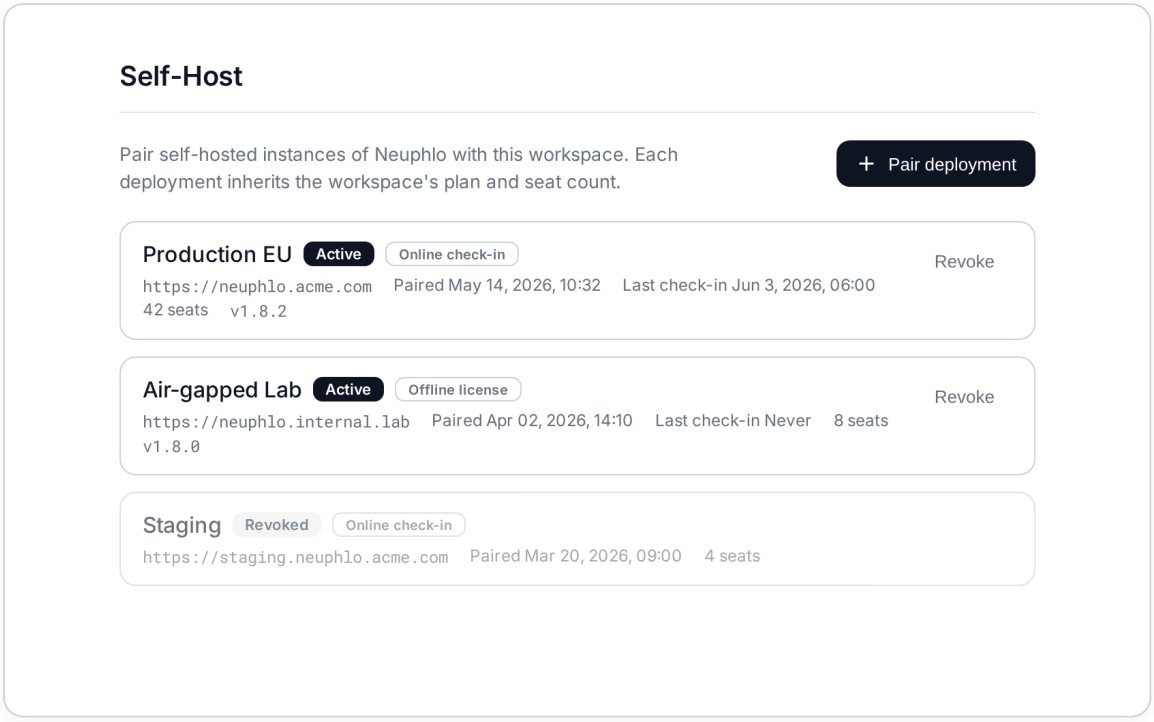
Online vs. offline at a glance

	ONLINE CHECK-IN	OFFLINE / AIR-GAPPED
Network	Outbound HTTPS to Neuphlo	None required after install
Validation	Daily check-in	Local ed25519 signature check
Setup	Device-style pairing code	Paste signed token, set env var
Renewal	Automatic on check-in	New token before 1-year expiry
Best for	Servers with internet access	Isolated or regulated networks

05 — STAYING IN CONTROL

§ Licensing & lifecycle management

Every paired instance appears in one place: the Self-Host panel in your workspace. From there an owner can see status, mode, seat usage and version — and pull the plug on any deployment.



Preferences › Self-Host. Active and revoked deployments, each showing its licensing mode, instance URL, pairing date, last check-in, seat count and version.

How enforcement works

A self-hosted instance runs in one of three states. **Unpaired:** the instance is up but has no license, so it serves only the setup surface and returns a 503 for everything else. **Paired:** a valid license is cached and the platform runs normally. **Invalid:** if a license is revoked or its grace period lapses, the instance stops serving the app but still allows re-validation so it can recover.

Revoking a deployment is immediate from the owner's side and takes effect on the instance at its next daily check-in. The license key is invalidated permanently — a revoked instance cannot quietly re-activate itself.

Grace period

If the license server is unreachable, an online instance keeps running on its cached license for up to **30 days** — brief outages never take your team offline.

Seat inheritance

Each deployment draws its plan tier and seat count from the workspace it is paired to, so entitlements and billing stay centralised.

§ Security & data sovereignty

The architecture is built so that **your content never transits Neuphlo's infrastructure**. In online mode the only outbound traffic is a small daily license check; in offline mode there is none. Application data — messages, files, call media, AI prompts — is processed and stored exclusively within your deployment.

- **TLS by default.** Caddy provisions and renews HTTPS certificates automatically for every public subdomain.
- **Locally verifiable licensing.** Offline tokens are signed with ed25519 and checked against a bundled public key — no trust-on-every-request call to a remote server.
- **Owner-gated pairing.** Attaching an instance to a workspace requires owner authentication on the cloud; single-use, ten-minute codes prevent replay.
- **Per-environment isolation.** Secrets (auth keys, sync secrets, database credentials) are unique per deployment and never shared across environments.
- **On-device AI option.** With bundled Ollama, AI never requires sending data to a third-party model provider.

§ Getting started

A complete stack runs comfortably on a modest single server. The published reference deployment runs on Hetzner Cloud for roughly **€13–14 per month** in infrastructure, and installs from a single command:

```
curl -sL https://get.neuphlo.com/setup | sh
```

An interactive installer is also available for guided setup of SMTP, OAuth providers, hostname and license mode. Everything the wizard configures can equally be set by editing environment variables and restarting the stack.

> **In short:** Neuphlo self-hosting gives you the full platform, on your own infrastructure, in your own jurisdiction — with a licensing model flexible enough for both a standard internet-connected server and a fully air-gapped secure network.

Neuphlo

Learn more & deploy → **neuphlo.com**