Practical architecture for combining AdGuard Home and Nginx Proxy Manager

Below is a **clear, actionable, scalable way** to combine **AdGuard Home** with **Nginx Proxy Manager** that mirrors the reasoning and quality of the Pi-hole + NPM setup you already reviewed — but adapted for the architectural and protocol differences that AdGuard Home introduces.(adguard.com)

In short:

- **AdGuard Home** becomes your DNS source of truth (and optionally encrypted DNS — DoH/DoT)
- **Nginx Proxy Manager (NPM)** handles HTTP/HTTPS for web interfaces and services
- You separate DNS from web traffic and avoid port conflicts

? Core Principles

1) Let AdGuard Home handle DNS, not HTTP

AdGuard Home's strength is network-wide DNS + filter/parental control features. It natively supports encrypted DNS protocols **DNS-over-HTTPS (DoH)** and **DNS-over-TLS (DoT)** if you enable them.(Virtualization Howto)

- DNS services: port 53 (UDP/TCP), DoH, DoT
- **Do not expose AdGuard Home's admin web UI directly on 80/443** ? NPM should own ports 80/443 so web traffic is centralized

---

2) Run AdGuard Home Admin UI on a different port

By default AdGuard Home binds its web UI on ports like 80/3000/any custom — this can conflict with NPM. Run AdGuard Home's web interface on a port **other than 80/443** (e.g., 8080, 3000,

8443).(Reddit)

Later NPM will proxy that internal port so external access can be via a friendly domain, with clean HTTPS.

---

3) Use AdGuard Home's DNS rewrites for local hostnames

Instead of wildcard DNS (which can cause SSL and service routing ambiguity), explicitly configure local DNS entries inside AdGuard Home:

```
myservice.lab.zn80.net   ? 192.168.10.105
pihole.lab.zn80.net     ? 192.168.10.105
```

These rewrites are equivalent to Pi-hole's A records and tell clients exactly where to send requests without relying on wildcard responses. (AdGuard Home supports rewrites but not exactly the same UI as Pi-hole — you use **DNS Rewrite Rules**.)(Chris Kirby)

---

4) Let NPM handle HTTP/HTTPS and certificates

All external web access, including admin panels and service frontends, should go through NPM:

For each service:

Public URL	NPM Proxy Host config	Internal Address
`https://adguard.lab.zn80.net`	Proxy to internal AdGuard Home admin port (e.g., 8080)	e.g., `192.168.10.105:8080`
`https://grafana.lab.zn80.net`	Proxy to Grafana	e.g., `192.168.10.110:3000`
etc.	…	…

This achieves:

- Single entry point for all web UIs
- Centralized SSL certificate generation & renewal (Let's Encrypt)
- Clean internal service structure

? Step-by-Step Setup

A) Configure AdGuard Home DNS

1. Install and run AdGuard Home on your homelab host.(adguard.com)

2. In AdGuard Home settings:

   - Enable normal DNS on 53
   - Optionally enable DoH/DoT if you want encrypted resolution
   - Set trusted proxies if using a reverse proxy for DoH (advanced)(GitHub)

3. Create **DNS rewrites** for your internal services ? local hostnames map to NPM IP.

Clients on your network should use AdGuard Home as their DNS server (via DHCP or router config). This ensures they resolve internal domains before going upstream.

B) Change AdGuard Home Admin UI Port

AdGuard Home lets you configure its HTTP admin port — do **not** leave it on 80/443 if NPM is going to manage those. It's common to use something like:

```
192.168.10.105:8080
```

(NPM will proxy this later.)

C) Configure Nginx Proxy Manager (NPM)

C.1 – Proxy Host Rules

For each service/web UI you want externally reachable:

1. Create a **Proxy Host** in NPM
2. Domain: e.g., `adguard.lab.zn80.net`
3. Forward hostname: `192.168.10.105`
4. Forward port: your AdGuard Home admin UI port (e.g., 8080)

5. Enable SSL ? request/renew certificate via Let’s Encrypt

Repeat for all other services (Home Assistant, Jellyfin, etc.).

> “ NPM will listen on ports **80/443** exclusively and terminate HTTPS.

D) Avoid Proxying DNS Port

Do **not** proxy port 53 through NPM — DNS is not an HTTP service and must stay direct.

E) SSL Certificates

- Use Let’s Encrypt through NPM for all publicly reachable subdomains.
- If you want internal-only names (e.g., `.lab.zn80.net`) with valid TLS and Let’s Encrypt fails:
  - Either use a DNS challenge
  - Or use a local/internal CA trusted by your clients
- Certificates must match the hostname used in the browser to avoid security warnings.

? Why This Is the Best Setup

? DNS features remain powerful and flexible

AdGuard Home can serve DNS, including local rewrites, redirection, and encrypted DNS protocols — all without conflicting with your web reverse proxy.(Virtualization Howto)

? NPM becomes the single source for secure web access

All internal UIs and apps are accessible through friendly URLs with proper certificates.

? Separation of concerns

- DNS: AdGuard Home

- HTTP/HTTPS: NPM This simplifies maintenance and improves security and debugging.

? Additional Notes

DNS over HTTPS (DoH) / DNS over TLS (DoT)

If you enable these in AdGuard Home and want clients to use them, they are handled independently of NPM. NPM can proxy HTTP-related admin UIs, but encrypted DNS paths are not HTTP in the same sense — you won't reverse-proxy them unless you deliberately expose DoH endpoints.

Trusted reverse proxy headers

If you proxy AdGuard Home through NPM and use features tied to client IPs (e.g., logging real client IPs), consider enabling forwarding headers (`X-Real-IP`, etc.) and configuring trusted proxies.(GitHub)

Local DNS resolution in NPM

Some setups report issues when NPM resolves local DNS from AdGuard Home in certain container environments — make sure containers can reach your DNS server or configure fallback resolvers.(forums.truenas.com)

? Final Architecture (Recommended)

```
[Router DHCP] ? gives clients DNS of AdGuard Home

[Client]
   |
   +-- DNS query to AdGuard Home --> resolves local services
   |
   +-- HTTP(S) request to NPM (80/443) -->
        |
        +-- Reverse proxy to internal services (AdGuard UI, apps)
```

This pattern keeps DNS resolution stable and web/HTTPS handling secure, centralized, and scalable as your services grow.

If you want, I can provide a **concrete example NPM proxy configuration** for AdGuard Home (with headers and SSL settings) along with how to handle DoH/DoT clients on Android or browsers —

just let me know your priorities.

Revision #7

Created 2025-12-15 19:52:54 UTC by Carsten

Updated 2026-02-14 15:36:00 UTC by Carsten