

h04-media- entertainment

- [00-chapter-intro.md](#)
- [01-plex.md](#)
- [02-jellyfin.md](#)
- [03-emby.md](#)
- [04-tautulli.md](#)
- [05-immich.md](#)
- [06-audiobookshelf.md](#)
- [07-seerr.md](#)
- [09-threadfin.md](#)

00-chapter-intro.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Chapter
Introduction tags: [media, plex,
jellyfin, immich, audiobookshelf,
stash]

Overview

This chapter covers all media consumption and photo management services on Centerpoint. Media files themselves are not stored locally — they live on the UnRAID NAS (`192.168.1.119`) and are bind-mounted into containers from the NFS shares at `/mnt/Multimedia` and `/mnt/Photos` .

Shared Storage

NFS Mount	Contents	Consumers
<code>/mnt/Multimedia</code>	Movies, TV, adult content, audiobooks, podcasts	Plex, Jellyfin, Emby, Audiobookshelf, Stash
<code>/mnt/Photos</code>	Photo library, Google Photos imports	Immich

Services in This Chapter

Service	Container(s)	Status	Purpose
Plex	plex	Active	Primary media server
Jellyfin	jellyfin	Active	Open-source media server (secondary)
Emby	emby	Offline	Legacy media server — not running
Tautulli	tautulli	Active	Plex analytics and monitoring
Immich	immich_server, immich_machine_learning, immich_postgres, immich_redis	Active	Photo & video library
Audiobookshelf	audiobookshelf	Active	Audiobooks and podcasts
Seerr	seerr	Active	Media request management
Stash	stash, stash-vr	Active	Adult media library + VR frontend
Threadfin	threadfin	Active	IPTV M3U proxy for Plex/Jellyfin

Last Updated: 2026-06-16

01-plex.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Plex tags:
[plex, media, streaming]

Overview

Plex Media Server is the primary media server for the homelab, serving movies, TV shows, and other content to Plex clients on the local network and remotely. It uses Plex's own account-based authentication — no Authentik ForwardAuth is applied. GeoBlock is active on the external route.

A second Plex instance running on another device (`192.168.1.186`) is routed via Traefik's file provider as `plex2.jeeves5454.ddns.net`.

Access

Type	URL	Notes
External	<code>https://plex.jeeves5454.ddns.net</code>	GeoBlock (CA/US/IN), no Authentik — Plex account auth
Direct	<code>http://192.168.1.85:32400</code>	LAN direct access

No `*.home.local` internal Traefik route — Plex is accessed externally or by direct IP on the LAN.

Configuration

Image: `lscr.io/linuxserver/plex:latest` **Compose project:** Standalone (managed via Portainer)

Ports

Port	Protocol	Purpose
<code>32400</code>	TCP	Plex Media Server API and web UI

Traefik Labels

```
traefik.enable: "true"
traefik.http.routers.plex.rule: Host(`plex.jeeves5454.ddns.net`)
traefik.http.routers.plex.entrypoints: websecure
traefik.http.routers.plex.tls.certresolver: letsencrypt
traefik.http.routers.plex.middlewares: plex-geoblock@file,plex-headers
traefik.http.middlewares.plex-headers.headers.customrequestheaders.X-Forwarded-Proto: https
traefik.http.services.plex.loadbalancer.server.port: 32400
```

The `plex-headers` middleware injects `X-Forwarded-Proto: https` — required for Plex to generate correct redirect and callback URLs when behind a reverse proxy.

Environment Variables

Variable	Value	Purpose
<code>PUID</code>	<code>1000</code>	Run as user ID 1000
<code>PGID</code>	<code>1000</code>	Run as group ID 1000
<code>TZ</code>	<code>America/Toronto</code>	Timezone
<code>VERSION</code>	<code>docker</code>	Use the latest Plex from Docker Hub
<code>ADVERTISE_IP</code>	<code>https://plex.jeeves5454.ddns.net:443</code>	External URL Plex advertises to clients

`ADVERTISE_IP` must match the externally reachable URL for remote streaming to work correctly when behind Traefik.

Volumes / Bind Mounts

Host Path	Container Path	Purpose
-----------	----------------	---------

/home/jeeves/docker/plex/config	/config	Plex database, metadata, settings
/mnt/Multimedia	/Multimedia	All media files (NFS from UnRAID)
/home/jeeves/docker	/docker	Utility bind (admin access)

Networks

Network	Purpose
traefik-net	Exposes Plex via Traefik external route

Dependencies

- NFS mount /mnt/Multimedia must be healthy — if UnRAID is down, Plex will show libraries as unavailable
- Internet connectivity for metadata scraping and Plex account authentication

Notes / Gotchas

- Plex uses its own authentication (Plex.tv account). Authentik ForwardAuth is not applied to the external route because it would break the Plex app auth flow.
- VERSION=docker always pulls the latest Plex release on container start — pin to a specific build number (e.g. VERSION=1.41.x.xxxx-xxxxxxxxxx) for stability.
- Plex transcoding temp files are written to /tmp inside the container. For heavy transcoding workloads, consider adding a bind mount for the transcode directory to a fast local path.
- Tautulli connects to Plex via the direct API at http://plex:32400 over the traefik-net bridge.

02-jellyfin.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Jellyfin tags:
[jellyfin, media, streaming, open-
source]

Overview

Jellyfin is a free and open-source media server used as a secondary streaming solution alongside Plex. It serves TV shows and movies from the same UnRAID NFS share and is accessible on the LAN only.

Access

Type	URL	Notes
Internal	<code>https://jellyfin.home.local</code>	LAN access via Step-CA TLS
Direct	<code>http://192.168.1.85:8096</code>	LAN direct access

No external (internet-facing) route.

Configuration

Image: `jellyfin/jellyfin:latest` **Compose project:** Standalone (managed via Portainer)

Ports

Port	Protocol	Purpose
8096	TCP	Jellyfin web UI and API (host-bound)

Traefik Labels

```
traefik.enable: "true"
traefik.http.routers.jellyfin.rule: Host(`jellyfin.home.local`)
traefik.http.routers.jellyfin.entrypoints: websecure
traefik.http.routers.jellyfin.tls: "true"
traefik.http.routers.jellyfin.tls.certresolver: step-ca
traefik.http.services.jellyfin.loadbalancer.server.port: 8096
```

Environment Variables

Variable	Value	Purpose
PUID	1000	User ID
PGID	1000	Group ID
TZ	America/Toronto	Timezone

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/jellyfin/config	/config	Jellyfin database and settings
/home/jeeves/docker/jellyfin/cache	/cache	Thumbnail and transcode cache
/mnt/Multimedia/TV	/media/tv:ro	TV series (NFS from UnRAID, read-only)
/mnt/Multimedia/Movies	/media/movies:ro	Movies (NFS from UnRAID, read-only)

Networks

Network	Purpose
traefik-net	Exposes Jellyfin via Traefik

Dependencies

- NFS mount `/mnt/Multimedia` must be healthy
- Step-CA for `jellyfin.home.local` TLS certificate

Notes / Gotchas

- Media mounts are read-only (`:ro`) — Jellyfin cannot modify or delete source files.
- Jellyfin and Plex both mount the same NFS directories. Metadata and library scans are independent between the two servers.
- Hardware transcoding is available on Jellyfin but not explicitly configured — the `jellyfin/jellyfin:latest` image supports it via FFmpeg with Intel/NVIDIA hooks. Configure in Admin → Dashboard → Playback if needed.
- Port `8096` is also used by Emby (currently offline). Only one can be host-bound at a time — ensure Emby is stopped before starting Jellyfin and vice versa.

Last Updated: 2026-06-16

03-emby.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Emby
(Offline) tags: [emby, media,
offline]

Overview

“ **Status: OFFLINE** — The Emby container is present on Centerpoint but is not currently running. It has been superseded by Jellyfin for open-source media serving needs. The container and its configuration are retained.

Emby was previously used as an alternative media server. Its configuration and data remain intact and the container can be restarted if needed, subject to the port `8096` conflict with Jellyfin (see Notes).

Access

Type	URL	Notes
Internal	<code>https://emby.home.local</code>	Traefik labels present — not routed while offline

Configuration

Image: `emby/embyserver:latest` **Status:** `exited` (container stopped)

Traefik Labels (inactive while container is stopped)

```
traefik.enable: "true"
traefik.http.routers.emby.rule: Host(`emby.home.local`)
traefik.http.routers.emby.entrypoints: websecure
traefik.http.routers.emby.tls.certresolver: step-ca
traefik.http.services.emby.loadbalancer.server.port: 8096
```

Volumes / Bind Mounts

Host Path	Container Path	Purpose
<code>/home/jeeves/docker/emby/config</code>	<code>/config</code>	Emby database and library config
<code>/mnt/Multimedia/ST</code>	<code>/data</code>	Media files (same NFS as Stash)

Notes / Gotchas

- Emby and Jellyfin both use port `8096` internally. Only one can be host-bound at a time — starting Emby while Jellyfin is running will fail with a port conflict.
- If reactivating Emby, stop Jellyfin first or remap Emby to a different host port.
- The `emby/embyserver:latest` image tag pulls the latest release — verify compatibility with the existing config before restarting after a long pause.

Last Updated: 2026-06-16

04-tautulli.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Tautulli tags:
[tautulli, plex, monitoring,
analytics]

Overview

Tautulli is the monitoring and analytics companion for Plex Media Server. It tracks play history, user activity, and library statistics, and can send notifications (email, Telegram, etc.) on media events such as new content additions, playback starts, or user logins.

Access

Type	URL	Notes
Internal	<code>https://tautulli.home.local</code>	LAN access via Step-CA TLS

No external route — LAN and Tailscale access only.

Configuration

Image: `lscr.io/linuxserver/tautulli:latest` **Compose project:** Standalone (managed via Portainer)

Traefik Labels

```
traefik.enable: "true"
traefik.http.routers.tautulli.rule: Host(`tautulli.home.local`)
traefik.http.routers.tautulli.entrypoints: websecure
traefik.http.routers.tautulli.tls.certresolver: step-ca
traefik.http.services.tautulli.loadbalancer.server.port: 8181
```

Environment Variables

Variable	Value	Purpose
PUID	1000	User ID
PGID	1000	Group ID
TZ	America/Toronto	Timezone

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/Tautulli/local_tautulli	/config	Tautulli database and configuration

Networks

Network	Purpose
traefik-net	Exposes Tautulli UI; also reaches Plex container

Dependencies

- Plex Media Server — Tautulli connects to the Plex API to retrieve playback data. Configure the Plex connection in Tautulli Settings → Plex Media Server using the container hostname `plex` and port `32400`.

Notes / Gotchas

- Tautulli's database (`tautulli.db`) contains full play history. Back it up before upgrades.
- Notification agents (Telegram, email, etc.) and their credentials are stored in the Tautulli database. If the `/config` bind mount is lost, all notification integrations must be reconfigured.

Last Updated: 2026-06-16

05-immich.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Immich tags:
[immich, photos, media, cuda,
gpu, self-hosted]

Overview

Immich is the self-hosted photo and video library for the homelab — a local alternative to Google Photos. It provides automatic mobile backup, face recognition, smart search, and album management. The machine learning container runs on the RTX 5080 via CUDA for accelerated facial recognition and CLIP-based smart search.

Photo storage lives on the UnRAID NAS NFS mount at `/mnt/Photos`.

Access

Type	URL	Notes
Internal	<code>https://photos.home.local</code>	LAN access via Step-CA TLS
External	<code>https://photos.jeevesconsults.ca</code>	GeoBlock (CA/US/IN) + CrowdSec — no Authentik ForwardAuth

Immich uses its own user authentication — Authentik ForwardAuth is not applied because it would break the mobile app OAuth flow.

Containers in This Stack

Container	Image	GPU	Role
immich_server	ghcr.io/immich-app/immich-server:release	No	Main API and web server
immich_machine_learning	ghcr.io/immich-app/immich-machine-learning:release-cuda	Yes	Face recognition + CLIP search
immich_postgres	ghcr.io/immich-app/postgres:14-vectorchord0.4.3-pgvectors0.2.0	No	PostgreSQL with pgvector.rs
immich_redis	valkey/valkey:8-bookworm	No	Job queue and cache

Configuration

Compose file: /home/jeeves/docker/immich/docker-compose.yml **Compose project:** immich

Ports

Port	Protocol	Purpose
2283	TCP	Immich web UI and API (host-bound)

Traefik Labels

```
# External route
traefik.http.routers.immich.rule: Host(`photos.jeevesconsults.ca`)
traefik.http.routers.immich.entrypoints: websecure
traefik.http.routers.immich.tls.certresolver: letsencrypt
traefik.http.routers.immich.middlewares: plex-geoblock@file,crowdsec-bouncer@file,immich-headers
traefik.http.middlewares.immich-headers.headers.customrequestheaders.X-Forwarded-Proto: https

# Internal route
traefik.http.routers.immich-internal.rule: Host(`photos.home.local`)
traefik.http.routers.immich-internal.entrypoints: websecure
```

traefik.http.routers.immich-internal.tls.certresolver: step-ca

traefik.http.services.immich.loadbalancer.server.port: 2283

Key Environment Variables (.env file)

Variable	Value / Notes
UPLOAD_LOCATION	/mnt/Photos/immich-library
DB_HOSTNAME	immich_postgres
DB_USERNAME	postgres
DB_PASSWORD	REDACTED
DB_DATABASE_NAME	immich
REDIS_HOSTNAME	immich_redis
TZ	America/Toronto
IMMICH_VERSION	release (pinned to latest stable)

Volumes / Bind Mounts

Host Path / Volume	Container Path	Purpose
/mnt/Photos/immich-library	/data	Primary upload library (NFS)
/mnt/Photos/Plex	/mnt/Photos/Plex:rw	Plex photo library (external library)
/mnt/Photos/Google_Photos	/mnt/Photos/Google_Photos:rw	Google Photos import folder
/etc/localtime	/etc/localtime:ro	Host timezone sync
model-cache (named volume)	/cache	ML model weight cache (machine learning container)
\${DB_DATA_LOCATION} (from .env)	/var/lib/postgresql/data	PostgreSQL data

Sub-section: Machine Learning (CUDA)

immich_machine_learning runs with runtime: nvidia, giving it access to the RTX 5080 for:

- **Face detection and recognition** — identifies and clusters faces across the library
- **CLIP embeddings** — powers smart search ("photos of dogs at the beach")

Environment Variable	Value	Purpose
----------------------	-------	---------

NVIDIA_VISIBLE_DEVICES	all	GPU access
NVIDIA_DRIVER_CAPABILITIES	compute,utility	CUDA compute caps
MACHINE_LEARNING_DEVICE_ID	0	Use GPU device 0

ML model files are cached in the `model-cache` named Docker volume. Models are downloaded from HuggingFace on first use and cached for subsequent runs.

Sub-section: PostgreSQL (pgvecto.rs)

Immich uses a custom PostgreSQL 14 image with the `pgvecto.rs` and `pgvectors` extensions pre-installed. These extensions power the vector similarity search that underlies CLIP smart search and face clustering.

```
ghcr.io/immich-app/postgres:14-vectorchord0.4.3-pgvectors0.2.0
```

The `shm_size: 128mb` allocation is required for PostgreSQL's shared memory.

Sub-section: Redis / Valkey

`immich_redis` uses Valkey (the Redis fork) as the job queue and cache backend. It handles background job scheduling for ML processing, thumbnail generation, and library scans.

Dependencies

- NFS mounts `/mnt/Photos` (UnRAID) must be healthy
- `immich_postgres` → `immich_redis` → `immich_server` startup order
- NVIDIA container runtime for `immich_machine_learning`

Notes / Gotchas

- Immich does not use Authentik ForwardAuth — it has built-in multi-user auth. The mobile app connects directly to `https://photos.jeevesconsults.ca`.
- The `X-Forwarded-Proto: https` header (`immich-headers` middleware) is required for Immich to generate correct share links and OAuth callbacks.
- The PostgreSQL image is pinned to a specific digest — do not change it arbitrarily. Immich releases are tied to specific database schema versions.
- Library scan and ML job processing can be CPU/GPU intensive. Schedule large library scans during off-peak hours via Admin → Jobs.

- External libraries (Plex photos, Google Photos) are read-only imports. Changes to files in these directories are picked up on the next library scan.
-

Last Updated: 2026-06-16

06-audiobookshelf.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page:
Audiobookshelf tags:
[audiobookshelf, audiobooks,
podcasts, media]

Overview

Audiobookshelf is a self-hosted audiobook and podcast server. It manages and streams audiobook and podcast content from the UnRAID NFS share, tracks listening progress across devices, and supports mobile apps for on-the-go listening.

A notable configuration detail: the Step-CA root certificate is injected into the container's trust store so that Audiobookshelf can make HTTPS calls to internal `*.home.local` services (e.g. for metadata lookups or integrations).

Access

Type	URL	Notes
External	<code>https://audio.jeeves5454.ddns.net</code>	GeoBlock (CA/US/IN) + CrowdSec — no Authentik

No *.home.local internal Traefik route — accessible externally or via direct LAN IP. Audiobookshelf uses its own account-based authentication.

Configuration

Image: ghcr.io/advplyr/audiobookshelf:latest **Compose project:** Standalone (managed via Portainer)

Traefik Labels

```
traefik.enable: "true"
traefik.http.routers.audio.rule: Host(`audio.jeeves5454.ddns.net`)
traefik.http.routers.audio.entrypoints: websecure
traefik.http.routers.audio.tls.certresolver: letsencrypt
traefik.http.routers.audio.middlewares: plex-geoblock@file,crowdsec-bouncer@file,audio-headers
traefik.http.middlewares.audio-headers.headers.customrequestheaders.X-Forwarded-Proto: https
traefik.http.services.audio.loadbalancer.server.port: 80
```

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/audiobookshelf/config	/config	App database and settings
/home/jeeves/docker/audiobookshelf/metadata	/metadata	Cover art and cached metadata
/mnt/Multimedia/Audio/Audio_Books	/audiobooks	Audiobook files (NFS from UnRAID)
/mnt/Multimedia/Audio/podcasts	/podcasts	Podcast episode files (NFS from UnRAID)
/home/jeeves/docker/step-ca/config/certs/root_ca.crt	/usr/local/share/ca-certificates/step-ca.crt:ro	Step-CA root cert trust injection

The Step-CA root certificate is bind-mounted into the container's CA trust directory, allowing Audiobookshelf to trust *.home.local TLS certificates when making outbound HTTPS requests to internal services.

Networks

Network	Purpose
traefik-net	Exposes the Audiobookshelf UI

Dependencies

- NFS mount `/mnt/Multimedia/Audio` must be healthy for content to be accessible
- Step-CA root certificate at `/home/jeeves/docker/step-ca/config/certs/root_ca.crt` must exist for the bind mount to succeed

Notes / Gotchas

- Audiobookshelf has its own user authentication — no Authentik ForwardAuth is applied. Manage users in the Admin panel.
- The `X-Forwarded-Proto: https` middleware is required for Audiobookshelf to generate correct callback and share URLs when behind Traefik.
- Podcast episode downloads are stored in `/mnt/Multimedia/Audio/podcasts`. Ensure UnRAID has sufficient free space for automatic podcast downloads.
- If Step-CA root cert is updated (e.g. after CA renewal), the container must be restarted to pick up the new cert from the bind mount.
- Listening progress is stored in the SQLite database under `/config`. Back up this directory before upgrades to preserve progress data.

Last Updated: 2026-06-16

07-seerr.md

kstack: book: Centerpoint Home
Lab chapter: Media &
Entertainment page: Seerr tags:
[seerr, overseerr, media-requests,
plex, jellyfin]

Overview

Seerr is a media request management application — a maintained fork of Overseerr. It provides a user-friendly interface for requesting movies and TV shows, which are then forwarded to the *arr stack (Radarr, Sonarr) for automated download and delivery to Plex and Jellyfin. It also surfaces Plex availability status so users can see what is already in the library before requesting.

Access

Type	URL	Notes
Internal	<code>https://seerr.home.local</code>	LAN access via Step-CA TLS
External	<code>https://seerr.jeeves5454.ddns.net</code>	Authentik SSO + GeoBlock + CrowdSec

Configuration

Image: ghcr.io/seerr-team/seerr:latest **Compose project:** arr stack (managed via Portainer alongside the *arr services)

Traefik Labels

```
# External route
traefik.http.routers.seerr-external.rule: Host(`seerr.jeeves5454.ddns.net`)
traefik.http.routers.seerr-external.entrypoints: websecure
traefik.http.routers.seerr-external.tls.certresolver: letsencrypt
traefik.http.routers.seerr-external.middlewares: authentik-auth@docker,plex-geoblock@file,crowdsec-bouncer@file
traefik.http.routers.seerr-external.service: seerr-svc

# Internal route
traefik.http.routers.seerr-internal.rule: Host(`seerr.home.local`)
traefik.http.routers.seerr-internal.entrypoints: websecure
traefik.http.routers.seerr-internal.tls.certresolver: step-ca
traefik.http.routers.seerr-internal.service: seerr-svc

traefik.http.services.seerr-svc.loadbalancer.server.port: 5055
```

Environment Variables

Variable	Value	Purpose
PORT	5055	Application port
LOG_LEVEL	debug	Logging verbosity
TZ	America/Toronto	Timezone
NODE_ENV	production	Runtime environment

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/arr/Seer/config	/app/config	Seerr database and configuration

Networks

Network	Purpose
<code>traefik-net</code>	Exposes Seerr UI
<code>media-network</code>	Internal network shared with *arr stack services (Sonarr, Radarr, etc.)

The `media-network` attachment allows Seerr to communicate directly with Sonarr, Radarr, and Plex by container hostname without going through Traefik.

Dependencies

- Plex Media Server — for library availability checks (connect via API)
- Sonarr, Radarr — to forward TV and movie requests
- Authentik for SSO on the external route

Notes / Gotchas

- Seerr is a fork of Overseerr — configuration and API are compatible but issue tracking and releases are at `github.com/seerr-team/seerr`.
- User permissions and notification settings are configured inside the Seerr UI. When Authentik is used for ForwardAuth, Seerr should be configured to use "Sign-in with Plex" or local accounts for user identity — not OIDC directly.
- Seerr's config database at `/app/config` contains API keys for Plex, Sonarr, and Radarr. Back it up before upgrades.

Last Updated: 2026-06-16

09-threadfin.md

kstack: book: Centerpoint Home

Lab chapter: Media &

Entertainment page: Threadfin

tags: [threadfin, iptv, m3u, xmltv, plex, jellyfin]

Overview

Threadfin is an M3U proxy and IPTV middleware that translates IPTV streams into a format consumable by Plex DVR and Jellyfin Live TV. It manages M3U playlists, XMLTV guide data, and presents a virtual tuner device to media servers via HDHR (HDHomeRun) emulation.

Access

Type	URL / Endpoint	Notes
Web UI	<code>http://192.168.1.85:34400</code>	Direct LAN access — no Traefik route
HDHR	<code>http://192.168.1.85:34400</code>	HDHomeRun device emulation endpoint

No Traefik route is configured for Threadfin — Plex and Jellyfin connect to it via direct IP and port on the LAN.

Configuration

Image: `fyb3roptik/threadfin:latest` **Compose project:** Standalone (managed via Portainer)

Ports

Port	Protocol	Purpose
<code>34400</code>	TCP	Threadfin web UI and HDHR endpoint (host-bound)

Environment Variables

Variable	Value	Purpose
<code>TZ</code>	<code>America/Toronto</code>	Timezone
<code>THREADFIN_PORT</code>	<code>34400</code>	Listen port
<code>THREADFIN_BRANCH</code>	<code>main</code>	Update channel
<code>THREADFIN_BIND_IP_ADDRESS</code>	<code>0.0.0.0</code>	Bind to all interfaces
<code>THREADFIN_DEBUG</code>	<code>0</code>	Debug logging off

Volumes / Bind Mounts

Host Path	Container Path	Purpose
<code>/home/jeeves/docker/threadfin/config</code>	<code>/home/threadfin/conf</code>	M3U playlists, XMLTV data, settings

Networks

Network	Purpose
<code>traefik-net</code>	Present (container is on traefik-net but no labels)

Threadfin is on `traefik-net` for network reachability but has no Traefik labels — it is accessed by direct IP.

Plex / Jellyfin Integration

Plex:

1. In Plex Settings → Live TV & DVR → Set Up Plex Tuner
2. Enter the Threadfin HDHR URL: `http://192.168.1.85:34400/device.xml`
3. Plex discovers the virtual tuner and available channels

Jellyfin:

1. Admin Dashboard → Live TV → Add Tuner Device
2. Select HDHomeRun and enter `http://192.168.1.85:34400`

Dependencies

- M3U source URL (IPTV provider) configured in Threadfin UI
- XMLTV guide source configured in Threadfin UI
- Plex or Jellyfin must be able to reach `192.168.1.85:34400` on the LAN

Notes / Gotchas

- M3U playlists and XMLTV guide data are refreshed on a schedule configured in the Threadfin UI. If the IPTV provider URL changes, update it in Threadfin settings — Plex and Jellyfin do not need to be reconfigured.
- Threadfin does not store or proxy video streams itself; it redirects the media server directly to the IPTV stream URL. Buffering and quality are determined by the stream source.
- Port `34400` is non-standard. Ensure no firewall rules block LAN-to-LAN traffic on this port between Plex/Jellyfin and Centerpoint.

Last Updated: 2026-06-16