

h07-finance

- [00-chapter-intro.md](#)
- [01-wallos.md](#)
- [02-splitpro.md](#)
- [03-ghostfolio.md](#)

00-chapter-intro.md

kstack: book: Centerpoint Home
Lab chapter: Finance page:
Chapter Introduction tags:
[finance, subscriptions, expenses,
investments, productivity]

Overview

This chapter covers personal finance and expense-tracking services. These tools handle subscription management, shared expense splitting, and investment portfolio tracking.

Services in This Chapter

Service	Container(s)	Purpose
Wallo	wallo	Subscription and recurring expense tracker
SplitPro	splitpro, splitpro-db	Shared expense splitting (Splitwise-alternative)
Ghostfolio	ghostfolio, ghostfolio-postgres, ghostfolio-redis	Open-source wealth management and portfolio tracker

01-wallos.md

kstack: book: Centerpoint Home
Lab chapter: Finance page: Wallos
tags: [wallos, subscriptions,
finance, sqlite]

Overview

Wallos is a self-hosted subscription and recurring payment tracker. It tracks monthly, annual, and custom-interval subscriptions, calculates total spending, and provides a visual dashboard of recurring costs. Data is stored in a SQLite database. Authentication is Wallos' own built-in user system — no Authentik integration. External-only route.

Access

Type	URL	Auth
External	<code>https://wallos.jeevesconsults.ca</code>	GeoBlock + CrowdSec (own auth)

No internal `home.local` route configured.

Configuration

Image: `bellamy/wallos:latest`

Runtime: PHP 8.3

Traefik Labels

```
traefik.http.routers.wallos.rule: Host(`wallos.jeevesconsults.ca`)
traefik.http.routers.wallos.entrypoints: websecure
traefik.http.routers.wallos.tls.certresolver: letsencrypt
traefik.http.routers.wallos.middlewares: plex-geoblock@file,crowdsec-bouncer@file,wallos-headers
traefik.http.middlewares.wallos-headers.headers.customrequestheaders.X-Forwarded-Proto: https
traefik.http.services.wallos.loadbalancer.server.port: 80
```

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/wallos/db	/var/www/html/db	SQLite database
/home/jeeves/docker/wallos/logos	/var/www/html/images/uploads/logos	Subscription logos

Wallos stores all data in a SQLite file within the `db` bind mount. No external database dependency.

Notes / Gotchas

- Wallos manages its own user accounts — no SSO or Authentik integration. Credentials are stored in the SQLite database.
- The `X-Forwarded-Proto: https` middleware is required for PHP to detect HTTPS correctly when behind a reverse proxy.
- Service logos are persisted in the `logos` bind mount. Both `db/` and `logos/` must be backed up together to fully restore the instance.
- No currency exchange rate API key is required by default — Wallos supports manual rate entry.

Last Updated: 2026-06-17

02-splitpro.md

kstack: book: Centerpoint Home

Lab chapter: Finance page:

SplitPro tags: [splitpro, expenses, splitting, authentik, postgres, step-ca]

Overview

SplitPro is a self-hosted shared expense splitting app — an open-source alternative to Splitwise. It tracks group expenses, calculates balances, and supports multi-currency via the Frankfurter exchange rate provider. Authentik OIDC provides authentication. Email invites are handled via Gmail SMTP. The Step-CA root certificate is injected so SplitPro can make HTTPS calls to internal `*.home.local` services. External-only route.

Access

Type	URL	Auth
External	<code>https://splitpro.jeevesconsults.ca</code>	Authentik OIDC + GeoBlock + CrowdSec

SplitPro handles its own OIDC redirect to Authentik — Traefik does not apply `authentik-auth@docker` ForwardAuth. Authentication is managed natively by the Next.js `next-auth` library.

Containers

Container	Image	Role
splitpro	ossapps/splitpro:latest	Web application
splitpro-db	ossapps/postgres:17.7-trixie	PostgreSQL 17 database

splitpro (application)

Runtime: Node.js 22 (Next.js)

Key environment variables:

Variable	Value / Notes
NEXTAUTH_URL	https://splitpro.jeevesconsults.ca
NEXTAUTH_SECRET	REDACTED
AUTHENTIK_ID	xA8pjyG7s4LsnmRKp9Wn9H9v8oUpawBAMvmEF9tT
AUTHENTIK_SECRET	REDACTED
AUTHENTIK_ISSUER	https://auth.jeevesconsults.ca/application/o/splitpro
OIDC_ALLOW_DANGEROUS_EMAIL_LINKING	1
DATABASE_URL	REDACTED (includes PostgreSQL password)
POSTGRES_USER	splitpro
POSTGRES_DB	splitpro
POSTGRES_PORT	5432
EMAIL_SERVER_HOST	smtp.gmail.com
EMAIL_SERVER_PORT	587
EMAIL_SERVER_USER	jeeves5454@gmail.com
EMAIL_SERVER_PASSWORD	REDACTED
FROM_EMAIL	splitpro@jeevesconsults.ca
ENABLE_SENDING_INVITES	true
DISABLE_EMAIL_SIGNUP	false
CURRENCY_RATE_PROVIDER	frankfurter
DEFAULT_HOMEPAGE	/balances
CACHE_RETENTION_INTERVAL	2 days
CLEAR_CACHE_CRON_RULE	0 2 * * 0 (Sunday 2am)
UPLOAD_MAX_FILE_SIZE_MB	10

Variable	Value / Notes
NODE_EXTRA_CA_CERTS	/etc/ssl/certs/step-ca-root.crt
PORT	3000

Bind mounts:

Host Path	Container Path	Purpose
/home/jeeves/docker/splitpro/uploads	/app/uploads	Receipt and document uploads
/home/jeeves/docker/step-ca/config/certs/root_ca.crt	/etc/ssl/certs/step-ca-root.crt	Step-CA root trust injection

splitpro-db (PostgreSQL 17)

Image: `ossapps/postgres:17.7-trixie`

A PostgreSQL 17 image published by the SplitPro project (based on the official `postgres:17` Debian image).

Variable	Value
POSTGRES_USER	splitpro
POSTGRES_DB	splitpro
POSTGRES_PASSWORD	REDACTED
POSTGRES_PORT	5432

Bind mounts:

Host Path	Container Path
/home/jeeves/docker/splitpro/db	/var/lib/postgresql/data

Traefik Labels

```
traefik.http.routers.splitpro-external.rule: Host(`splitpro.jeevesconsults.ca`)
traefik.http.routers.splitpro-external.entrypoints: websecure
traefik.http.routers.splitpro-external.tls.certresolver: letsencrypt
traefik.http.routers.splitpro-external.middlewares: plex-geoblock@file,crowdsec-bouncer@file
traefik.http.routers.splitpro-external.service: splitpro-svc
traefik.http.services.splitpro-svc.loadbalancer.server.port: 3000
```

Notes / Gotchas

- `OIDC_ALLOW_DANGEROUS_EMAIL_LINKING: 1` allows Authentik OIDC logins to link to existing accounts by matching email address. Required if users were created before OIDC was enabled, or if local signup is still active.
 - `NODE_EXTRA_CA_CERTS` injects the Step-CA root certificate into Node.js' TLS trust chain. This is required if SplitPro makes any HTTPS calls to internal `*.home.local` endpoints (e.g., if integrated with other internal services).
 - `DATABASE_URL` in the container environment contains the PostgreSQL password in plaintext — treat this as sensitive.
 - Currency rates are fetched from `api.frankfurter.app` (public, no API key required).
-

Last Updated: 2026-06-17

03-ghostfolio.md

kstack: book: Centerpoint Home
Lab chapter: Finance page:
Ghostfolio tags: [ghostfolio,
investments, portfolio, finance,
postgres, redis]

Overview

Ghostfolio is an open-source wealth management and investment portfolio tracker. It aggregates holdings across accounts, tracks performance, and visualises asset allocation. Internal-only access via Step-CA TLS — no external route is configured. Ghostfolio manages its own user authentication. The stack is three containers: the application, a PostgreSQL 15 database, and Redis for caching.

Access

Type	URL	Auth
Internal	<code>https://ghostfolio.home.local</code>	Ghostfolio own auth (Step-CA TLS)

No external route — accessible only on the LAN.

Containers

Container	Image	Role
ghostfolio	ghostfolio/ghostfolio:latest	Web application
ghostfolio-postgres	postgres:15-alpine	PostgreSQL 15 database
ghostfolio-redis	redis:alpine	Cache and session store

ghostfolio (application)

Runtime: Node.js 22

Key environment variables:

Variable	Value / Notes
DATABASE_URL	REDACTED (PostgreSQL connection string with password)
REDIS_HOST	ghostfolio-redis
REDIS_PORT	6379
ACCESS_TOKEN_SALT	REDACTED
JWT_SECRET_KEY	REDACTED
NODE_ENV	production
TZ	America/Toronto

No bind mounts — application state is stored entirely in PostgreSQL.

ghostfolio-postgres (PostgreSQL 15)

Image: postgres:15-alpine

Variable	Value
POSTGRES_DB	ghostfoliodb
POSTGRES_USER	ghostfoliouser
POSTGRES_PASSWORD	REDACTED
TZ	America/Toronto

Bind mounts:

Host Path	Container Path
/home/jeeves/docker/ghostfolio/postgres_data	/var/lib/postgresql/data

ghostfolio-redis

Image: `redis:alpine`

No authentication configured — network-isolated to the Ghostfolio internal stack network.

Bind mounts:

Host Path	Container Path
<code>/home/jeeves/docker/ghostfolio/redis_data</code>	<code>/data</code>

Traefik Labels

```
traefik.http.routers.ghostfolio.rule: Host(`ghostfolio.home.local`)
traefik.http.routers.ghostfolio.entrypoints: websecure
traefik.http.routers.ghostfolio.tls.certresolver: step-ca
traefik.http.services.ghostfolio.loadbalancer.server.port: 3333
```

Notes / Gotchas

- `ACCESS_TOKEN_SALT` and `JWT_SECRET_KEY` are critical secrets. If lost, all existing sessions and API tokens will be invalidated and cannot be recovered without reconfiguring Ghostfolio.
- Ghostfolio fetches market data from various external providers (Yahoo Finance, Coingecko, etc.). Market data API rate limits may apply depending on configured data sources.
- Portfolio data import supports CSV and JSON formats compatible with common brokerages. Check the Ghostfolio docs for the required format.
- Redis stores only ephemeral cache — data is not critical and does not need to be backed up. The `postgres_data` bind mount is the primary backup target.
- Port 3333 is the default Ghostfolio application port.

Last Updated: 2026-06-17