

AI-03095 Docker Self-Hosted Apps

👤 Owner	© Rigel Arcayan
☰ Tags	Project
🕒 Created time	@October 21, 2025 10:38 AM

If Immich impressed you, there's a whole world of self-hosted Docker apps that are just as useful — especially for media, backups, file organization, home automation, and personal productivity. Here are powerful apps most people don't know exist until they dive into self-hosting.

Photo & Memory Management (Like Immich)

App	Purpose
Photoprism	AI-powered photo organizer, alternative to Google Photos.
LibrePhotos	Open-source fork of Photoprism, privacy-focused.
Lychee	Simple, elegant self-hosted photo gallery for public/private albums.

Media Streaming (Your Own Netflix/Spotify)

App	Usage
Jellyfin	Free media server (movies, series, music, live TV).
Plex	Popular, polished version of Jellyfin (some paid features).
Emby	Media server with good UI; sits between Plex & Jellyfin.
Navidrome	Lightweight music streaming – your private Spotify.

Plex Meta Manager	Automatically updates Plex/Jellyfin libraries, posters, metadata.
-------------------	---

File Sharing, Cloud Storage & Backup

App	Purpose
Nextcloud	Self-hosted Google Drive + Calendar + Contacts + Notes + Tasks.
FileBrowser	Web-based file explorer for managing folders on your server.
Syncthing	Peer-to-peer file syncing (better than Dropbox, no cloud).
MinIO	Self-hosted Amazon S3 storage for large photo/video backups.
Duplicati	Backup your folders to cloud/NAS with encryption and scheduling.
Restic + Restic-Web	Secure backup system, works well for server or NAS backups.

Bookmarking, Notes & Knowledge Base

App	Purpose
Joplin Server	Self-hosted Evernote/Notion-style note app with encryption.
Paperless-ngx	Scans, organizes, OCR text — turns documents into a searchable archive.
Mealie	Recipe manager with meal plans & grocery lists.
LinkAce / Shaarli	Bookmark manager — organize links, tags, searchable.

Home Network, Automation & Monitoring

App	Purpose
Home Assistant	Smart home automation — control IoT devices locally.
Portainer	Web dashboard to manage Docker containers easily.
Nginx Proxy Manager	Easy HTTPS + reverse proxy for all your self-hosted apps.
Watchtower	Automatically updates Docker containers.

Uptime Kuma	Monitor if websites or servers go offline (beautiful UI).
-------------	---

Communication & Collaboration

App	Purpose
Mattermost / Rocket.Chat	Self-hosted Slack.
Matrix Synapse + Element	Decentralized messaging like Discord + WhatsApp.
Freshrss / Miniflux	Self-hosted RSS reader to follow blogs/news.
Vaultwarden	Self-hosted Bitwarden (password manager).

Top 5 You Should Start With (If You Like Immich)

Priority	App	Why You'll Love It
★ 1	Jellyfin	Organize & stream movies/music to any device.
★ 2	Nextcloud	Your own Google Drive + Notes + Calendar.
★ 3	Paperless-ngx	Scan receipts/documents — find them via search, not folders.
★ 4	Navidrome	Store and stream all your music like Spotify.
★ 5	Uptime Kuma	Know when any server or service is down — instant alerts.

Want Deployment Templates Like Immich?

I can generate:

- Full docker-compose.yml
- Directory structure
- Reverse proxy config (Nginx Proxy Manager)
- SSL + Domain

- Auto-backup setup

Just tell me which app you want to install next.

Would you like:

- 📦 A curated personal homelab setup?
- 🛡️ Secure remote access plan?
- 📁 One dashboard to manage everything?

Let me know — I'll build it for you.

Docker For Part Inventory

Yes — you can absolutely use a Docker-based app to manage parts or inventory, and there are several excellent open-source options. Here are two top-recommendations plus tips to help you choose and deploy.

Recommended Apps

1. InvenTree

- Description: InvenTree is designed for “parts management and stock control”.
- Key Features:
 - Manage “parts” as first-class objects (ideal if you have many components).
 - Stock levels by location, build/BOM support (bill of materials) so you can manage assemblies and their sub-parts.
 - REST API + plugin support for custom integrations.
-
- Deployment: There is a Docker image and a “docker stack / docker-compose” template available.
- Good for: A semi-professional or serious hobby inventory of parts/components (electronics, manufacturing, large-scale storeroom) rather than just a few items.

2. Homebox

- Description: Homebox is built for home users, inventorying “stuff” — boxes, tools, components, items in storage.
- Key Features:
 - Simple UI, custom fields, items + locations + attachments. ✓
 - QR code / label generator support mentioned for easier item tracking.
 - Lightweight: uses SQLite by default, minimal overhead.
-
- Good for: If you want to track inventory/parts in a workshop, home storage, garage, without enterprise-scale overhead.

Choosing Which One

Here are some decision factors to help:

Factor	If you need it → Choose InvenTree	If you need it → Choose Homebox
Many parts/components, need build/BOM/assemblies	✓	—
Simple home inventory, many categories, but no complex parts build	—	✓
Need barcode / label printer / enterprise features	✓	Possibly but simpler
Lightweight, minimal setup	Homebox is lighter	✓
Want large scale stock control, multiple locations, suppliers	InvenTree is better	Homebox simpler

Quick Deployment Example (for InvenTree)

Here’s a basic outline of how you’d deploy InvenTree with Docker (you’ll need to adapt for your network, domain, storage).

1. Create folder structure:

```
mkdir -p ~/docker/inventree/{data,media,static}
```

1. Create a docker-compose.yml:

```
version: '3.8'
```

```
services:
```

```
inventree:
```

```
image: inventree/inventree:latest
```

```
container_name: inventree
```

```
volumes:
```

```
- ./data:/home/inventree/data
```

```
- ./media:/home/inventree/media
```

```
- ./static:/home/inventree/static
```

```
ports:
```

```
- "8000:8000"
```

```
environment:
```

```
- INVENTREE_SITE_URL=http://your-domain.example.com
```

```
# add DB settings, etc
```

```
restart: unless-stopped
```

```
# optionally database service if not using external DB
```

```
db:
```

```
image: postgres:15
```

```
container_name: inventree-db
```

```
environment:
```

```
- POSTGRES_DB=inventree
```

```
- POSTGRES_USER=inventree
```

```
- POSTGRES_PASSWORD=YourStrongPassword
```

```
volumes:
```

```
- ./postgres_data:/var/lib/postgresql/data
```

restart: unless-stopped

1. Start it:

```
cd ~/docker/inventree
```

```
docker compose pull
```

```
docker compose up -d
```

1. Configure via web UI, set up users, add parts/locations etc.

Note: This is a starter template. For production you'll want HTTPS (via a reverse proxy like Nginx Proxy Manager or Traefik), backup strategy, maybe dedicated DB, etc.

Considerations Before You Start

- Data backup: Inventory data + media + DB should be backed up regularly.
- Storage & performance: Depending on scale (thousands of parts, many transactions) you may need decent hardware.
- Labels / barcodes: If you want scanning/barcode/label printing, check which app supports it (InvenTree has many integrations; Homebox has QR/label support).
- User access / roles: If multiple people will use the inventory, check permissions support.
- Custom fields / workflow: Do you need tracking of batch/lot numbers, expiry, supplier links? Choose accordingly.
- Integration: Do you want to integrate with other systems (ERP, automation)? InvenTree is more extensible.

If you like, I can generate a full Docker-Compose template for whichever system you pick (InvenTree or Homebox) customised for your server (Cebu City /

Philippines environment) and include backup strategy + reverse-proxy + SSL.
Would you like me to prepare that?

Docker for project management

If you're looking to host a project-management application using Docker, there are several excellent open-source solutions that work well in a self-hosted environment. Here are some top picks, along with quick overviews + deployment pointers to help you choose and get started.

Top Project-Management Apps (Docker-Friendly)

1. OpenProject

Overview:

- Full-featured project management tool that supports classic, agile or hybrid workflows (Gantt charts, Kanban/boards, time tracking, team planner).
- Open-source, self-hosted edition available.
- Has Docker installation instructions.

Good for: Teams that want robust project planning, multi-team collaboration, mixture of waterfall + agile.

Considerations: Slightly heavier to deploy; will need proper resources and backup.

2. Focalboard

Overview:

- Open-source alternative to Trello / Asana / Notion for self-hosted use.
- Supports Kanban boards, list views, customizable tasks.

Good for: Smaller teams or individuals who want a simple, lightweight PM tool for tasks/projects.

Considerations: Might lack some of the more complex planning features (Gantt, resource management) that bigger tools have.

3. Taiga

Overview:

- Agile-centric open-source project management (Scrum/Kanban/backlog) tool.
- Can be self-hosted via Docker with pre-built setups.
Good for: Software teams focusing on agile workflows, sprints, backlog management.
Considerations: If you want heavy resource-/Gantt-/waterfall planning, this may be less suited.

4. Kanboard

Overview:

- Lightweight Kanban board project management tool, open-source.
- Very simple UI; easy to self-host with Docker.
Good for: Very simple task/project tracking, small teams, minimal overhead.
Considerations: Less powerful than full PM suites (fewer built-in features for complex workflows).

5. Vikunja

Overview:

- Open-source self-hostable to-do / project management app supporting multiple views (list, Kanban, Gantt).
Good for: If you want something flexible for tasks + projects but still relatively light.
Considerations: Might require more initial configuration compared to simpler board tools.



Which One Should You Pick?

Here's a quick decision guide:

Requirement	Tool Recommendation
Full-fledged PM with Gantt, teams, hybrid workflows	OpenProject
Lightweight Kanban for small team	Kanboard or Focalboard
Agile software-development (Sprint/backlog)	Taiga

Sample Docker Deployment (Example: OpenProject)

Here's a basic outline to deploy OpenProject via Docker.

1. Create a folder (e.g., /home/dockerlab/openproject)
2. Create a docker-compose.yml like:

```
version: "3.8"
```

```
services:
```

```
  openproject:
```

```
    image: openproject/community:latest
```

```
    container_name: openproject
```

```
    ports:
```

```
      - "8080:8080"
```

```
    volumes:
```

```
      - ./pgdata:/var/openproject/pgdata
```

```
      - ./attachments:/var/openproject/attachments
```

```
    environment:
```

```
      OPENPROJECT_SECRET_KEY_BASE: "YourVerySecureSecretHere"
```

```
      OPENPROJECT_HTTPS: "false"
```

```
      OPENPROJECT_HOST: "your.domain.com"
```

```
    restart: unless-stopped
```

1. Run:

```
cd /home/dockerlab/openproject
```

```
docker compose pull
```

```
docker compose up -d
```

1. Access via `http://your-server-ip:8080`, then apply reverse proxy + HTTPS later for production.

(Note: adjust volumes, ports, environment variables for your specific needs & domain)

Tips for Self-Hosted Project-Management Tools

- Use a reverse proxy (e.g., Nginx Proxy Manager) + Let's Encrypt SSL so you can access from anywhere securely.
- Enable backup of both the database and attachments/files — your project data matters.
- Give user roles and permissions properly (team vs viewer vs admin).
- Consider integrations (e.g., Git service, chat, notifications) if required.
- Monitor container updates and apply patches for security.

If you like, I can generate a full `docker-compose.yml` template for one of these tools (you pick which) tailored to your server in Cebu City / Philippines, and include reverse proxy + HTTPS and backup strategy. Would that be helpful?

List top 10 downloaded docker applications from docker hub

Here's a list of commonly cited Docker Hub images that rank among the most-pulled/downloaded. Because Docker Hub doesn't publicly publish a full definitive real-time "top 10 by pull count" list, consider these as very popular rather than strictly precise ranking.

#	Image	Why so popular
1	nginx	A very widely-used web server and reverse proxy image.
2	alpine	Ultra-lightweight base image used as a foundation for many containers.

3	ubuntu	One of the most common OS base images used in containers.
4	redis	Popular in-memory data store / cache image.
5	mysql	Widely-used relational database image in containerized setups.
6	postgres	Popular open-source relational database image for container use.
7	memcached	Lightweight caching layer image used in many stack setups.
8	httpd (Apache HTTP Server)	Classic web server image for many web-application deployments.
9	mariadb	A popular drop-in replacement for MySQL, used in container form.
10	busybox	Extremely minimal image, often used for toolkits or small tasks.

Notes & Caveats

- These images are mostly base/utility/infra images (OS, web server, database etc.).
- The “pull count” ranking is approximate because Docker Hub doesn’t make a transparent, up-to-date ranking easily accessible. For example, one discussion states you can query:
https://hub.docker.com/v2/repositories/library/?page=1&page_size=100
- The list excludes many application-specific images (self-hosted services, etc.) which may also have very high pull counts but less consistently tracked in “official” lists.
- Popularity doesn’t imply suitability — always check for security, upstream maintenance, versioning.

If you like, I can pull the latest data (via API) and compile an updated list of top 20 Docker Hub images by pull count, including custom/self-hosted service images. Would you like that?