

AI-03092 Docker Immich Deployment

👤 Owner	© Rigel Arcayan
☰ Tags	Project
🕒 Created time	@October 19, 2025 12:57 PM

Can I use Immich to organize and serve all my photo library

Yes — Immich is designed exactly for that, and it can be a great self-hosted solution to organize, view, and serve your entire photo and video library across devices.

What Immich Can Do for Your Photo Library



1. Centralized Photo Storage & Browsing

- You can upload or import all existing photos and videos from hard drives, NAS, or cloud exports (Google Photos, iCloud, etc.).
- Immich displays your media in a timeline view, grouped by date, location, or album.



2. Smart Organization

Immich automatically organizes your media using:

- Face recognition & People tagging
- Location mapping (GPS EXIF data → shows on map)
- Album creation & custom collections
- AI-based object search (e.g., "dog", "sunset", "beach")



3. Access Anywhere






- Works on Web, iOS, and Android.
- Supports remote access if you're using NGINX Proxy Manager + HTTPS domain.
- Can be shared with family members (multi-user with private & shared libraries).



4. Performance & Self-Hosting Freedom

- Stores original files + generates optimized Web-ready copies.
- Everything runs locally on your server or NAS — no cloud subscription required.

Things to Know Before Using Immich for All Photos

Factor	Details
 Large Libraries	Supports 100k+ photos/videos if using SSD + enough RAM (8GB or more recommended).
 Keeps Folder Structure	You can import from your existing directory layout.
 Backup Responsibility	Immich is not a backup tool — you need to back up your data folder and database.
 RAW Photos	Supported, but preview might be slower.
 GPU/AI	Features like face recognition are faster with a GPU (but not required).

Recommended Setup for Full Photo Library Management

1. Organize your external drives/NAS structure first (optional but ideal).
2. Deploy Immich using Docker (as previously provided).
3. Import photos by:
 - Web upload
 - Mobile auto-upload
 - Or mount external folders to immich-data/library
- 4.
5. Enable AI features (face detection, object tags).
6. Configure backups → Use rsync, Synology Hyper Backup, or BorgBackup.

Final Answer

Yes. Immich is perfectly capable of organizing, managing, and serving your entire photo collection with smart AI search, face recognition, albums, and mobile access — as long as you take care of backups and give it enough storage and server resources.

Would you like:

- Guide to import existing Google Photos/iCloud data?
- Auto-backup plan?
- NGINX HTTPS domain to access anywhere?

I just installed Immich via docker on my ubuntu server vm in Hyper-v, I have my pictures in Windows e:\windows folder, how to setup Immich to use this file location for both upload and library.

To allow Immich (running in Docker on your Ubuntu Server VM in Hyper-V) to access your Windows `E:\windows` folder for both uploads and library indexing, you'll need to bridge the gap between your Windows host and the Ubuntu VM. Here's a step-by-step protocol tailored for your setup:

Protocol: Linking Windows Folder to Immich in Ubuntu VM

1. Share the Windows Folder

- On your Windows host:
 - Right-click `E:\windows` → Properties → Sharing tab → Share the folder.
 - Note the network path (e.g., `\\YourPCName\E_windows`).

2. Mount the Windows Share in Ubuntu VM

- Inside your UbuntuServer VM:
 1. Install CIFS utilities:

```
sudo apt update
sudo apt install cifs-utils
```

2. Create a mount point:

```
sudo mkdir -p /mnt/windows_photos
```

3. Mount the share (replace with your actual credentials and hostname):

```
sudo mount -t cifs -o username=your_windows_user,password=your_
password,uid=1000,gid=1000 //YourPCName/E_windows /mnt/window
s_photos
```

4. To make it persistent, add to `/etc/fstab` :

```
//YourPCName/E_windows /mnt/windows_photos cifs username=your_
_windows_user,password=your_password,uid=1000,gid=1000 0 0
```

3. Bind-Mount the Folder into Immich Docker Container

- Modify your `docker-compose.yml` for Immich:

```
services:
  immich-server:
    volumes:
      - /mnt/windows_photos:/usr/src/app/upload
```




- This makes `/mnt/windows_photos` accessible inside Immich as `/usr/src/app/upload`.

4. Configure Immich to Use the Mounted Folder

- In Immich's web UI or config:
 - Set the upload and library paths to `/usr/src/app/upload`.
 - You may need to restart the container:

```
docker-compose down
docker-compose up -d
```

Tips for Legacy-Grade Stability

-  Use a dedicated Windows user with read-only access for safety.
-  Avoid storing plaintext passwords in `/etc/fstab`; consider using a credentials file.
-  Document this as part of your DockerLab legacy archive with restore instructions.

Would you like a zipped README walkthrough for this setup, tailored for gifting to your son or mentees? Or a decision matrix comparing this method vs syncing via rsync or SMB containers?

Great question, Rigel — and your instincts are spot-on for resource stewardship. When you mount the Windows folder inside your UbuntuServer VM using CIFS (or SMB), **the pictures do not get copied into the VM**. They are simply **referenced** via the network share.




Think of it like this:

Your Ubuntu VM is acting like a guest peeking into the Windows host's photo album — it sees the contents, can read/write if permissions allow, but **doesn't store them locally**.

Storage Behavior Summary

Action	Result in Ubuntu VM	Impact on VM Disk Space
Mounting Windows folder	Creates a reference to remote data	Minimal (just mount point)
Uploading via Immich	If configured to use mounted path	Stored on Windows drive
Indexing/thumbnail caching	Small cache may be stored locally	Negligible (few MBs)

Legacy-Safe Notes

-  Your 74GB of photos will **stay on your Windows E: drive**.
-  Immich will treat the mounted folder as its upload and library location, so no duplication occurs.
-  Thumbnail previews and metadata may be cached inside the container, but this is minor.

Would you like to log this as a zipped vignette in your DockerLab archive — maybe titled *"Immich Mount Protocol: Legacy-Safe Photo Access"*? I can help you formalize it for gifting or mentoring.