

# Immich

## Mastering Media Management with Immich

Are you looking for an efficient and user-friendly solution to manage your personal photo and video library? Look no further than Immich. Immich is an open-source, self-hosted photo and video backup solution designed to give you full control over your media files. In this blog post, we will explore the key features of Immich, provide installation instructions using Docker Compose, and guide you through the basic setup.

### What is Immich?

Immich is a self-hosted photo and video backup solution that allows you to store, manage, and access your media files securely. It aims to provide a powerful and privacy-focused alternative to commercial cloud storage services. With Immich, you have complete control over your data, ensuring your photos and videos are stored safely and accessible only to you.

## Key Features of Immich

### 1. Self-Hosted Solution

- **Data Privacy:** Keep your media files private and secure by hosting them on your own server.
- **Full Control:** Manage your storage and access settings without relying on third-party services.

### 2. User-Friendly Interface

- **Intuitive UI:** Easy-to-navigate web interface for managing and viewing your media files.
- **Mobile Apps:** Access your media library on the go with Immich's mobile applications.

### 3. Automatic Backup

- **Scheduled Backups:** Configure automatic backups to ensure your media files are regularly saved.

- **Incremental Backups:** Only new or modified files are backed up, saving time and storage space.

## 4. Organizational Features

- **Albums and Tags:** Organize your media files into albums and add tags for easy retrieval.
- **Search Functionality:** Quickly find specific photos or videos using keywords and tags.

## 5. Sharing Options

- **Secure Sharing:** Share your photos and videos with friends and family through secure links.
- **Access Controls:** Set permissions to control who can view or download your media files.

## 6. Extensible with Plugins

- **Plugin Support:** Enhance functionality with a variety of plugins to meet your specific needs.
- **Community Contributions:** Benefit from plugins developed by the Immich community.

# Installing Immich Using Docker Compose

To install Immich using Docker Compose, follow these steps:

## Prerequisites

- Docker and Docker Compose installed on your server
- Basic understanding of Docker and Docker Compose

## Step-by-Step Installation

### 1. Create a Docker Compose File

Create a directory for Immich and navigate into it. Then, create a `docker-compose.yml` file with the following content:

```
services:
  immich-server:
    container_name: immich_server
    environment:
      - PUID=${PUID}
      - PGID=${PGID}
      - TZ=${TZ}
    image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
```

```
command: [ "start.sh", "immich" ]
volumes:
  - /path/to/photos/upload:/usr/src/app/upload
#   - ${EXTERNAL_PATH1}:/usr/src/app/external/photos
#   - ${EXTERNAL_PATH2}:/usr/src/app/external/.xxx
#   - ${EXTERNAL_PATH3}:/usr/src/app/external/Videos/m&d
env_file:
  - .env
ports:
  - ${HTTP_PORT}:3001
depends_on:
  - redis
  - database
restart: always

immich-microservices:
  container_name: immich_microservices
  environment:
    - PUID=${PUID}
    - PGID=${PGID}
    - TZ=${TZ}
  image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
  extends:
    file: hwaccel.yml
    service: hwaccel
  command: [ "start.sh", "microservices" ]
  volumes:
    - /path/to/photos/upload:/usr/src/app/upload
#   - ${EXTERNAL_PATH1}:/usr/src/app/external/photos
#   - ${EXTERNAL_PATH2}:/usr/src/app/external/.xxx
#   - ${EXTERNAL_PATH3}:/usr/src/app/external/Videos/m&d
  env_file:
    - .env
  depends_on:
    - redis
    - database
  restart: always

immich-machine-learning:
```

```
container_name: immich_machine_learning
environment:
  - PUID=${PUID}
  - PGID=${PGID}
  - TZ=${TZ}
image: ghcr.io/immich-app/immich-machine-learning:${IMMICH_VERSION:-release}
volumes:
  - ${docker}/immich/model-cache:/cache
env_file:
  - .env
restart: always
```

#### redis:

```
container_name: immich_redis
environment:
  - PUID=${PUID}
  - PGID=${PGID}
  - TZ=${TZ}
```

```
image: redis:6.2-
```

```
alpine@sha256:70a7a5b641117670beae0d80658430853896b5ef269ccf00d1827427e3263fa3
```

```
restart: always
```

#### database:

```
container_name: immich_postgres
```

```
image: registry.hub.docker.com/tensorchord/pgvecto-rs:pg14-
```

```
v0.2.0@sha256:90724186f0a3517cf6914295b5ab410db9ce23190a2d9d0b9dd6463e3fa298f0
```

```
env_file:
```

```
- .env
```

```
environment:
```

```
- PUID=${PUID}
- PGID=${PGID}
- TZ=${TZ}
- POSTGRES_PASSWORD=${DB_PASSWORD}
- POSTGRES_USER=${DB_USERNAME}
- POSTGRES_DB=${DB_DATABASE_NAME}
```

```
volumes:
```

```
- /home/thesabear/docker/immich/pgdata:/var/lib/postgresql/data
```

```
restart: always
```

## 2. Deploy the Containers

Open a terminal, navigate to the directory containing your `docker-compose.yml` file, and run:

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

This command will download the necessary images and start the Immich and PostgreSQL containers.

## 3. Access Immich

Once the containers are running, you can access Immich by navigating to

`http://your_server_ip:3000` in your web browser.

# Basic Setup Instructions

## Step 1: Initial Configuration

- Upon first accessing Immich, you will be guided through the initial setup process.
- Create an admin account by providing a username, email, and password.

## Step 2: Adding Media

- Log in with your admin account.
- Navigate to the "Upload" section.
- Click on the "Upload" button to add your photos and videos.
- Uploaded media will be automatically organized and indexed.

## Step 3: Organizing Media

- Create albums to organize your media files.
- Add tags to your photos and videos for easier retrieval.

## Step 4: Setting Up Backups

- Configure automatic backups by navigating to the "Settings" section.
- Set the schedule and parameters for your backups to ensure your media files are regularly saved.

## Step 5: Sharing Media

- Navigate to the "Share" section to create secure sharing links.
- Set permissions to control who can view or download your media files.

# Useful Links

- [Immich GitHub Repository](#) – Access the source code and contribute to the project.
- [Immich Documentation](#) – Detailed documentation for advanced configuration and usage.

# Conclusion

Immich is a powerful and flexible solution for managing your personal photo and video library. With features like automatic backups, easy organization, secure sharing, and a user-friendly interface, Immich provides an efficient and privacy-focused alternative to commercial cloud storage services. By following the installation and setup instructions provided, you can quickly deploy Immich and start managing your media files more effectively.

---

Revision #4

Created 2024-07-01 06:53:26 UTC by thesabear

Updated 2024-09-17 13:38:08 UTC by thesabear