Immich

Link

Docker Compose [Recommended]

Docker Compose is the recommended method to run Immich in production. Below are the steps to deploy Immich with Docker Compose.

Step 1 - Download the required files

Create a directory of your choice (e.g. `./immich-app`) to hold the `docker-compose.yml` and `.env` files.

Move to the directory you created

```
mkdir ./immich-app
cd ./immich-app
```

Download `docker-compose.yml` and `example.env`, either by running the following commands:

Get docker-compose.yml file

```
wget https://github.com/immich-app/immich/releases/latest/download/docker-compose.yml
```

Get .env file

```
wget -O .env https://github.com/immich-app/immich/releases/latest/download/example.env
```

(Optional) Get hwaccel.yml file

```
wget https://github.com/immich-app/immich/releases/latest/download/hwaccel.yml
```

or by downloading from your browser and moving the files to the directory that you created.

Note: If you downloaded the files from your browser, also ensure that you rename `example.env` to `.env`.

> Optionally, you can use the `hwaccel.yml` file to enable hardware acceleration for transcoding. See the Hardware Transcoding guide for info on how to set this up.

Step 2 - Populate the .env and .yml files with custom values

Example .yml content

```
version: "3.8"

services:
  immich-server:
    container_name: immich_server
    image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
    command: [ "start.sh", "immich" ]
    volumes:
      - ${UPLOAD_LOCATION}:/usr/src/app/upload
    env_file:
      - .env
    depends_on:
      - redis
      - database
      - typesense
    restart: always

  immich-microservices:
    container_name: immich_microservices
    image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
    # extends:
    #   file: hwaccel.yml
    #   service: hwaccel
    command: [ "start.sh", "microservices" ]
    volumes:
      - ${UPLOAD_LOCATION}:/usr/src/app/upload
    env_file:
      - .env
    depends_on:
      - redis
      - database
      - typesense
    restart: always

  immich-machine-learning:
    container_name: immich_machine_learning
    image: ghcr.io/immich-app/immich-machine-learning:${IMMICH_VERSION:-release}
    volumes:
      - model-cache:/cache
    env_file:
      - .env
    restart: always

  immich-web:
    container_name: immich_web
    image: ghcr.io/immich-app/immich-web:${IMMICH_VERSION:-release}
    env_file:
      - .env
    restart: always

  typesense:
    container_name: immich_typesense
    image: typesense/typesense:0.24.1@sha256:9bcff2b829f12074426ca044b56160ca9d777a0c488303469143dd9f8259d4dd
    environment:
      - TYPESENSE_API_KEY=${TYPESENSE_API_KEY}
      - TYPESENSE_DATA_DIR=/data
      # remove this to get debug messages
      - GLOG_minloglevel=1
    volumes:
      - /srv/path/Files/Immich/tsdata:/data
    restart: always
  redis:
    container_name: immich_redis
    image: redis:6.2-alpine@sha256:70a7a5b641117670beae0d80658430853896b5ef269ccf00d1827427e3263fa3
    restart: always

  database:
    container_name: immich_postgres
    image: postgres:14-alpine@sha256:28407a9961e76f2d285dc6991e8e48893503cc3836a4755bbc2d40bcc272a441
    env_file:
      - .env
    environment:
      POSTGRES_PASSWORD: ${DB_PASSWORD}
      POSTGRES_USER: ${DB_USERNAME}
      POSTGRES_DB: ${DB_DATABASE_NAME}
    volumes:
      - /srv/path/Files/Immich/pgdata:/var/lib/postgresql/data
    restart: always

  immich-proxy:
    container_name: immich_proxy
    image: ghcr.io/immich-app/immich-proxy:${IMMICH_VERSION:-release}
    environment:
      # Make sure these values get passed through from the env file
      - IMMICH_SERVER_URL
      - IMMICH_WEB_URL
    ports:
      - 2283:8080
    depends_on:
      - immich-server
      - immich-web
    restart: always

volumes:
  pgdata:
  model-cache:
  tsdata:
```

Example .env content

```
# You can find documentation for all the supported env variables at https://immich.app/docs/install/environment-variables

# The location where your uploaded files are stored
UPLOAD_LOCATION=/srv/path/Files/Immich/Upload

# The Immich version to use. You can pin this to a specific version like "v1.71.0"
IMMICH_VERSION=release

# Connection secrets for postgres and typesense. You should change these to random passwords
TYPESENSE_API_KEY=complicatedrandomkey
DB_PASSWORD=anothercomplicatedrandomkey

## Needed
IMMICH_WEB_URL=http://immich-web:3000
IMMICH_SERVER_URL=http://immich-server:3001
IMMICH_MACHINE_LEARNING_URL=http://immich-machine-learning:3003

# The values below this line do not need to be changed
###################################################################################
DB_HOSTNAME=immich_postgres
DB_USERNAME=postgres
DB_DATABASE_NAME=immich

REDIS_HOSTNAME=immich_redis
```

- Populate custom database information if necessary.
- Populate `UPLOAD_LOCATION` with your preferred location for storing backup assets.
- Consider changing `DB_PASSWORD` to something randomly generated
- Consider changing `TYPESENSE_API_KEY` to something randomly generated

Step 3 - Start the containers

From the directory you created in Step 1, (which should now contain your customized `docker-compose.yml` and `.env` files) run `docker-compose up -d`.

Start the containers using docker compose command

```
docker-compose up -d     # or `docker compose up -d` based on your docker-compose version
```

> For more information on how to use the application, please refer to the Post Installation guide.

> Note that downloading container images might require you to authenticate to the GitHub Container Registry (steps here).

Step 4 - Upgrading

If `IMMICH_VERSION` is set, it will need to be updated to the latest or desired version.

When a new version of Immich is released, the application can be upgraded with the following commands, run in the directory with the `docker-compose.yml` file:

Upgrade Immich

```
docker-compose pull && docker-compose up -d     # Or `docker compose up -d`
```

> Automatic Updates

> Immich is currently under heavy development, which means you can expect breaking changes and bugs. Therefore, we recommend reading the release notes prior to updating and to take special care when using automated tools like Watchtower.

Portainer

1. Go to "**Stacks**" in the left sidebar.
2. Click on "**Add stack**".
3. Give the stack a name (i.e. Immich), and select "**Web Editor**" as the build method.
4. Copy the content of the `docker-compose.yml` file from the GitHub repository.
5. Replace `.env` with `stack.env` for all containers that need to use environment variables in the web editor.

Dot Env Example

8. Click on "**Advanced Mode**" in the **Environment Variables** section.

Dot Env Example

9. Copy the content of the `example.env` file from the GitHub repository and paste into the editor.
10. Switch back to "**Simple Mode**".

Dot Env Example

- Populate custom database information if necessary.
- Populate `UPLOAD_LOCATION` with your preferred location for storing backup assets.

11. Click on "**Deploy the stack**".

> For more information on how to use the application, please refer to the Post Installation guide.

Revision #6
Created 27 August 2023 13:35:58 by Admin
Updated 21 November 2024 22:16:20 by Admin