Immich docker-compose.yml

Installing

For installing Immich with a docker container follow the below instructions.

Step 1 - Making directory

Go to the docker folder where you create your containers and make a folder.

```
mkdir immich-app
cd immich-app
```

Step 2 - Create the docker-compose.yml

```
nano docker-compose.yml
```

and copy paste the below yaml config, you can change it according to your needs.

```
#
# WARNING: To install Immich, follow our guide: https://immich.app/docs/install/docker-compose
#
# Make sure to use the docker-compose.yml of the current release:
#
# https://github.com/immich-app/immich/releases/latest/download/docker-compose.yml
#
# The compose file on main may not be compatible with the latest release.

name: immich

services:
  immich-server:
    container_name: immich_server
    image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
    # extends:
    #   file: hwaccel.transcoding.yml
    #   service: cpu # set to one of [nvenc, quicksync, rkmpp, vaapi, vaapi-wsl] for
accelerated transcoding
    labels:
```

```
      - "com.centurylinklabs.watchtower.enable=false"
    deploy:
      resources:
        limits:
          cpus: '1'
          memory: 4096M
        reservations:
          cpus: '1'
          memory: 4096M
    volumes:
      # Do not edit the next line. If you want to change the media storage location on your
system, edit the value of UPLOAD_LOCATION in the .env file
      - ${UPLOAD_LOCATION}:/usr/src/app/upload
      - /etc/localtime:/etc/localtime:ro
    env_file:
      - .env
    ports:
      - '2283:2283'
    depends_on:
      - redis
      - database
    restart: always
    healthcheck:
      disable: false

  immich-machine-learning:
    container_name: immich_machine_learning
    # For hardware acceleration, add one of -[armnn, cuda, openvino] to the image tag.
    # Example tag: ${IMMICH_VERSION:-release}-cuda
    image: ghcr.io/immich-app/immich-machine-learning:${IMMICH_VERSION:-release}
    # extends: # uncomment this section for hardware acceleration - see
https://immich.app/docs/features/ml-hardware-acceleration
    #   file: hwaccel.ml.yml
    #   service: cpu # set to one of [armnn, cuda, openvino, openvino-wsl] for accelerated
inference - use the `-wsl` version for WSL2 where applicable
    labels:
      - "com.centurylinklabs.watchtower.enable=false"
    deploy:
      resources:
        limits:
```

```
          cpus: '1'
          memory: 4096M
        reservations:
          cpus: '1'
          memory: 4096M
    volumes:
      - model-cache:/cache
    env_file:
      - .env
    restart: always
    healthcheck:
      disable: false

  redis:
    container_name: immich_redis
    image: docker.io/redis:6.2-
alpine@sha256:148bb5411c184abd288d9aaed139c98123eeb8824c5d3fce03cf721db58066d8
    labels:
      - "com.centurylinklabs.watchtower.enable=false"
    deploy:
      resources:
        limits:
          cpus: '1'
          memory: 4096M
        reservations:
          cpus: '1'
          memory: 4096M
    healthcheck:
      test: redis-cli ping || exit 1
    restart: always

  database:
    container_name: immich_postgres
    image: docker.io/tensorchord/pgvecto-rs:pg14-
v0.2.0@sha256:90724186f0a3517cf6914295b5ab410db9ce23190a2d9d0b9dd6463e3fa298f0
    labels:
      - "com.centurylinklabs.watchtower.enable=false"
    deploy:
      resources:
        limits:
```

```
          cpus: '1'
          memory: 4096M
        reservations:
          cpus: '1'
          memory: 4096M
    environment:
      POSTGRES_PASSWORD: ${DB_PASSWORD}
      POSTGRES_USER: ${DB_USERNAME}
      POSTGRES_DB: ${DB_DATABASE_NAME}
      POSTGRES_INITDB_ARGS: '--data-checksums'
    volumes:
      # Do not edit the next line. If you want to change the database storage location on your
system, edit the value of DB_DATA_LOCATION in the .env file
      - ${DB_DATA_LOCATION}:/var/lib/postgresql/data
    healthcheck:
      test: >-
        pg_isready --dbname="$${POSTGRES_DB}" --username="$${POSTGRES_USER}" || exit 1;
        Chksum="$$(psql --dbname="$${POSTGRES_DB}" --username="$${POSTGRES_USER}" --tuples-
only --no-align
        --command='SELECT COALESCE(SUM(checksum_failures), 0) FROM pg_stat_database')";
        echo "checksum failure count is $$Chksum";
        [ "$$Chksum" = '0' ] || exit 1
      interval: 5m
      start_interval: 30s
      start_period: 5m
    command: >-
      postgres
      -c shared_preload_libraries=vectors.so
      -c 'search_path="$$user", public, vectors'
      -c logging_collector=on
      -c max_wal_size=2GB
      -c shared_buffers=512MB
      -c wal_compression=on
    restart: always

volumes:
  model-cache:
```

To save and exit do the follow
press `Ctrl + x` and the press `y` and `enter`

Step 3 - Create and populate the .env file with custom values

```
nano .env
```

and copy paste the below yaml config, you can change it according to your needs.

```
# 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=./library
# The location where your database files are stored
DB_DATA_LOCATION=./postgres

# To set a timezone, uncomment the next line and change Etc/UTC to a TZ identifier from this
list: https://en.wikipedia.org/wiki/List_of_tz_database_time_zones#List
TZ=Europe/Athens

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

# Connection secret for postgres. You should change it to a random password
# Please use only the characters `A-Za-z0-9`, without special characters or spaces
DB_PASSWORD=yourpasswordhere

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

To save and exit do the follow
press `Ctrl + x` and the press `y` and `enter`

Step 4 - Start the containers

Start the containers

```
docker compose up -d
```

Check the logs if there are any errors.

Upgrading

Step 1 - Stop the containers

Go to the folder where is the immich-app and run the below commands.

```
docker compose down
```

Step 2 - Upgrade and restart Immich

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

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

Step 3 - Clean up unused Docker images (optional)

To clean up disk space, the old version's obsolete container images can be deleted with the following command:

```
docker image prune
```

That's it you immich-app is now upgraded.

If you find this useful please leave a comment.

Ins0mniA

Revision #3
Created 2025-03-03 17:31:32 EET by Green
Updated 2025-09-10 19:44:14 EEST by Green