

# Librephotos

## Link

```
mkdir librephotos
```

```
cd librephotos
```

```
touch .env
```

```
nano .env
```

### **.env**

```
# This file contains all the things you need to change to set up your Libre Photos.
# There are a few items that must be set for it to work such as the location of your photos.
# After the mandatory entries there are some optional ones that you may set.

# Start of mandatory changes.

# Location of your photos.
scanDirectory=./librephotos/pictures

# Internal data of LibrePhotos
data=./librephotos/data

# -----

# Wow, we are at the optional now. Pretty easy so far. You do not have to change any of the below.

# Set this value if you have a custom domain name. This allows uploads and django-admin access. If you do
not have a custom domain name, leave this blank.
csrfTrustedOrigins=

#What port should Libre Photos be accessed at (Default 3000)
httpPort=3000
```

# What branch should we install the latest weekly build or the development branch (dev)

tag=latest

# Number of workers, which take care of the request to the api. This setting can dramatically affect the ram usage.

# A positive integer generally in the 2-4 x \$(NUM\_CORES) range.

# You'll want to vary this a bit to find the best for your particular workload.

# Each worker needs 800MB of RAM. Change at your own will. Default is 2.

gunniWorkers=2

# You can set the database name. Did you know Libre Photos was forked from OwnPhotos?

dbName=librephotos

# Here you can change the user name for the database.

dbUser=docker

# The password used by the database.

dbPass=AaAa1234

# Default minimum rating to interpret as favorited. This default value is used when creating a new user.

# Users can change this in their settings (Dashboards > Library).

DEFAULT\_FAVORITE\_MIN\_RATING=4

# Database host. Only change this if you want to use your own existing Postgres server. If using your own server, you can remove the 'db' container from docker-compose.yml. If you're changing the name of the DB's container name (DB\_CONT\_NAME further down), you need to set this variable to match that name too.

dbHost=db

# Set the names of the docker containers to your own entries. Or don't, I'm not your dad.

# Changing these will require you to `make rename` to rename the services, and start the system with your chosen `docker-compose up -d` invocation again.

# Note that changing the DB\_CONT\_NAME will also need you to set the `dbHost` variable to the same value.

DB\_CONT\_NAME=db

BACKEND\_CONT\_NAME=backend

FRONTEND\_CONT\_NAME=frontend

PROXY\_CONT\_NAME=proxy

PGADMIN\_CONT\_NAME=pgadmin

# -----

```
# If you are not a developer ignore the following parameters: you will never need them.
```

```
# Where shall we store the backend and frontend code files.
```

```
codedir=./librephotos/code
```

```
# Location for pgAdmin
```

```
pgAdminLocation=./librephotos/pgadmin
```

```
touch docker-compose.yml
```

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

 and don't edit it

## docker-compose.yml

```
# DO NOT EDIT
```

```
# The .env file has everything you need to edit.
```

```
# Run options:
```

```
# 1. Use prebuilt images (preferred method):
```

```
# run cmd: docker-compose up -d
```

```
# 2. Build images on your own machine:
```

```
# build cmd: COMPOSE_DOCKER_CLI_BUILD=1 DOCKER_BUILDKIT=1 docker-compose build
```

```
# run cmd: docker-compose up -d
```

```
version: "3.8"
```

```
services:
```

```
  proxy:
```

```
    image: reallibrephotos/librephotos-proxy:${tag}
```

```
    container_name: proxy
```

```
    restart: unless-stopped
```

```
  volumes:
```

```
    - ${scanDirectory}:/data
```

```
    - ${data}/protected_media:/protected_media
```

```
  ports:
```

```
    - ${httpPort}:80
```

```
  depends_on:
```

```
    - backend
```

```
    - frontend
```

db:

image: postgres:13

container\_name: db

restart: unless-stopped

environment:

- POSTGRES\_USER=\${dbUser}
- POSTGRES\_PASSWORD=\${dbPass}
- POSTGRES\_DB=\${dbName}

volumes:

- \${data}/db:/var/lib/postgresql/data

command: postgres -c fsync=off -c synchronous\_commit=off -c full\_page\_writes=off -c

random\_page\_cost=1.0

healthcheck:

test: psql -U \${dbUser} -d \${dbName} -c "SELECT 1;"

interval: 5s

timeout: 5s

retries: 5

frontend:

image: reallibrephotos/librephotos-frontend:\${tag}

container\_name: frontend

restart: unless-stopped

backend:

image: reallibrephotos/librephotos:\${tag}

container\_name: backend

restart: unless-stopped

volumes:

- \${scanDirectory}:/data
- \${data}/protected\_media:/protected\_media
- \${data}/logs:/logs
- \${data}/cache:/root/.cache

environment:

- SECRET\_KEY=\${shhhhKey:-}
- BACKEND\_HOST=backend
- ADMIN\_EMAIL=\${adminEmail:-}
- ADMIN\_USERNAME=\${userName:-}
- ADMIN\_PASSWORD=\${userPass:-}
- DB\_BACKEND=postgresql

```
- DB_NAME=${dbName}
- DB_USER=${dbUser}
- DB_PASS=${dbPass}
- DB_HOST=${dbHost}
- DB_PORT=5432
- MAPBOX_API_KEY=${mapApiKey:-}
- WEB_CONCURRENCY=${gunniWorkers:-1}
- SKIP_PATTERNS=${skipPatterns:-}
- ALLOW_UPLOAD=${allowUpload:-false}
- CSRF_TRUSTED_ORIGINS=${csrfTrustedOrigins:-}
- DEBUG=0
- HEAVYWEIGHT_PROCESS=${HEAVYWEIGHT_PROCESS:-}
```

depends\_on:

db:

condition: service\_healthy

Then `docker-compose up -d`

Revision #1

Created 14 March 2024 15:19:11 by Admin

Updated 21 November 2024 22:16:20 by Admin