Setting up Immich

Setup

Set up the server

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` by running the following commands:

Get docker-compose.yml file

```
wget -O docker-compose.yml 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
```

You can alternatively download these two files from your browser and move them to the directory that you created, in which case ensure that you rename `example.env` to `.env`.

Step 2 - Populate the .env file with custom values

Default environmental variable content

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

# The location where your uploaded files are stored
UPLOAD_LOCATION=./library
```

```
# The location where your database files are stored. Network shares are not supported for the database
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=Etc/UTC

# The Immich version to use. You can pin this to a specific version like "v2.1.0"
IMMICH_VERSION=v2

# 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=postgres

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

- Populate `UPLOAD_LOCATION` with your preferred location for storing backup assets. It should be a new directory on the server with enough free space.
- Consider changing `DB_PASSWORD` to a custom value. Postgres is not publicly exposed, so this password is only used for local authentication. To avoid issues with Docker parsing this value, it is best to use only the characters `A-Za-z0-9`. `pwgen` is a handy utility for this.
- Set your timezone by uncommenting the `TZ=` line.
- Populate custom database information if necessary.

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 the following command to start Immich as a background service:

Start the containers

```
docker compose up -d
```

All of this documentation can also be found on https://docs.immich.app/overview/quick-start/

Docker install 1st

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker `apt` repository. Afterward, you can install and update Docker from the repository.

1. Set up Docker's `apt` repository.

   ```
   # Add Docker's official GPG key:
   sudo apt update
   sudo apt install ca-certificates curl
   sudo install -m 0755 -d /etc/apt/keyrings
   sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
   /etc/apt/keyrings/docker.asc
   sudo chmod a+r /etc/apt/keyrings/docker.asc

   # Add the repository to Apt sources:
   sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
   Types: deb
   URIs: https://download.docker.com/linux/ubuntu
   Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")
   Components: stable
   Signed-By: /etc/apt/keyrings/docker.asc
   EOF

   sudo apt update
   ```

2. Install the Docker packages.

   Latest | Specific version

   To install the latest version, run:

   ```
    sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
   compose-plugin
   ```

   > Note
   > The Docker service starts automatically after installation. To verify that Docker is running, use:
   >
   > ```
   > sudo systemctl status docker
   > ```

> Some systems may have this behavior disabled and will require a manual start:
>
> ```
> sudo systemctl start docker
> ```

3. Verify that the installation is successful by running the `hello-world` image:

```
sudo docker run hello-world
```