Instalação Nginx Proxy Manager

- Full Setup Instructions installations

Full Setup Instructions installations

Link: https://nginxproxymanager.com/setup/

Running the App

Create a `docker-compose.yml` file:

yml

```
version: '3.8'
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      # These ports are in format <host-port>:<container-port>
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      - '81:81' # Admin Web Port
      # Add any other Stream port you want to expose
      # - '21:21' # FTP

    # Uncomment the next line if you uncomment anything in the section
    # environment:
      # Uncomment this if you want to change the location of
      # the SQLite DB file within the container
      # DB_SQLITE_FILE: "/data/database.sqlite"

      # Uncomment this if IPv6 is not enabled on your host
      # DISABLE_IPV6: 'true'
```

```
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

Then:

bash

```
docker compose up -d
```

Using MySQL / MariaDB Database

If you opt for the MySQL configuration you will have to provide the database server yourself. You can also use MariaDB. Here are the minimum supported versions:

- MySQL v5.7.8+
- MariaDB v10.2.7+

It's easy to use another docker container for your database also and link it as part of the docker stack, so that's what the following examples are going to use.

Here is an example of what your `docker-compose.yml` will look like when using a MariaDB container:

yml

```
version: '3.8'
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      # These ports are in format <host-port>:<container-port>
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      - '81:81' # Admin Web Port
      # Add any other Stream port you want to expose
      # - '21:21' # FTP
    environment:
      # Mysql/Maria connection parameters:
      DB_MYSQL_HOST: "db"
      DB_MYSQL_PORT: 3306
```

```
      DB_MYSQL_USER: "npm"
      DB_MYSQL_PASSWORD: "npm"
      DB_MYSQL_NAME: "npm"
      # Uncomment this if IPv6 is not enabled on your host
      # DISABLE_IPV6: 'true'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
    depends_on:
      - db

  db:
    image: 'jc21/mariadb-aria:latest'
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: 'npm'
      MYSQL_DATABASE: 'npm'
      MYSQL_USER: 'npm'
      MYSQL_PASSWORD: 'npm'
      MARIADB_AUTO_UPGRADE: '1'
    volumes:
      - ./mysql:/var/lib/mysql
```

WARNING

Please note, that `DB_MYSQL_*` environment variables will take precedent over `DB_SQLITE_*` variables. So if you keep the MySQL variables, you will not be able to use SQLite.

Running on Raspberry PI / ARM devices

The docker images support the following architectures:

- amd64
- arm64
- armv7

The docker images are a manifest of all the architecture docker builds supported, so this means you don't have to worry about doing anything special and you can follow the common instructions above.

Check out the dockerhub tags for a list of supported architectures and if you want one that doesn't exist, create a feature request.

Also, if you don't know how to already, follow this guide to install docker and docker-compose on Raspbian.

Please note that the `jc21/mariadb-aria:latest` image might have some problems on some ARM devices, if you want a separate database container, use the `yobasystems/alpine-mariadb:latest` image.

Initial Run

After the app is running for the first time, the following will happen:

1. JWT keys will be generated and saved in the data folder
2. The database will initialize with table structures
3. A default admin user will be created

This process can take a couple of minutes depending on your machine.

Default Administrator User

```
Email:    admin@example.com
Password: changeme
```

Immediately after logging in with this default user you will be asked to modify your details and change your password.