

HumHub on Linode

```
version: '3'

networks:
  nginx_proxy_manager:

services:
  nginxproxymanager:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    container_name: nginx-proxy-manager
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - /home/docker/nginxproxymanager/data: /data
      - /home/docker/nginxproxymanager/letsencrypt: /etc/letsencrypt
    networks:
      nginx_proxy_manager:

  portainer-ce:
    ports:
      - '9000:9000'
      - '8000:8000'
    container_name: portainer
    restart: always
    volumes:
      - '/var/run/docker.sock:/var/run/docker.sock'
      - 'portainer_data:/data'
    image: 'portainer/portainer-ce:latest'
    networks:
      nginx_proxy_manager:

  db:
    image: mariadb:10.2
```

```
container_name: humhubdb
environment:
  MYSQL_ROOT_PASSWORD: root
  MYSQL_DATABASE: humhub
  MYSQL_USER: humhub
  MYSQL_PASSWORD: humhub
volumes:
  - humhubdb3: /var/lib/mysql
networks:
  nginx_proxy_manager:
```

humhub:

```
image: mriedmann/humhub:latest
container_name: humhub
links:
  - db: db
ports:
  - 8080: 80
volumes:
  - /home/docker/humhub/config: /var/www/localhost/htdocs/protected/config
  - /home/docker/humhub/uploads: /var/www/localhost/htdocs/uploads
  - /home/docker/humhub/modules: /var/www/localhost/htdocs/protected/modules
environment:
  HUMHUB_DB_USER: humhub
  HUMHUB_DB_PASSWORD: humhub
networks:
  nginx_proxy_manager:
```

volumes:

```
humhubdb3:
portainer_data:
```

To avoid issues during the setup process, you'll want to create the following file structure:

- /home/docker/humhub
- /home/docker/humhub/config
- /home/docker/humhub/modules
- /home/docker/humhub/uploads
- /home/docker/humhub/uploads/profile_image

I placed my docker-compose.yml file in /home/docker and then ran `docker-compose up -d` to spin up the containers.

Revision #4

Created 27 April 2022 15:29:05 by DB T3CH

Updated 27 April 2022 19:29:18 by DB T3CH