

HumHub

With the current state of social media, I have, more than once, thought to myself, "I should just host my own social media platform for me and my friends to use." That's what we're going to take a look at in this post!

I set up a Docker Linode with a Shared CPU/2GB RAM configuration.

While the Linode was deploying I made sure that I had a domain available on Linode for this project.

After the Linode deployed, I pointed the domain to the Linode.

Once that was done, I SSHed into my Linode and created some folders and then a docker-compose.yml file.

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

```
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:

- humhubdb: /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:
  humhubdb:
  portainer_data:
```

After everything has deployed, I went to <http://my.linode.ip.address:81> and logged into Nginx Proxy Manager (admin@example.com/changeme) and created an SSL for the domain name.

Next, I headed over to <http://my.linode.ip.address:9000>, created a user, and logged into Portainer.

Then I found the container IP address for the humhub container and headed back over to Nginx Proxy Manager and created a Proxy Host with the the domain I setup in Linode and the IP address I found in Portainer.

I selected the SSL that we created earlier and deployed the Proxy Host.

I was then able to go to my domain name and go through the set up process of HumHub to configure the database and other settings/features of the install.

That's really just about all there was to it!

Revision #3

Created 27 April 2022 20:17:46 by DB T3CH

Updated 27 April 2022 20:31:21 by DB T3CH