Flarum Forum

- Install Flarum Forum

Install Flarum Forum

https://www.youtube.com/embed/M9oP4ztUpQo

Flarum is a brilliantly built, light-weight, powerful forum software that may just be perfect for your community. I've used Discourse for years for my Linux Users Group, but have had it crash on me lately several times. I've had to bring it back from an older backup, and while it's fairly low usage, we did lose a few posts here and there. I started looking for an alternative, and Flarum seems to be it. It's similar in look and modern feel, but also very light weight on resources.

Let's walk through getting Flarum up and running.

What You'll Need

- A server or virtual machine to run the software on (preferably Linux, but you can run Docker on Windows too).
- Docker and Docker compose installed on the machine (we'll go through it).
- A Fully Qualified Domain Name (FQDN) (e.g. community.mygreatdomain.com)
- (optional) a Reverse Proxy (I use NGinX Proxy Manager)

Installing Docker and Docker Compose (on Linux)

If you are on a Linux server, then installing Docker and Docker Compose can generally be accomplished with a simple one line command: Just open a terminal window, and paste the following in it:

```
curl https://get.docker.com | sh
```

You'll be prompted for your super user password, then the install shuld commence. When it's done, we'll want to make ourselves part of the 'docker' group. We can do this with the command:

```
sudo usermod -aG docker <your-username>
```

Replace <your-username> with your actual username in the command above.

Now you can either log out and right back in so your docker group is activated, or you can use these two commands:

`newgrp`

`newgrp docker`

Install Flarum

Next, let's make a folder structure for our docker applications to live in.

`mkdir -p docker/flarum`

Next, we'll move into this folder and create a new file:

`cd docker/flarum`

`nano compose.yaml`

Inside this new file we want to paste the following:

```
---
services:
  flarum-app:
    image: tiredofit/flarum
    container_name: flarum-app
    links:
      - flarum-db
    volumes:
      - ./data:/data
      - ./logs:/www/logs
    environment:
      - TIMEZONE=America/Chicago
      - CONTAINER_NAME=flarum-app
      - ADMIN_USER=<your-username>
      - ADMIN_PASS=<a-long-strong-password-for-your-web-ui-admin>
      - ADMIN_EMAIL=<you@yourdomain.com>
      - SITE_TITLE=<YourSite>
      - SITE_URL=<https://your-site-name.yourdomain.com>
      - DB_HOST=flarum-db
      - DB_NAME=flarum
      - DB_USER=flarum
      - DB_PASS=<a-nice-strong-password-for-the-database> # <-- must match the DB_PASS in the
two sections below
    restart: unless-stopped
```

```
  flarum-db:
    image: tiredofit/mariadb:10.8-latest
    container_name: flarum-db
    volumes:
      - ./db:/var/lib/mysql
    environment:
      - TIMEZONE=America/Chicago
      - CONTAINER_NAME=flarum-db
      - ROOT_PASS=<a-different-long-strong-password-for-root>
      - DB_NAME=flarum
      - DB_USER=flarum
      - DB_PASS=<a-nice-strong-password-for-the-database>
    restart: unless-stopped

  flarum-db-backup:
    container_name: flarum-db-backup
    image: tiredofit/db-backup
    links:
     - flarum-db
    volumes:
      - ./dbbackup:/backup
    environment:
      - TIMEZONE=America/Chicago
      - CONTAINER_NAME=flarum-db-backup
      - DB_HOST=flarum-db
      - DB_TYPE=mariadb
      - DB_NAME=flarum
      - DB_USER=flarum
      - DB_PASS=<a-nice-strong-password-for-the-database>
      - DB01_BACKUP_INTERVAL=1440
      - DB01_BACKUP_BEGIN=0000
      - DB_CLEANUP_TIME=8640
    restart: unless-stopped
```

In the above file, you should replace any value surrounded by "<" and ">" with your own proper values. Usernames, passwords, and URLs should all be setup correctly for your install.

Once all edits have been made, you can save the file with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Now we'll pull down the images we need with

```
docker compose pull
```

A-Record in DNS

Next, we need to add an A-record for our site. Let's presume we'll be calling it "community.awesomeness.com". We need to go to our domain registrar (GoDaddy, Hover, Ghandi.net, etc) and in their DNS settings for the "awesomeness.com" domain we will need to add an A-Record.

In the A-Record we'll put our subdomain, "community", then in the IP address field, we need to put the Public IP address of our server. If possible, have this change take effect in 600 Seconds (10 minutes0. Save.

Reverse Proxy Entry

If you are running this service inside of a Private Network, or on a server where you run multiple services, a reverse proxy is often a really great way to make these services more secure. You can also use a reverse proxy to help get, and maintain LetsEncrypt certificates for SSL.

I use NGinX Proxy Manager, but feel free to use any reverse proxy you like.

I show on the video how to setup the reverse proxy entry, if needed, so make sure to check that out.

Bring Up our Flarum Application

Let's bring up our application with the commands:

```
docker compose up -d && docker compose logs -f
```

These two command (concatenated by the && symbols, tell docker compose to bring up the containers from the images we pulled down earlier, and to set the settings we specified in the docker compose file. Next, it will show us the live logs (-f follow the logs) as the containers are started up. We are just looking for any big error messages, or issues. Hopefully all goes well.

Flarum does take a few minutes to get started, so be patient while the servers all start up.

Once, you've given things a few minutes to start, you can quit out of the live logs with CTRL + C.

Now, open your favorite modern web browser, and go to your domain. You should be greeted by the Flarum start page. You can click Login, and use the admin username and password you setup in the compose file.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Buy me a Beer / Coffee:

https://paypal.me/BrianMcGonagill