Install with Docker and Docker Compose

First, you'll need docker / docker compose installed on your host system. Luckily, you can get docker installed on Windows, MacOS, and Linux.

I'm a Linux user, so that's what these directions will focus on, but if you are familiar with docker on Windows or Mac you should be able to adapt them fairly easily.

If you don't have docker and docker compose installed yet, good news is that on Linux it just takes one command:

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

You'll need curl installed in order to use the above command. I'll leave that up to you.

Once you've installed docker (it now comes with docker compose btw), you'll want to create a folder to keep your Get My docker compose and data folder in.

```
mkdir -p docker/get_my
```

Now move into that folder with

```
cd docker/get_my
```

Next, let's create our compose.yaml file, which is what docker compose uses to bring up our Get My application.

```
nano compose.yaml
```

In this new file, you'll want to copy the text block below, then make a few changes for your own needs:

> NOTE: if you have a CPU without the AVX instruction set, you can change the line 'image: mongo' to be 'image: mongo:4.4'in the compose.yaml file below.

```
---
services:
  get_my:
    container_name: get_my
```

```
    image: bmcgonag/get_my:latest
    ports:
      - "80:3000"     # you can change the left side to a less common port number if needed
    healthcheck:
      test: curl --fail -s http://localhost:3000/ || exit 1
      interval: 30s
      timeout: 10s
      retries: 3
    links:
      - mongo
    depends_on:
      - mongo
    restart: unless-stopped
  mongo:
    container_name: get_my-mongo
    image: mongo
    volumes:
      - ./data:/data/db     # this will store the application data in the same directory where
 you have the docker-compose.yml file. the directory on your host will be called 'data'.
    restart: unless-stopped
```

If you already have the host port 80 in use on your host server, you can change 80 in the port mapping `80:3000` to be some other port number that is not in use. I suggest using a port above 8200 in order to not conflict with ports that the system may use by default.

Once you've made any needed changes, save the file with CTRL + O, then press Enter to confirm, and exit the nano text editor with CTRL + X.

We are ready to startup the application:

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

The one-liner above, is really two commands concatenated with the &&. The first says to pull down the images, and start the needed containers with the settings we provided in the compose.yaml file. The second part says we want to watch the logs as things are starting up.

All we are really doing is checking to see if there are any errors in the log.

Assuming you've seen no errors, you can now browse to the IP address of your host machine, on the port you specified on the left side of the port mapping in the compose.yaml file.

If the page loads make sure to check out the Registering Users information in this documentation.

Revision #1
Created 15 August 2024 16:14:09 by Brian McGonagill