**Docker Apache and Mysql**

**Description**

You may accomplish the following however you see fit, but using `docker-compose` will make things vastly easier.

**Build images**

For this project, you should build 2 docker images as follows:

- **Image1**
  - Should be based on `ubuntu`
  - Should install `apache2`, `php7.0`, `php-mysql`, `libapache2-mod-php`
  - copy starter files from [here](#). Note that this is expecting you to set the following variables:
    - `DB_HOST`
    - `DB_USER`
    - `DB_PASS`
    - `DB_NAME`
  - Runs `apache2ctl` in the foreground
  - Exposes port 80

- **Image 2**
  - Should be based on `mysql`
  - Should set the environment variables `MYSQL_DATABASE`, `MYSQL_USER`, `MYSQL_PASSWORD` and `MYSQL_RANDOM_ROOT_PASSWORD`. See the documentation [here](#) for examples.

**Create a user-defined network**

Create a user-defined network and have an instance of both images that can connect to it. (Docker compose will do this automatically)

Make sure that you can access port 80 of the apache image and visit `/water/install.php` of the downloaded pokemon files to install the tables and see if the app works. After it works, you can just visit the base url of your app.

**HINTS**

- You set an environment variable like `DB_HOST=joesmysqlserver`. (You cannot have any spaces around the equal sign.)
- One idea would be to use the COPY command to copy the files to the correct location on your server (after you have downloaded and extracted in a location next to your Dockerfile). Another idea would be to do something like: `RUN wget http://whateverfile.tgz && tar -xvf whateverfile.tgz -C /var/www/html && rm *.tgz` (I am not sure if it is the `-C` option for tar that puts it in a location or not)
- You can name your instance by doing something like `docker run -d --name=joesmysqlserver mymysqlbuiltimage`.

**Upload the images to DockerHub**

Upload both new images to dockerhub.

**To Pass off**

Send me the name of your dockerhub images.