LAB: Container-Orchestration using Docker-Compose

You need:

- An AWS Account

Duration of the Lab: 30 Minutes.

Difficulty: medium

Update the PHP Script to connect to a Database

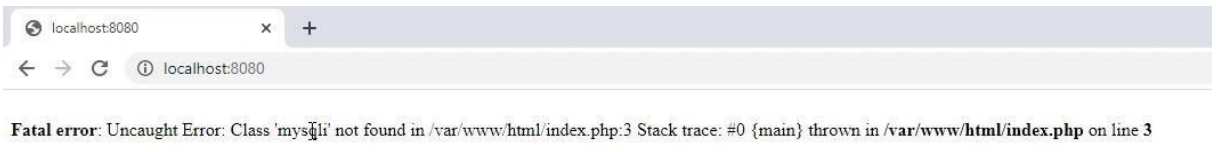
Use the following script to try and connect to a database. If the connection fails it will show you an error, if it's successful it will show "Connection Successful":

```php
<?php

// Create connection
$connection = new mysqli($servername, $username, $password);
// Check connection
if ($connection->connect_error) {
    die("Connection failed: " . $connection->connect_error);
} else {
    echo "connection successful";
    $connection->close();
}
```

And try to run this script inside a php:7.3-apache container without your custom Image:

```
docker run --rm -v ${PWD}:/var/www/html -p 8080:80 php:7.3-apache
```

And open your browser on localhost:8080. It will show you an error that mysqli is not installed:



```
Fatal error: Uncaught Error: Class 'mysqli' not found in /var/www/html/index.php:3 Stack trace: #0 {main} thrown in /var/www/html/index.php on line 3
```

Stop your running container (ctrl+c, docker ps, docker stop [containerID])

Install the mysqli extension

For the php script to work, we have to install a mysqli extension. We can only do this with a custom Dockerfile.

Update your Dockerfile

Add the following command to install mysqli inside a Dockerfile:

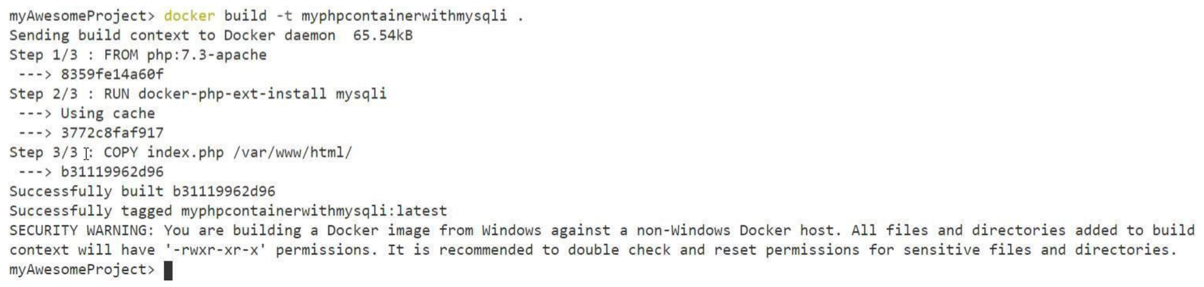
```
FROM php:7.3-apache
RUN docker-php-ext-install mysqli
COPY index.php /var/www/html/
EXPOSE 80
```

Build the Image

Now we have build the image, for example with this command:

```
docker build -t myphpcontainerwithmysqli .
```

This will build a new image called myphpcontainerwithmysqli image.



```
myAwesomeProject> docker build -t myphpcontainerwithmysqli .
Sending build context to Docker daemon  65.54kB
Step 1/3 : FROM php:7.3-apache
 ---> 8359fe14a60f
Step 2/3 : RUN docker-php-ext-install mysqli
 ---> Using cache
 ---> 3772c8faf917
Step 3/3 : COPY index.php /var/www/html/
 ---> b31119962d96
Successfully built b31119962d96
Successfully tagged myphpcontainerwithmysqli:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build
context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
myAwesomeProject>
```

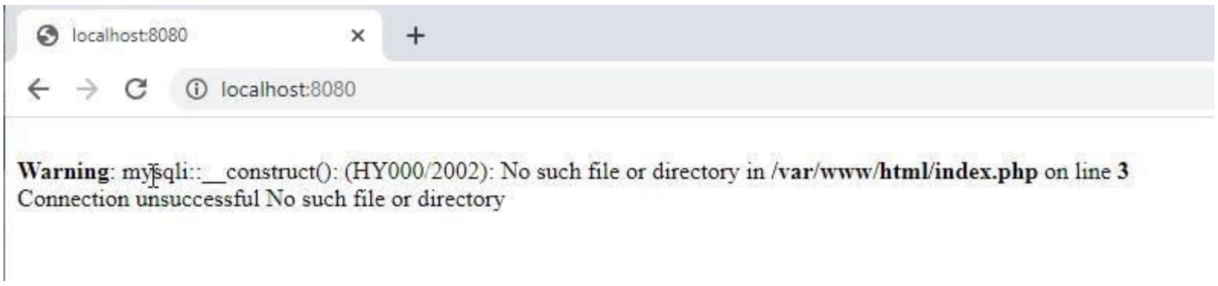
Run the Container

Now it's time to run the container:

```
docker run --rm -v ${PWD}:/var/www/html -p 8080:80 myphpcontainerwithmysqli
```

Then open the browser on localhost:8080

The error message is different now, because we don't have any mysql database running:



Warning: mysqli::__construct(): (HY000/2002): No such file or directory in **/var/www/html/index.php** on line **3**
Connection unsuccessful No such file or directory

Stop your Container again (ctrl+c, docker ps, docker stop [ID])

Introduce Environment Variables in the PHP file

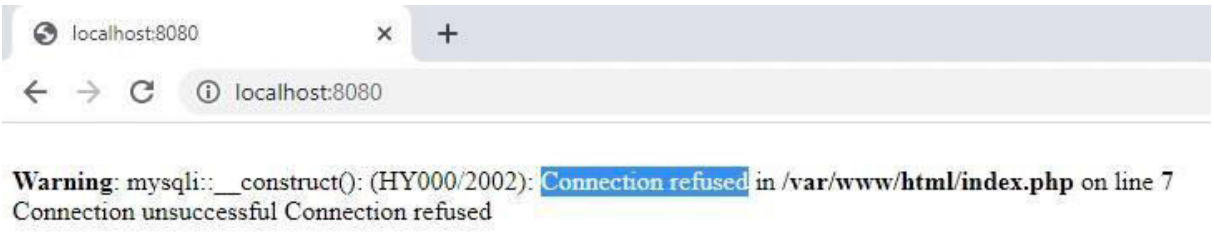
Update the PHP file with the following content:

```php
<?php

$servername = getenv("DB_HOST");
$username = getenv("DB_USERNAME");
$password = getenv("DB_PASSWORD");

// Create connection
$connection = new mysqli($servername, $username, $password);
// Check connection
if ($connection->connect_error) {
    die("Connection failed: " . $connection->connect_error);
} else {
    echo "connection successful";
    $connection->close();
}
```

Run the container with environment variables:

docker run --rm -v ${PWD}:/var/www/html -p 8080:80 -e DB_HOST='127.0.0.1' -e DB_USER='myuser' -e DB_PASSWORD='mypass' myphpcontainerwithmysqli

But still, there is no connection possible because there is simply no database running.



```
Warning: mysqli::__construct(): (HY000/2002): Connection refused in /var/www/html/index.php on line 7
Connection unsuccessful Connection refused
```

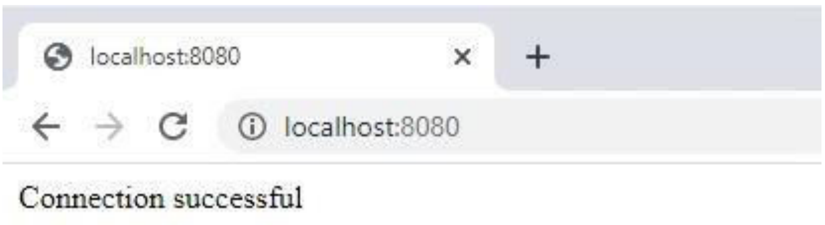
Stop the container again (ctrl+c,…)

We need a MySQL Database. But instead of running it in just another container, let’s directly orchestrate this via docker-compose.

Create a Docker-Compose.yaml File

```
version: '3'
services:

  #PHP Service
  app:
    build:
      context: .
    container_name: app
    environment:
      DB_HOST: db
      DB_PASSWORD: your_mysql_root_password
      DB_USERNAME: root
    depends_on:
      - db
    ports:
      - "8080:80"
    volumes:
       - ./:/var/www/html


  #MySQL Service
  db:
    image: mysql:5.7.22
    container_name: db
    restart: unless-stopped
    tty: true
    ports:
      - "3306:3306"
    environment:
      MYSQL_DATABASE: phpapp
      MYSQL_ROOT_PASSWORD: your_mysql_root_password
    volumes:
      - dbdata:/var/lib/mysql

#Volumes
volumes:
  dbdata:
    driver: local
```

Then run the environment with “docker-compose up” and it should come up.



Connection successful

Open your browser on localhost:8080 again.

Stop docker-compose with ctrl+c.

Lab End
