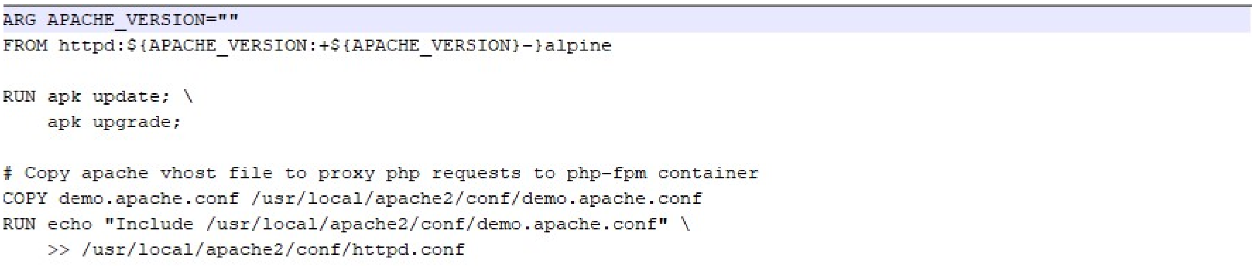
Technical Report
4/21/2019
Installation of LAMP Stack Server utilizing Docker Toolbox
Group 4 – Jeffrey Jones, Andre Ibarrondo, Jacob Hansen

Development Progress:

- Successfully installed and connected the following services:
    1.) PHP: v5.6
    2.) Apache: v.2.4.32
    3.) MySQL: v5.7
- Successfully utilized decoupling by utilizing containers with PID 1, allowing us to run our services without the interference of other unspecified services/processes.
    - Apache and PHP are successfully decouple, only able to interact with each other when permitted.
    - Apache Dockerfile

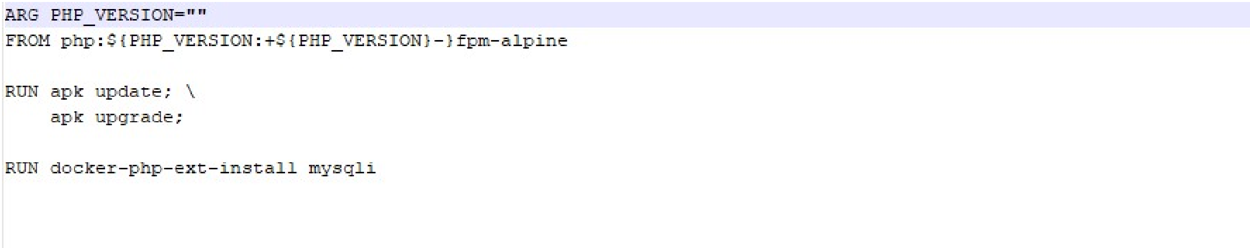


```
ARG APACHE_VERSION=""
FROM httpd:${APACHE_VERSION:+${APACHE_VERSION}-}alpine

RUN apk update; \
    apk upgrade;

# Copy apache vhost file to proxy php requests to php-fpm container
COPY demo.apache.conf /usr/local/apache2/conf/demo.apache.conf
RUN echo "Include /usr/local/apache2/conf/demo.apache.conf" \
    >> /usr/local/apache2/conf/httpd.conf
```

    - PHP Dockerfile



```
ARG PHP_VERSION=""
FROM php:${PHP_VERSION:+${PHP_VERSION}-}fpm-alpine

RUN apk update; \
    apk upgrade;

RUN docker-php-ext-install mysqli
```

    - Used PHP FPM to connect both containers
        - Apache uses port 80
        - PHP container utilizes port 9000
        - Utilized existing apache.conf outline that takes requests through a proxy for our PHP files to our PHP container.
        - Utilizing front end and back end in docker-compose.conf so only the apache container is exposed on selected port.
        - Demo for website uses port 8080

- Utilized Volumes for code distribution
    - Apache and PHP both have access the same specified volume
    - Mapping from local system to containers as to make changes to repository file more easily.

Development so far:

- With our services set up, and running independently from each other in separate containers, we have our original goal of a working LAMP Server that allows us to continue with our extended goal of the development of a web service that displays images selected from the MySQL Database.
- Have test website working on local machines
    - Index.php file displays:
        - Welcome Message.
        - Showing if MySQL was successfully connected to.
        - Demo of image displaying capability.
- Our files are uploaded onto Docker Hub and currently uploaded as a repository on Git Hub.
    - Working on moving images onto Git Hub for integration onto Cloud Lab

Future Development Tasks

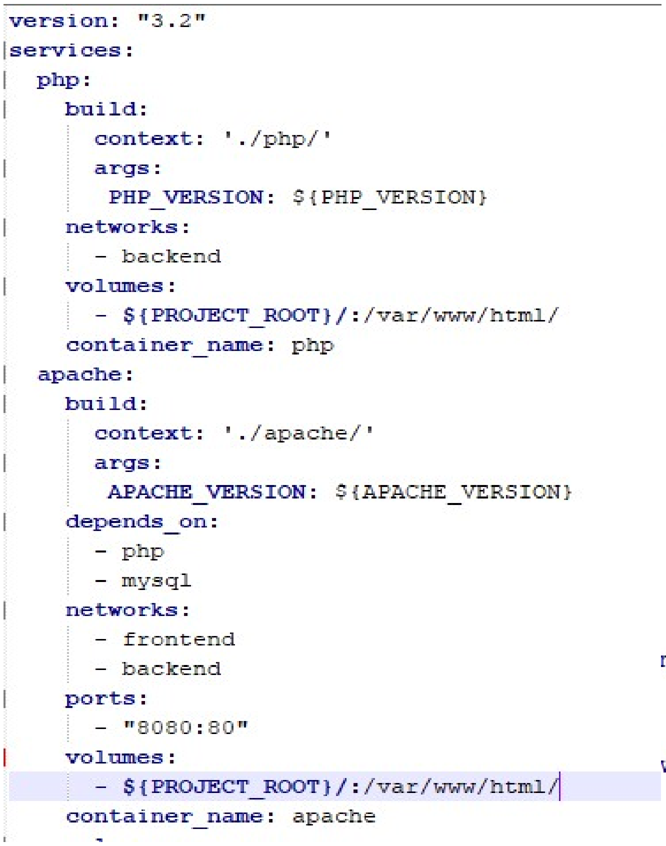
- Development of selecting images to be displayed onto our web page using PHP and MySQL functionality is now underway.
    - ETA on delivery is unknown, secondary goal after having a fully secure LAMP Stack
- Establishing a fully automated profile on Cloud Lab/Git Hub

Conclusion:

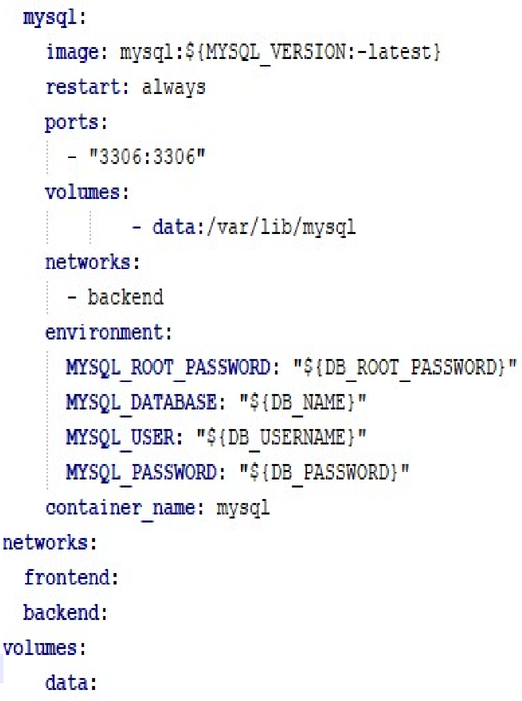
- **Deliverable 2**
    - As a team, we believe we have met with the requirements of the deliverable 2, as well as succeeded in our goal of establishing a fully functional LAMP Stack Service. We have utilized information and instructions from online resources in an efficient manner, allowing us to produce a demo of our desired extended goal of developing a Web page showing pictures selected from database container on docker.
- **Deliverable 3**

  - We believe that we are on track to complete the requirements for deliverable 3 which is to have a fully automated profile on Cloud Lab/Git Hub.

docker-compose.yml

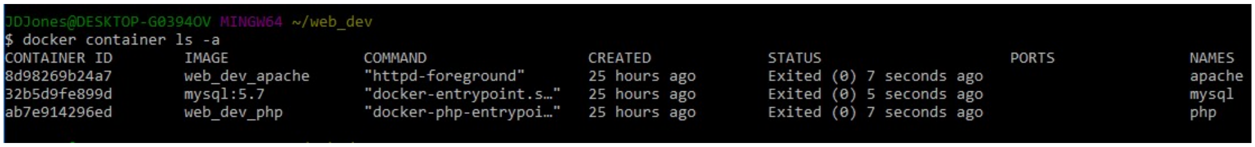


```
version: "3.2"
services:
  php:
    build:
      context: './php/'
      args:
       PHP_VERSION: ${PHP_VERSION}
    networks:
      - backend
    volumes:
      - ${PROJECT_ROOT}/:/var/www/html/
    container_name: php
  apache:
    build:
      context: './apache/'
      args:
       APACHE_VERSION: ${APACHE_VERSION}
    depends_on:
      - php
      - mysql
    networks:
      - frontend
      - backend
    ports:
      - "8080:80"
    volumes:
      - ${PROJECT_ROOT}/:/var/www/html/
    container_name: apache
```



```
  mysql:
    image: mysql:${MYSQL_VERSION:-latest}
    restart: always
    ports:
      - "3306:3306"
    volumes:
          - data:/var/lib/mysql
    networks:
      - backend
    environment:
      MYSQL_ROOT_PASSWORD: "${DB_ROOT_PASSWORD}"
      MYSQL_DATABASE: "${DB_NAME}"
      MYSQL_USER: "${DB_USERNAME}"
      MYSQL_PASSWORD: "${DB_PASSWORD}"
    container_name: mysql
networks:
  frontend:
  backend:
volumes:
    data:
```

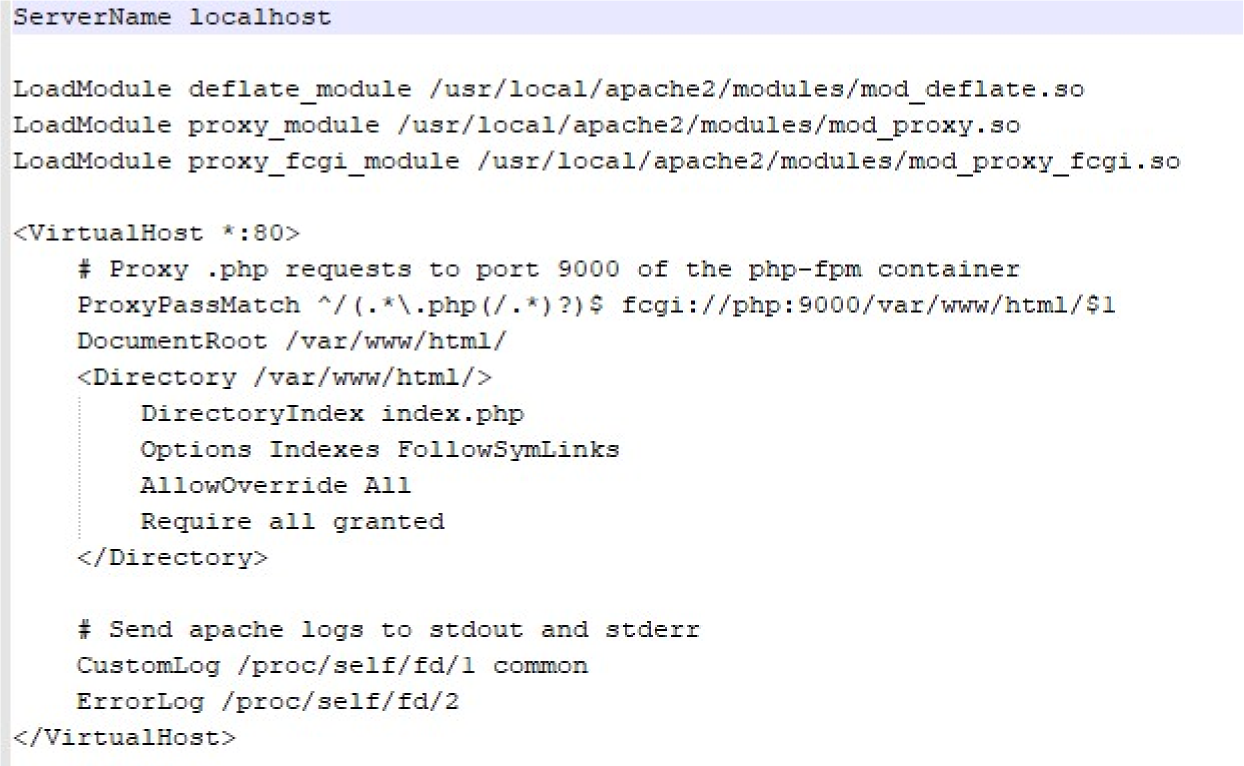
Docker Containers: Apache, PHP, MySQL



```
JDJones@DESKTOP-G0394OV MINGW64 ~/web_dev
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                     PORTS               NAMES
8d98269b24a7        web_dev_apache      "httpd-foreground"       25 hours ago        Exited (0) 7 seconds ago                       apache
32b5d9fe899d        mysql:5.7           "docker-entrypoint.s…"   25 hours ago        Exited (0) 5 seconds ago                       mysql
ab7e914296ed        web_dev_php         "docker-php-entrypoi…"   25 hours ago        Exited (0) 7 seconds ago                       php
```

Technical Report
4/21/2019
Installation of LAMP Stack Server utilizing Docker Toolbox
Group 4 – Jeffrey Jones, Andre Ibarrondo, Jacob Hansen

apache.conf



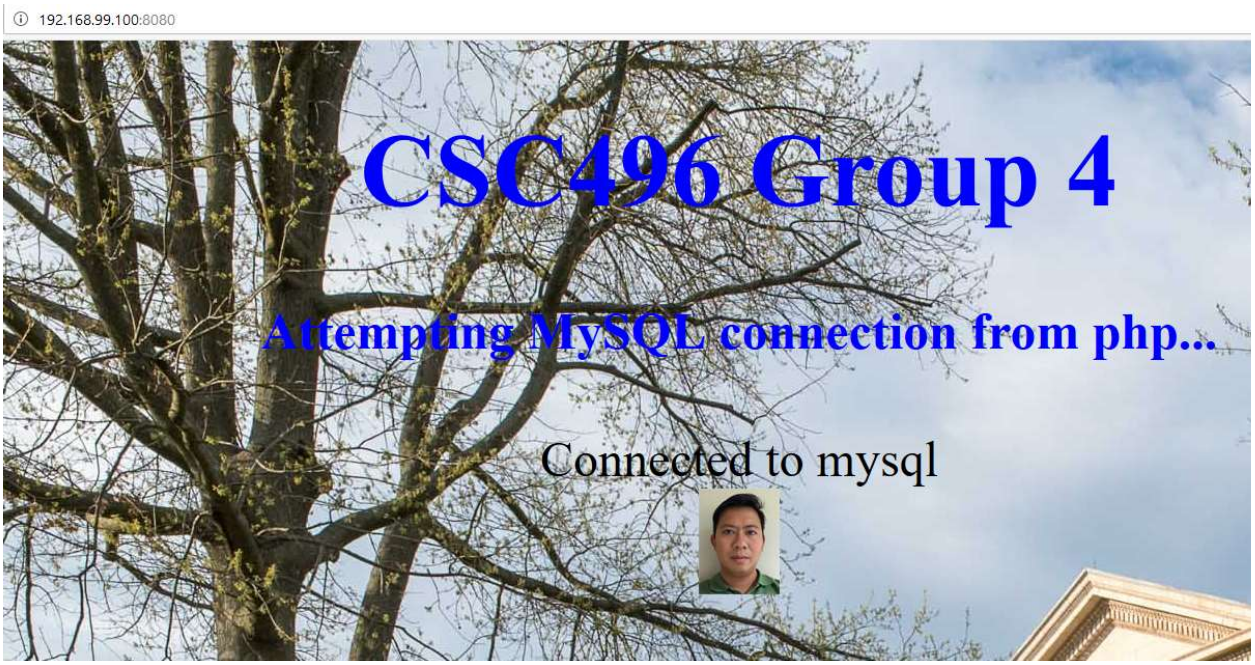
```
ServerName localhost

LoadModule deflate_module /usr/local/apache2/modules/mod_deflate.so
LoadModule proxy_module /usr/local/apache2/modules/mod_proxy.so
LoadModule proxy_fcgi_module /usr/local/apache2/modules/mod_proxy_fcgi.so

<VirtualHost *:80>
    # Proxy .php requests to port 9000 of the php-fpm container
    ProxyPassMatch ^/(.*\.php(/.*)?)$ fcgi://php:9000/var/www/html/$1
    DocumentRoot /var/www/html/
    <Directory /var/www/html/>
        DirectoryIndex index.php
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # Send apache logs to stdout and stderr
    CustomLog /proc/self/fd/1 common
    ErrorLog /proc/self/fd/2
</VirtualHost>
```

Demo



192.168.99.100:8080

CSC496 Group 4

Attempting MySQL connection from php...

Connected to mysql