

Experiment No: 06

Title: Running php application and MySQL server over the docker.

Aim: To run the php application and MySQL server over the docker.

Theory:

Container technology is growing every day. It's a technology that makes application development much easier and faster. It has a clean architecture that ensures application services utilize resources sustainably by dividing an application into smaller services called images. This allows you to set up each service independently without affecting how the other services run.

In this case, Docker provides a docker-compose file that allows you to set all your application environments and run a few commands to fully set up your application in a more elegant and faster approach.

Let's take the case of running a PHP application. You would have to install all environments that you need to run PHP scripts. You need an apache server installed in your server/system and probably a MySQL database. Then set up each environment in a way that will allow you to run your PHP-driven website.

With Docker, things are much more manageable. Docker allows you to set your application with each service running as a microservice. This way, you set a single YML file that will isolate all the services that your application needs to run. The file sets up the PHP Apache server and MySQL database for you. All you need is to specify the parameters that you need your application to run on.

The main advantage that Containers provides, is a scalable environment to run your application services. It ensures that the practices of continuous integration and continuous delivery (CI/CD) pipelines are enhanced across the team. So you only need to share this YML file with every team member. This will set all the necessary environments across the team regardless of the operating system they are running on. Thus team members can synchronize their work without breaking the code.

This guide will show you how we can use Docker development environment to:

Setup and run a local PHP Apache server instance.

Serve a dynamic PHP-driven website.

Setup a MySQL database to run SQL scripts, fetch records, and print them in a PHP-driven website.

We will use the Docker hub images to set up a containerized PHP development environment.

Prerequisites

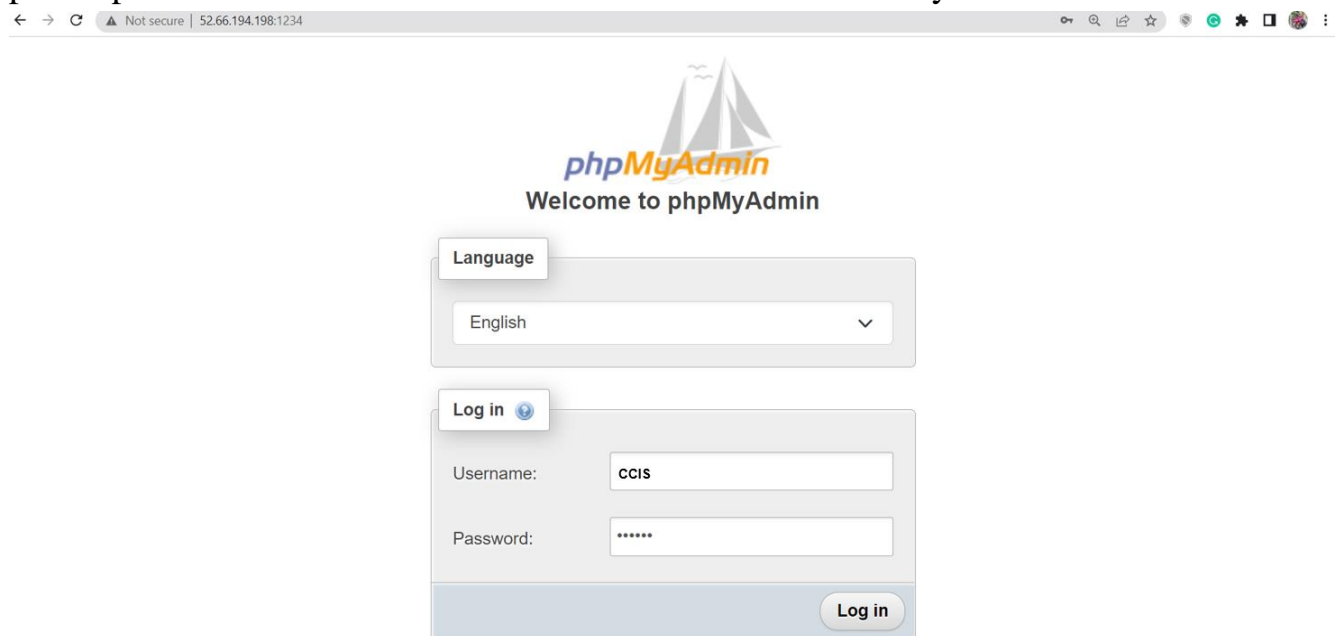
- Ensure that Docker demons are installed on your computer.
- Basic knowledge of PHP and SQL queries.
- Fundamental understating of how to build and run Docker hub images from a Docker file.
- Understand how containers work.
- Basic knowledge on how to execute Docker and docker-compose commands.

Practical:

To set a docker-compose, you need first to select the Docker version you want to use, the services you want to provide, and the containers you want to run.

```
root@ip-172-31-44-248:~# ls
docker-compose.yml  exp5
root@ip-172-31-44-248:~# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS                               NAMES
0e03f4bfe182   phpmyadmin/phpmyadmin "/docker-entrypoint..." 12 minutes ago Up 12 minutes 0.0.0.0:1234->80/tcp, :::1234->80/tcp root_phpos_1
alaf93e1f4b3   drupal:latest         "docker-php-entrypoi..." 12 minutes ago Up 12 minutes 0.0.0.0:8080->80/tcp, :::8080->80/tcp root_drupalos_1
4b83af9ae80a   mysql:5.7             "docker-entrypoint.s..." 12 minutes ago Up 12 minutes 3306/tcp, 33060/tcp              root_sqlos_1
root@ip-172-31-44-248:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>             8e523d6fc8f9       40 minutes ago     991MB
drupal              latest             5e91efff5965       2 days ago         555MB
mysql               5.7                eb175b0743cc       2 days ago         433MB
node                latest             35ff1df466e8       5 days ago         991MB
phpmyadmin/phpmyadmin latest             4a4023c7e22a       5 months ago       510MB
root@ip-172-31-44-248:~#
```

pen <http://localhost:8080/> on the browser to access the PHPMYAdmin.



← → ↻ Not secure | 52.66.194.198:1234

phpMyAdmin
Welcome to phpMyAdmin

Language

English

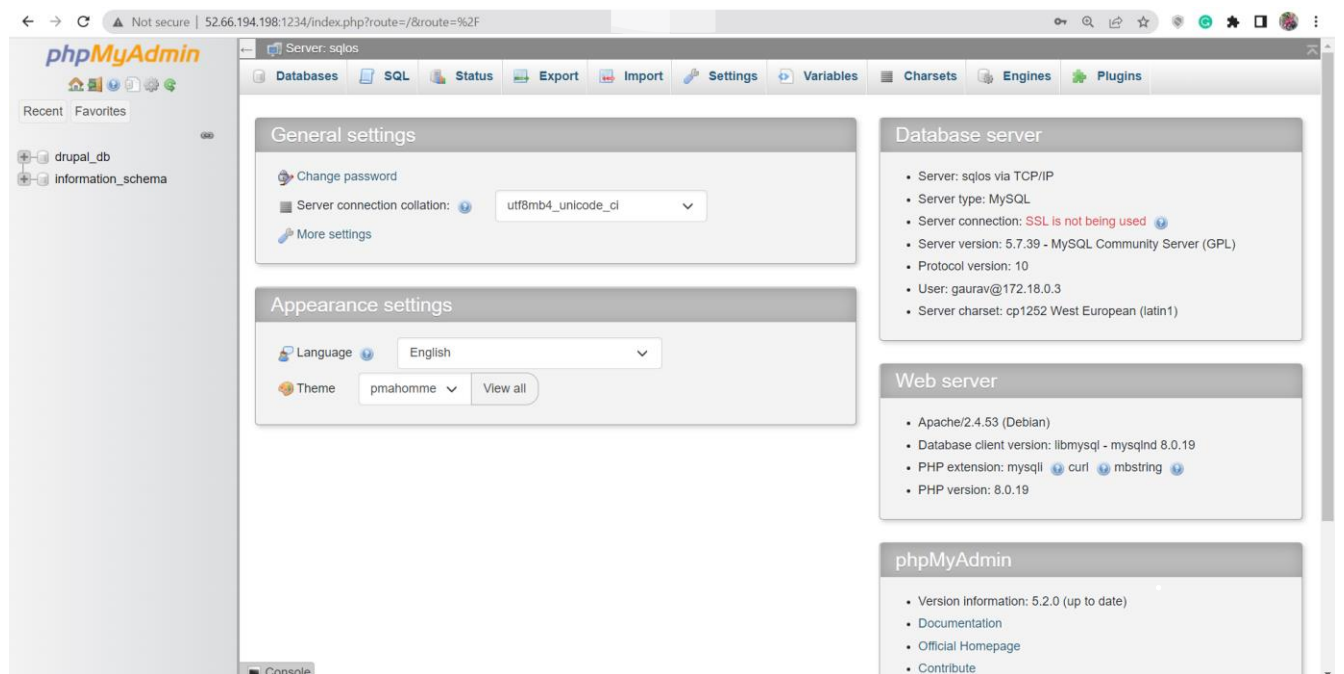
Log in

Username: ccis

Password: *****

Log in

To login to the Phpmyadmin panel, use username as root and password as MYSQL_ROOT_PASSWORD. The password was already set in the MySQL environment variables (MYSQL_ROOT_PASSWORD: MYSQL_ROOT_PASSWORD)



← → ↻ Not secure | 52.66.194.198:1234/index.php?route=/&route=%2F

phpMyAdmin

Recent Favorites

drupal_db
information_schema

Server: sqlos

Databases SQL Status Export Import Settings VariablesCharsetsEnginesPlugins

General settings

Change password

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme View all

Database server

- Server: sqlos via TCP/IP
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 5.7.39 - MySQL Community Server (GPL)
- Protocol version: 10
- User: gaurav@172.18.0.3
- Server charset: cp1252 West European (latin1)

Web server

- Apache/2.4.53 (Debian)
- Database client version: libmysql - mysqlnd 8.0.19
- PHP extension: mysqli curl mbstring
- PHP version: 8.0.19

phpMyAdmin

- Version information: 5.2.0 (up to date)
- Documentation
- Official Homepage
- Contribute

Console

You can now see the database we defined is already set as MYSQL_DATABASE, and you can start interacting with Phpmyadmin.

Drupal 9.4.8

Choose language

Choose profile

Verify requirements

Set up database

Install site

Configure site

Database configuration

Database type *

☒ MySQL, MariaDB, Percona Server, or equivalent

☐ PostgreSQL

☐ SQLite

Database name *

drupal_db

Database username *

CCIS

Database password

.....

▼ ADVANCED OPTIONS

Host *

sqls

Port number

3306

Table name prefix

.....

If more than one application will be sharing this database, a unique table name prefix - such as *drupal_* - will prevent collisions.

Save and continue

Show desktop

Conclusion: Thus we have successfully executed php application and MySQL server on docker.