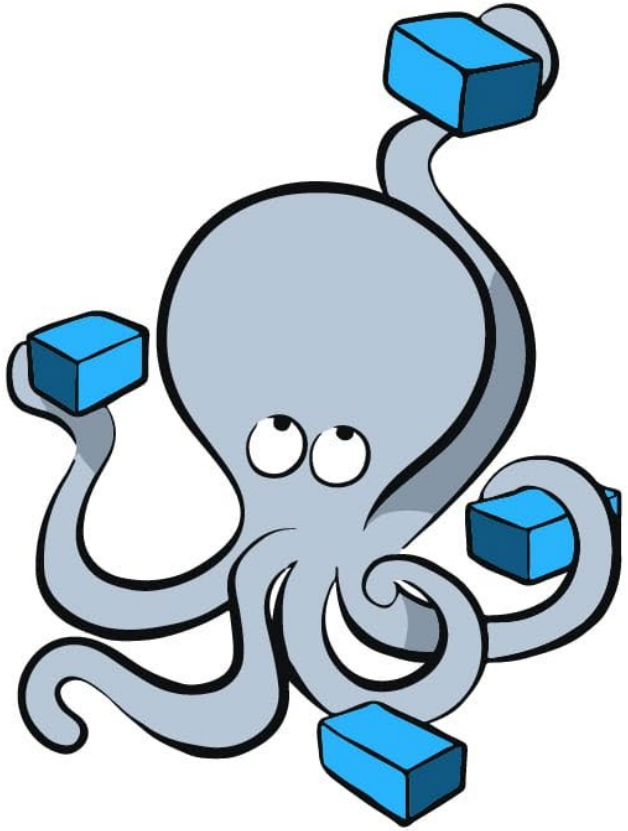

WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

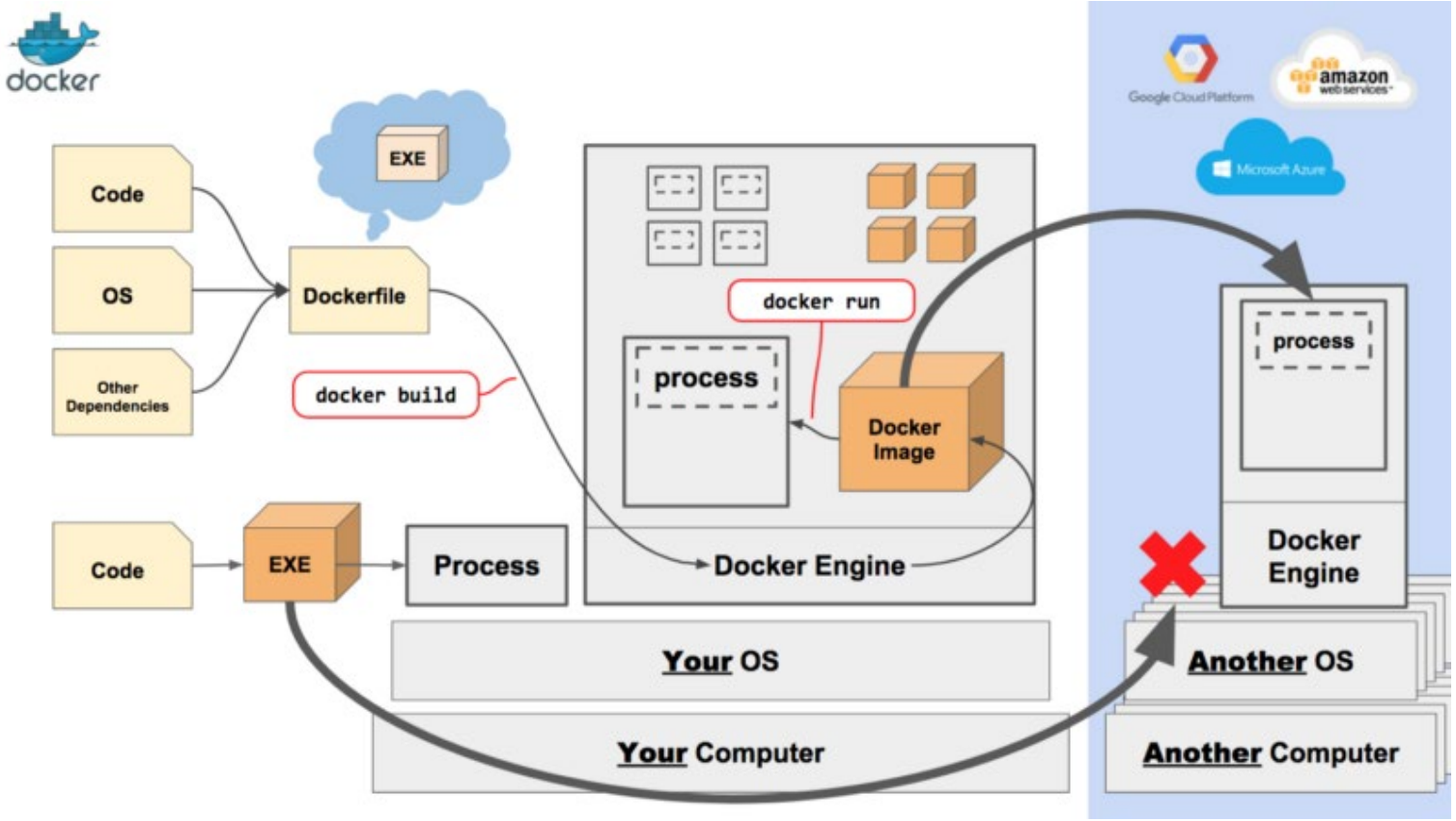
DevOps

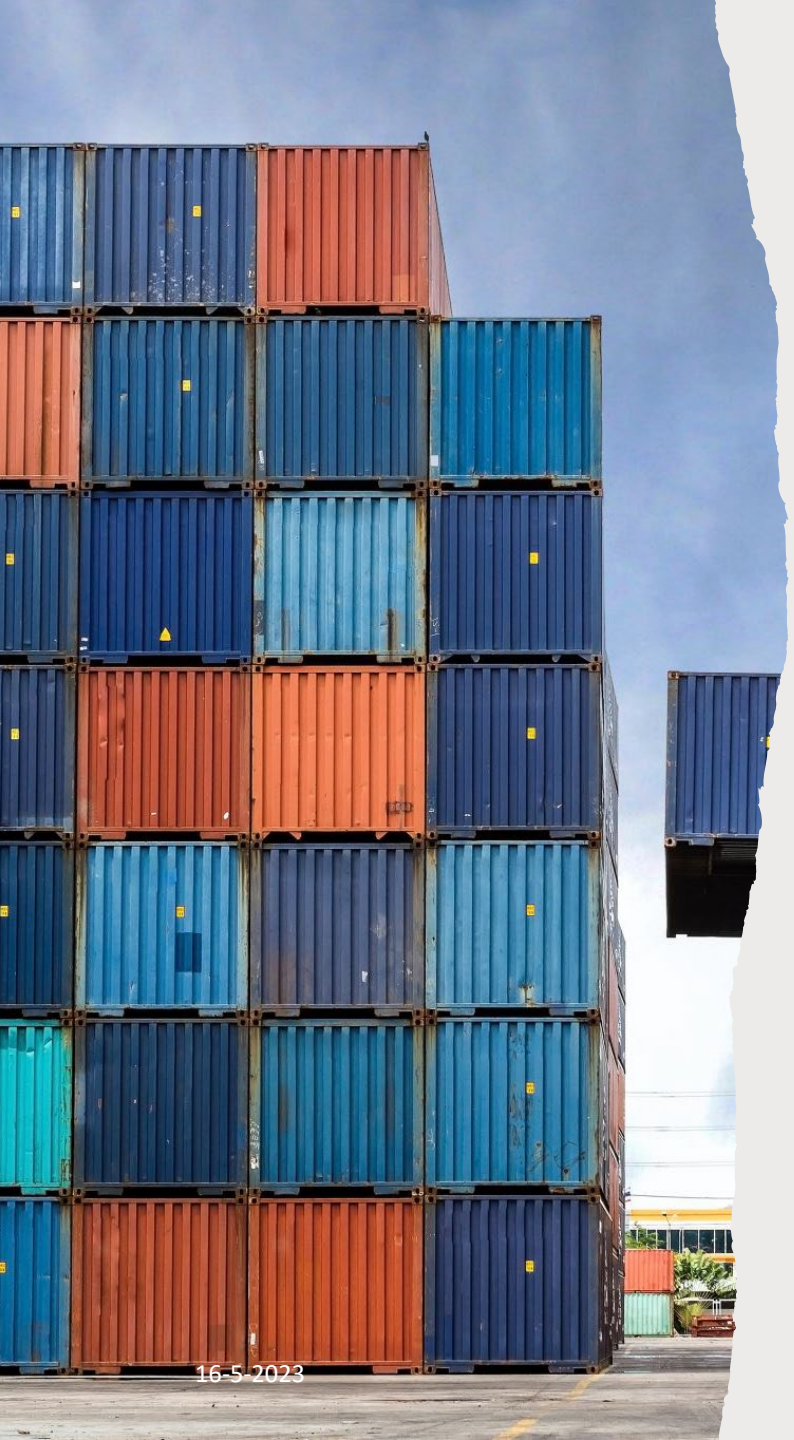
Framework Project 2



docker Compose

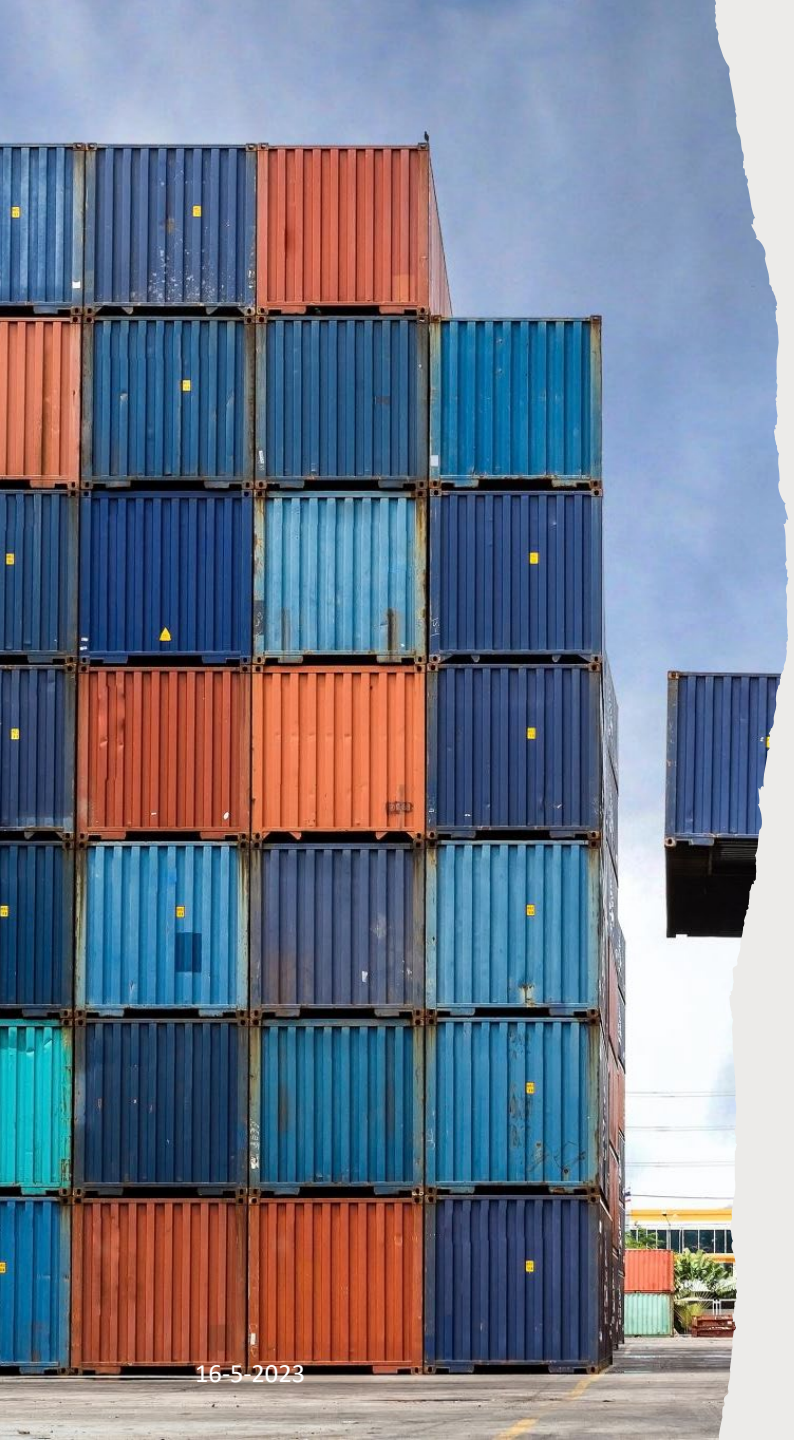
Docker!


docker
Code
OS
Other Dependencies
Dockerfile
EXE
docker build
docker run
process
Docker Image
Docker Engine
Code
EXE
Process
Your OS
Your Computer
Google Cloud Platform
amazon web services
Microsoft Azure
process
Docker Engine
Another OS
Another Computer



Docker

- Containers
- Images
- Dockerfiles
- Compose
- Kubernetes (k8s)
- Orchestration



Docker

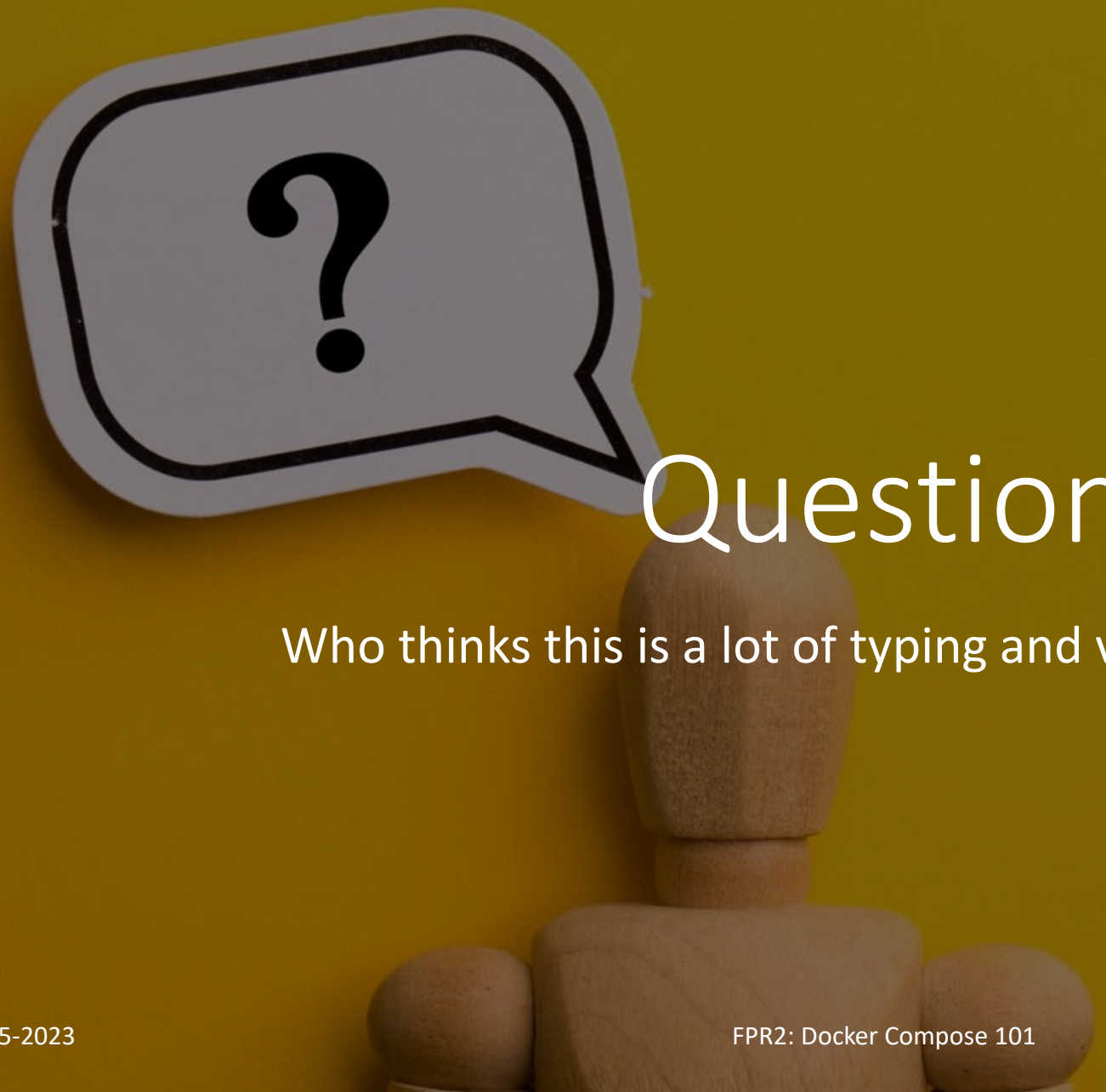
• Containers	Last lecture + the one before
• Images	Last lecture
• Dockerfiles	Last lecture
• Compose	Today
• Kubernetes (k8s)	Won’t do
• Orchestration	Won’t do

Part 2 – Step 7

Start new database and php app containers and connect them to the network

```
docker run --name myserver --network my-network -e MYSQL_ROOT_PASSWORD=ActuallyNotThatBadAPassword database

# # Powershell: use ${PWD}
docker run -v --network my-network $PWD:/var/www/html example-app
```



Question

Who thinks this is a lot of typing and very error-prone?

Today

- What is a Docker compose?
- How does it relate to images and containers?
- How do you use it?

Exercise for last sprint

- Execute DevOps assignment 1
- It's not meant to be easy
- Spend some time on it (40 points is about 18 hours of work!)

- (optional) Find https://github.com/HZ-HBO-ICT/docker-course, read and execute *Lecture 2* again

How is it going?

What is Docker Compose?

Less typing, more doing

Remember all those lengthy commands you needed to type last time?

```
docker run --name myserver --network my-network -e MYSQL_ROOT_PASSWORD=ActuallyNotThatBadAPassword database

# # Powershell: use ${PWD}
docker run -v --network my-network $PWD:/var/www/html example-app
```

Docker Compose replaces that with

```
docker compose up


# Or if you want to keep your terminal clear
docker compose up -d
```


THE BEATLES
ALL TOGETHER NOW
APPLE
STEREO

Please note

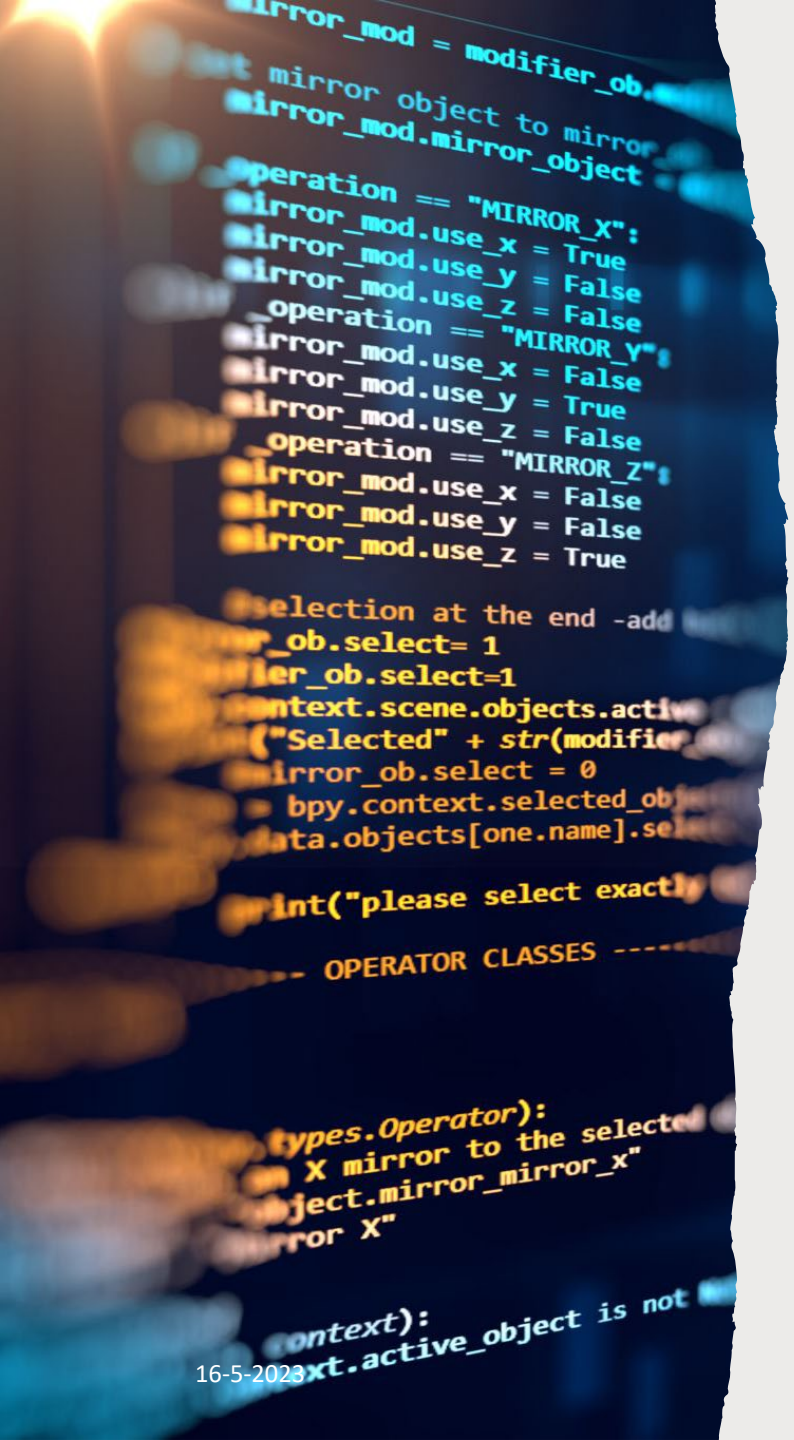
If you still haven't installed Docker yet

Just watch with a classmate

Installing it takes too long to wait for you

And you should have done your homework ☺

(And you're bulldozing your assignments too much!)



Goal

Part 1: create a simple Docker setup that contains PhpMyAdmin and a MySQL server

Part 2: Compose-ify our setup

Part 1 – Step 1

Create a folder

Add a Dockerfile

```
FROM phpmyadmin:latest
```

Part 1 – Step 2

Build the Dockerfile

```
docker build -t pma ./
```

Part 1 – Step 3

Run the containers!

(last time typing so much, pinky swear!)

```
# Start MySQL
docker run --name mysql-server -e MYSQL_ROOT_PASSWORD=badpassword mysql:8

# Start PhpMyAdmin
# --link is shortcut for creating and adding to a Docker network
docker run --name pma --link mysql-server:db -p 8081:80 pma
```

Part 1 – Step 4

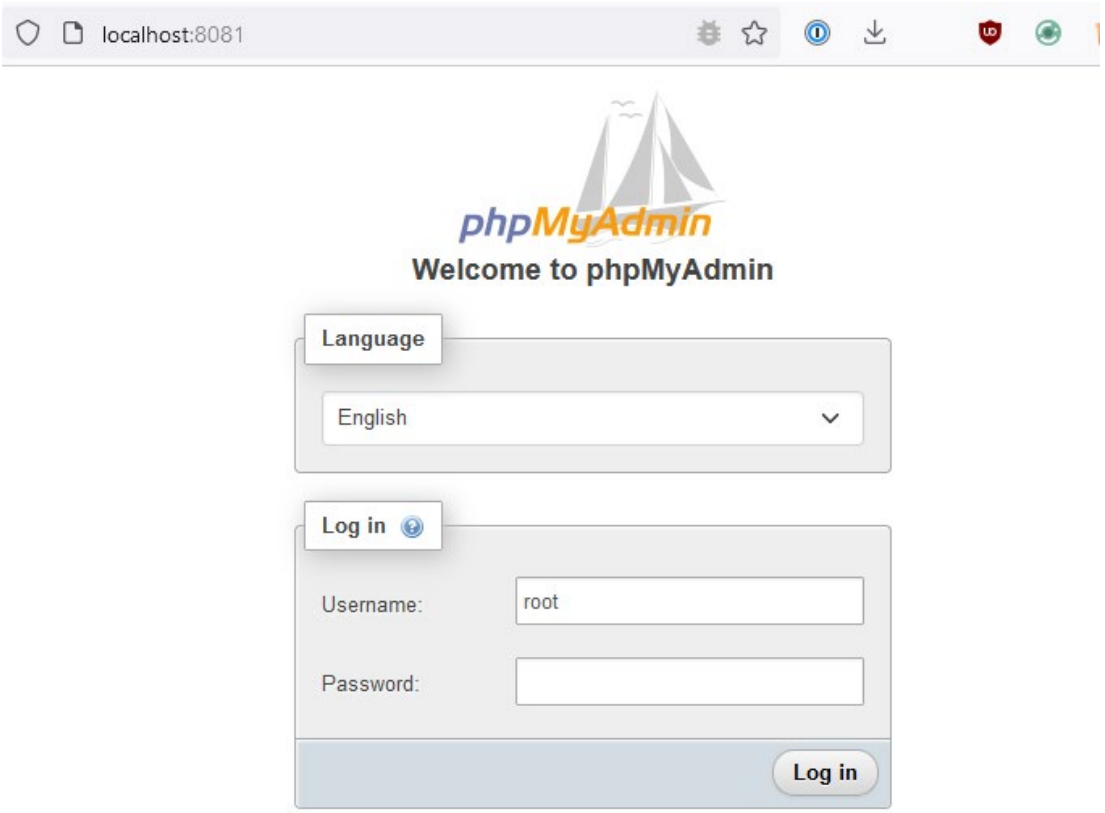
Browse to localhost:8081

(or given IP address in Linux)

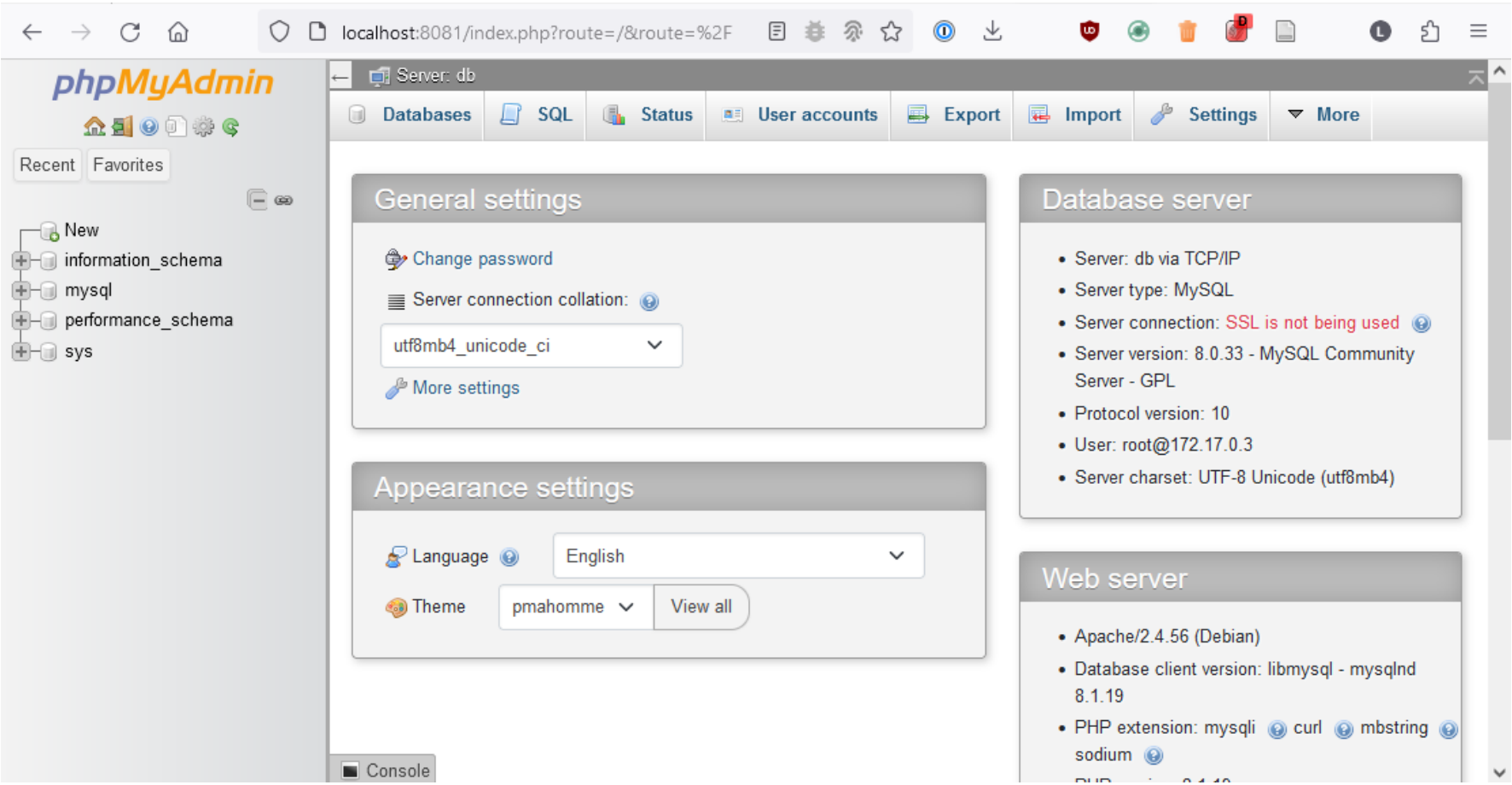
Log in with

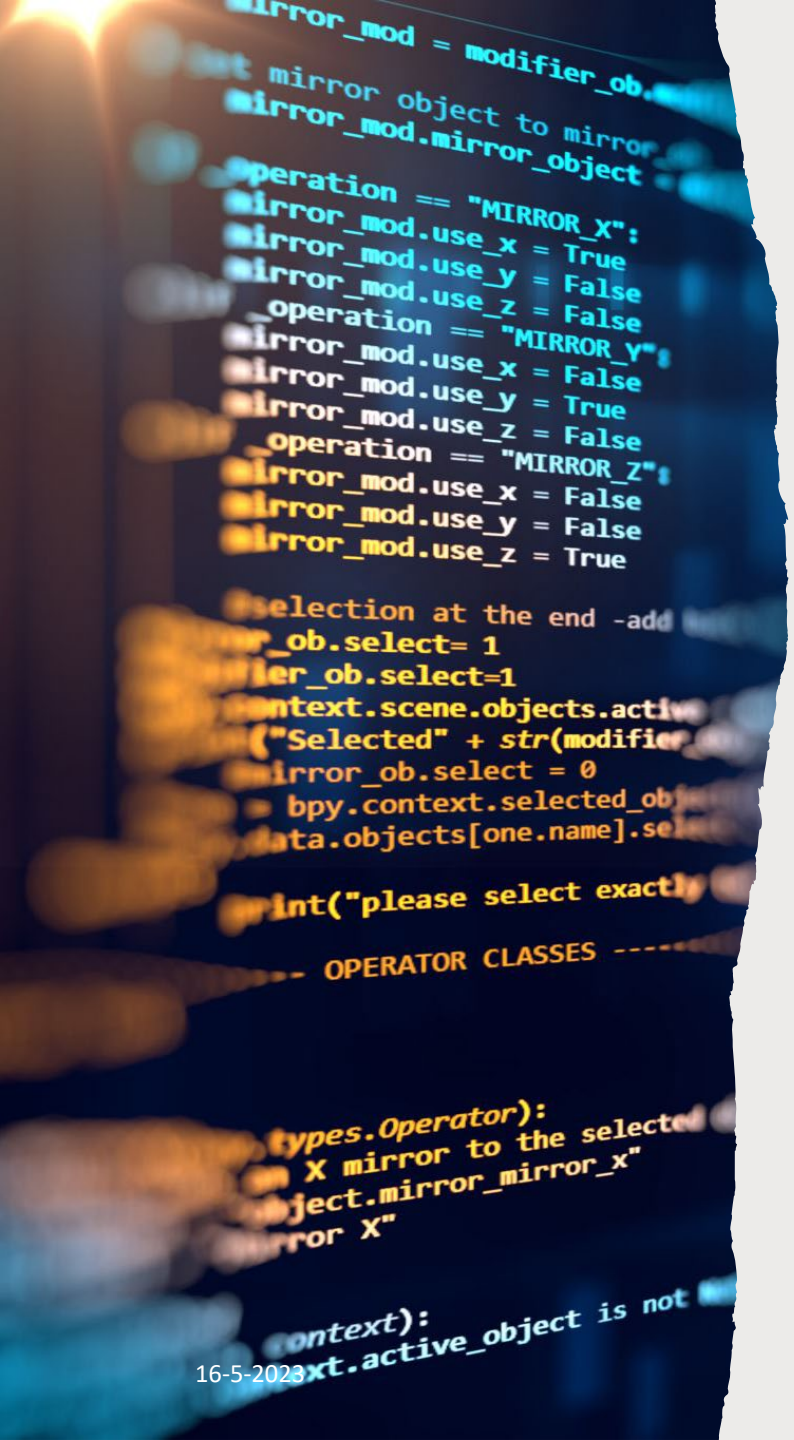
User: root

Pass: password you gave
in docker command



Part 1 – Step yaaaay





Goal

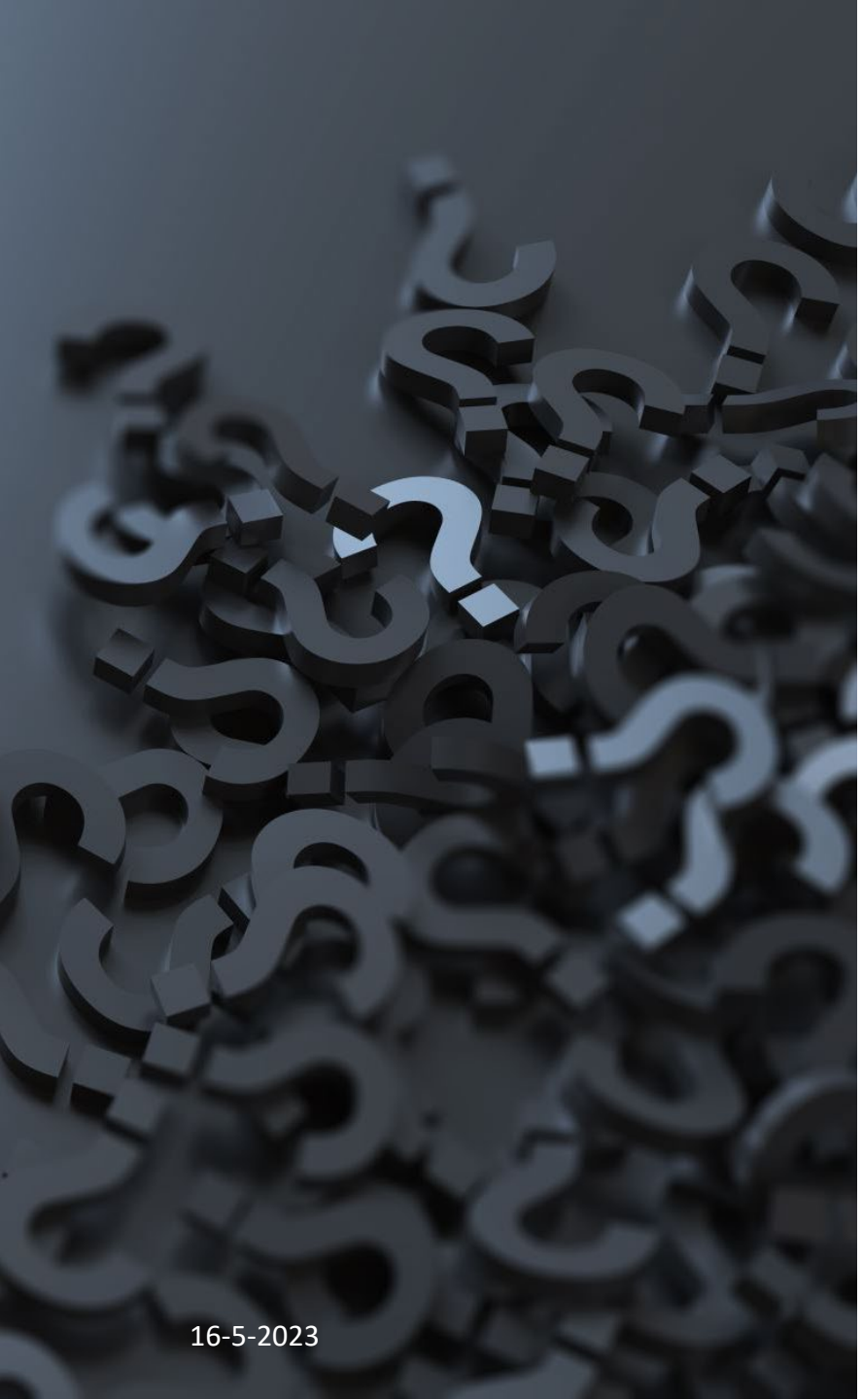
~~**Part 1**: create a simple Docker setup that contains PhpMyAdmin and a MySQL server~~

Part 2: Compose-ify our setup

Part 2 – Step 1

Create a file “docker-compose.yml”

Without capital D!



Interlude

“yml” stands for “YAML”

YAML is space sensitive:

2 spaces means something else than 4

Indents have meaning

Be careful!

Part 2 – Step 2

Add to compose file

```yaml
version: '3'

networks:
    mysql-pma-network:
        driver: bridge

# Containers are now called services for some reason
services:
    mysql-server:
        image: mysql:8
        environment:
            MYSQL_ROOT_PASSWORD: badpassword
            # Let's actually add a database now
            MYSQL_DATABASE: lecture
            MYSQL_USER: db_username
            MYSQL_PASSWORD: db_password
        networks:
            - mysql-pma-network
```

Part 2 – Step 3

Add to same indent level as mysql-server

```yaml
20    pma:
21        # Make sure MySQL starts before PhpMyAdmin
22        depends_on:
23            - mysql-server
24        # Build the Dockerfile in the current directory
25        build: ./
26        ports:
27            - 8081:80
28        environment:
29            PMA_HOST: mysql-server
30        networks:
31            - mysql-pma-network
```

Part 2 – Step 4 – The magic!

```
docker compose up


# Or if you want to keep your terminal clear
docker compose up -d
```

Part 2 – Step 5

Wait until MySQL is started

```
lecture-3-mysql-server-1  | 2023-05-16 14:05:08+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
lecture-3-mysql-server-1  |
lecture-3-mysql-server-1  | 2023-05-16T14:05:08.472967Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated
 and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
lecture-3-mysql-server-1  | 2023-05-16T14:05:08.475105Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.33) starting as
process 1
lecture-3-mysql-server-1  | 2023-05-16T14:05:08.485130Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
lecture-3-mysql-server-1  | 2023-05-16T14:05:08.719236Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
lecture-3-mysql-server-1  | 2023-05-16T14:05:09.068051Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
lecture-3-mysql-server-1  | 2023-05-16T14:05:09.068097Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS.
 Encrypted connections are now supported for this channel.
lecture-3-mysql-server-1  | 2023-05-16T14:05:09.075999Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Locat
ion '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
lecture-3-mysql-server-1  | 2023-05-16T14:05:09.122292Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Vers
ion: '8.0.33'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
lecture-3-mysql-server-1  | 2023-05-16T14:05:09.122111Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address:
 '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
```

Part 2 – Step 5

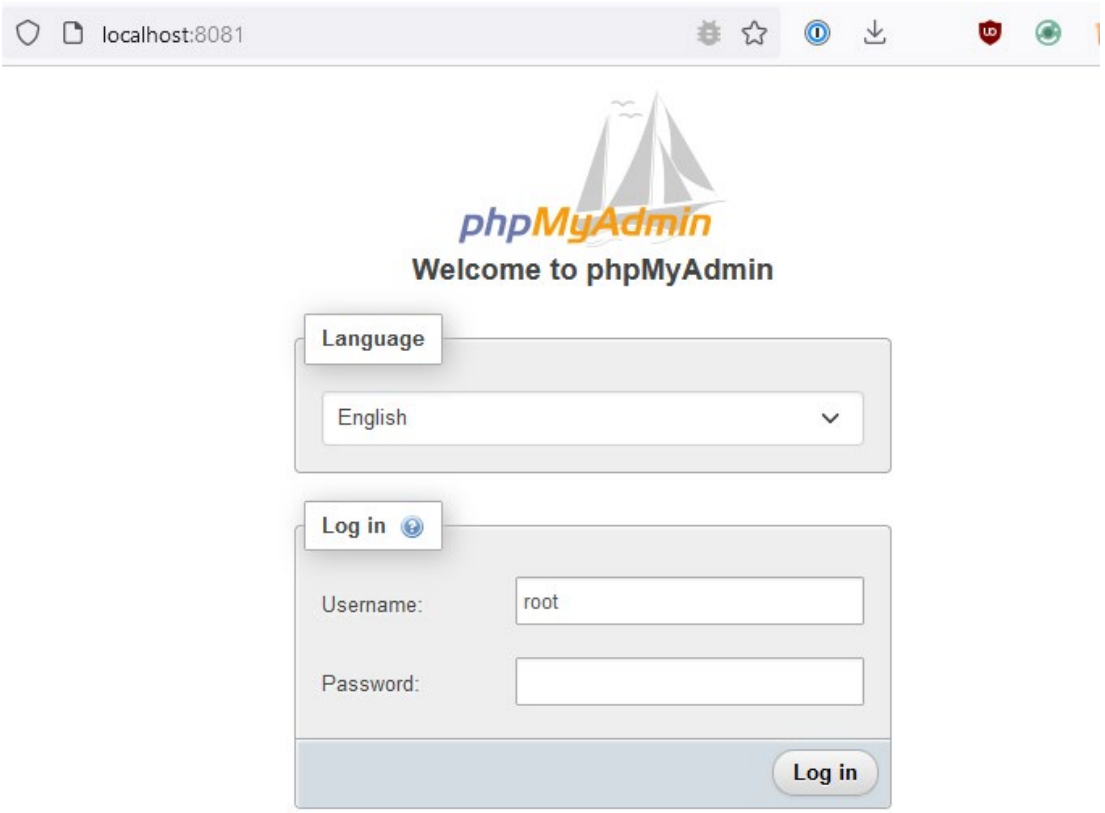
Browse to localhost:8081

(or given IP address in Linux)

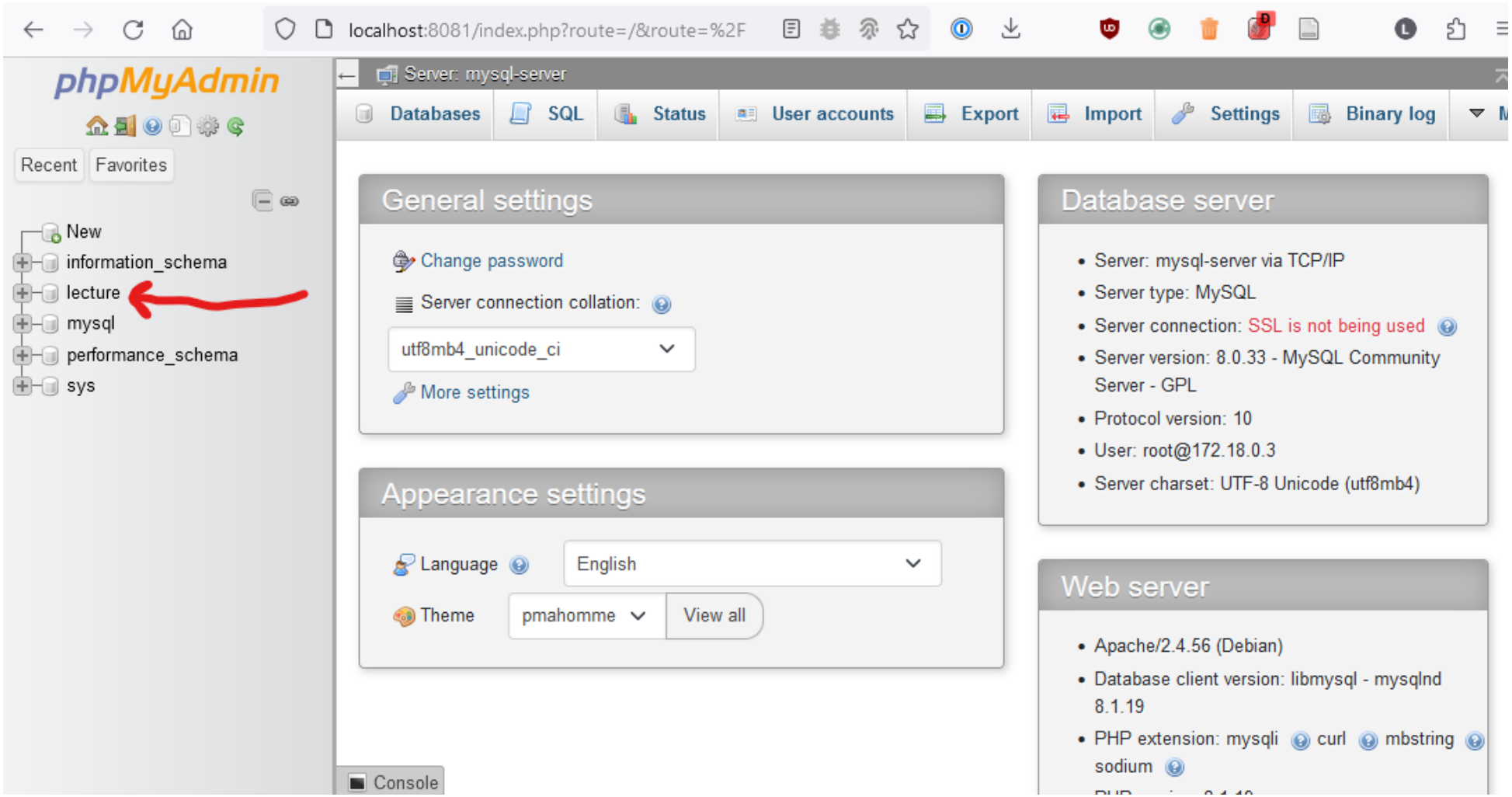
Log in with

User: root

Pass: password you gave
in docker compose



Part 2 – Step 6


localhost:8081/index.php?route=/&route=%2F
phpMyAdmin
Recent
Favorites
New
information_schema
lecture
mysql
performance_schema
sys
Server: mysql-server
Databases
SQL
Status
User accounts
Export
Import
Settings
Binary log
General settings
Change password
Server connection collation:
utf8mb4_unicode_ci
More settings
Appearance settings
Language
English
Theme
pmahomme
View all
Console
Database server
Server: mysql-server via TCP/IP
Server type: MySQL
Server connection: SSL is not being used
Server version: 8.0.33 - MySQL Community Server - GPL
Protocol version: 10
User: root@172.18.0.3
Server charset: UTF-8 Unicode (utf8mb4)
Web server
Apache/2.4.56 (Debian)
Database client version: libmysql - mysqlnd 8.1.19
PHP extension: mysqli curl mbstring sodium

Careful

The name of a service is it’s URL on the Docker network

So “mysql-server” is actually a URL on the Docker Network (it’s only a process on a network namespace after all!)

```yaml
services:
    mysql-server:
        image: mysql:8
```

Question

Who likes Docker more now?

Next lecture

Exercise for this sprint

- Execute DevOps assignment 2
- It's probably easier than the first
- Spend some time on it (40 points is about 18 hours of work!)

- (optional) Find https://github.com/HZ-HBO-ICT/docker-course, read and execute *Lecture 3* again