

# **Université Episcopale d'Haïti**

## **UNEPH**



**Faculté des Sciences Informatiques**

**Cours Virtualisation**

**Professeurs : Pr Elie TAGNE FUTE**

**Dr. Miguel Foko Sind Joung**

**Rapport du projet Lamp & l'application web sur  
l'environnement Docker**

**Préparé par : Carlos Philippe**

Avril 202

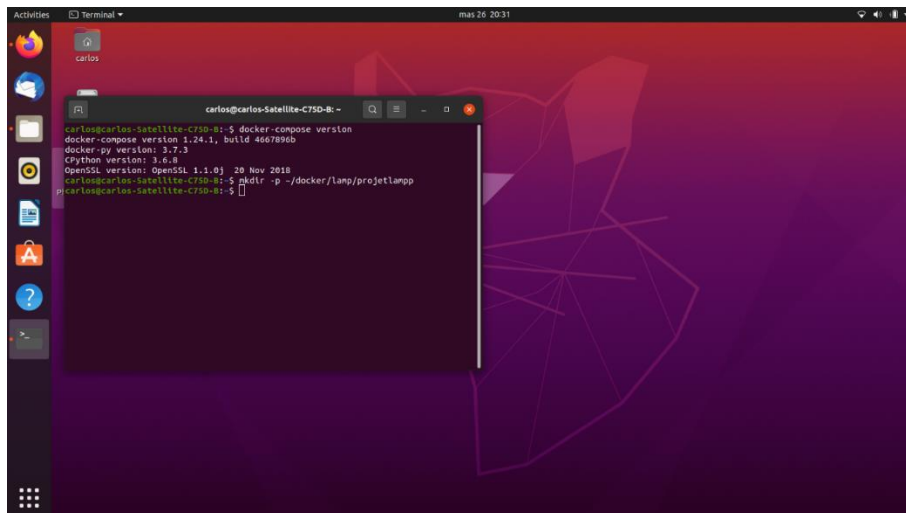
## Introduction

Dans le cadre du cours de virtualisation à l'Université Episcopale d'Haïti (UNEPH) on nous a demandé d'installer Lamp (Linux, Apache, MySql et PHP) sur un environnement Docker et de développer une application web dans cet environnement.

Voici les étapes que nous utilisons pour réaliser ce projet.

- 1- Création d'un répertoire de projet ~/docker/lamp (disons) et un répertoire html/ dans le répertoire du projet pour conserver les fichiers du site Web (c'est-à-dire php, html, css, js, etc.) comme suit :

```
$ mkdir -p ~/docker/lamp/html
```



Accédons au répertoire du projet ~/docker/lamp comme suit :

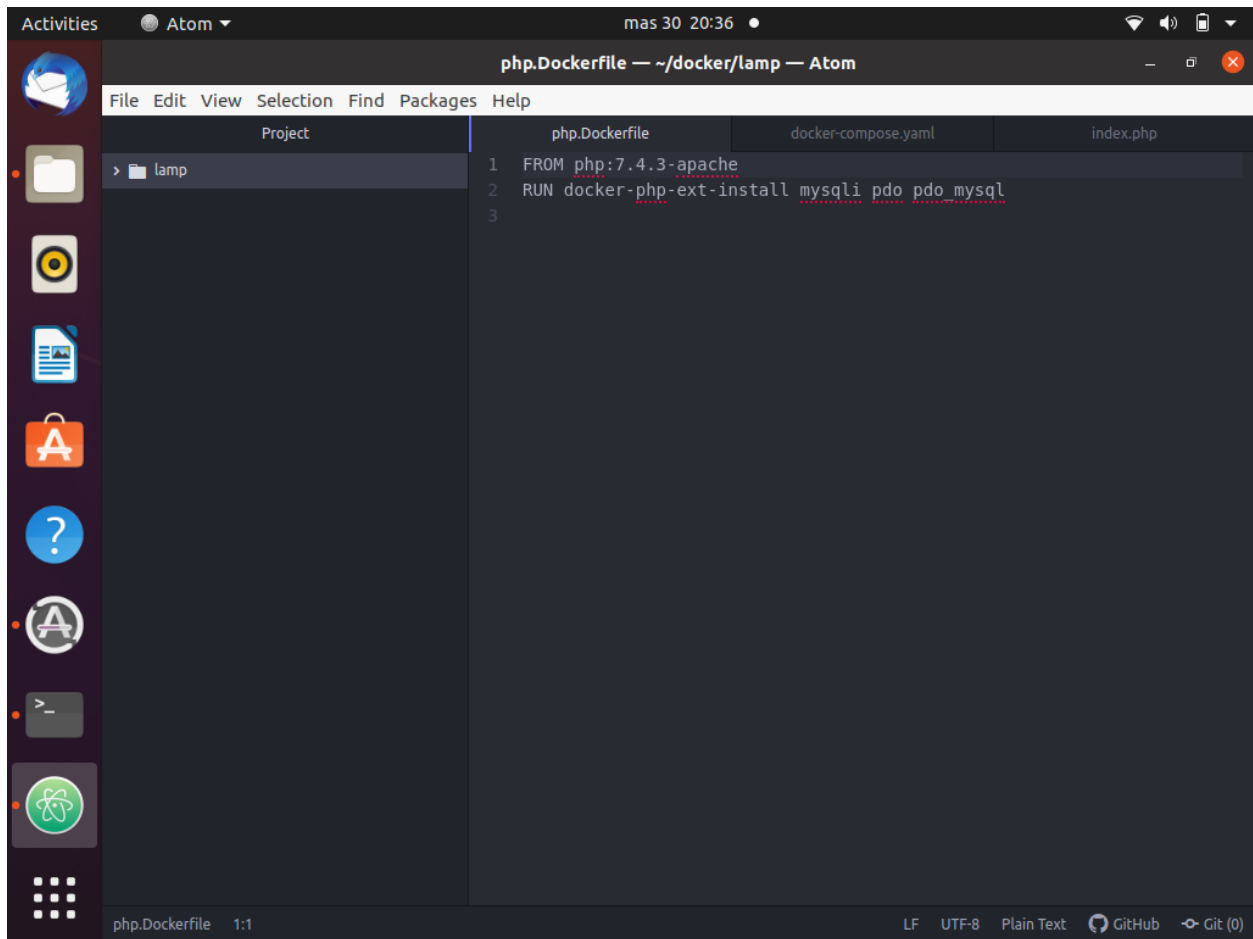
```
$ cd ~/docker/lamp
```

Créons un fichier **php.Docker** dans le répertoire du projet ~/docker/lamp. Il s'agit d'un Dockerfile qui active les extensions mysqli et PDO php dans l'image php:7.4.3-apache de Docker Hub et crée une image Docker personnalisée à partir de celle-ci.

Le contenu du php.Dockerfile est donné ci-dessous.

```
FROM php:7.4.3-apache
```

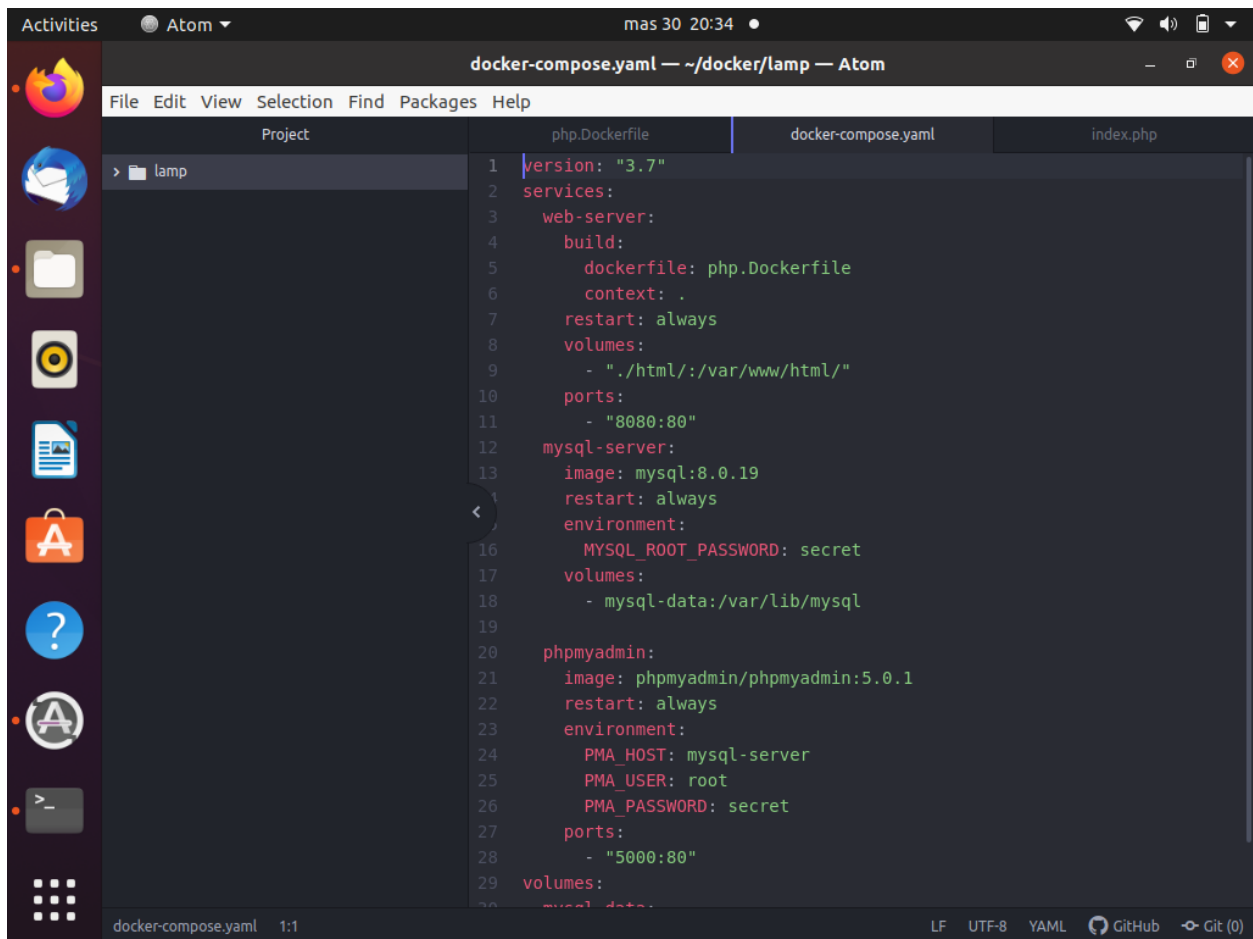
```
RUN docker-php-ext-install mysqli pdo pdo_mysql
```



Maintenant, créons un fichier **docker-compose.yaml** dans le répertoire du projet `~/docker/lamp` et saisissons les lignes suivantes dans le fichier `docker-compose.yaml`.

```
version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "/html:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - mysql-data:/var/lib/mysql
```

```
phpmyadmin:
  image: phpmyadmin/phpmyadmin:5.0.1
  restart: always
  environment:
    PMA_HOST: mysql-server
    PMA_USER: root
    PMA_PASSWORD: secret
  ports:
    - "5000:80"
volumes:
  mysql-data:
```

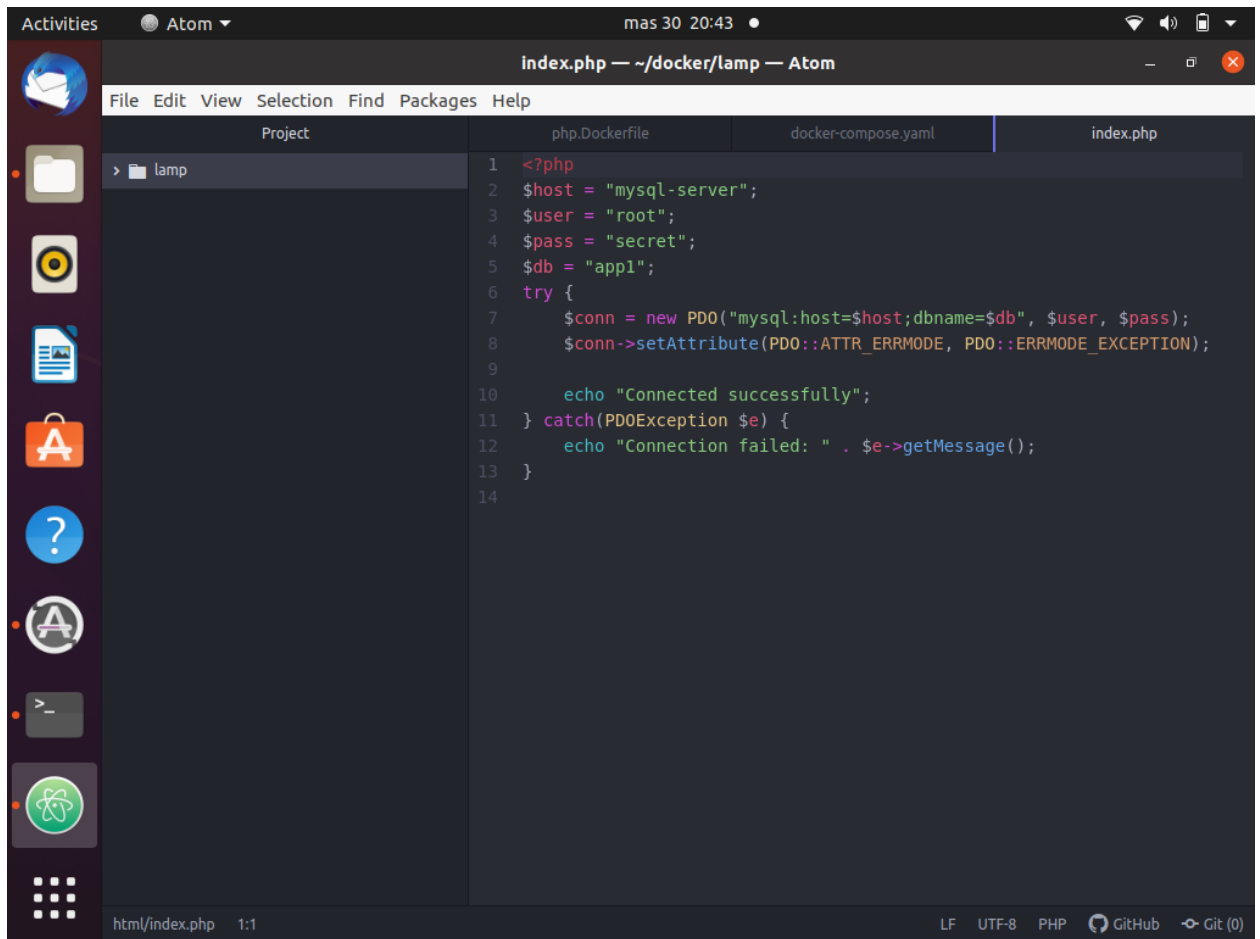


```
version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "./html:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - mysql-data:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: secret
    ports:
      - "5000:80"
    volumes:
      - mysql-data:
```

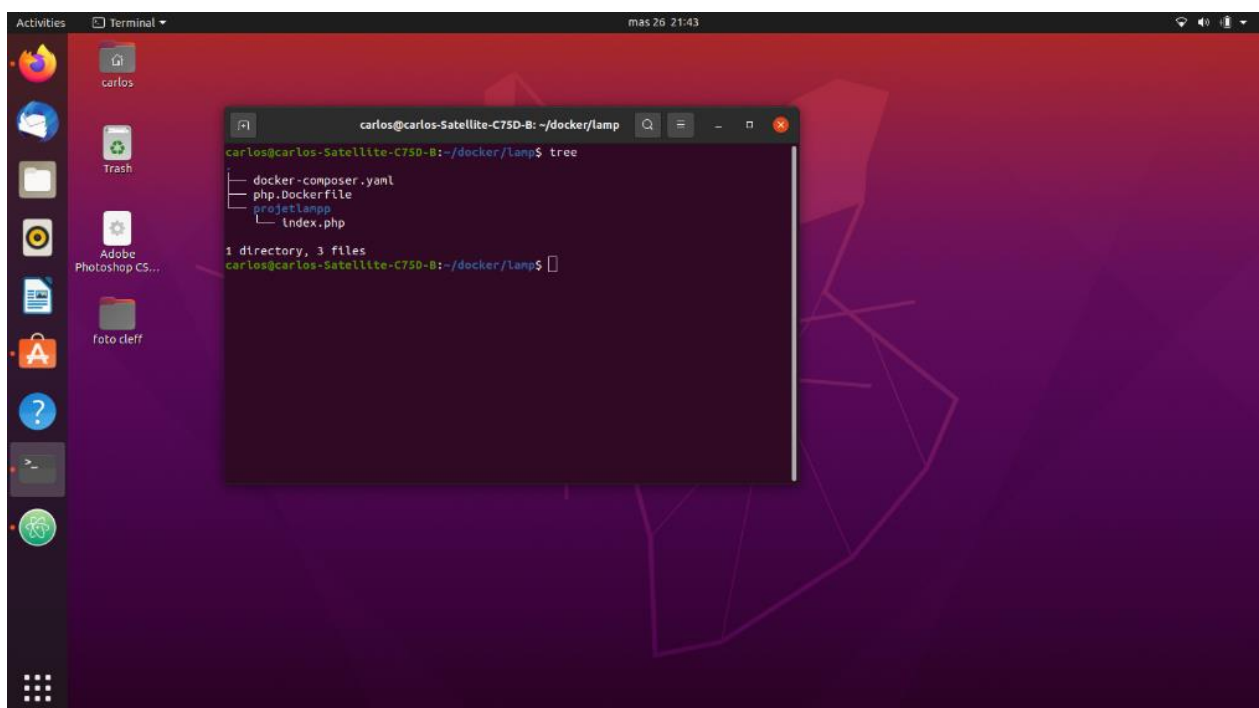
Nous avons créé 3 services **web-server**, **mysql-server** et **phpmyadmin**.  
Nous Créons également un fichier **index.php** dans le répertoire html/ pour tester le serveur LAMP.

4

```
<?php
$host = "mysql-server";
$user = "root";
$pass = "secret";
$db = "app1";
try {
    $conn = new PDO("mysql:host=$host;dbname=$db", $user, $pass);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```



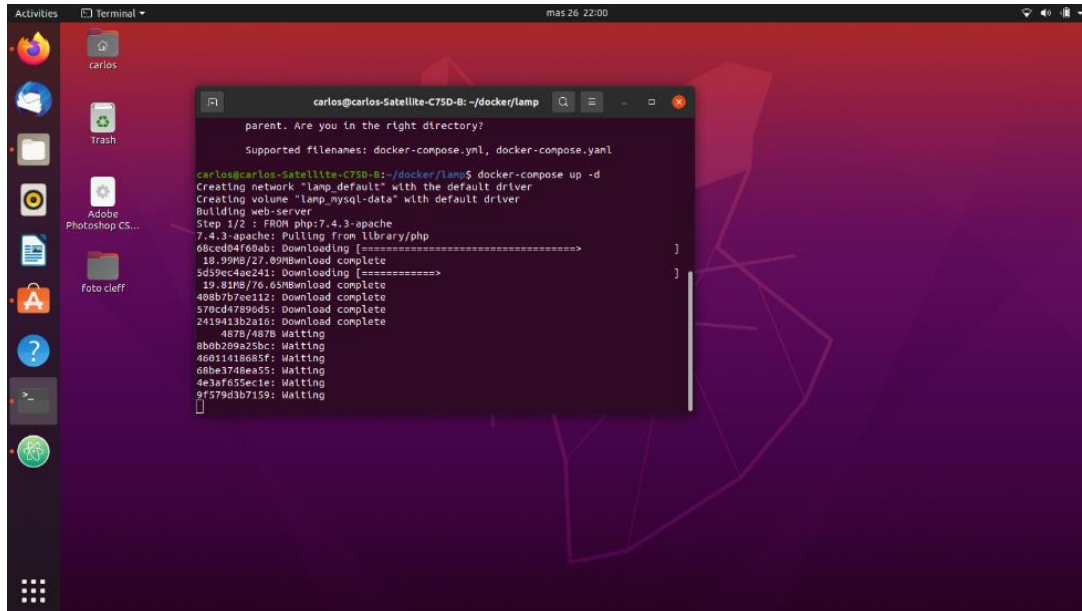
Enfin, le répertoire du projet `~/docker/lamp` devrait ressembler à ceci :



## Démarrage le serveur LAMP :

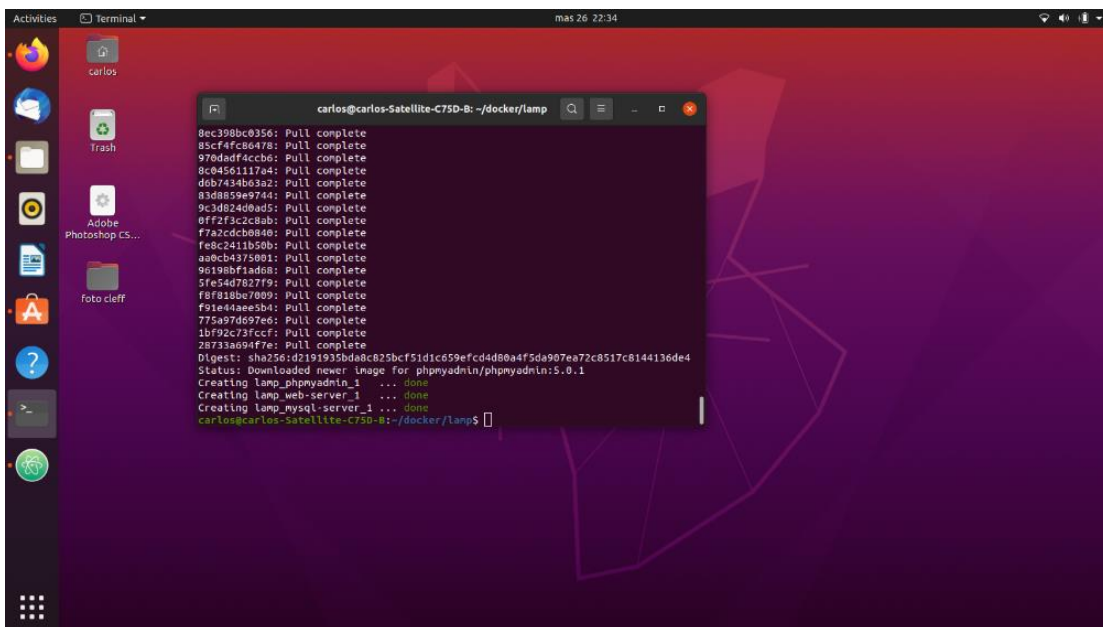
Maintenant, pour démarrer les services **web-server**, **mysql-server** et **phpmyadmin**, exécutons la commande suivante :

```
$ docker-compose up -d
```



```
carlos@carlos-Satellite-C75D-B: ~/docker/lamp
parent. Are you in the right directory?
Supported filenames: docker-compose.yml, docker-compose.yaml

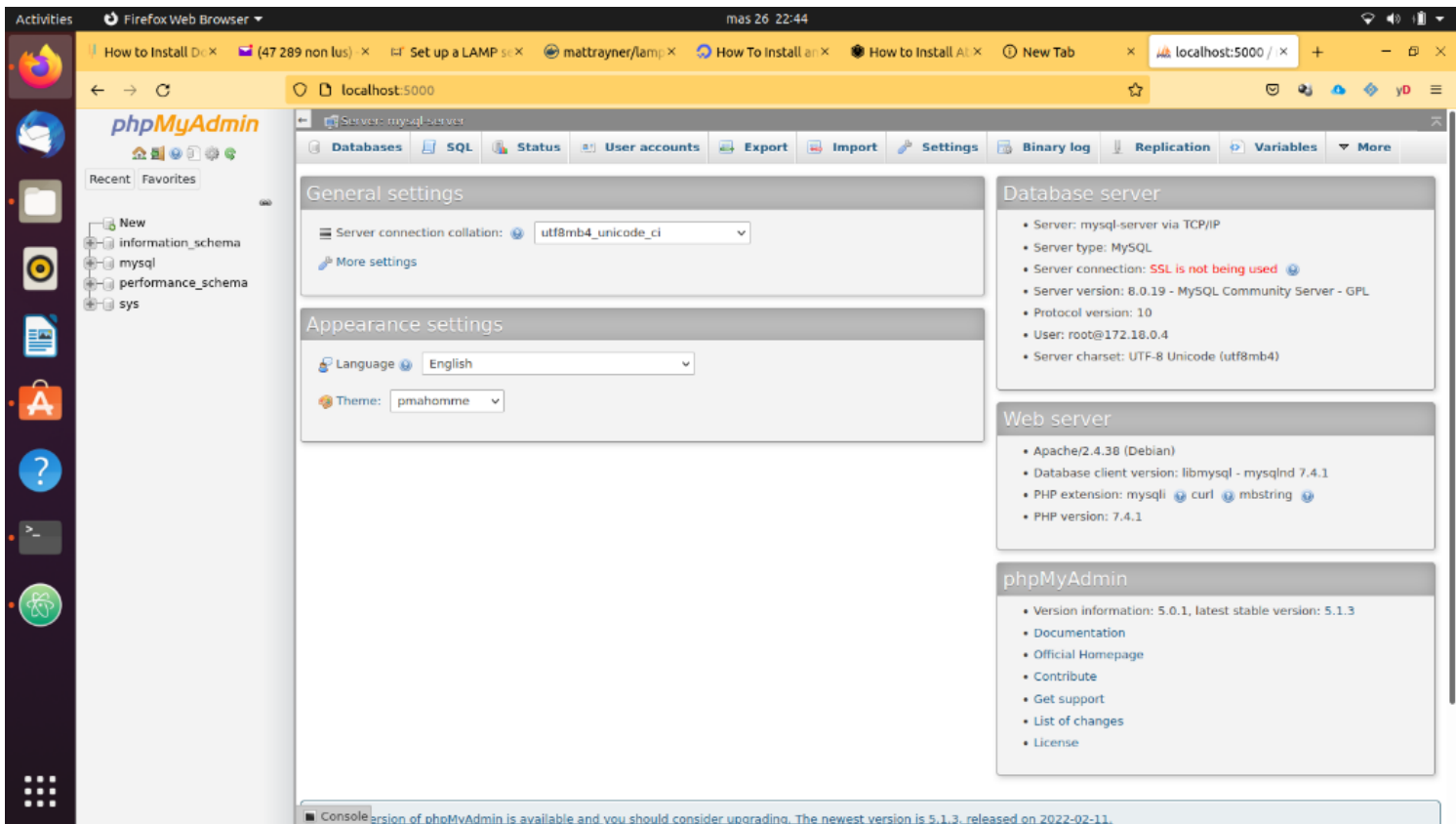
carlos@carlos-Satellite-C75D-B:~/docker/lamp$ docker-compose up -d
Creating network "lamp_default" with the default driver
Creating volume "lamp_mysql-data" with default driver
Building web-server
Step 1/2 : FROM php:7.4.3-apache
7.4.3-apache: Pulling from library/php
68ced94f60ab: Downloading [=====]
18.99MB/27.09MB Download complete
5d59ec4ae241: Downloading [=====]
19.83MB/76.05MB Download complete
480b7b7ee112: Download complete
570cd47896d5: Download complete
2419413b2a16: Download complete
4878/4878 Waiting
8b0b299225bc: Waiting
46011418685f: Waiting
68be3748ea55: Waiting
4e3af655ec1e: Waiting
97579d3b7159: Waiting
[]
```



```
carlos@carlos-Satellite-C75D-B: ~/docker/lamp
8ec390bc0356: Pull complete
85cf4fc86478: Pull complete
970dndf4ccb6: Pull complete
8c045611174d: Pull complete
d0b7434b63a2: Pull complete
83d8859e9744: Pull complete
9c3d82400ad5: Pull complete
0ff2f92cc8ab: Pull complete
f7a2cdcb0840: Pull complete
fe8c2411b50b: Pull complete
aa0cb4375001: Pull complete
9619b0f1ad0a: Pull complete
5fe54d7827f9: Pull complete
f8f810be7009: Pull complete
f91e44aee5b4: Pull complete
775a97d097ed: Pull complete
1b192c73fc9f: Pull complete
28733a604f7e: Pull complete
Digest: sha256:d2191935bda8c025bcf51dc659efcd4d80a4f5da907ea72c8517c8144136de4
Status: Downloaded newer image for phpmyadmin/phpmyadmin:5.0.1
Creating lamp_phpmyadmin_1 ... done
Creating lamp_web-server_1 ... done
Creating lamp_mysql-server_1 ... done
carlos@carlos-Satellite-C75D-B:~/docker/lamp$
```

Nous pouvons accéder à phpMyAdmin 5 et au serveur Web à partir d'un navigateur Web.

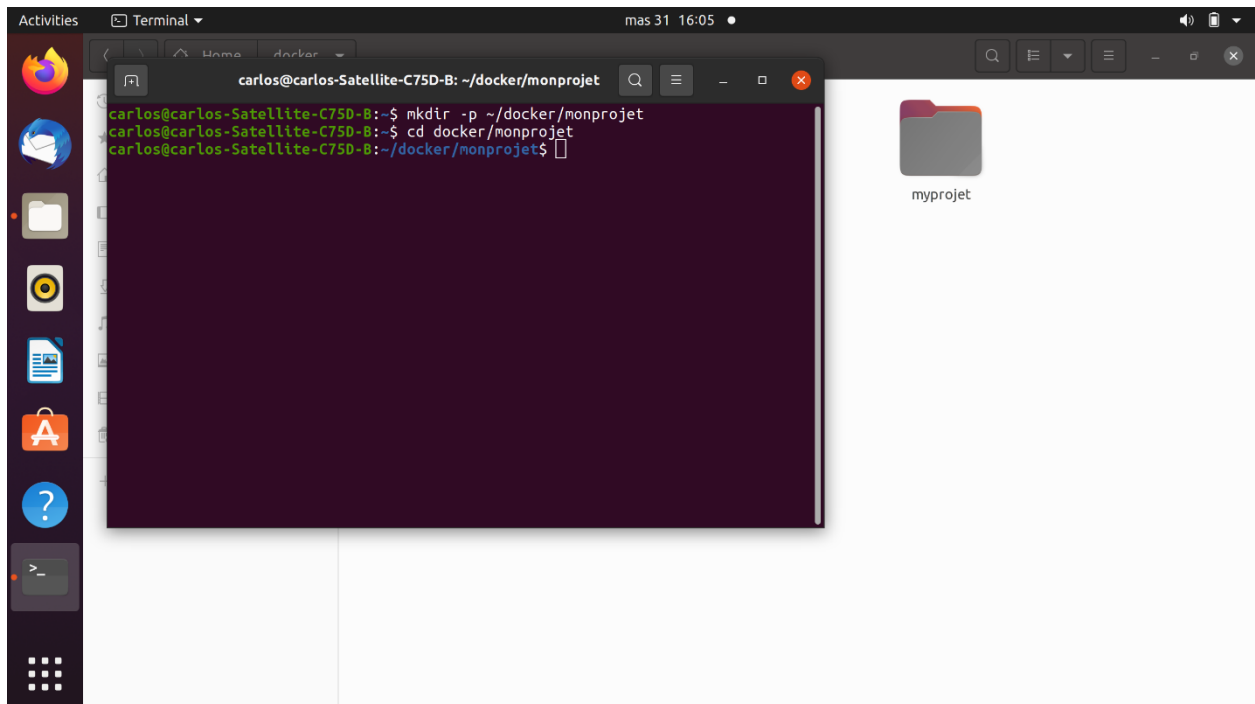
Pour accéder à phpMyAdmin 5, ouvrons un navigateur Web et visitons <http://localhost:5000>



## II- Réalisation de l'application web

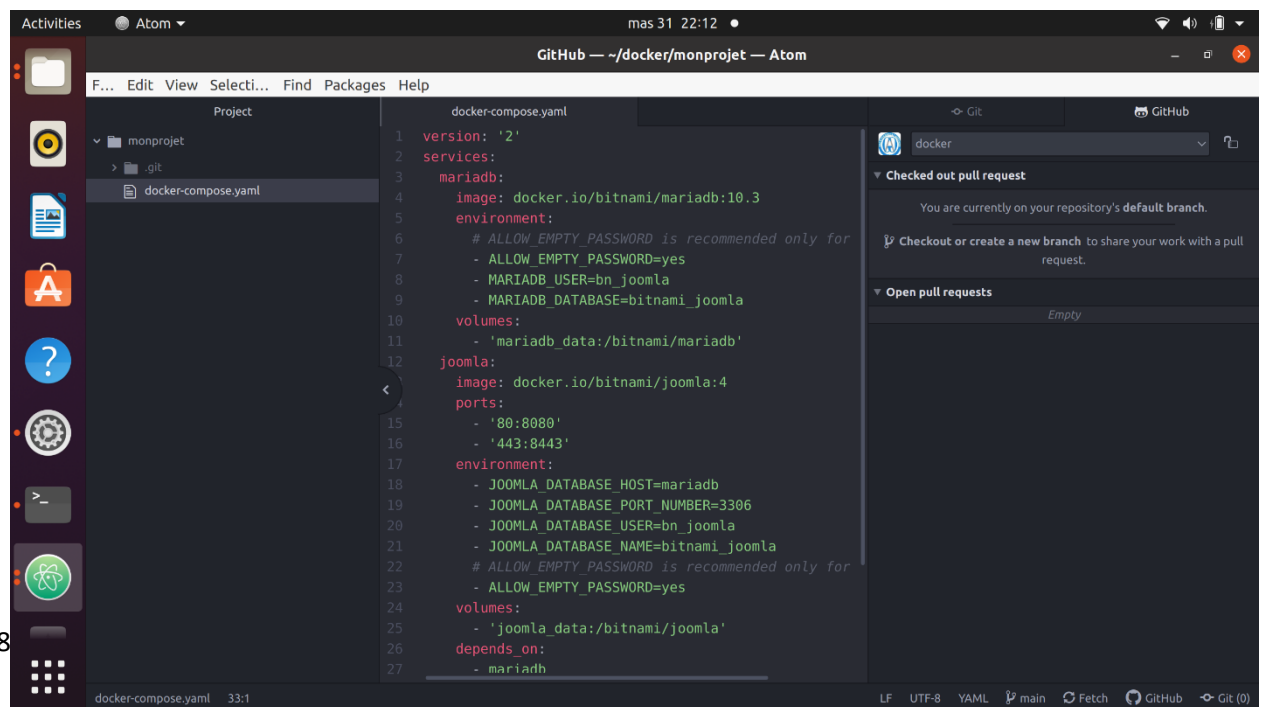
### 1- Création du répertoire de l'application

```
$ mkdir -p monprojet
```



### 2- Création du fichier **docker-compose.yaml**

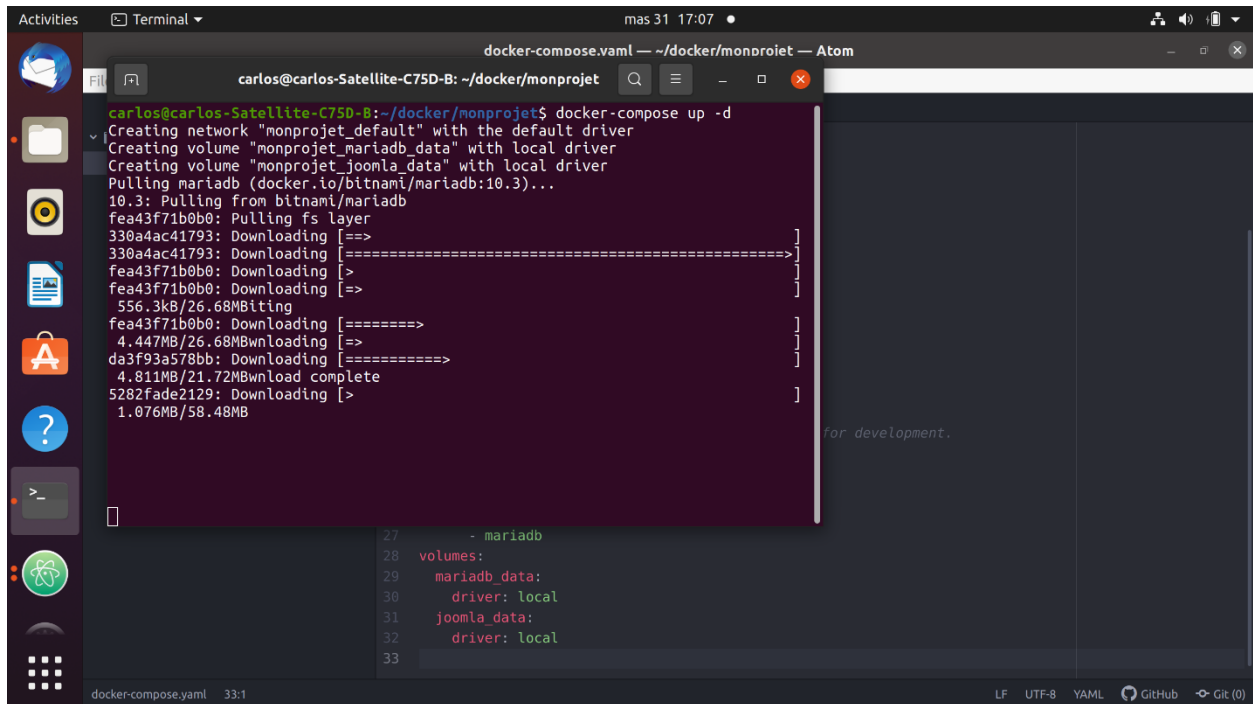
Pour notre application web on a utilisé le CMS joomla et nous avons installer sur l'environnement docker, en utilisant le fichier **docker-compose.yaml**



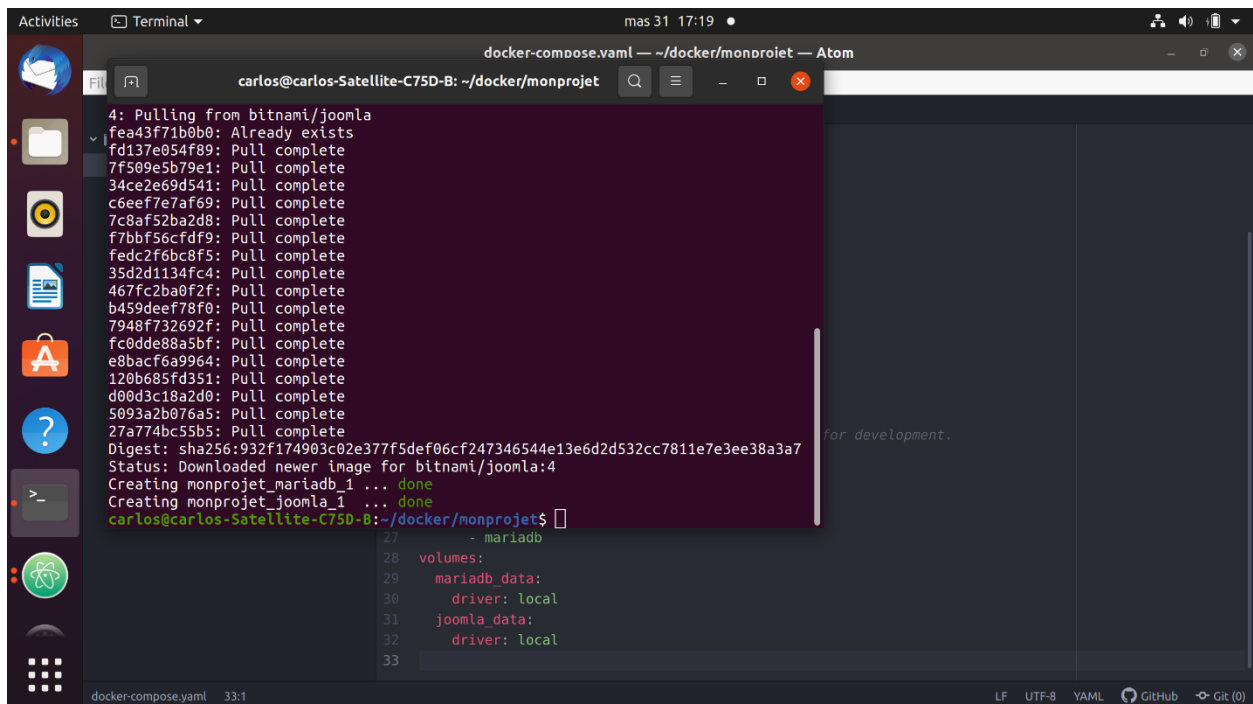


3- Installons le CMS joomla en utilisant la commande :

```
$ docker-compose up -d
```

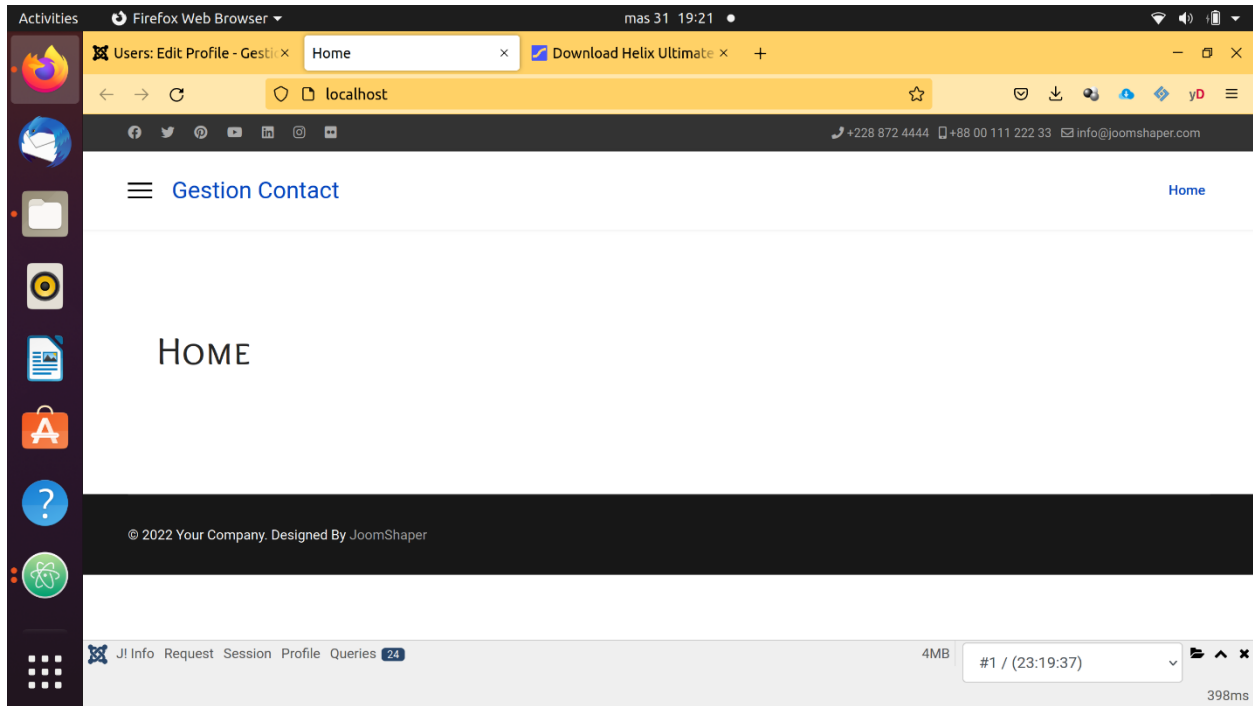


```
carlos@carlos-Satellite-C75D-B: ~/docker/monprojet
carlos@carlos-Satellite-C75D-B:~/docker/monprojet$ docker-compose up -d
Creating network "monprojet_default" with the default driver
Creating volume "monprojet_mariadb_data" with local driver
Creating volume "monprojet_joomla_data" with local driver
Pulling mariadb (docker.io/bitnami/mariadb:10.3)...
10.3: Pulling from bitnami/mariadb
fea43f71b0b0: Pulling fs layer
330a4ac41793: Downloading [==>]
330a4ac41793: Downloading [=====]
fea43f71b0b0: Downloading [==>]
fea43f71b0b0: Downloading [=====]
556.3kB/26.68MBbiting
fea43f71b0b0: Downloading [=====]
4.447MB/26.68MBwnloading [==>]
da3f93a578bb: Downloading [=====]
4.811MB/21.72MBwnload complete
5282fade2129: Downloading [==>]
1.076MB/58.48MB
27 - mariadb
28 volumes:
29 mariadb_data:
30 driver: local
31 joomla_data:
32 driver: local
33
```



```
carlos@carlos-Satellite-C75D-B: ~/docker/monprojet
4: Pulling from bitnami/joomla
fea43f71b0b0: Already exists
fd137e054f89: Pull complete
7f509e5b79e1: Pull complete
34ce2e69d541: Pull complete
c6eef7e7af69: Pull complete
7c8af52ba2d8: Pull complete
f7bbf56cfd9: Pull complete
fedc2f6bc8f5: Pull complete
35d2d1134fc4: Pull complete
467fc2ba0f2f: Pull complete
b459deef78f0: Pull complete
7948f732692f: Pull complete
fc0dde88a5bf: Pull complete
e8bacf6a9964: Pull complete
120b685fd351: Pull complete
d00d3c18a2d0: Pull complete
5093a2b076a5: Pull complete
27a774bc55b5: Pull complete
Digest: sha256:932f174903c02e377f5def06cf247346544e13e6d2d532cc7811e7e3ee38a3a7
Status: Downloaded newer image for bitnami/joomla:4
Creating monprojet_mariadb_1 ... done
Creating monprojet_joomla_1 ... done
carlos@carlos-Satellite-C75D-B:~/docker/monprojet$
27 - mariadb
28 volumes:
29 mariadb_data:
30 driver: local
31 joomla_data:
32 driver: local
33
```

4- A travers un navigateur web lançons le cms joomla en tapant localhost :



Finalement, nous utilisons un module de formulaire pour la création de la page d'accueil, il faut être s'inscrire pour avoir accès à la page d'accueil.

