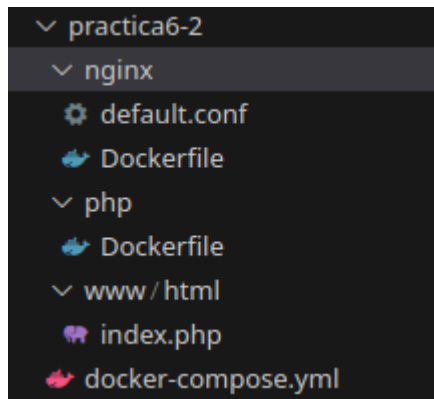

Práctica 6.2

Índice

- Estructura de directorios.....3
- Creación de un contenedor Nginx.....3
- Creación de un contenedor PHP.....3
- Creación de un contenedor para datos.....5
- Creación de un contenedor MySQL.....5
- Verificación de conexión a la base de datos.....7

Estructura de directorios

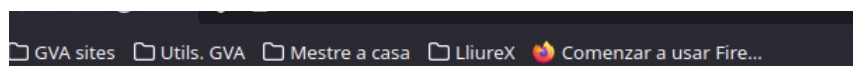
Para empezar necesitamos crear una estructura de directorios que tendrá la siguiente forma



Creación de un contenedor Nginx

Creamos un archivo docker-compose.yml con el siguiente código y usamos el comando docker-compose.yml para iniciar el contenedor

```
DEAW > UD6 > practica6-2 > docker-compose.yml
1  services:
2    nginx:
3      image: nginx:latest
4      container_name: nginx-container
5      ports:
6      - 8081:80
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Creación de un contenedor PHP

Ahora creamos index.html en la carpeta html

```
DEAW > UD6 > practica6-2 > www > html > index.php > body
1  <!DOCTYPE html>
2
3  <head>
4  |   <title>¡Hola mundo!</title>
5  </head>
6
7  <body>
8  |   <h1>¡Hola mundo!</h1>
```

Después creamos el archivo de configuración para que Nginx pueda correr PHP

```
DEAW > UD6 > practica6-2 > nginx > default.conf
1  server {
2
3      listen 80 default_server;
4      root /var/www/html;
5      index index.html index.php;
6
7      charset utf-8;
8
9      location / {
10         try_files $uri $uri/ /index.php?$query_string;
11     }
12
13     location = /favicon.ico { access_log off; log_not_found off; }
14     location = /robots.txt { access_log off; log_not_found off; }
15
16     access_log off;
```

Y por último creamos el Dockerfile para copia el archivo de configuración al contenedor correspondiente.

```
DEAW > UD6 > practica6-2 > nginx > Dockerfile
1  FROM nginx:latest
2  COPY ./default.conf /etc/nginx/conf.d/default.conf
```

Modificando el docker-compose.yml podremos obtener el siguiente resultado

```

DEAW > UD6 > practica6-2 > docker-compose.yml
1  services:
2    nginx:
3      build: ./nginx/
4      container_name: nginx-container
5      ports:
6        - 81:80
7      links:
8        - php
9      volumes:
10       - ./www/html:/var/www/html/
11
12    php:
13      image: php:7.0-fpm
14      container_name: php-container
15      expose:
16        - 9000
17      volumes:
18        - ./www/html:/var/www/html/
19

```

Creación de un contenedor para datos

A continuación vamos a crear un contenedor para datos para ello agregamos las siguientes líneas en el docker-compose

¡Hola mundo!

Estamos corriendo PHP, version: 7.0.33

```

app-data:
  image: php:7.0-fpm
  container_name: app-data-container
  volumes:
    - ./www/html:/var/www/html/
  command: "true"

```

Cuando verificamos que todo esta corriendo con docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
019d0389e6f9	practica6-2_nginx	"/docker-entrypoint..."	19 seconds ago	Up 16 seconds	0.0.0.0:81->80/tcp, :::81->80/tcp	nginx-container
c4f1e267b0c7	practica6-2_php	"docker-php-entrypoi..."	19 seconds ago	Up 17 seconds	9000/tcp	php-container
a9b620c580c9	mysql:5.7	"docker-entrypoint.s..."	4 days ago	Up 17 seconds	3306/tcp, 33060/tcp	mysql-container

Creación de un contenedor MySQL

Ahora en el Dockerfile del directorio php ponemos lo siguiente:

```
DEAW > UD6 > practica6-2 > php > Dockerfile
1 FROM php:7.0-fpm
2 RUN docker-php-ext-install pdo_mysql
```

y por ultimo en el docker-compose añadimos dos contenedores uno para MySQL y otro para los datos de MySQL

```
mysql:
  image: mysql:5.7
  container_name: mysql-container
  volumes_from:
    - mysql-data
  environment:
    MYSQL_ROOT_PASSWORD: secret
    MYSQL_DATABASE: mydb
    MYSQL_USER: myuser
    MYSQL_PASSWORD: password

mysql-data:
  image: mysql:5.7
  container_name: mysql-data-container
  volumes:
    - /var/lib/mysql
  command: "true"
```

Modificamos el index.html

```
<body>
  <h1>iHola mundo!</h1>
  <p><?php echo 'Estamos corriendo PHP, version: '
  <?
  $database = "mydb";
  $user = "root";
  $password = "secret";
  $host = "mysql";

  $connection = new PDO("mysql:host={$host};dbname={$database}");
  $query = $connection->query("SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = '{$database}'");
  $tables = $query->fetchAll(PDO::FETCH_COLUMN);

  if (empty($tables)) {
    echo "<p>No hay tablas en la base de datos \";
  } else {
    echo "<p>La base de datos \"{$database}\" tiene las siguientes tablas:
    echo "<ul>";
    foreach ($tables as $table) {
      echo "<li>{$table}</li>";
    }
    echo "</ul>";
  }
}
```

Y comprobamos

```
ALU2W@PC112:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ffadaf9e60d0	practica6-2_nginx	"/docker-entrypoint..."	28 seconds ago	Up 24 seconds	0.0.0.0:8081->80/tcp, :::8081->80/tcp	nginx-container
c4f1e267b9c7	practica6-2_php	"docker-php-entrypoi..."	7 minutes ago	Up 25 seconds	9000/tcp	php-container
3439f5204aea	php:7.4-apache	"docker-php-entrypoi..."	3 days ago	Exited (0) 3 days ago		porfolioapp-web-1
a9b620c580c9	mysql:5.7	"docker-entrypoint.s..."	4 days ago	Up 26 seconds	3306/tcp, 33060/tcp	mysql-container
4762cae5b673	mysql:5.7	"docker-entrypoint.s..."	4 days ago	Exited (0) 27 seconds ago		mysql-data-container
23acd93e2ded	php:7.0-fpm	"docker-php-entrypoi..."	4 days ago	Exited (0) 27 seconds ago		app-data-container
6c87a4f34342	httpd:2.4	"httpd-foreground"	5 days ago	Exited (0) 5 days ago		confident_black
bd0f1022da74	php:7.4-apache	"docker-php-entrypoi..."	5 days ago	Exited (0) 5 days ago		porfolioapp04102023-web-1

Verificación de conexión a la base de datos

Ahora si accedemos nos saldrá lo siguiente:

← → ↻ ⓘ localhost:8081

¡Hola mundo!

Estamos corriendo PHP, version: 7.0.33

No hay tablas en la base de datos "mydb".

Pero en realidad si tenemos tablas solo que somos un usuario normal y cambiamos a root veremos las tablas

¡Hola mundo!

← → ↻ ⓘ localhost:8081

¡Hola mundo!

Estamos corriendo PHP, version: 7.0.33

La base de datos "mydb" tiene las siguientes tablas:

- columns_priv
- db
- engine_cost
- event
- func
- general_log
- gtid_executed
- help_category
- help_keyword
- help_relation
- help_topic
- innodb_index_stats
- innodb_table_stats
- ndb_binlog_index
- plugin
- proc
- procs_priv
- proxies_priv
- server_cost
- servers
- slave_master_info
- slave_relay_log_info
- slave_worker_info