

TRABAJO INTEGRADOR DOCKER

Presentación

Se subió el trabajo final a la Plataforma GitHub. El link para descargar el proyecto es <https://github.com/lalombrizsolitaria/ipap-docker>.

Se ofrece una solución (o receta) que permite establecer un entorno estándar preconfigurado para un sistema desarrollado en el framework de PHP Symfony.

El trabajo incluye un directorio “.docker” donde se incluye el “docker-compose.yml” y los archivos/directorios con lo necesario para levantar el entorno.

En la receta, y dadas las condiciones con las que habitualmente nos encontramos los que desarrollamos en PHP, se generarán cuatro contenedores (uno opcional):

- Un contenedor de PHP 8.1, con un volumen persistente donde se aloja el código fuente (directorio padre)
- Un contenedor NGinx alpine como alternativa al servidor apache, con el mismo volumen incluido.
- Un contenedor con una base de datos MySQL, donde persistirá la información la aplicación.
- Un contenedor opcional de PhpMyAdmin, para administrar y ver la base de datos a medida que se desarrolla.

Los contenedores comparten la misma red virtual o “network” para que puedan enlazarse entre ellos. La base de datos MySQL se alojará en un directorio “db”.

A continuación se adjuntarán las capturas del procedimiento para dejar andando el entorno, con los pasos adicionales para Symfony.

[illegible]

3 – VERIFICAR ENTORNO

Con “docker ps -a” puedo listar todos los contenedores. En el listado veo los cuatro contenedores creados, con su información. Se puede ver que los cuatro se están ejecutando, sus puertos (anfitrión:contenedor) , sus ids, imágenes y nombres.

```
laloMacBookPro ~ % docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
6fc95fc721e   ipap-sy-php    "/docker-entrypoint..." 43 seconds ago Up 41 seconds 9000/tcp          ipap-sy-php-1
1742ad6658d7   ipap-sy-nginx  "/docker-entrypoint..." 43 seconds ago Up 41 seconds 0.0.0.0:80->80/tcp, 443/tcp      ipap-sy-nginx-1
67f747e9d3c2   mysql:8.0.23   "/docker-entrypoint.s..." 43 seconds ago Up 41 seconds 0.0.0.0:3306->3306/tcp, 33060/tcp ipap-sy-db-1
e72abec9cfad   phpmyadmin     "/docker-entrypoint..." 44 seconds ago Up 41 seconds 0.0.0.0:8881->88/tcp          ipap-sy-phpmyadm
laloMacBookPro ~ %
```

Con “docker volumen ls” puedo ver los volúmenes configurados, y con “docker network ls” puedo ver las redes establecidas.

```
laloMacBookPro ~ % docker volume ls
DRIVER    VOLUME NAME
local     25081ff3b90bc002a70d9f0283cc0a7d6f7631e31388c85a3941b5088cb46d4
laloMacBookPro ~ % docker network ls
NETWORK ID   NAME          DRIVER    SCOPE
2bfc621f932b bridge        bridge    local
2574e3808770 host           host      local
9a9d54a1c52f ipap-sy_symfony bridge     local
54bcece8a28d none          null      local
laloMacBookPro ~ %
```

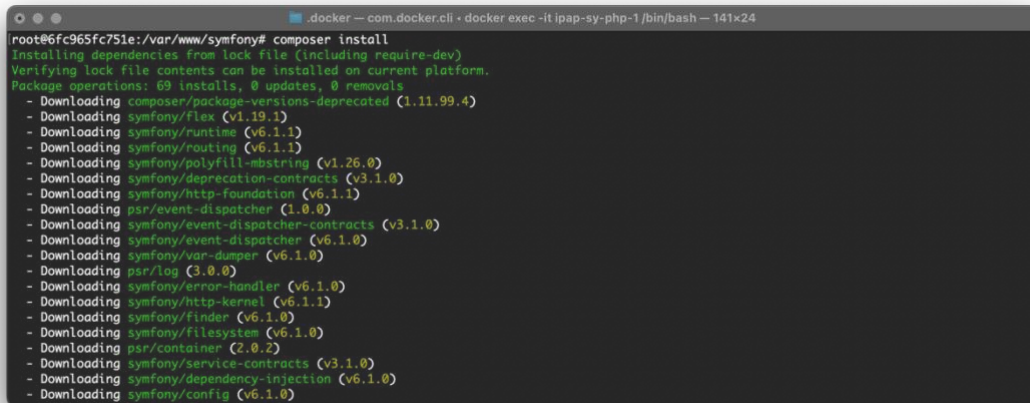
3 – INGRESO A CONTENEDOR

Puedo ejecutar “docker exec -it ipap-sy-php-1 /bin/bash” para ingresar a la consola del contenedor de “php”. El parámetro “-it” establece que será interactivo, lo que nos permitirá quedarnos en esa consola, e “ipap-sy-php-1” es el nombre del contenedor donde queremos ejecutar el comando “/bin/bash”.

```
laloMacBookPro ~ % docker exec -it ipap-sy-php-1 /bin/bash
root@6fc95fc731e:/var/www/symfony#
```

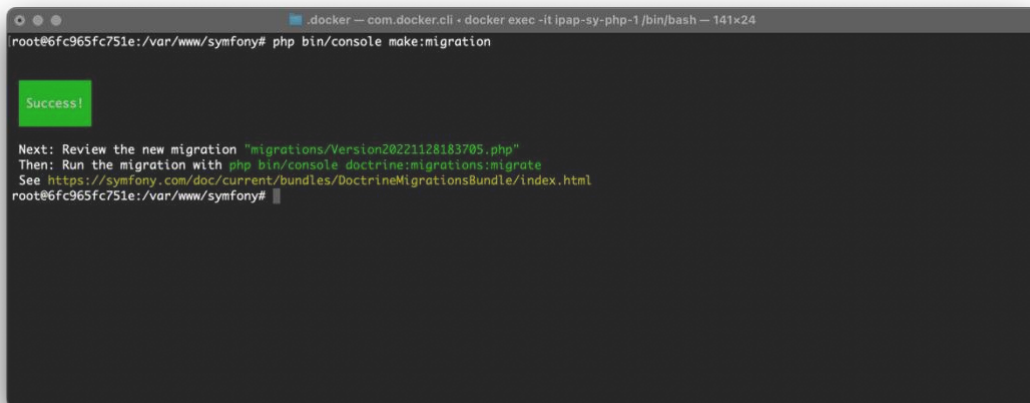
4 – CONFIGURACION ADICIONAL PARA SYMFONY

Ejecutamos “composer install” para instalar las dependencias y librerías necesarias para el proyecto Symfony que estamos utilizando. Sin este paso, “el código estará incompleto”.



```
root@6fc965fc751e:/var/www/symfony# composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Package operations: 69 installs, 0 updates, 0 removals
- Downloading composer/package-versions-deprecated (1.11.99.4)
- Downloading symfony/flex (v1.19.1)
- Downloading symfony/runtime (v6.1.1)
- Downloading symfony/routing (v6.1.1)
- Downloading symfony/polyfill-mbstring (v1.26.0)
- Downloading symfony/deprecation-contracts (v3.1.0)
- Downloading symfony/http-foundation (v6.1.1)
- Downloading psr/event-dispatcher (1.0.0)
- Downloading symfony/event-dispatcher-contracts (v3.1.0)
- Downloading symfony/event-dispatcher (v6.1.0)
- Downloading symfony/var-dumper (v6.1.0)
- Downloading psr/log (3.0.0)
- Downloading symfony/error-handler (v6.1.0)
- Downloading symfony/http-kernel (v6.1.1)
- Downloading symfony/finder (v6.1.0)
- Downloading symfony/filesystem (v6.1.0)
- Downloading psr/container (2.0.2)
- Downloading symfony/service-contracts (v3.1.0)
- Downloading symfony/dependency-injection (v6.1.0)
- Downloading symfony/config (v6.1.0)
```

Luego ejecutamos “php bin/console make:migration” y “php bin/console doctrine:migrations:migrate” para actualizar la base de datos con el modelo de datos establecido en Symfony. De esta forma, se crea la estructura de datos que define el sistema.



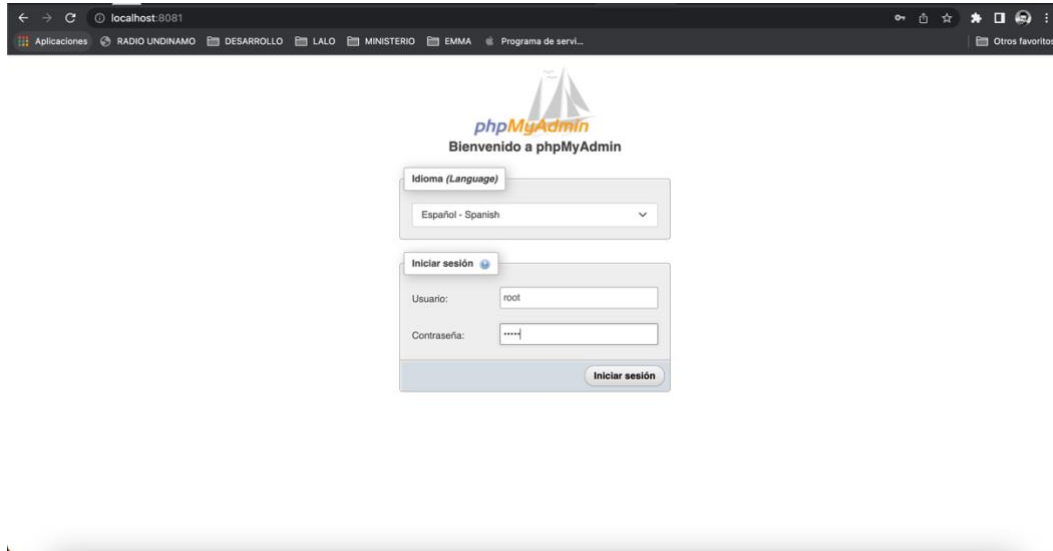
```
root@6fc965fc751e:/var/www/symfony# php bin/console make:migration

Success!

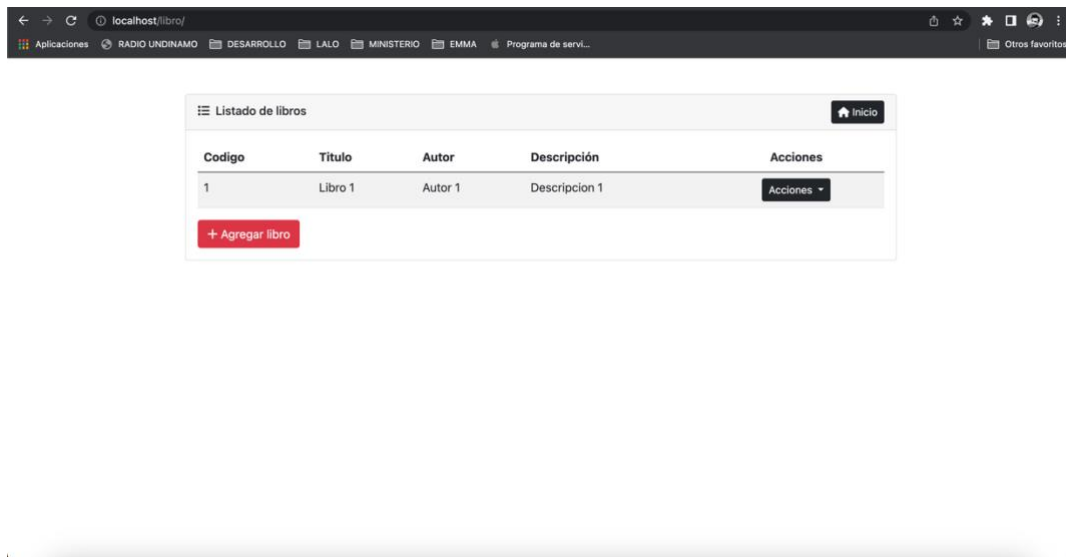
Next: Review the new migration "migrations/Version20221128183705.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html
root@6fc965fc751e:/var/www/symfony#
```

5 – PRUEBA FINAL

Si accedo por navegador a “localhost:8081” puedo acceder a PhpMyAdmin.



Si accedo por navegador a localhost (localhost:80) accedo a la página principal del sistema. Si accedo a localhost/libro se mostrará la siguiente ventana.



Aclaración final:

En la definición del servidor nginx, se establece un parámetro NGINX_BACKEND_DOMAIN, donde se establece un virtual host al que se puede acceder si lo definimos en el archivos de hosts del anfitrión.

En este caso, al agregarlo (127.0.0.1 ipap.test) al archivo de hosts del sistema operativo del anfitrión, podemos acceder por “localhost” o por “ipap.test”.

```
nginx:
  build:
    # INDICAO DONDE BUSCO EL DOCKERFILE
    context: ./nginx
    # VOLUMEN PERSISTENTE DONDE SE ENCUENTRA EL CODIGO FUENTE (Symfony) : WWW EN CONTENEDOR
    volumes:
      - ../:/var/www/symfony/
    # MAPEO DE PUERTOS INTERNOS 80 DEL CONTENEDOR AL 80 DEL ANFITRION (ACCEDO POR localhost)
    ports:
      - 80:80
    # ENLAZO CONTENEDORES EN RED VIRTUAL "symfony"
    networks:
      - symfony
    # SE UTILIZAR ESE ARCHIVO PARA DEFINIFIR EL VIRTUAL HOST PARA LA APP
    environment:
      NGINX_BACKEND_DOMAIN: ipap.test
```