

# Infoargo

## Manual de Instalación

17/02/2023

Óscar, Adrián, Gil Pablo, Rubén

<b>1. Docker</b>	<b>2</b>
<b>2. MySQL</b>	<b>3</b>
2.1 Descarga de la imagen	4
2.2 Creación de volúmenes	4
2.3 Despliegue de contenedor	4
<b>3. PHPMysqlAdmin</b>	<b>5</b>
3.1 Descarga de la imagen	6
3.2 Creación de volúmenes	6
3.3 Despliegue de contenedor	6
<b>4. Nginx</b>	<b>6</b>
4.1 Descarga de la imagen	7
4.2 Creación de volúmenes	7
4.3 Despliegue de contenedor	7

# 1. Docker

Para la implementación de la solución usaremos la herramienta Docker.

Docker nos permite crear contenedores, los cuales contienen la imagen de cualquier servicio que se quiera implementar. También nos permite añadir volúmenes, donde se guarda la información importante (datos de cada imagen o configuración de cada una).

En nuestro caso hemos creado tres contenedores:

- MySQL
- PHPMyAdmin
- Nginx + PHP + FPM

Y cuatro volúmenes:

- mysql-conf
- mysql-data
- nginx-conf
- nginx-data

## 2. MySQL

Como herramienta de Base de Datos se ha usado MySQL, puesto que es con la que más experiencia tenemos y se adapta a nuestras necesidades.

### 2.1 Descarga de la imagen

Página de Docker Hub dónde se ha extraído la imagen:

[https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql)

Comando para “pullar” o descargar la imagen:

```
docker pull mysql
```

### 2.2 Creación de volúmenes

Necesitaremos dos volúmenes orientados a MySQL:

- **“mysql-data”**: lo orientaremos a los “datos”, es decir, el directorio donde alojaremos los datos de la aplicación web: **/var/lib**
- **“mysql-conf”**: lo orientaremos a la configuración de MySQL, para modificar el comportamiento del SGBD como se desea: **/etc/mysql**

Por ahora, lo único que se necesita es desplegar estos dos volúmenes con los siguientes comandos:

```
docker volume create mysql-data
```

```
docker volume create mysql-conf
```

## 2.3 Despliegue de contenedor

Para desplegarlo, usaremos el siguiente comando:

```
docker run --name mysql -p 3306:3306 -v mysql-data:/var/lib -v mysql-conf:/etc/mysql -e MYSQL_ROOT_PASSWORD=alumno -d mysql
```

A continuación se explica cada parte del comando:

- **--name mysql**: nombre con el que se distingue el contenedor.
- **-p 3306:3306**: puerto que usará el contenedor.
- **-v mysql-data:/var/lib** y **-v mysql-conf:/etc/mysql**: volúmenes que usará el contenedor, los detalles están explicados anteriormente.
- **-e MYSQL\_ROOT\_PASSWORD=alumno**: contraseña del usuario root.
- **-d**: ejecutará el comando en segundo plano, por lo que nos dejará la terminal libre.

### 3. PHPMyAdmin

Como herramienta de gestión para la Base de Datos, se ha usado PHPMyAdmin, puesto que es con la que más experiencia tenemos y se adapta a nuestras necesidades.

#### 3.1 Descarga de la imagen

Página de Docker Hub dónde se ha extraído la imagen:

[https://hub.docker.com/\\_/phpmyadmin](https://hub.docker.com/_/phpmyadmin)

Para “pullear” o descargar la imagen:

```
docker pull phpmyadmin
```

#### 3.2 Creación de volúmenes

Para PHPMyAdmin no se necesitan volúmenes

#### 3.3 Despliegue de contenedor

Para desplegarlo, usaremos el siguiente comando:

```
docker run --name myadmin --link mysql:db -p 8080:80 -d phpmyadmin
```

A continuación se explica cada parte del comando:

- **“--name myadmin”**: nombre con el que se distingue el contenedor.
- **“--link mysql:db”**: permite que, PHPMyAdmin, se conecte a la base de datos con ese nombre.
- **“-p 8080:80”**: indica el puerto que usará el contenedor.
- **“-d”**: ejecutará el comando en segundo plano, por lo que nos dejará la terminal libre.

## 4. Nginx

Como servidor, se ha usado Nginx, puesto que para la solución, se requería. La imagen escogida es una combinación de Nginx + PHP + FPM.

### 4.1 Descarga de la imagen

Página de Docker Hub dónde se ha extraído la imagen:

<https://hub.docker.com/r/richarvey/nginx-php-fpm>

Para “pullear” o descargar la imagen:

```
docker pull richarvey/nginx-php-fpm
```

### 4.2 Creación de volúmenes

Necesitaremos dos volúmenes orientados a Nginx:

Uno lo orientaremos a los “datos”, es decir, el directorio donde alojaremos la aplicación web: **/var/www/html**

Y el segundo, lo orientaremos a la configuración de Nginx, para modificar el comportamiento del servidor como se desea: **/etc/nginx/sites-enabled**

Por ahora, lo único que se necesita es desplegar estos dos volúmenes con los siguientes comandos:

```
docker volume create nginx-data
```

```
docker volume create nginx-conf
```

## 4.3 Despliegue de contenedor

Para desplegarlo, usaremos el siguiente comando:

```
docker run --name nginx -p 80:80 -p 443:443 -v nginx-data:/var/www/html -v
nginx-conf:/etc/nginx/sites-enabled -d richarvey/nginx-php-fpm
```

A continuación se explica cada parte del comando:

- **--name nginx**: nombre con el que se distingue el contenedor.
- **-p 80:80** y **-p 443:443**: puerto que usará el contenedor.
- **-v nginx-data:/var/www/html** y **-v nginx-conf:/etc/nginx/sites-enabled**: volúmenes que usará el contenedor, los detalles están explicados anteriormente.
- **-d**: ejecutará el comando en segundo plano, por lo que nos dejará la terminal libre.

### 4.3.1 Archivo de configuración

En el archivo de configuración del volumen **nginx-conf**, se tendrá que hacer un archivo con el siguiente comando:

```
touch infoargo.conf
```

Para modificarlo:

```
nano infoargo.conf
```

Dentro del archivo, habrá que pegar lo siguiente:

```
server {
    listen    443; ## listen for ipv4; this line is default and implied
    ssl on;
    ## Esta ruta debe equivaler a la estructura de nginx-data
    root /var/www/html/Sarabastall1;
    index index.php index.html index.htm;

    # Make site accessible from http://localhost/
    server_name _;

    # Disable sendfile as per
https://docs.vagrantup.com/v2/synced-folders/virtualbox.html
    sendfile off;
```



```

# Add stdout logging
error_log /dev/stdout info;
access_log /dev/stdout;

ssl_certificate      /etc/nginx/conf.d/cert.crt;
ssl_certificate_key  /etc/nginx/conf.d/key.key;
ssl_protocols        TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers           HIGH:!aNULL:!MD5;

# block access to sensitive information about git
location /.git {
    deny all;
    return 403;
}

# .htaccess /App/
autoindex off;

# .htaccess ./
location / {
    rewrite ^/$ /public/ break;
    rewrite ^(.*)$ /public/$1 break;
    try_files $uri $uri/ /public/index.php?url=$1;
}

# .htaccess /public/
location /Sarabastall1/public/ {
    if (!-e $request_filename){
        rewrite ^/Sarabastall1/public/(.+) $
/Sarabastall1/public/index.php?url=$1 break;
    }
}

# pass the PHP scripts to FastCGI server listening on socket
#
location ~ /\.php$ {
    try_files $uri =404;
    fastcgi_split_path_info ^(.+\.(php))(/.+) $;
    fastcgi_pass unix:/var/run/php-fpm.sock;
    fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    fastcgi_index index.php;

```

```
        include fastcgi_params;
    }

location img/ {
    root /public;
}

# deny access to . files, for security
#
location ~ /\. {
    log_not_found off;
    deny all;
}

location ^~ /.well-known {
    allow all;
    auth_basic off;
}
}
```

Cada línea relevante tiene su explicación comentada encima.

## 5. Importar Aplicación Web

### 5.1 Aplicación Web

Para importar el apartado de la aplicación se usará el comando SCP:

```
scp -r "ruta de tu equipo" "equipo al que se le importa"@ip a la que se le importa":"ruta a la que se importa (ejemplo: "/home/daw/")"
```

Por lo general esta sería la ruta a la que se importa en el caso de haber creado los volúmenes anteriores:

```
/var/lib/docker/volumes/nginx-data/_data/
```

### 5.2 Base de Datos

Para importar el apartado de la Base de Datos, una vez dentro de PHPMyAdmin, en el menú superior colocaremos en el botón de "Importar":



#### Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.

Un archivo comprimido tiene que terminar en **.[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador:  No se ha seleccionado ningún archivo. (Máximo: 2,048KB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

Examinar > "El archivo de la base de datos."

Una vez seleccionado el archivo, le daremos a continuar.

## 6. Exportar Aplicación Web

### 6.1 Aplicación Web

Para exportar el apartado de la aplicación se usará el comando SCP:

```
scp -r "ruta del equipo" "usuario al que se le exporta"@ip al que se le exporta:"ruta al que se le exporta (ejemplo: "/home/daw/")"
```

Por lo general esta es la ruta de donde exportar si se han seguido los anteriores pasos:

```
/var/lib/docker/volumes/nginx-data/_data/
```

### 6.2 Base de Datos

Para exportar el apartado de la Base de Datos, una vez dentro de PHPMyAdmin, en el menú superior está situado el botón de "Exportar":



## Exportando tablas de la base de datos "infoargo"

### Exportar plantillas:

Nueva plantilla:

Plantillas existentes:

Plantilla:

### Método de exportación:

- ☒ Rápido - mostrar sólo el mínimo de opciones de configuración
- ☐ Personalizado - mostrar todas las opciones de configuración posibles

### Formato:

Como método de exportación, en nuestro caso, se puede dejar por predeterminado en "Rápido" y no supondría problema, pero si se quiere cambiar algún ajuste, en "Personalizado" se podría cambiar.

## 7. Certificado SSL

### 7.1 Comando de generación de certificado

Con el siguiente comando se genera un nuevo certificado:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/key.key -out /etc/ssl/certs/cert.crt
```

Para incluir este certificado en el docker de Nginx se usa el siguiente comando:

```
docker cp /etc/ssl/certs/cert.crt nginx:/etc/nginx/conf.d/cert.crt docker cp /etc/ssl/private/key.key nginx:/etc/nginx/conf.d/key.key
```

### 7.2 Cambios en archivo de configuración

Para que se pueda acceder al servidor de Nginx por el puerto 443 (https), hay que añadir las siguientes líneas en el archivo de configuración:

```
listen 443;  
ssl on;  
ssl_certificate /etc/nginx/conf.d/cert.crt;  
ssl_certificate_key /etc/nginx/conf.d/key.key;  
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
ssl_ciphers HIGH:!aNULL:!MD5;
```

Lo recomendable, teniendo en cuenta que ya tenemos el servidor abierto al puerto 443, sería comentar o eliminar la línea:

```
listen 80;
```