

Comandos Importantes

```
docker run -d -p 8080:80 --name app1 -v /home/ciclost/Documentos/app1:/var/www/html php:8.2-apache
```

- El comando en cuestión es un comando de Docker que se utiliza para ejecutar un contenedor. Aquí está una explicación paso a paso:
 1. **"docker run"** es el comando principal que se utiliza para ejecutar un contenedor.
 2. **"-d"** indica que el contenedor debe ejecutarse en segundo plano (en modo desatendido).
 3. **"-p 8080:80"** indica que el puerto 8080 de tu máquina local se debe vincular al puerto 80 del contenedor. Esto significa que puedes acceder a la aplicación dentro del contenedor a través del puerto 8080 en tu máquina local.
 4. **"--name app1"** le da un nombre al contenedor. En este caso, el nombre del contenedor será "app1".
 5. **"-v /home/ciclost/Documentos/app1:/var/www/html"** indica que quieres montar un volumen. Un volumen en Docker es una forma de compartir archivos y directorios entre tu máquina local y el contenedor. En este caso, estás montando el directorio `/home/ciclost/Documentos/app1` de tu máquina local en el directorio `/var/www/html` dentro del contenedor.
 6. **"php:8.2-apache"** es la imagen de Docker que se utilizará para crear el contenedor. En este caso, estás utilizando la imagen "php:8.2-apache", que es una imagen que contiene PHP 8.2 ejecutándose en un servidor Apache.
- En resumen, este comando de Docker ejecutará un contenedor utilizando la imagen "php:8.2-apache", montando un volumen para compartir archivos entre tu máquina local y el contenedor, y vinculando el puerto 8080 de tu máquina local al puerto 80 del contenedor. Esto permitirá que accedas a la aplicación PHP dentro del contenedor a través del puerto 8080 en tu máquina local.
 - localhost:8080

Lo primero que se hace al crear un proyecto (phpstorm) es crear el archivo docker-compose.yaml.

- **docker-compose.yaml**

- El archivo *docker-compose.yaml* es un archivo de configuración utilizado por Docker Compose para definir y administrar múltiples contenedores en una aplicación.

- Dentro del archivo:

```
# Services
services:
  # Apache + PHP
  apache_php:
    image: php:8.2-apache
  # Preparamos un volumen para almacenar nuestro código
  volumes:
    - ./src/:/var/www/html
  expose:
    - 80
  ports:
    - 8080:80
```

```
# Services
services:

  nginx:
    image: nginx:1.25.2
    ports:
      - 8081:80
    volumes:
      - ./src:/var/www/php
      - ../config/docker/nginx/conf.d:/etc/nginx/conf.d # cargamos la
configuración de un fichero externo
    depends_on:
      - php # enlazamos nginx con php

  php:
    image: php:8.2-fpm # puerto por defecto: 9000
    working_dir: /var/www/php
    volumes:
      - ./src:/var/www/php
```

- Crear archivos ".config, docker, nginx, conf.d" si no se puede modificar el conf.d que crea docker.
 - Dentro de nuestro conf.d crear el archivo php.conf.
- Contenido del php.conf:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root    /var/www/php;
    index   index.php;

    location ~* \.php$ {
        fastcgi_pass    php:9000;
        include          fastcgi_params;
        fastcgi_param    SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param    SCRIPT_NAME     $fastcgi_script_name;
    }
}
```

Práctica app3: contenedor de docker con apache + nginx

1. Crear proyecto app3
2. Crear archivo docker-compose.yaml
3. Dentro del archivo:

```
# Services
services:

    nginx:
        image: nginx:1.25.2
        ports:
            - 8080:80
        volumes:
            - ./src:/var/www/php
            - ../config/docker/nginx/conf.d:/etc/nginx/conf.d # cargamos la
configuración de un fichero externo
        depends_on:
            - php # enlazamos nginx con php

    php:
        image: php:8.2-fpm # puerto por defecto: 9000
        working_dir: /var/www/php
        volumes:
            - ./src:/var/www/php
```

4. Crear carpeta src
5. Crear index.php con un mensaje dentro (en src)
6. Crear directorio .config
7. Dentro, directorio docker

8. Dentro, directorio nginx
9. Dentro, directorio conf.d
10. Dentro, archivo php.conf
11. Dentro del archivo:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root    /var/www/php;  
    index  index.php;  
  
    location ~* \.php$ {  
        fastcgi_pass    php:9000;  
        include          fastcgi_params;  
        fastcgi_param    SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        fastcgi_param    SCRIPT_NAME     $fastcgi_script_name;  
    }  
}
```

12. Ejecutar el docker-compose.yaml

Práctica app3.2: contenedor de docker con apache

1. Crear proyecto app3.2
2. Crear archivo docker-compose.yaml
3. Dentro del archivo:

```
# Services  
services:  
  # Apache + PHP  
  apache_php:  
    image: php:8.2-apache  
  # Preparamos un volumen para almacenar nuestro código  
  volumes:  
    - ./src:/var/www/html  
  expose:  
    - 80  
  ports:  
    - 8080:80
```

4. Crear carpeta src
5. Crear index.php con un mensaje dentro (en src)
6. Ejecutar el docker-compose.yaml