

## ÍNDICE:

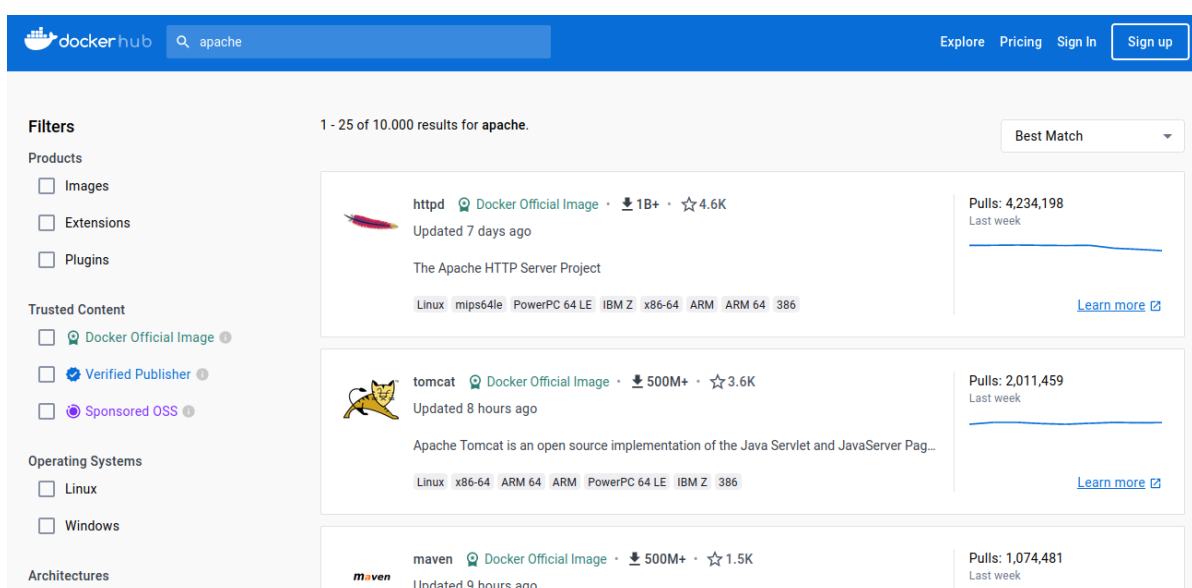
- [Consideraciones previas](#)
- [Lo que yo hice](#)
- [Instalación del contenedor](#)

## CONSIDERACIONES PREVIAS:

Para la creación del docker tenemos 2 opciones:

- Usar un contenedor virgen: una imagen de debian o ubuntu vacía.
  - En este método instalamos la imagen elegida e instalamos todos los servicios necesarios. Estos incluyen un editor de texto y cualquier otro servicio o paquete que no sea una dependencia directa del servicio a instalar.
  - # Docker funciona por capas, y cada cambio que hagamos añade una capa a la imagen. Así, instalar paquetes añade capas al contenedor.
- Usar un contenedor con una imagen que ya contenga parte o la totalidad de los servicios que necesitamos.
  - Tenemos una gran variedad de imágenes que incluyan php y apache. Las podemos buscar en tanto con docker search como en dockerhub. El primero es más cómodo el segundo más detallado.

```
[kenneth@arch-kenneth ~]$ docker search php
NAME                DESCRIPTION                               STARS   OFFICIAL   AUTOMATED
php                  While designed for web development, the PHP ... 7276   [OK]
adminer              Database management in a single PHP file.      874    [OK]
composer             Composer is a dependency manager written in ... 1013   [OK]
phpmyadmin           phpMyAdmin - A web interface for MySQL and M... 876    [OK]
mediawiki             MediaWiki is a free software open source wik... 515    [OK]
yourls                YOURLS is a set of PHP scripts that will all... 259    [OK]
php-zendserver        Zend Server - the integrated PHP application... 212    [OK]
newrelic/php-daemon  New Relic's PHP daemon gathers APM data from... 6
bitnami/phpmyadmin   Bitnami Docker Image for phpMyAdmin           42     [OK]
circleci/php         CircleCI images for PHP                       34
bitnami/php-fpm      Bitnami PHP-FPM Docker Image                  177    [OK]
bitnami/phpbb        Bitnami Docker Image for phpBB                29     [OK]
phplist/phplist      phplist, Open Source newsletter manager. htt... 16
cimg/php              cimg/php                                       2
phpcollab/phpcollab  phpcollab is an open source internet-enabled... 7
phplist/postfix      Postfix Container for use with phplist in te... 1
bitnami/phpPgAdmin-archived  A copy of the container images of the deprec... 0
webdevops/php-apache-dev  PHP with Apache for Development (eg. with xd... 156    [OK]
webdevops/php-apache  Apache with PHP-FPM (based on webdevops/php)  138    [OK]
webdevops/php-nginx    Nginx with PHP-FPM                            264    [OK]
bitnamicharts/phpmyadmin  bitnamicharts/phpmyadmin                     0
dockage/phpPgAdmin     phpPgAdmin Docker image, phpPgAdmin is a web... 21     [OK]
phpsu/phpsu           phpsu/phpsu                                   0
okteto/php            okteto/php                                     0
mirantis/phpldapadmin  mirantis/phpldapadmin                         4
```



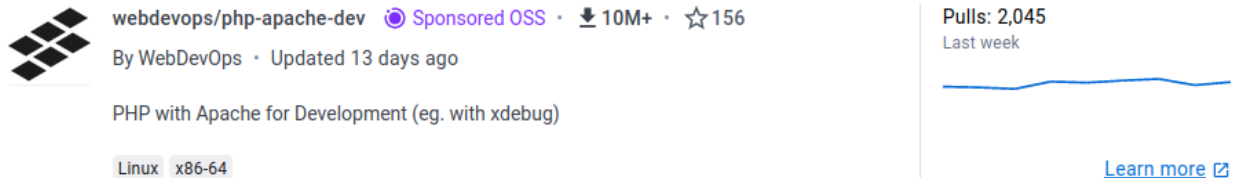
- Estas imágenes tienen ya una configuración predefinida, con la que podemos pelearnos o no. En cualquier caso, es recomendable que mirar la

entrada de dockerhub de la imagen que usemos para saber si tiene cambios de estructura y cuáles son.

- Muchas imágenes tienen variables de entorno definidas y una gran variedad de otros parámetros que no sé usar ni me hacen falta para lo que voy a usarlas de momento.

## LO QUE YO HICE

En mi caso personal usé esta imagen:



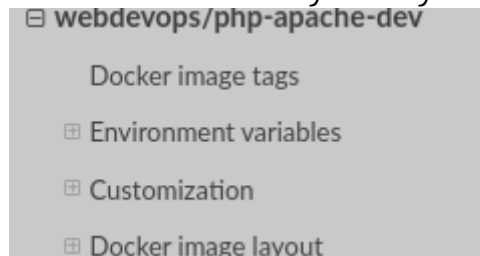
The screenshot shows the Docker Hub page for the image 'webdevops/php-apache-dev'. It includes the following information:

- Repository name: webdevops/php-apache-dev
- Labels: Sponsored OSS, 10M+ downloads, 156 stars
- Author: By WebDevOps
- Updated: 13 days ago
- Description: PHP with Apache for Development (eg. with xdebug)
- Architecture: Linux x86-64
- Activity: Pulls: 2,045 Last week
- Link: [Learn more](#)

Dado que implementa su propia estructura personalizada, todos los directorios activos están cambiados. Los directorios por defecto siguen ahí, pero no están en uso, por lo que `/var/www/html` existe pero para que lo use tenemos que tocar archivos de configuración, los cuales también están altamente editados.

Para saber qué archivos y directorios hacen qué cosa, accedemos a su página de dockerhub y buscamos su documentación.

Nos interesan las secciones de Customización y de Layout.



Sin embargo, mis conocimientos en la materia son demasiado bajos todavía como para poder aplicar los cambios que necesitaba aplicar, así que decidí usar la estructura que ya venía y olvidarme de problemas.

#Observación: Sería interesante ver cómo modificar estos pathings ya que tiene que ver más con la administración de sistemas que la programación en php, que si bien a mí me resulta lo más atractivo, no es donde vamos a llevar a cabo nuestro trabajo principalmente.

## INSTALACIÓN DEL CONTENEDOR

Se puede construir un dockerfile\*, o bien hacer esto:

```
docker run -d --name php-clase -p 60080:80 -v /home/kenneth/php:/app webdevops/php-apache-dev
```

¿Qué hace eso?

-p redirige puertos. En este caso para evitar conflictos con otros usos del puerto 80 uso el 60080 de mi máquina física. En principio tengo el puerto 80 libre, pero por si acaso no está de más tomar precauciones previas.

-v crea un volumen. Es un concepto que si bien es sencillo de entender, es complicado de explicar, pero resumiendo: monta lo que está en /home/kenneth/php los contenidos de /app (dentro del contenedor), que es donde se encuentran los archivos php que usa el servicio en esta configuración. Así puedo acceder a los contenidos de esa carpeta del contenedor desde mi máquina, y si apago o borro el contenedor todos esos contenidos persisten en el host.

Los volúmenes son muy interesantes y hay otras formas de crearlos y gestionarlos, pero para este caso así vamos bien.

NOTA IMPORTANTE: hay que crear la carpeta de la ruta /home/kenneth/php y darnos permisos con recursividad sobre ella.

Si queremos que el contenedor se inicie al iniciar el ordenador por algún motivo introducimos:

```
docker container update --restart unless-stopped <id_contenedor | nombre_contenedor>
```

\*Con dockerfile solo podremos crear un volumen sin estar montado en nuestra máquina física, por lo que tendremos que montar el volumen manualmente en nuestra máquina, así que por simplicidad es mejor tirar de docker run o crear un script.