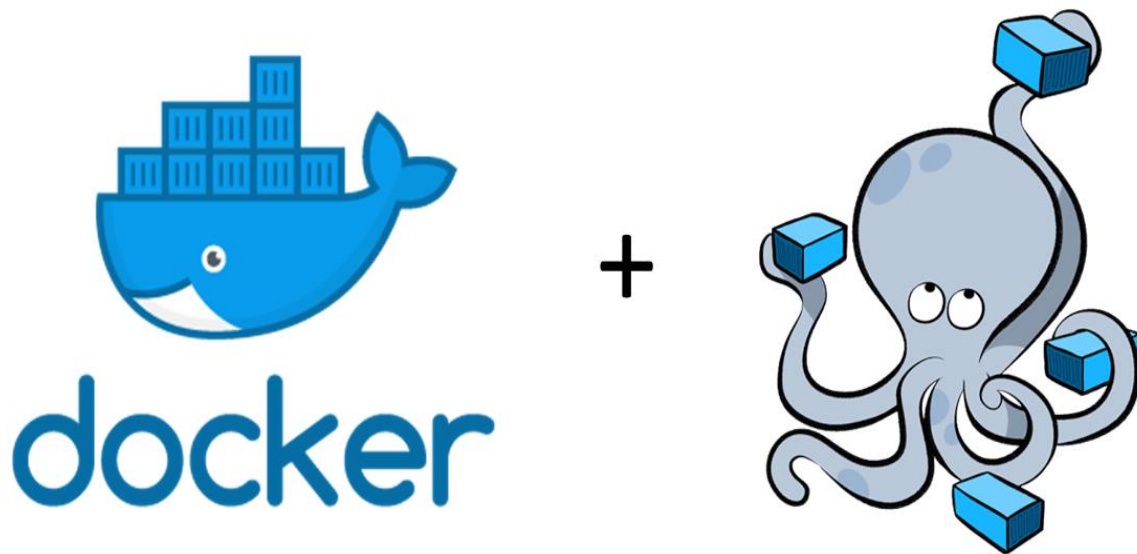


Docker docker- compose: Apache2, PHP7 y PostgreSQL



ÍNDICE:

Instalación de Docker y Docker-Compose: [página 3](#)

Crear directorios y ficheros: [página 6](#)

- despliegue (nombre de la carpeta)
- Dockerfile
- docker-compose.yml
- 000-default.conf
- web
- index.html
- index.php

Configuración de directorios y ficheros: [página 8](#)

Visualización index.html e index.php: [página 14](#)

Instalación de Docker y Docker-Compose

Poner la máquina virtual en red NAT.

Después utilizar estos comandos para actualizar.

```
root@alumno-virtualbox:/despliegue# apt update
```

```
root@alumno-virtualbox:/despliegue# apt upgrade
```

Instalar Docker.

```
root@alumno-virtualbox:/despliegue# apt install docker.io
```

Ver estado de Docker.

```
root@alumno-virtualbox:/despliegue# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-02-06 22:40:42 CET; 54min ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 924 (dockerd)
       Tasks: 13
      Memory: 111.0M
      CGroup: /system.slice/docker.service
              └─924 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 06 23:10:58 alumno-virtualbox dockerd[924]: time="2023-02-06T23:10:58.182292746+01:00" level=info msg="ignoring"
Feb 06 23:11:08 alumno-virtualbox dockerd[924]: time="2023-02-06T23:11:08.154740034+01:00" level=info msg="Containerd"
Feb 06 23:11:08 alumno-virtualbox dockerd[924]: time="2023-02-06T23:11:08.178736546+01:00" level=info msg="ignoring"
Feb 06 23:11:13 alumno-virtualbox dockerd[924]: time="2023-02-06T23:11:13.559695587+01:00" level=info msg="No"
Feb 06 23:11:13 alumno-virtualbox dockerd[924]: time="2023-02-06T23:11:13.561131837+01:00" level=info msg="IPV"
```

Localizar versión de Docker.

```

root@alumno-virtualbox:/despliegue# docker version
Client:
Version:           20.10.12
API version:       1.41
Go version:        go1.16.2
Git commit:        20.10.12-0ubuntu2~20.04.1
Built:             Wed Apr  6 02:14:38 2022
OS/Arch:           linux/amd64
Context:           default
Experimental:      true

Server:
Engine:
Version:           20.10.12
API version:       1.41 (minimum version 1.12)
Go version:        go1.16.2
Git commit:        20.10.12-0ubuntu2~20.04.1
Built:             Thu Feb 10 15:03:35 2022

```

Mirar información de Docker.

```

root@alumno-virtualbox:/despliegue# docker info
Client:
Context:    default
Debug Mode: false

Server:
Containers: 0
Running: 0
Paused: 0
Stopped: 0
Images: 12
Server Version: 20.10.12
Storage Driver: overlay2
Backing Filesystem: extfs

```

Mirar información de Docker con Debian.

```

root@alumno-virtualbox:/despliegue# docker search debian
NAME                DESCRIPTION                STARS     OFFICIAL    AUTOMATED
ubuntu              Ubuntu is a Debian-based Linux operating sys... 15549     [OK]
debian              Debian is a Linux distribution that's compos... 4564      [OK]
bitnami/minideb     A minimalist Debian-based image built specif... 126
neurodebian          NeuroDebian provides neuroscience research s... 98        [OK]
treehouses/debian   2
bitnami/debian-base-buildpack  Debian base compilation image 2          [OK]
ustclug/debian      Official Debian Image with USTC Mirror 2
rancher/debianconsole 1
osrf/debian_arm64    Debian arm64 Base Images 1
eclipse/debian_jdk8_node  Node (+ Angular stuff) + JDK8. Debian based 1          [OK]
eclipse/debian_jdk8     Debian, JDK8, Maven 3, git, curl, nmap, mc, ... 1          [OK]
dokken/debian-11     Debian 11 image for use with kitchen-dokken 1

```

Localizar las imágenes que hayamos descargado para Docker.

```
root@alumno-virtualbox:/despliegue# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
despliegue_proyecto  latest         85cb4bb37109   6 hours ago    430MB
<none>              <none>         aab3aa0c5824   7 hours ago    410MB
<none>              <none>         a844df436c4b   7 hours ago    410MB
postgres            latest         f462f91720c0   2 days ago     379MB
mysql               latest         05b458cc32b9   5 days ago     517MB
php                 7.2-apache     c61d277263e1   2 years ago    410MB
root@alumno-virtualbox:/despliegue#
```

Instalar docker-compose para la práctica.

```
root@prueba-1:/home# sudo apt install docker-compose
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

Crear directorios y ficheros

Empezamos a crear nuestro directorio de trabajo.

```
root@alumno-virtualbox:/# mkdir despliegue
```

Una vez dentro creamos fichero 000-default.conf

```
root@alumno-virtualbox:/# cd /despliegue/  
root@alumno-virtualbox:/despliegue# touch 000-default.conf
```

Creamos fichero Dockerfile

```
root@alumno-virtualbox:/# cd /despliegue/  
root@alumno-virtualbox:/despliegue# touch Dockerfile
```

Creamos fichero docker.compose.yml

```
root@alumno-virtualbox:/# cd /despliegue/  
root@alumno-virtualbox:/despliegue# touch docker-compose.yml
```

Creamos directorio web para ubicar nuestro index.php, index.html o ambos.

```
root@alumno-virtualbox:/despliegue# mkdir web
```

```
root@alumno-virtualbox:/despliegue# cd web/  
root@alumno-virtualbox:/despliegue/web# touch index.php
```

```
root@alumno-virtualbox:/despliegue/web# ls  
index.html index.php
```

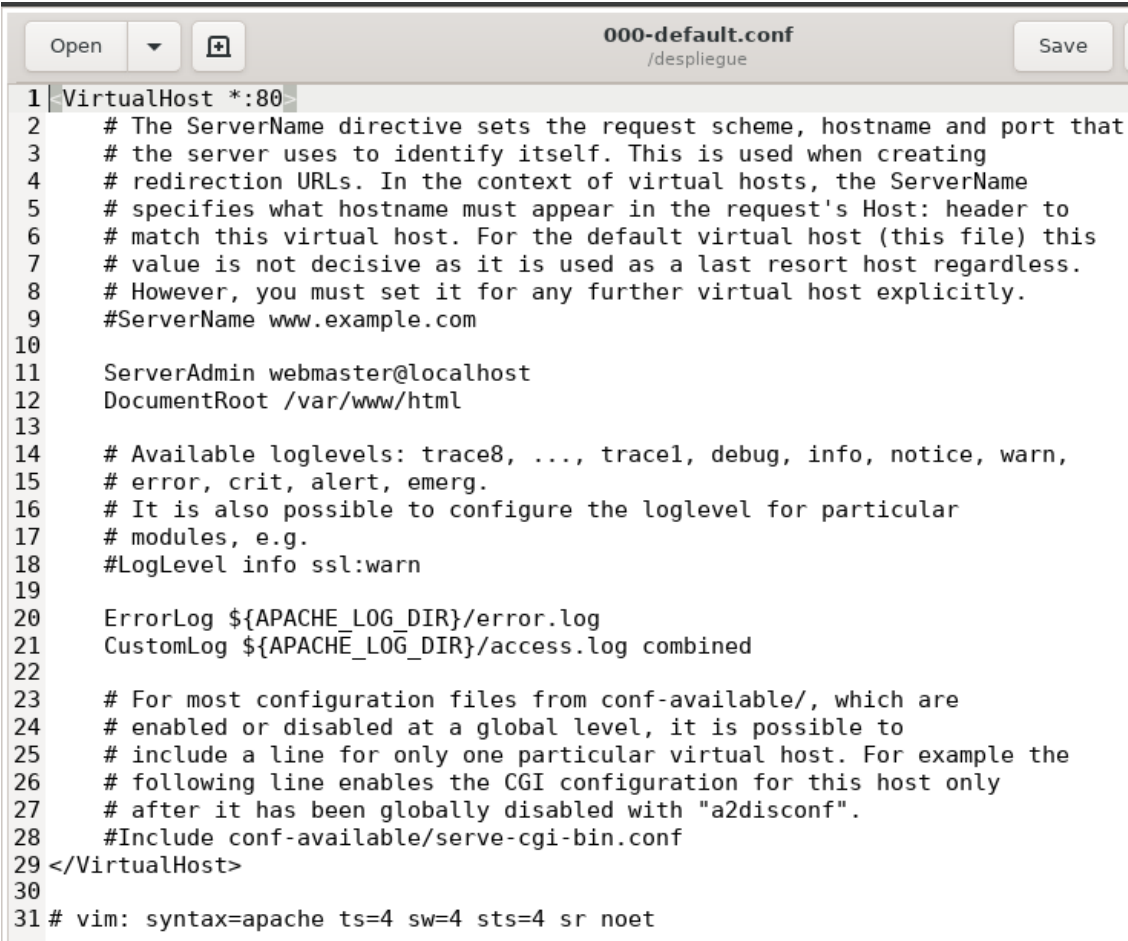
Debe de quedar la estructura así:

Dentro del directorio web está index.php, index.html o ambas.

```
root@alumno-virtualbox:/despliegue# ls -l  
total 16  
-rw-r--r-- 1 root root 1332 feb  6 16:42 000-default.conf  
-rw-r--r-- 1 root root  497 feb  6 23:09 docker-compose.yml  
-rw-r--r-- 1 root root  264 feb  6 23:57 Dockerfile  
drwxr-xr-x 2 root root 4096 feb  6 23:12 web  
root@alumno-virtualbox:/despliegue#
```

Configuración de directorios y ficheros

1. Para el fichero 000-default.conf nombrado anteriormente, lo copiamos su información de /etc/apache2/sites-available/ y quedaría así, con esto podrá funcionar la práctica.



```
1 VirtualHost *:80
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName www.example.com
10
11     ServerAdmin webmaster@localhost
12     DocumentRoot /var/www/html
13
14     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15     # error, crit, alert, emerg.
16     # It is also possible to configure the loglevel for particular
17     # modules, e.g.
18     #LogLevel info ssl:warn
19
20     ErrorLog ${APACHE_LOG_DIR}/error.log
21     CustomLog ${APACHE_LOG_DIR}/access.log combined
22
23     # For most configuration files from conf-available/, which are
24     # enabled or disabled at a global level, it is possible to
25     # include a line for only one particular virtual host. For example the
26     # following line enables the CGI configuration for this host only
27     # after it has been globally disabled with "a2disconf".
28     #Include conf-available/serve-cgi-bin.conf
29 </VirtualHost>
30
31 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```


2. En el fichero docker-compose.yml nombrado anteriormente empezamos a poner varias cosas:

Con este fichero lo que hacemos es realizar un despliegue de contenedores, dos en este caso:

-Apache con PHP.

-PostgreSQL.

El fichero sería así:

-En él se muestra la versión de docker-compose.

-Los servicios que ofrece (2).

-El primer contenedor que se ha llamado proyecto.

-La build es muy importante para poder ejecutar Dockerfile y la carpeta web.

-El puerto en el que escuchará 8016:80.

-Volumes donde se mapeará el index.php, index.html o ambos.

-Networks para ofrecer una ip personalizada y estén en la misma red **(172.20.254.153)**

-El segundo contenedor que se ha llamado db.

-La imagen que se usará que será postgres de PostgreSQL.

-Muy importante para el index.php -> Networks para ofrecer una ip personalizada y estén en la misma red **(172.20.254.152)**.

-Por defecto Postgres utiliza el puerto 5432, no se pone.

-Environment para ubicar POSTGRES_DB (nombre base de datos), POSTGRES_USER (nombre de usuario) y POSTGRES_PASSWORD (contraseña).

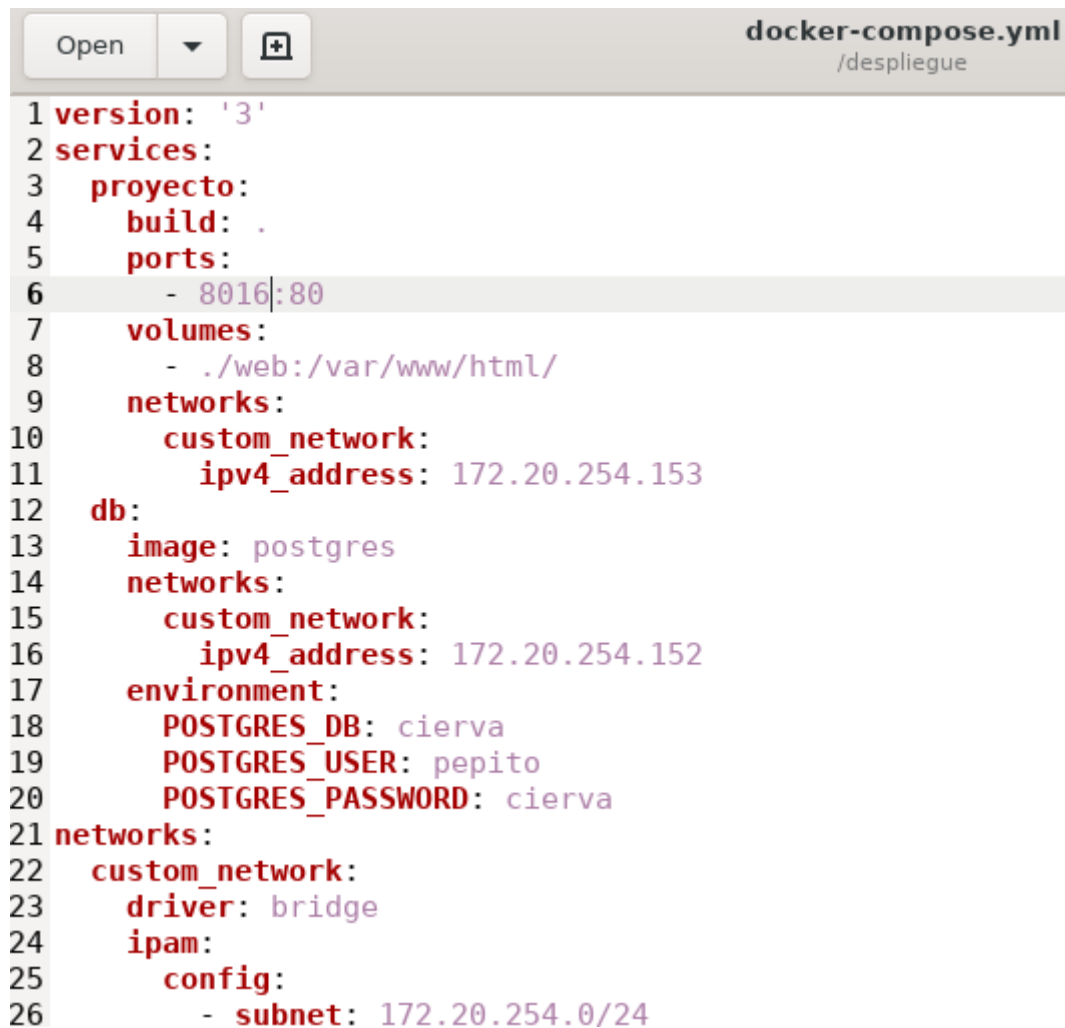
Sus valores son:

POSTGRES_DB: cierva

POSTGRES_USER: pepito

POSTGRES_PASSWORD: cierva

Y ya lo último identificar en que red se ubica docker-compose.yml que sería **172.20.254.0/24**.



```
1 version: '3'
2 services:
3   proyecto:
4     build: .
5     ports:
6     - 8016:80
7     volumes:
8     - ./web:/var/www/html/
9     networks:
10      custom_network:
11      ipv4_address: 172.20.254.153
12   db:
13     image: postgres
14     networks:
15      custom_network:
16      ipv4_address: 172.20.254.152
17     environment:
18      POSTGRES_DB: cierva
19      POSTGRES_USER: pepito
20      POSTGRES_PASSWORD: cierva
21 networks:
22   custom_network:
23     driver: bridge
24     ipam:
25       config:
26       - subnet: 172.20.254.0/24
```

3. En el fichero Dockerfile nombrado anteriormente empezamos a poner varias cosas:

- FROM php:7.4-apache-buster

Esto es para el primer contenedor para poder instalar y ejecutar un sistema operativo junto con Apache y PHP.

**- RUN apt-get update && apt-get install -y **
**libpq-dev **

&& docker-php-ext-install pgsql

Comandos para actualizar, instalar y pueda ser compatible Apache, PHP y PostgreSQL.

RUN a2enmod rewrite

Con este comando activa/sobrescribe Apache.

COPY 000-default.conf /etc/apache2/sites-available/

Copia el fichero 000-default.conf para el funcionamiento de la práctica.

COPY /web/index.php /var/www/html/

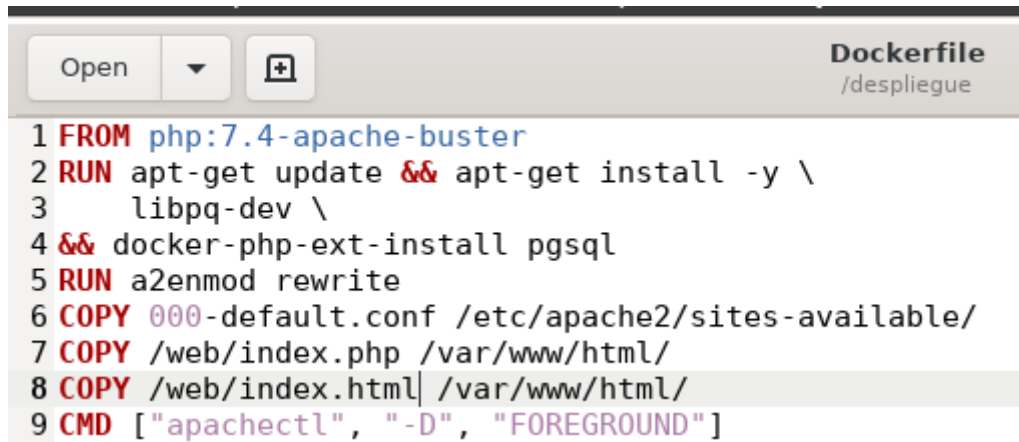
Copia el fichero index.php para el funcionamiento de la práctica.

COPY /web/index.html /var/www/html/

Copia el fichero index.html para el funcionamiento de la práctica.

CMD ["apachectl", "-D", "FOREGROUND"]

Ejecuta Apache y con ellos los demás ficheros.



```
1 FROM php:7.4-apache-buster
2 RUN apt-get update && apt-get install -y \
3     libpq-dev \
4     && docker-php-ext-install pgsql
5 RUN a2enmod rewrite
6 COPY 000-default.conf /etc/apache2/sites-available/
7 COPY /web/index.php /var/www/html/
8 COPY /web/index.html /var/www/html/
9 CMD ["apachectl", "-D", "FOREGROUND"]
```

Una vez tengamos estos 3 ficheros listos (000-default.conf, docker-compose.yml y Dockerfile), pasaremos al directorio web.

En él como se ha nombrado antes contendrá index.html e index.php.

Visualizamos index.html



```
1 <h1> Ruben| Perez Lara </h1>
```

Visualizamos index.php

Este fichero revisará si estos datos existen en PostgreSQL:

- a.) Si existen: se mostrará la página de PHP con `phpinfo()`; y nombre de alumno.
- b.) Si no existe: se mostrará `connection failed` y nombre alumno pero no mostrará `phpinfo()`;



```
1 <h1>Rubén Pérez Lara</h1>
2 ?php
3 function getdb() {
4     $db = pg_connect('host=172.20.254.152 port=5432 dbname=cierva user=pepito
5 password=cierva connect_timeout=8') or die('connection failed');
6     return $db;
7 }
8 echo getdb();
9 phpinfo();
10 ?>
```

Visualización index.html e index.php

Una vez tenemos nuestra carpeta despliegue que contiene dentro 000-default.conf, docker-compose.yml, Dockerfile y una carpeta web que contiene index.html e index.php, ya podemos encender los contenedores así

```
root@alumno-virtualbox:/despliegue# docker-compose up -d
Creating network "despliegue_custom_network" with driver "bridge"
Creating despliegue_db_1 ... done
Creating despliegue_proyecto_1 ... done
root@alumno-virtualbox:/despliegue#
```

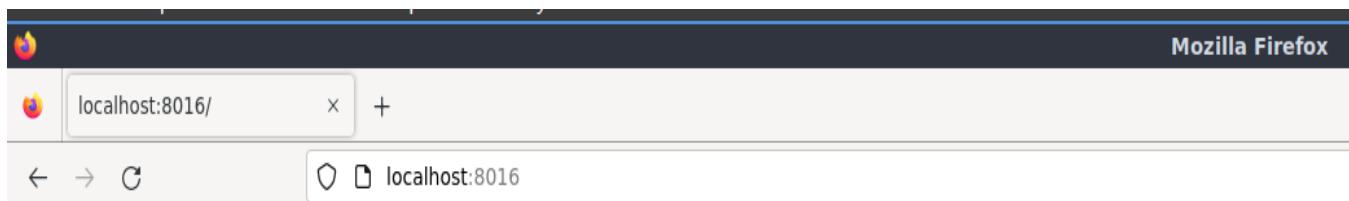
Miramos el estado de los contenedores

El primer container tiene Apache con PHP (la imagen la coge de Dockerfile)

El segundo container tiene PostgreSQL

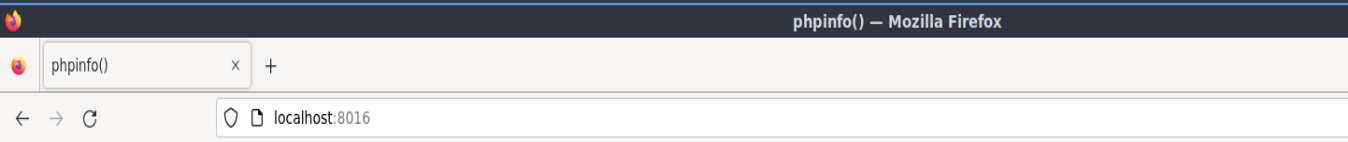
```
root@alumno-virtualbox:/despliegue# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
656682fba07d   despliegue_proyecto   "docker-php-entrypoi..." 54 seconds ago Up 53 seconds 0.0.0.0:8016->80/tcp, :::8016->80/tcp despliegue_proyecto_1
934ab18e4d04   postgres        "docker-entrypoint.s..." 54 seconds ago Up 53 seconds 5432/tcp                                           despliegue_db_1
root@alumno-virtualbox:/despliegue#
```

Visualizamos index.html (hay que mover o borrar index.php) (con tildes no se muestra bien el nombre en el index.html).




Ruben Perez Lara

Y por último visualizamos index.php (hay que mover o borrar index.html).



Rubén Pérez Lara

Resource id #2

PHP Version 7.2.34	
	
System	Linux 66c8d9c3f650 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:17 UTC 2021 x86_64
Build Date	Dec 11 2020 10:50:00
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-pgsql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Rebuild Build	no