2º DAMS

Esther Arias Donadiós

SISTEMAS DE GESTIÓN EMPRESARIAL

UNIDAD 1.
INTRODUCCIÓN A DOCKER.

ÍNDICE

Parte 3. Caso práctico 2. Instalando LAMP + Wordpress en un contenedor.

Preparando el contenedor:

```
esther@DESKTOP-TF5UUC2:~$ docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
35807b77a593: Pull complete
Digest: sha256:9d6a8699fb5c9c39cf08a0871bd6219f0400981c570894cd8cbea30d3424a31f
Status: Downloaded newer image for ubuntu:latest
root@9599f089553d:/#
```

Actualizamos la lista de paquetes e instalamos los necesarios para LAMP y Wordpress:

```
root@9599f089553d: /
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
35807b77a593: Pull complete
Digest: sha256:9d6a8699fb5c9c39cf08a0871bd6219f0400981c570894cd8cbea30d3424a31f
Status: Downloaded newer image for ubuntu:latest
root@9599f089553d:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [30.1 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [529 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [791 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1082 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1071 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [33.3 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1528 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [576 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6310 B]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2668 B]
Fetched 19.1 MB in 11s (1669 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@9599f089553d:/#
```

```
root@9599f089553d: /
-network-sk_SK.mo admin-network-sl_SI.mo admin-network-sq.mo admin-network-sr_RS.mo admin-network-sv_SE.mo admin-network
-ug_CN.mo admin-network-zh_CN.mo admin-network-zh_TW.mo admin-ru_RU.mo admin-sk_SK.mo admin-sl_SI.mo admin-sq.mo admin-s
r_RS.mo admin-sv_SE.mo admin-ug_CN.mo admin-zh_CN.mo admin-zh_TW.mo ar.mo bg_BG.mo bn_BD.mo bs_BA.mo ca.mo ckb.mo contin
ents-cities-bg_BG.mo continents-cities-bs_BA.mo continents-cities-cs_CZ.mo continents-cities-cy.mo continents-cities-da_
DK.mo continents-cities-de_DE.mo continents-cities-el.mo continents-cities-en_CA.mo continents-cities-es_PE.mo continent
s-cities-et.mo continents-cities-eu.mo continents-cities-fa_IR.mo continents-cities-fi.mo continents-cities-fr_FR.mo con
tinents-cities-gd.mo continents-cities-he_IL.mo continents-cities-hr.mo continents-cities-hu_HU.mo continents-cities-it_
IT.mo continents-cities-ja.mo continents-cities-ka_GE.mo continents-cities-ko_KR.mo continents-cities-lv.mo continents-c
ities-mk_MK.mo continents-cities-my_MM.mo continents-cities-nb_NO.mo continents-cities-nn_NO.mo continents-cities-pl_PL.
mo continents-cities-pt_BR.mo continents-cities-pt_PT.mo continents-cities-ro_RO.mo continents-cities-ru_RU.mo continent
s-cities-sk_SK.mo continents-cities-sl_SI.mo continents-cities-sq.mo continents-cities-sr_RS.mo continents-cities-su_ID.
mo continents-cities-sv_SE.mo continents-cities-ta_LK.mo continents-cities-tr_TR.mo continents-cities-ug_CN.mo continent
s-cities-vi.mo continents-cities-zh_CN.mo continents-cities-zh_TW.mo cs_CZ.mo cy.mo da_DK.mo de_DE.mo el.mo en_CA.mo eo.
mo es_CL.mo es_ES.mo es_PE.mo et.mo eu.mo fa_IR.mo fi.mo fr_FR.mo gd.mo gl_ES.mo he_IL.mo hr.mo hu_HU.mo id_ID.mo it_IT.
mo ja.mo jv_ID.mo ka_GE.mo ko_KR.mo lv.mo mk_MK.mo ms_MY.mo my_MM.mo nb_NO.mo nl.mo nl_NL.mo pl_PL.mo pt_BR.mo pt_PT.mo
ro_RO.mo ru_RU.mo ru_UA.mo si_LK.mo sk_SK.mo sl_SI.mo sq.mo sr_RS.mo su_ID.mo sv_SE.mo sw.mo ta_LK.mo th.mo tr_TR.mo ug_
CN.mo uk.mo ur.mo vi.mo zh_CN.mo zh_HK.mo zh_TW.mo
Setting up php (2:7.4+75) ...
Setting up wordpress-l10n (5.3.2+dfsg1-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for ca-certificates (20210119~20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.6) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.6) ...
root@9599f089553d:/#
```

Lanzamos el servicio Apache:

```
root@9599f089553d:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'Serv
erName' directive globally to suppress this message
 *
root@9599f089553d:/#
```

Creamos el fichero wordpress.conf:

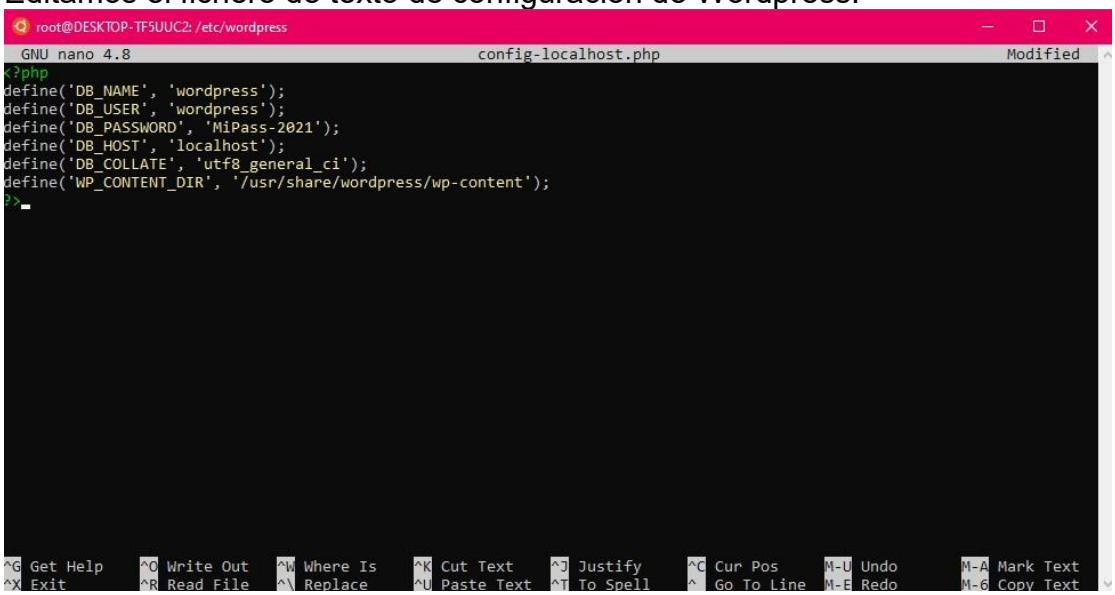
```
root@9599f089553d: /etc/apache2/sites-available
root@9599f089553d:/# ls etc/apache2/sites-available
000-default.conf  default-ssl.conf
root@9599f089553d:/# cd etc
root@9599f089553d:/etc# cd apache2
root@9599f089553d:/etc/apache2# cd sites-available
root@9599f089553d:/etc/apache2/sites-available# nano
root@9599f089553d:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf  wordpress.conf
root@9599f089553d:/etc/apache2/sites-available#
```

Preparamos el servicio SQL:

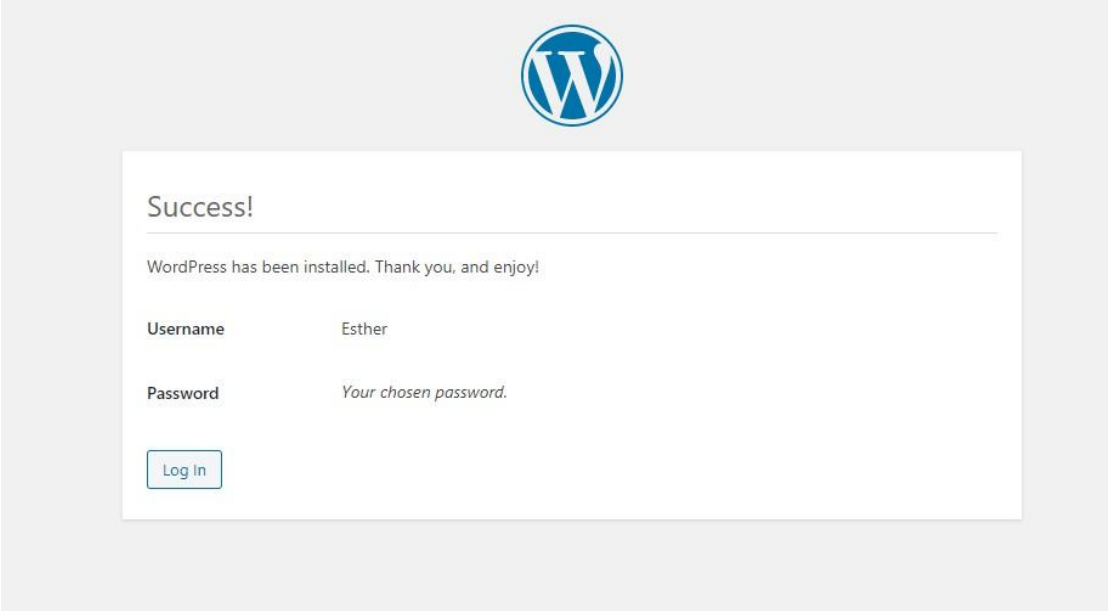
```
root@DESKTOP-TF5UUC2: /etc/apache2/sites-available
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
root@DESKTOP-TF5UUC2:/etc/apache2/sites-available#
```

```
root@DESKTOP-TF5UUC2: /etc/apache2/sites-available

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
mysql> CREATE USER 'wordpress'@'%' IDENTIFIED BY 'MiPass-2021';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

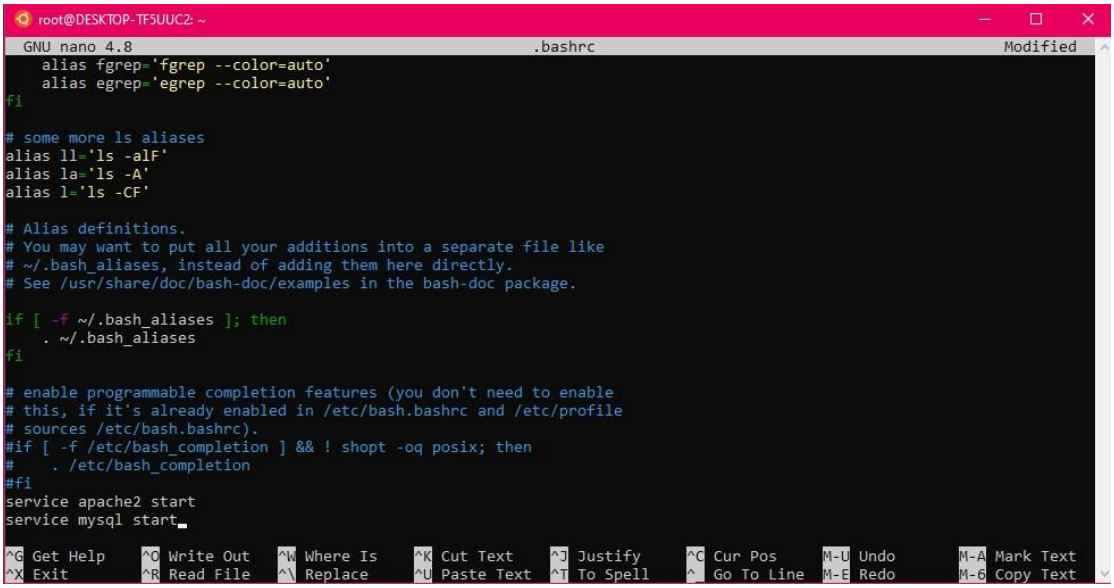
Editamos el fichero de texto de configuración de Wordpress:



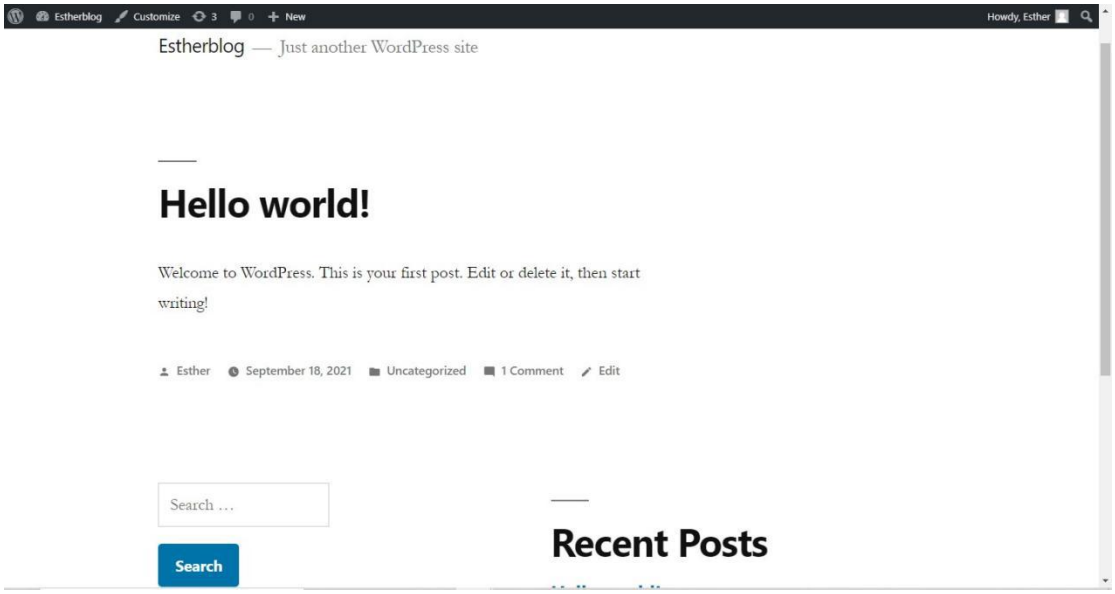
Configuramos nuestro sitio Wordpress:



Modificamos el archivo .bashrc:

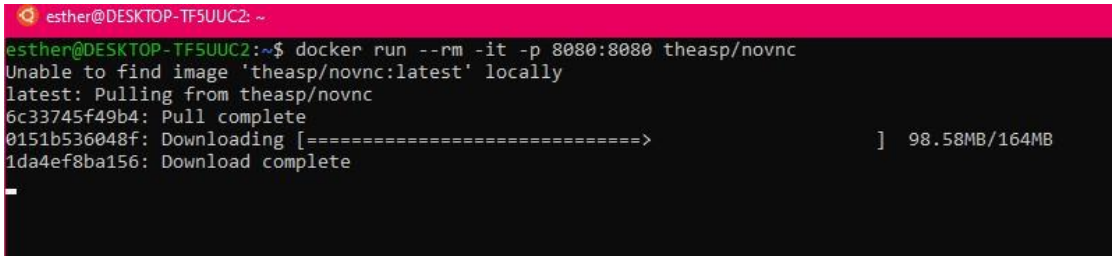


Comprobamos que todo haya funcionado correctamente:

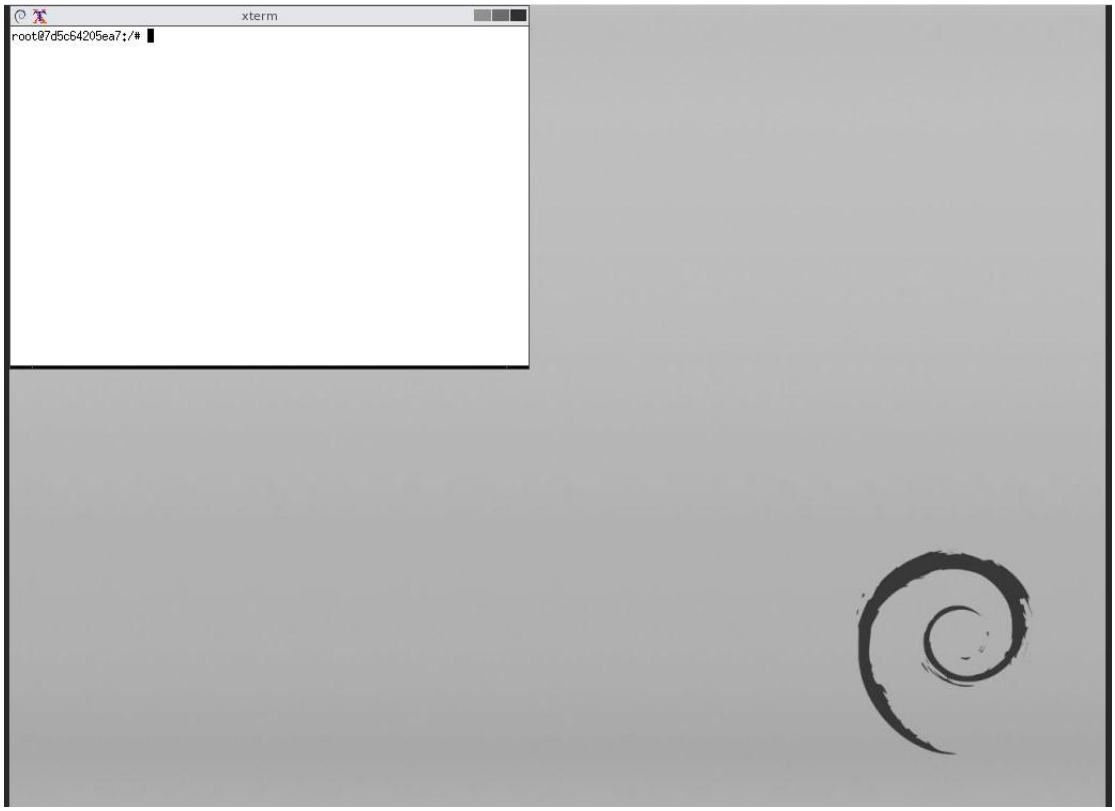


Parte 3. Caso práctico 3. Accediendo a interfaz gráfica con NoVNC.

Creamos el contenedor:

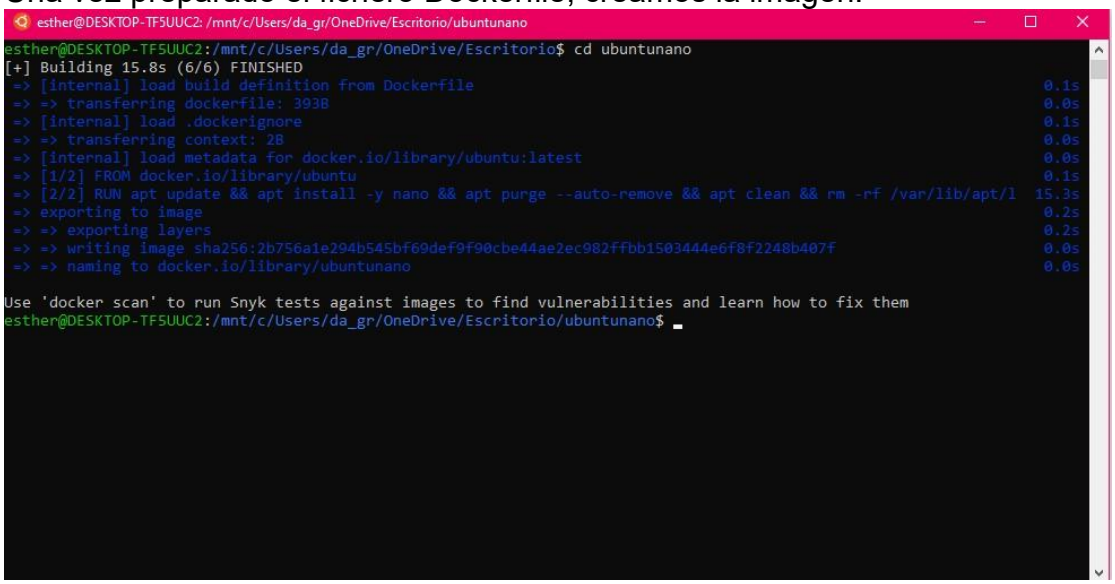


Al abrir la url vemos esto:

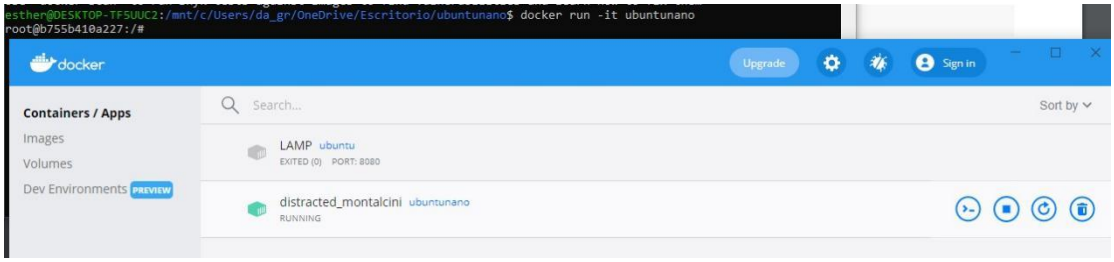


Parte 4. Caso práctico 1. Creando imagen Ubuntu con nano.

Una vez preparado el fichero Dockerfile, creamos la imagen:

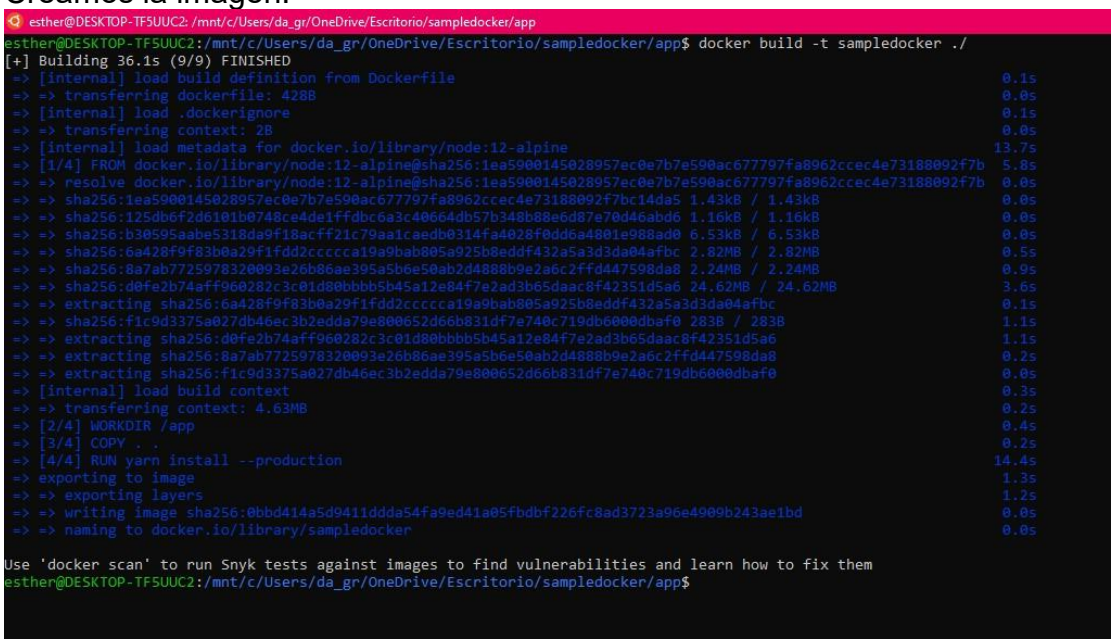


Comprobamos que se ha creado correctamente:

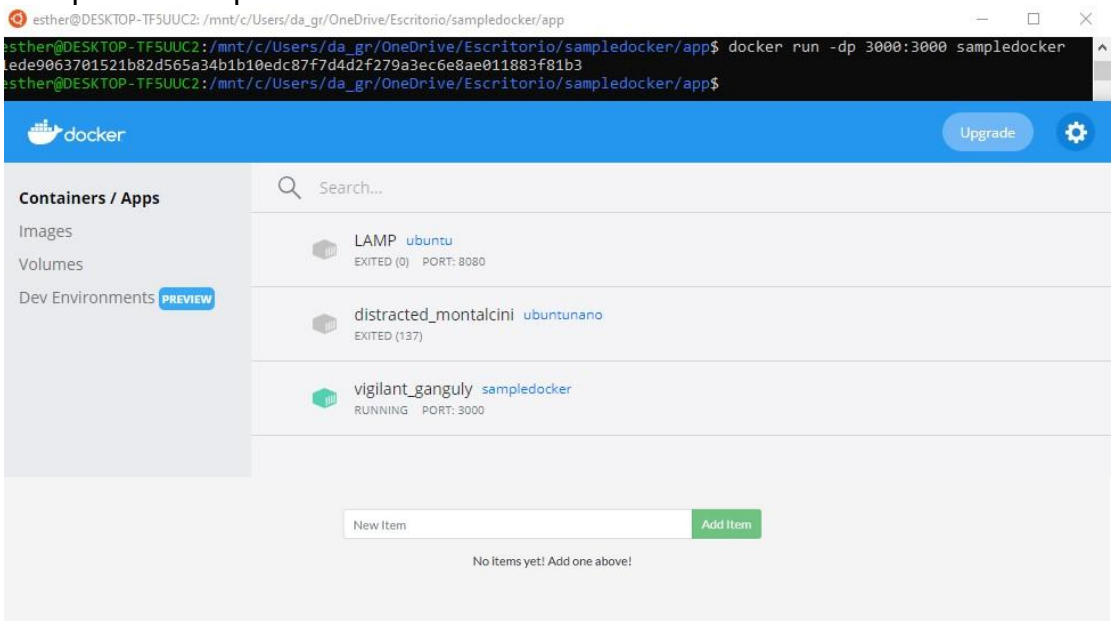


Parte 4. Caso práctico 2. Creando imagen con APP ejemplo Docker en Node.

Creamos la imagen:

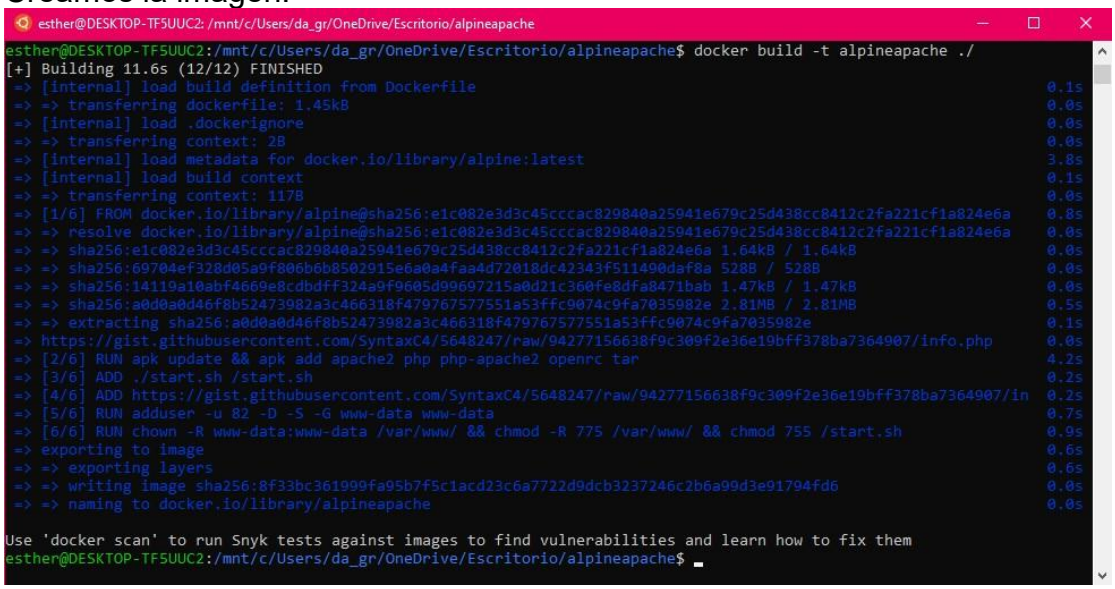


Comprobamos que ha funcionado:

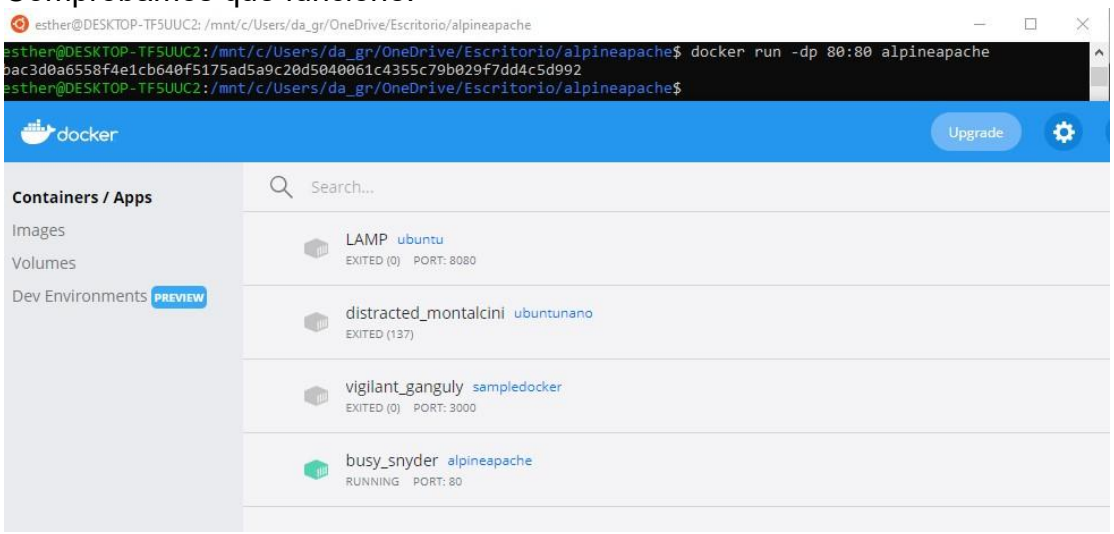


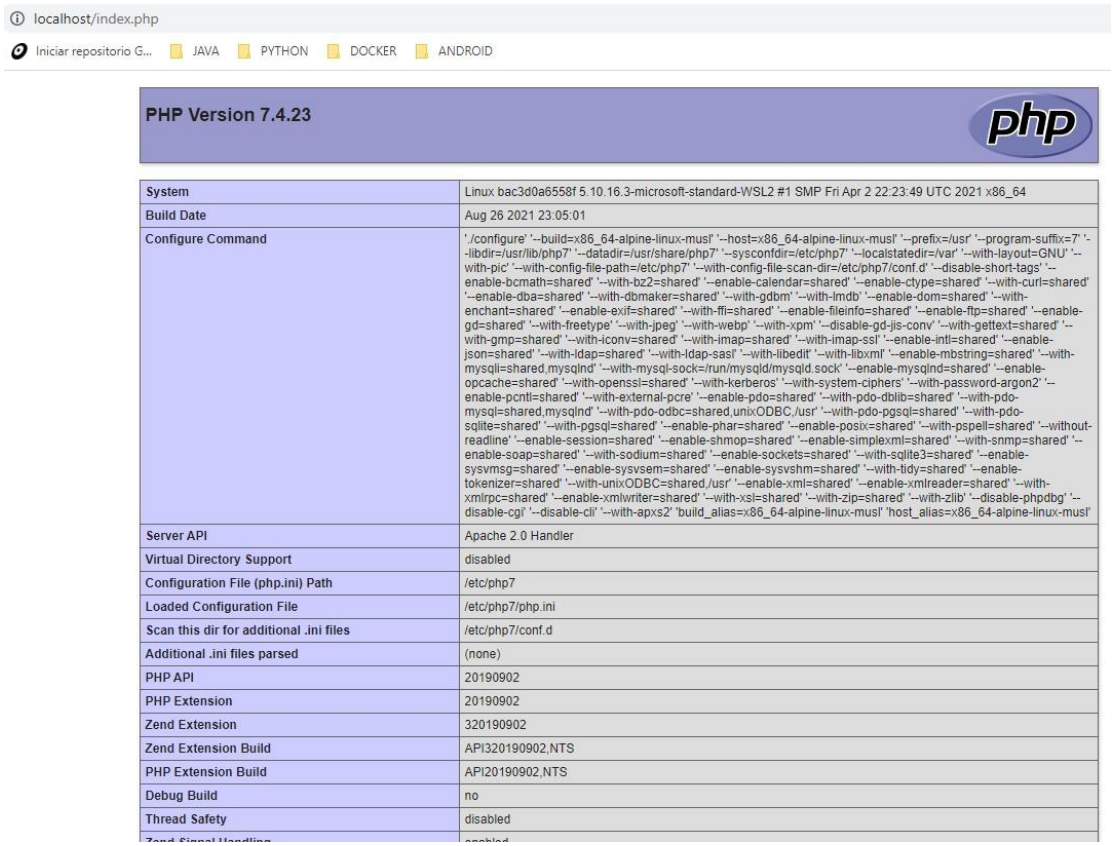
Parte 4. Caso práctico 3. Apache 2 con PHP desde Alpine.

Creamos la imagen:



Comprobamos que funcione:





System	Linux bac3d0a6558f 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64
Build Date	Aug 26 2021 23:05:01
Configure Command	'./configure' '--build=x86_64-alpine-linux-musl' '--host=x86_64-alpine-linux-musl' '--prefix=/usr' '--program-suffix=7' '--libdir=/usr/lib/php7' '--datadir=/usr/share/php7' '--sysconfdir=/etc/php7' '--localstatedir=/var' '--with-layout=GNU' '--with-pic' '--with-config-file-path=/etc/php7' '--with-config-file-scan-dir=/etc/php7/conf.d' '--disable-short-tags' '--enable-bcmath=shared' '--with-bz2=shared' '--enable-calendar=shared' '--enable-ctype=shared' '--with-curl=shared' '--enable-dba=shared' '--with-dbmaker=shared' '--with-gdbm' '--with-lmdb' '--enable-dom=shared' '--with-enchant=shared' '--enable-exif=shared' '--with-ffi=shared' '--enable-fileinfo=shared' '--enable-ftp=shared' '--enable-gd=shared' '--with-freetype' '--with-jpeg' '--with-webp' '--with-xpm' '--disable-gd-jis-conv' '--with-gettext=shared' '--with-gmp=shared' '--with-iconv=shared' '--with-imap=shared' '--with-imap-ssl' '--enable-intl=shared' '--enable-json=shared' '--with-ldap=shared' '--with-ldap-sasl' '--with-libedit' '--with-libxml' '--enable-mbstring=shared' '--with-mysqli=shared,mysqlnd' '--with-mysql-sock=/run/mysqld/mysqld.sock' '--enable-mysqlnd=shared' '--enable-opcache=shared' '--with-openssl=shared' '--with-kerberos' '--with-system-ciphers' '--with-password-argon2' '--enable-pcntl=shared' '--with-external-pcre' '--enable-pdo=shared' '--with-pdo-dblib=shared' '--with-pdo-mysql=shared,mysqlnd' '--with-pdo-odbc=shared,unixODBC,/usr' '--with-pdo-pgsql=shared' '--with-pdo-sqlite=shared' '--with-pgsql=shared' '--enable-phar=shared' '--enable-posix=shared' '--with-pspell=shared' '--without-readline' '--enable-session=shared' '--enable-shmop=shared' '--enable-simplexml=shared' '--with-snmp=shared' '--enable-soap=shared' '--with-sodium=shared' '--enable-sockets=shared' '--with-sqlite3=shared' '--enable-sysvmsg=shared' '--enable-sysvsem=shared' '--enable-sysvshm=shared' '--with-tidy=shared' '--enable-tokenizer=shared' '--with-unixODBC=shared,/usr' '--enable-xml=shared' '--enable-xmlreader=shared' '--with-xmlrpc=shared' '--enable-xmlwriter=shared' '--with-xsl=shared' '--with-zip=shared' '--with-zlib' '--disable-phpdbg' '--disable-cgi' '--disable-cli' '--with-apxs2' 'build_alias=x86_64-alpine-linux-musl' 'host_alias=x86_64-alpine-linux-musl'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php7
Loaded Configuration File	/etc/php7/php.ini
Scan this dir for additional .ini files	/etc/php7/conf.d
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled

Parte 5. Caso práctico 1. Wordpress + MySQL.

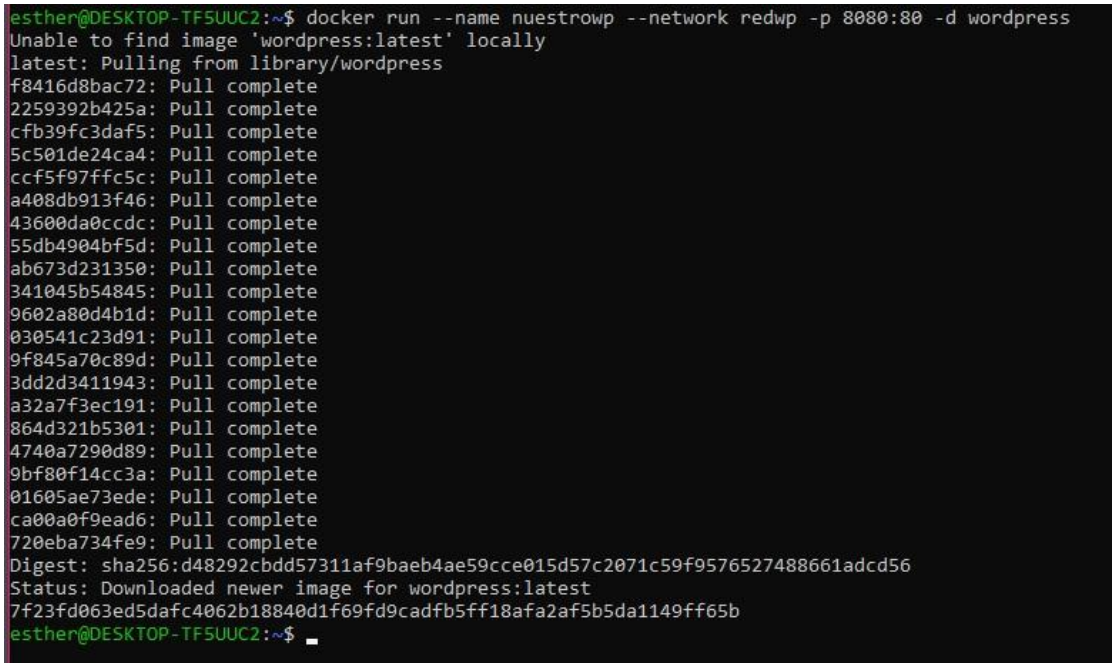
Primero creamos la red que van a compartir los contenedores:

```
esther@DESKTOP-TF5UUC2:~$ docker network create redwp
4cd06d0a7153af3eee8d0fac7fdd536cee26619616c8a47777f5257807e008de
esther@DESKTOP-TF5UUC2:~$
```

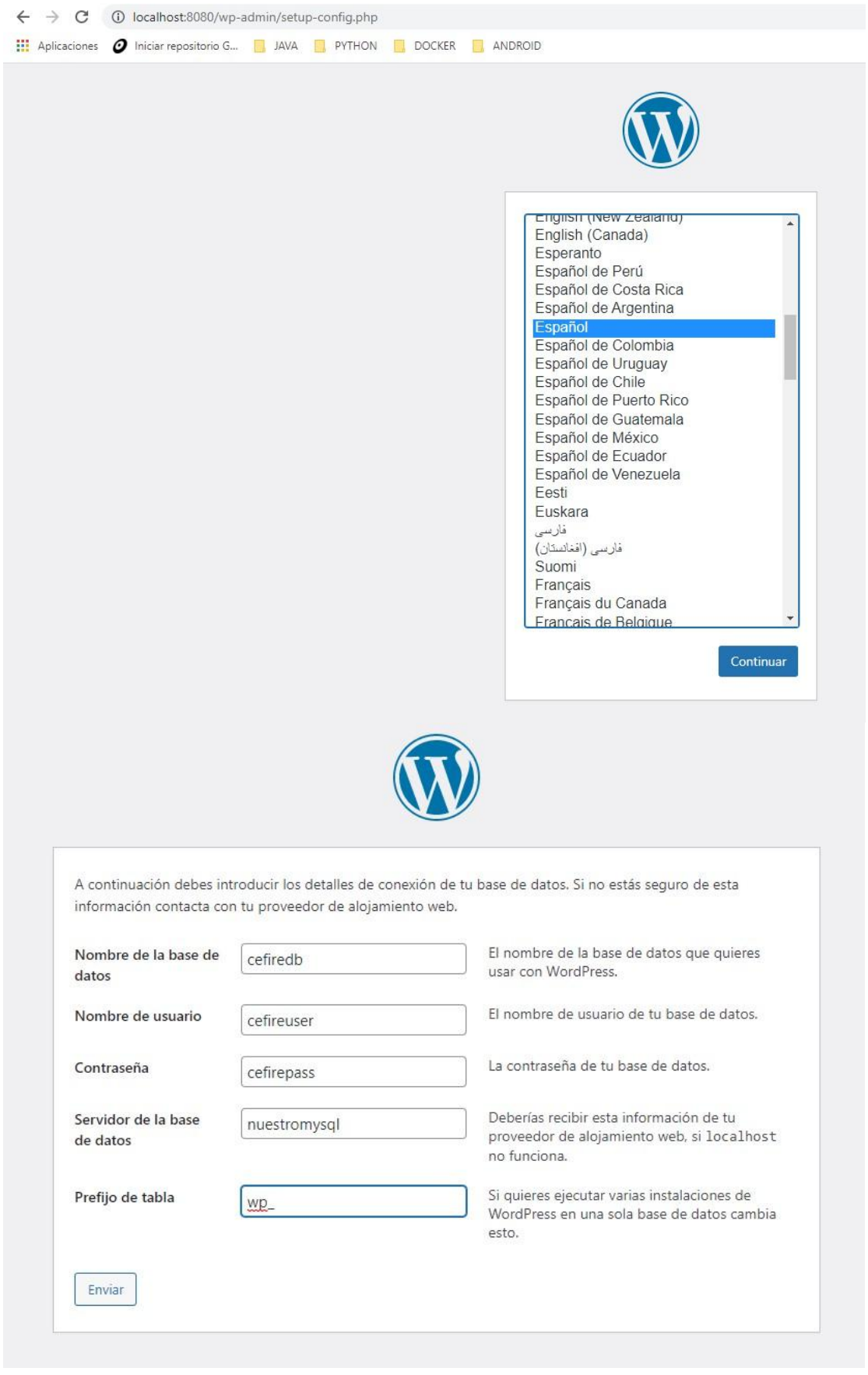
Luego el contenedor MySQL:

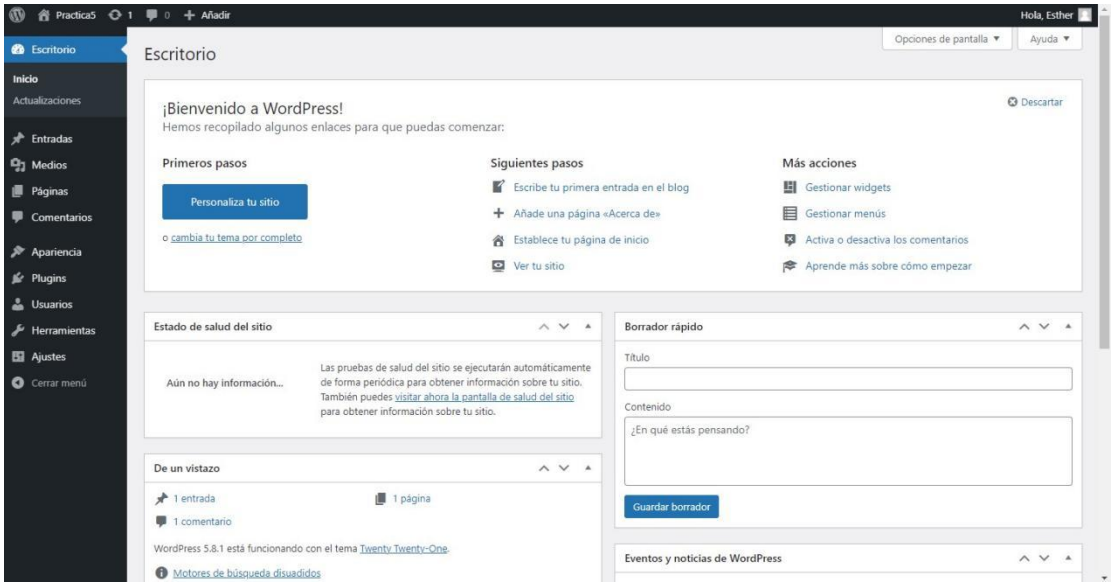
```
esther@DESKTOP-TF5UUC2:~$ docker run --name nuestromysql --network redwp -v /home/sergi/mysqldata:/var/lib/mysql -e MYSQ
L_ROOT_PASSWORD=cefireroot -e MYSQL_USER=cefireuser -e MYSQL_PASSWORD=cefirepass -e MYSQL_DATABASE=cefiredb -d mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
442547fc262c: Pull complete
2bf716144687: Pull complete
e8b3b16588b1: Pull complete
c89f7ee6da81: Pull complete
091490fb32f5: Pull complete
6eeb696bc30f: Pull complete
8a92263747b2: Pull complete
07097cad43f1: Pull complete
e09f00a44ec7: Pull complete
18f954e29df7: Pull complete
b46b7702c2b2: Pull complete
Digest: sha256:35aa66157963240633625d6490d940069a1311fdfc022bf32116cbf95b90b541
Status: Downloaded newer image for mysql:5.6
5d0d753ca8e41df025442e95044965cd4104dcb9c6aadcb83800752f59da117c
esther@DESKTOP-TF5UUC2:~$
```

Ahora creamos el contenedor que contendrá Apache, PHP y Wordpress:



Instalamos y configuramos Wordpress:


localhost:8080/wp-admin/setup-config.php
Aplicaciones
Iniciar repositorio G...
JAVA
PYTHON
DOCKER
ANDROID
English (New Zealand)
English (Canada)
Esperanto
Español de Perú
Español de Costa Rica
Español de Argentina
Español
Español de Colombia
Español de Uruguay
Español de Chile
Español de Puerto Rico
Español de Guatemala
Español de México
Español de Ecuador
Español de Venezuela
Eesti
Euskara
فارسی
فارسی (افغانستان)
Suomi
Français
Français du Canada
Français de Belgique
Continuar
A continuación debes introducir los detalles de conexión de tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.
Nombre de la base de datos
cefiredb
El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario
cefireuser
El nombre de usuario de tu base de datos.
Contraseña
cefirepass
La contraseña de tu base de datos.
Servidor de la base de datos
nuestromysql
Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla
wp_
Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.
Enviar



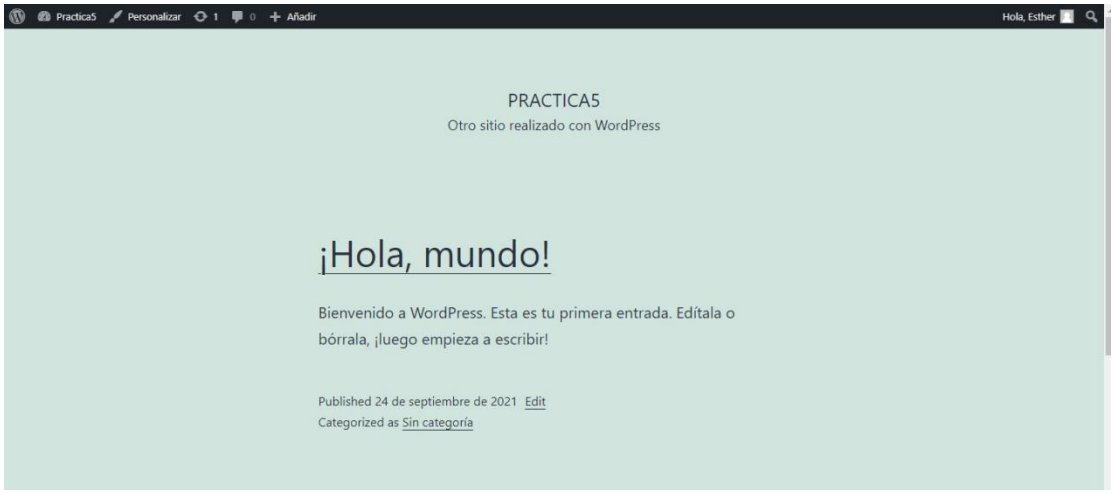
Y ahora migramos el contenedor MySQL de 5.6 a 5.7, para ello primero paramos y eliminamos el contenedor “nuestromysql”:

```
esther@DESKTOP-TF5UUC2:~$ docker stop nuestromysql
nuestromysql
esther@DESKTOP-TF5UUC2:~$ docker rm nuestromysql
nuestromysql
esther@DESKTOP-TF5UUC2:~$
```

Creamos el nuevo contenedor pasándole los datos que habíamos guardado previamente del otro:

```
esther@DESKTOP-TF5UUC2:~$ docker run --name nuestromysql --network redwp -v /home/sergi/mysqldata:/var/lib/mysql -d mysq
l:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
a330b6cecb98: Pull complete
9c8f656c32b8: Pull complete
88e473c3f553: Pull complete
062463ea5d2f: Pull complete
daf7e3bdf4b6: Pull complete
1839c0b7aac9: Pull complete
cf0a0cfee6d0: Pull complete
fae7a809788c: Pull complete
dae5a82a61f0: Pull complete
7063da9569eb: Pull complete
51a9a9b4ef36: Pull complete
Digest: sha256:d9b934cdf6826629f8d02ea01f28b2c4ddb1ae27c32664b14867324b3e5e1291
Status: Downloaded newer image for mysql:5.7
913cc4e85242f73a122ddeb99e27e37c1a1149377d4b4ea1b845b7346b299889
esther@DESKTOP-TF5UUC2:~$
```

Comprobamos que todo ha salido bien accediendo a Wordpress:

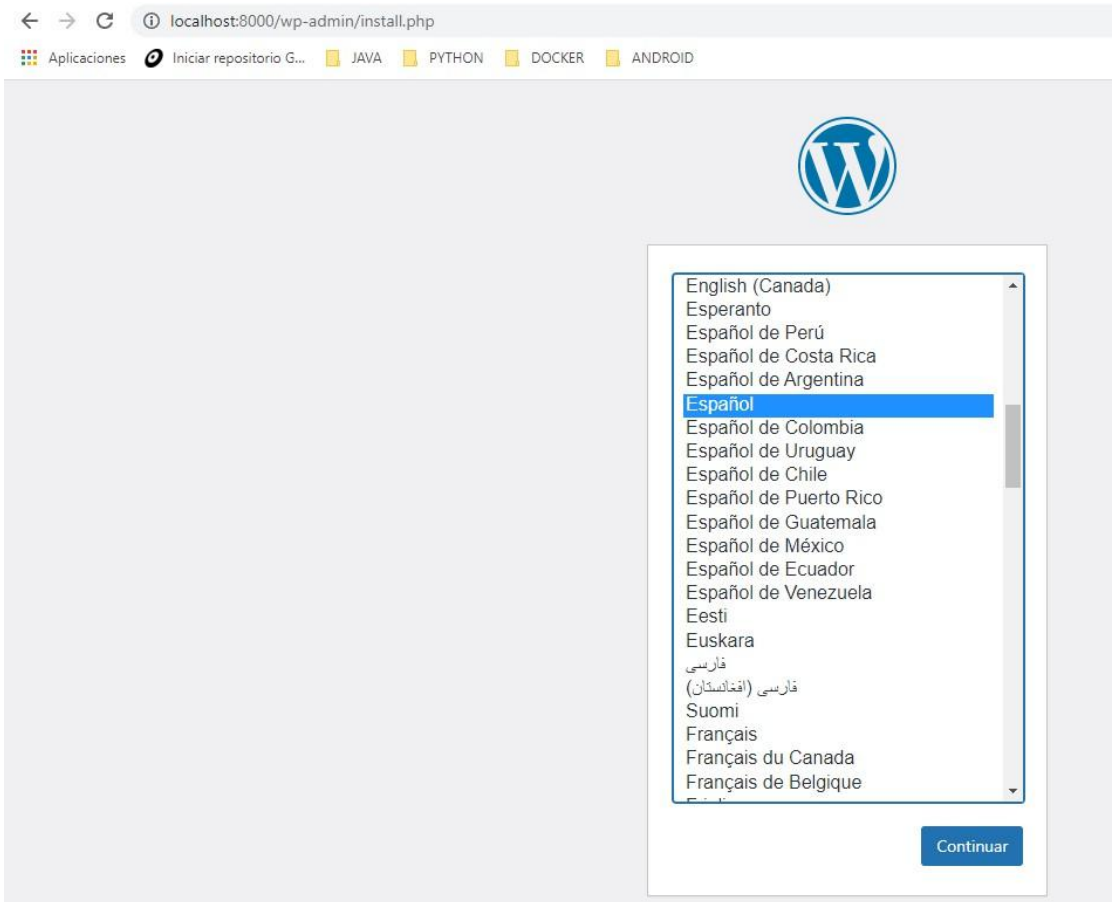


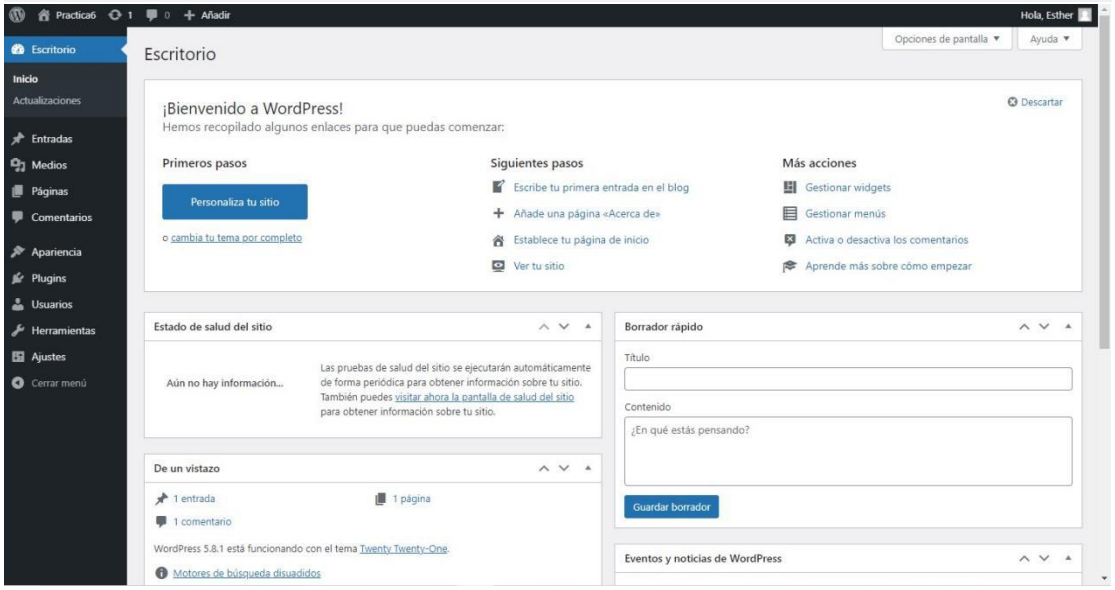
Parte 6. Caso práctico 1. Wordpress con Docker Compose.

Nos situamos en la carpeta donde tenemos el fichero docker-compose.yml y lo lanzamos:

```
esther@DESKTOP-TF5UUC2:/mnt/c/Users/da_gr/OneDrive/Escritorio$ docker-compose up -d
[+] Running 4/4
 ⠿ Network escritorio_default        Created                                    0.8s
 ⠿ Volume "escritorio_db_data"       Created                                    0.0s
 ⠿ Container escritorio_db_1         Started                                    2.2s
 ⠿ Container escritorio_wordpress_1  Started                                    3.2s
esther@DESKTOP-TF5UUC2:/mnt/c/Users/da_gr/OneDrive/Escritorio$
```

Para comprobar que ha funcionado accedemos a wordpress y configuramos el sitio:



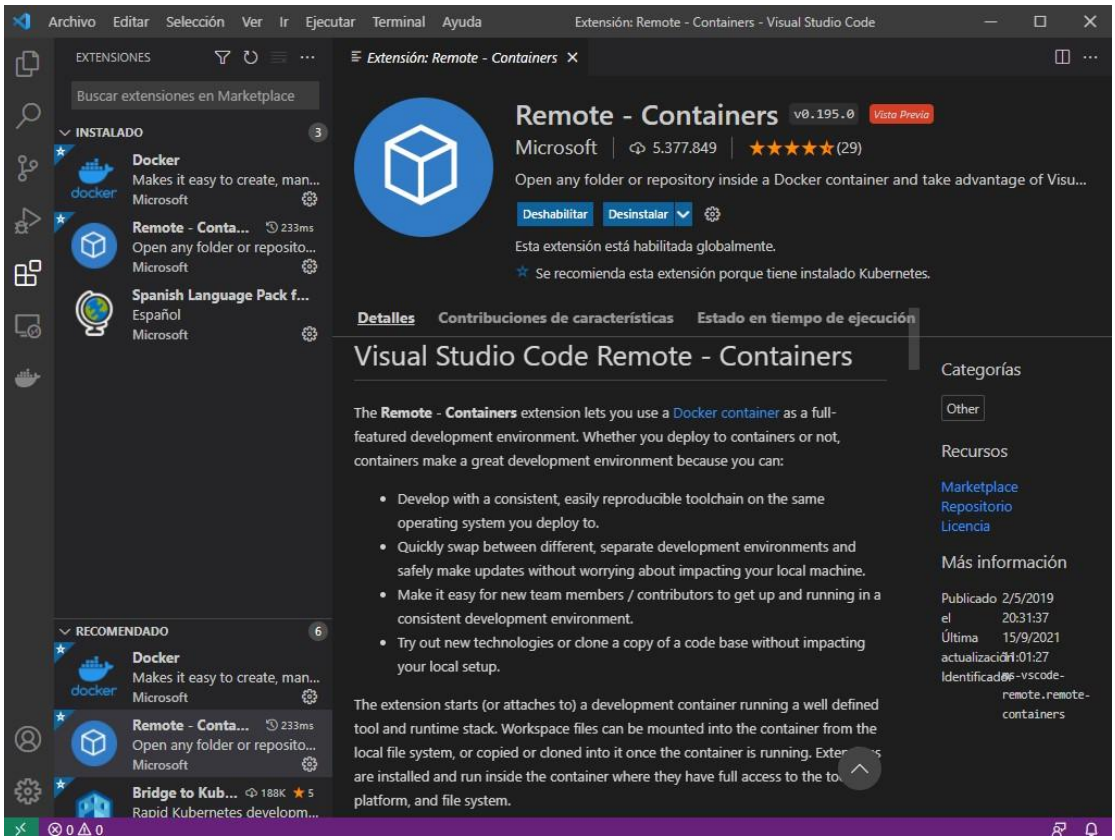


Comprobamos que podemos parar y relanzar el sistema cuando queramos:

```
esther@DESKTOP-TF5UUC2:/mnt/c/Users/da_gr/OneDrive/Escritorio$ docker-compose down
[+] Running 3/3
 ⠿ Container escritorio_wordpress_1  Removed                                    2.8s
 ⠿ Container escritorio_db_1         Removed                                    2.1s
 ⠿ Network escritorio_default        Removed                                    0.8s
esther@DESKTOP-TF5UUC2:/mnt/c/Users/da_gr/OneDrive/Escritorio$ docker-compose up -d
[+] Running 3/3
 ⠿ Network escritorio_default        Created                                    0.8s
 ⠿ Container escritorio_db_1         Started                                    2.5s
 ⠿ Container escritorio_wordpress_1  Started                                    3.6s
esther@DESKTOP-TF5UUC2:/mnt/c/Users/da_gr/OneDrive/Escritorio$
```

Parte 7. Caso práctico 1. Desarrollando con Visual Studio Code en un contenedor.

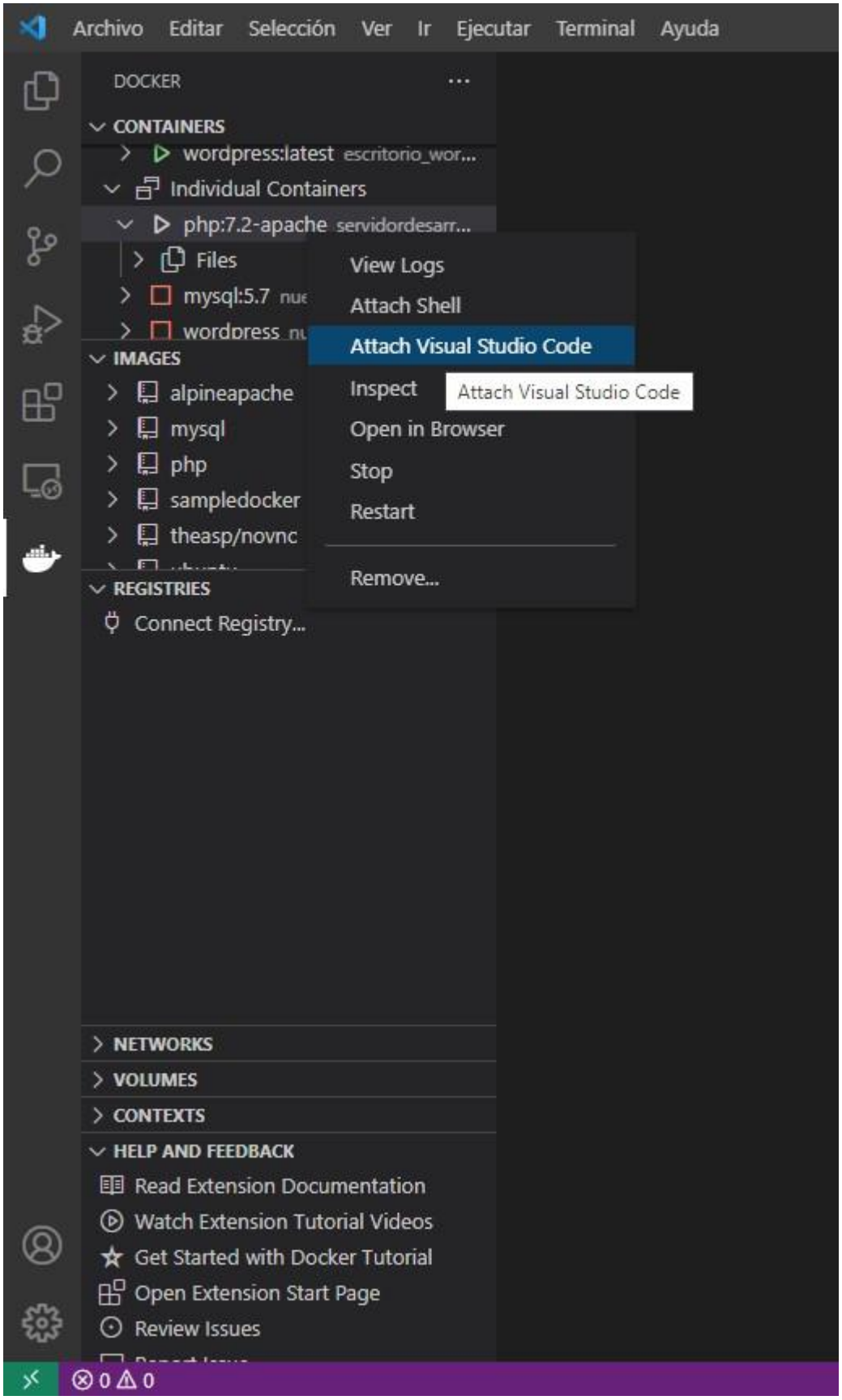
Primero instalamos Visual Studio y los plugins necesarios para usarlo con Docker:



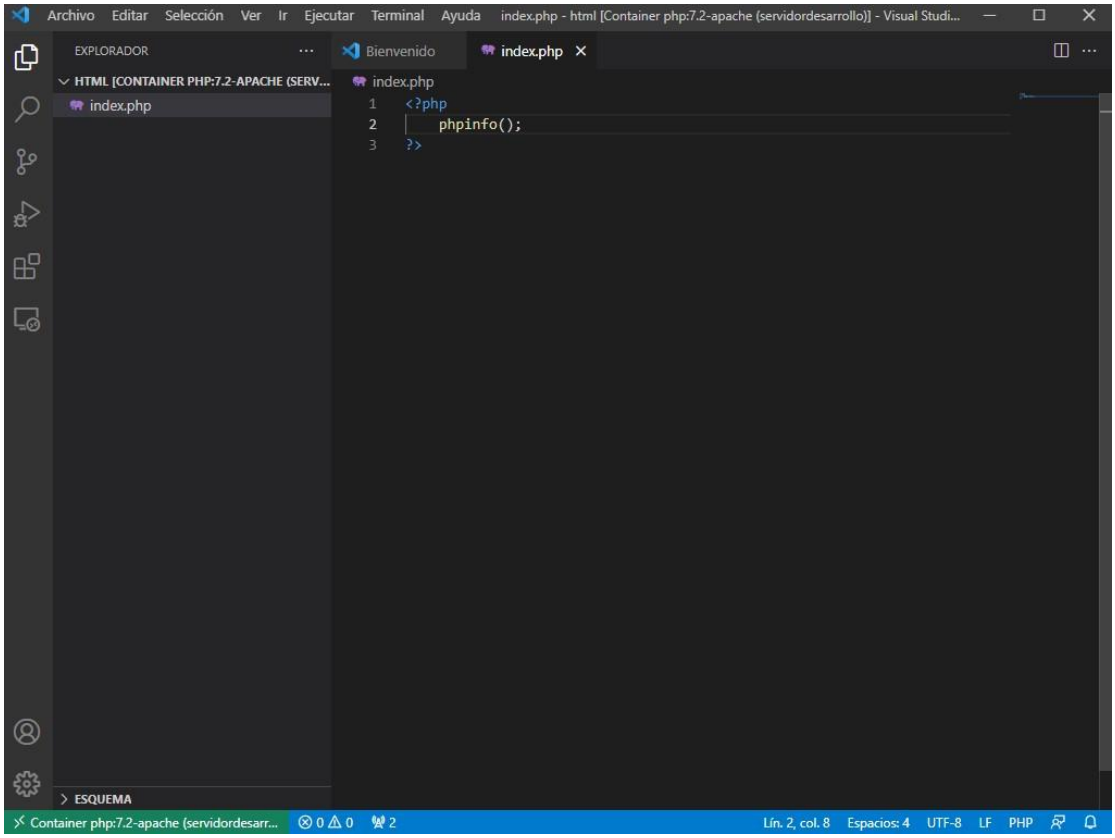
Creamos el contenedor:

```
esther@DESKTOP-TF5UUC2:~$ docker run -d --name servidordesarrollo -p 8080:80 php:7.2-apache
Unable to find image 'php:7.2-apache' locally
7.2-apache: Pulling from library/php
6ec7b7d162b2: Pull complete
db606474d60c: Pull complete
afb30f0cd8e0: Pull complete
3bb2e8051594: Pull complete
4c761b44e2cc: Pull complete
c2199db96575: Pull complete
1b9a9381eea8: Pull complete
fd07bbc59d34: Pull complete
72b73ab27698: Pull complete
983308f4f0d6: Pull complete
6c13f026e6da: Pull complete
e5e6cd163689: Pull complete
5c5516e56582: Pull complete
154729f6ba86: Pull complete
Digest: sha256:4dc0f0115acf8c2f0df69295ae822e49f5ad5fe849725847f15aa0e5802b55f8
Status: Downloaded newer image for php:7.2-apache
8da37be62e8a91ef9754f3ec777423c17457ea329dee1df0da4e5715e58c0cee
esther@DESKTOP-TF5UUC2:~$
```

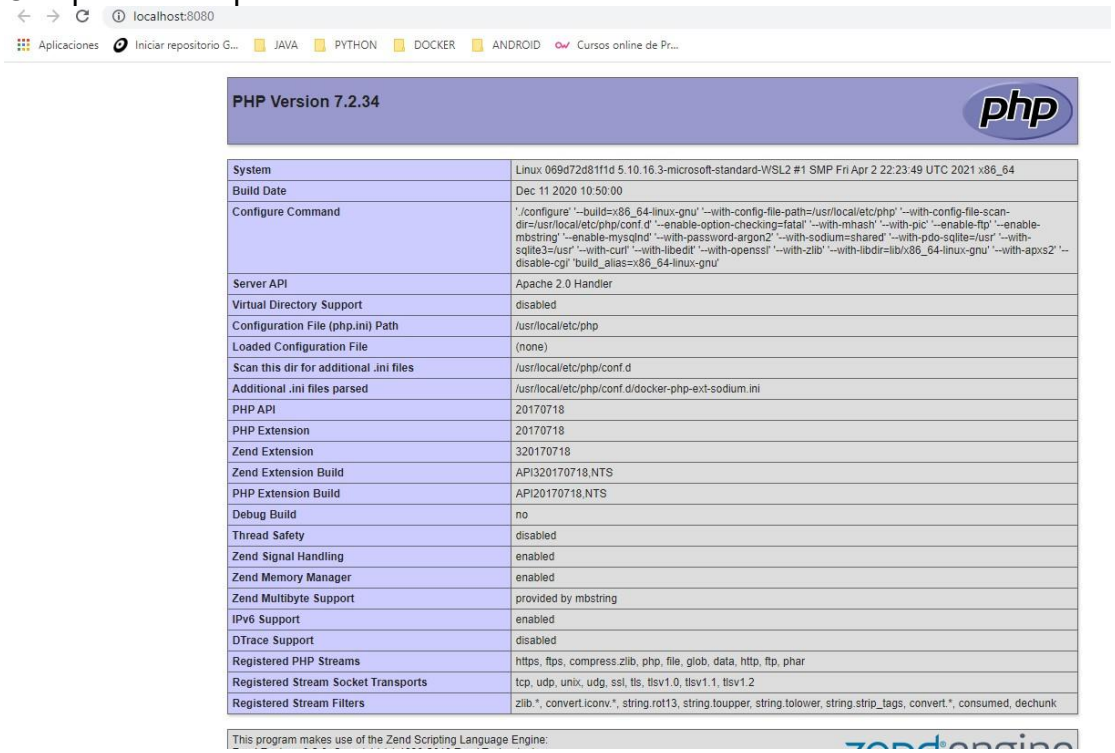
Dentro de Visual Studio, en el icono de la ballena, podremos ver un listado de nuestros contenedores, seleccionamos el que acabamos de crear y hacemos click derecho, entonces pulsamos en Attach Visual Studio Code para “enlazarlo” a la aplicación:



Una vez hecho esto nos colocamos en el directorio var/www/html y creamos el fichero index.php:



Comprobamos que todo ha salido bien entrando en el localhost:

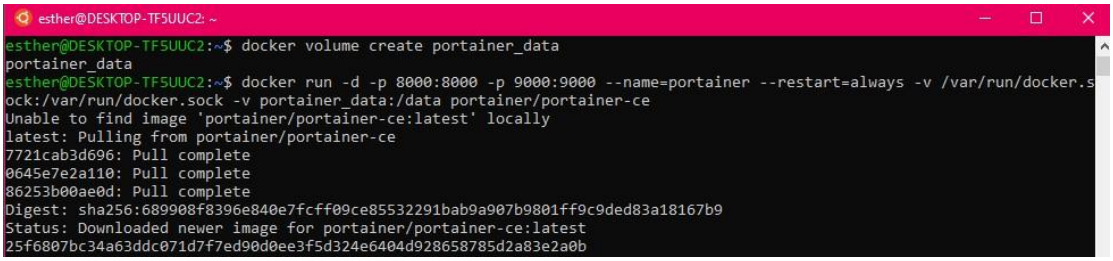


System	Linux 069d72d81f1d 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64
Build Date	Dec 11 2020 10:50:00
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--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-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

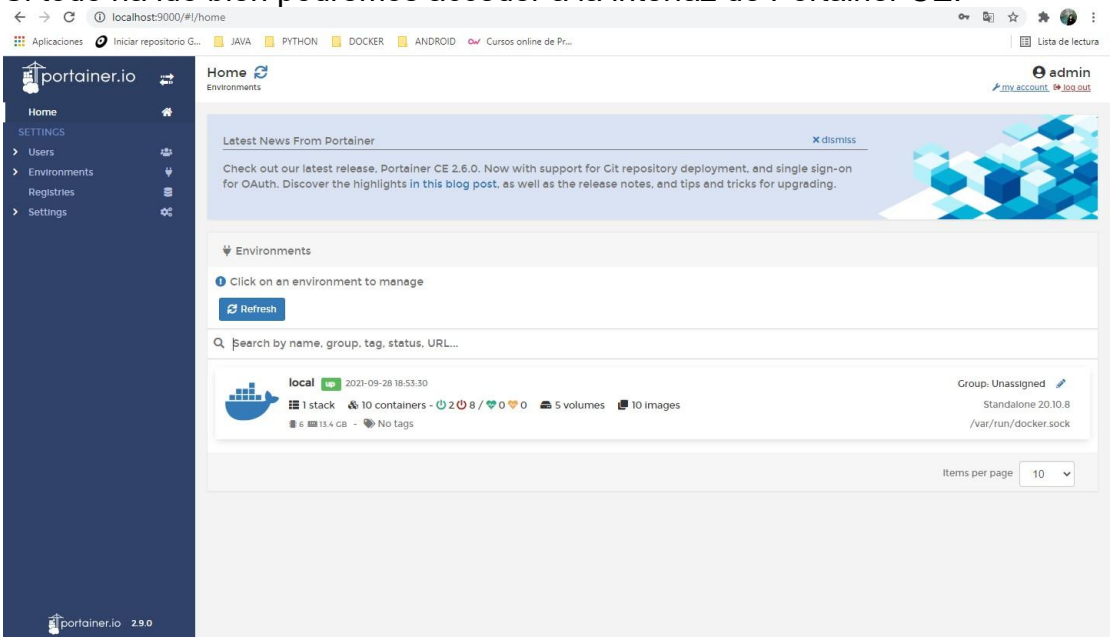
This program makes use of the Zend Scripting Language Engine:

Parte 7. Caso práctico 2. Gestionando Docker con Portainer CE.

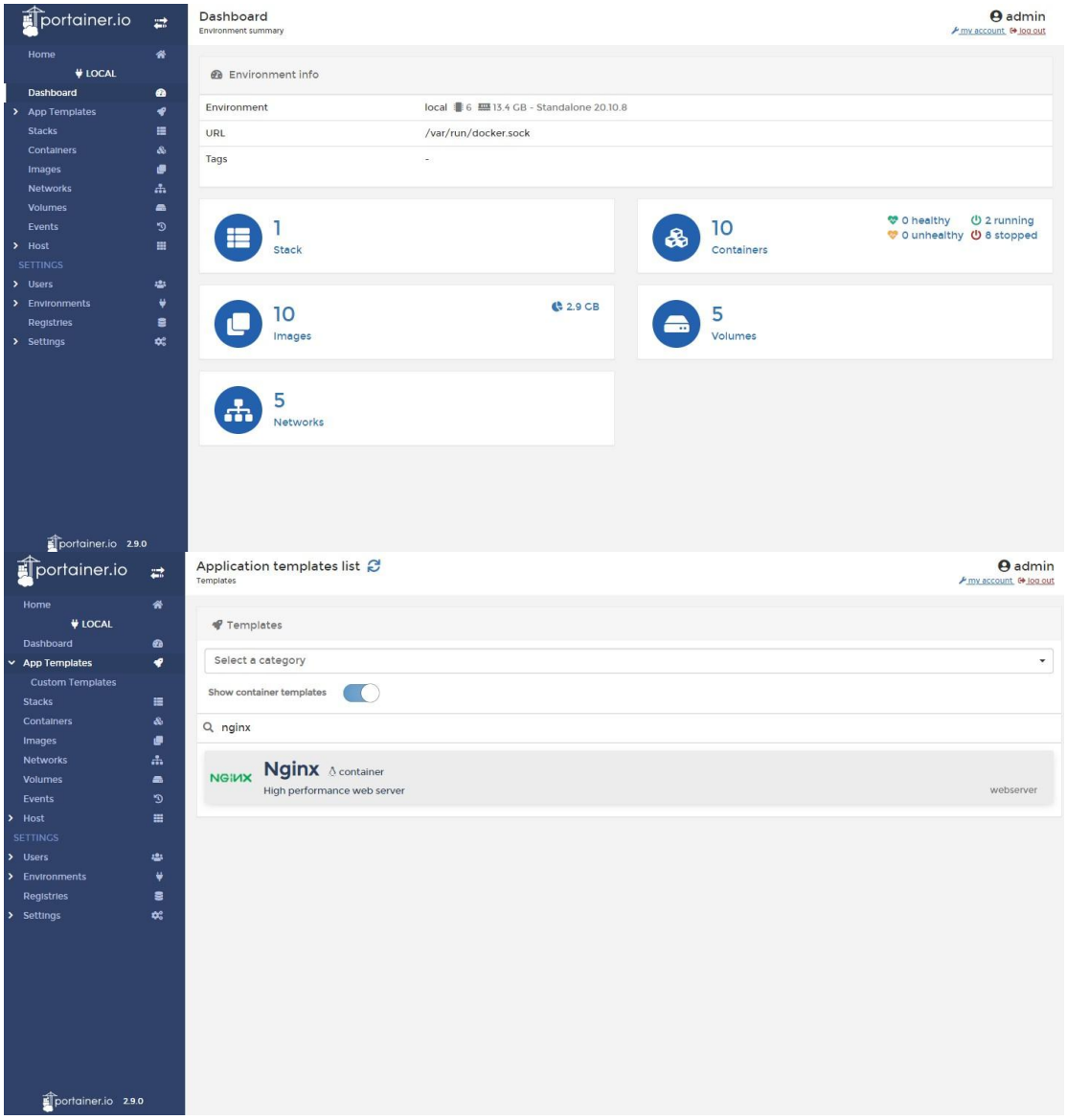
Creamos y lanzamos el contenedor que tiene todo lo necesario para que funcione Portainer CE:



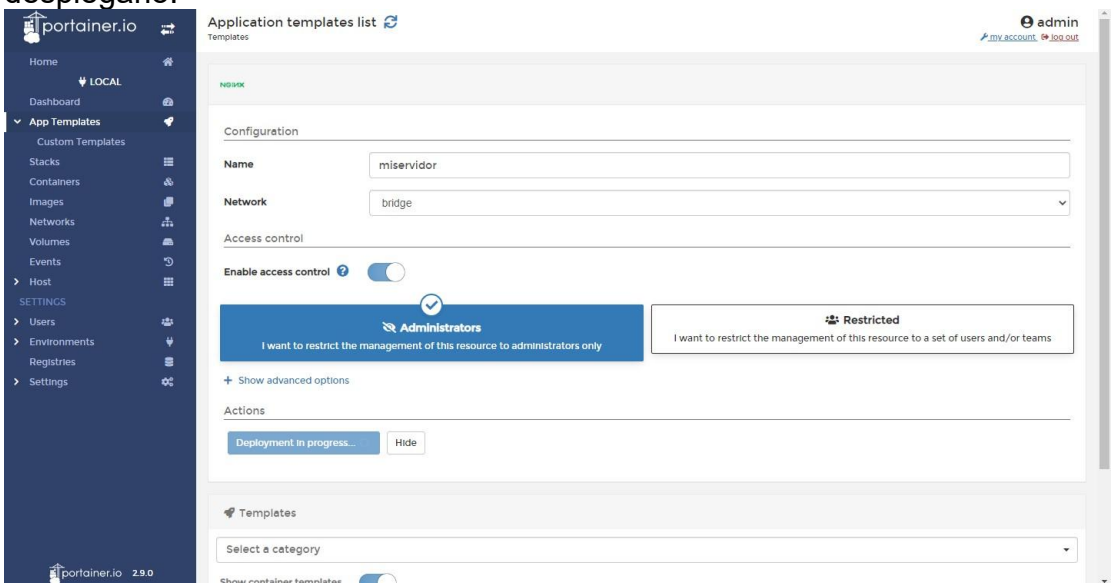
Si todo ha ido bien podremos acceder a la interfaz de Portainer CE:



Entramos en local y en el menú de la izquierda en App Templates, escribimos nginx en la barra de búsqueda:



Preparamos el contenedor y hacemos click en Deploy the container para desplegarlo:



Ahora podremos ver una lista con todos nuestros contenedores, entre ellos el que acabamos de crear a través de la interfaz: