Manual WordPress

DESPLEGANDO EL CMS WORDPRESS

Índice

Introducción:

En esta práctica se despliega el CMS WordPress, utilizando dos contenedor y una red que los conecta

Instalación:

Estos son los comandos necesarios para montar el entorno de Wordpress sobre dos contenedores conectados a la red, red_wp.

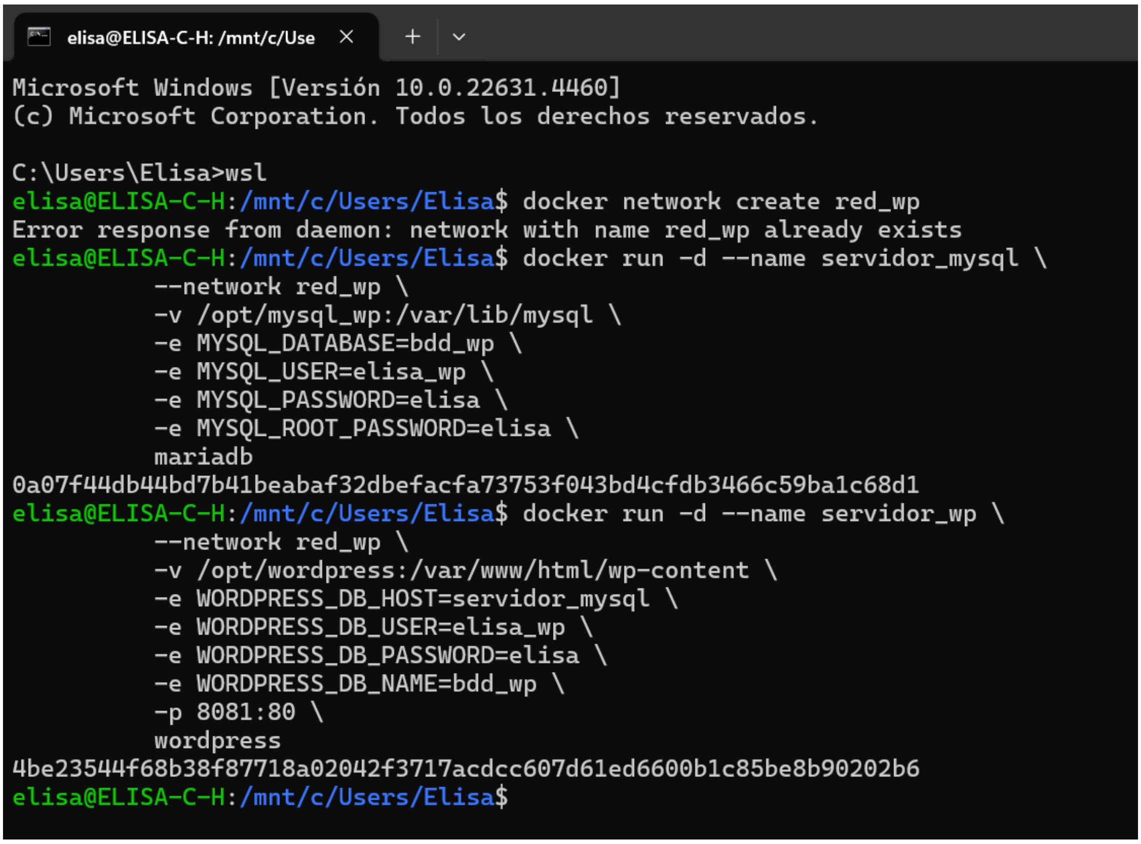
```
$ docker network create red_wp docker run -d --name servidor_mysql
--network red_wp
-v /opt/mysql_wp:/var/lib/mysql
-e MYSQL_DATABASE=bdd_wp
-e MYSQL_USER=elisa_wp
-e MYSQL_PASSWORD=elisa
-e MYSQL_ROOT_PASSWORD=elisa
mariadb
```

```
 $docker run -d --name servidor_wp
--network red_wp
-v /opt/wordpress:/var/www/html/wp-content
-e WORDPRESS_DB_HOST=servidor_mysql
-e WORDPRESS_DB_USER=user_wp
-e WORDPRESS_DB_PASSWORD=asdasd
-e WORDPRESS_DB_NAME=bd_wp
-p 80:80
wordpress
```

Elementos importantes:

El contenedor servidor_mysql ejecuta un script docker-entrypoint.sh que es el encargado de configurar la base de datos y termina ejecutando el servidor mariadb. Al crear la imagen mariadb tiene que permitir la conexión desde otra máquina, por lo que en la configuración tenemos comentado el parámetro bind-address. Bind-address

Del contenedor servidor_wp ejecuta un script docker-entrypoint.sh, a partir de las variables de entorno, ha creado el fichero wp-config.php de wordpress.



```
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Elisa>wsl
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker network create red_wp
Error response from daemon: network with name red_wp already exists
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker run -d --name servidor_mysql \
          --network red_wp \
          -v /opt/mysql_wp:/var/lib/mysql \
          -e MYSQL_DATABASE=bdd_wp \
          -e MYSQL_USER=elisa_wp \
          -e MYSQL_PASSWORD=elisa \
          -e MYSQL_ROOT_PASSWORD=elisa \
          mariadb
0a07f44db44bd7b41beabaf32dbefacfa73753f043bd4cfdb3466c59ba1c68d1
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker run -d --name servidor_wp \
          --network red_wp \
          -v /opt/wordpress:/var/www/html/wp-content \
          -e WORDPRESS_DB_HOST=servidor_mysql \
          -e WORDPRESS_DB_USER=elisa_wp \
          -e WORDPRESS_DB_PASSWORD=elisa \
          -e WORDPRESS_DB_NAME=bdd_wp \
          -p 8081:80 \
          wordpress
4be23544f68b38f87718a02042f3717acdcc607d61ed6600b1c85be8b90202b6
elisa@ELISA-C-H:/mnt/c/Users/Elisa$
```

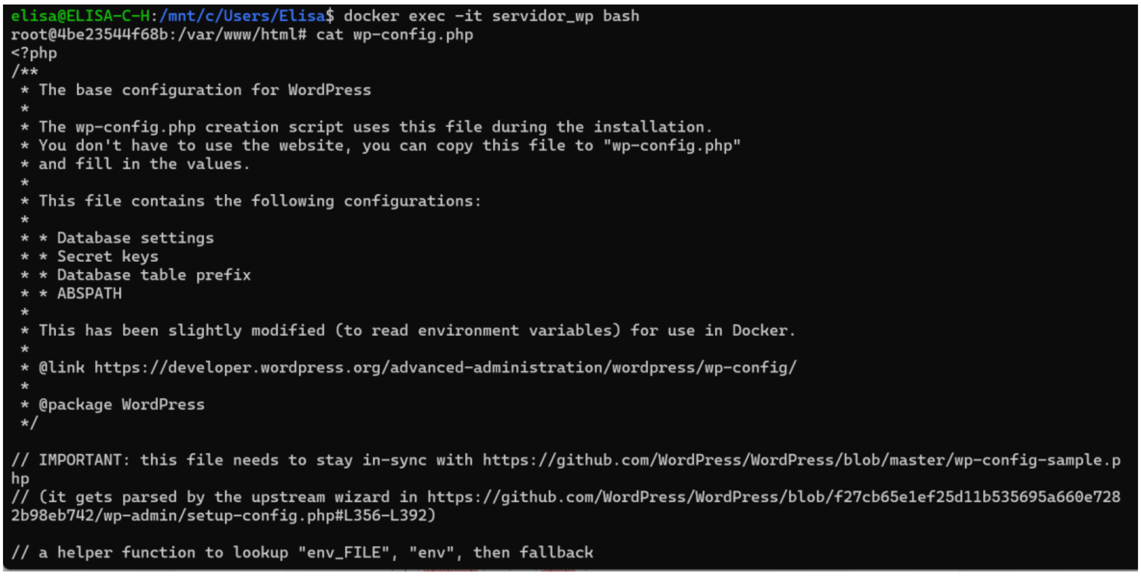
Desarrollo de la práctica:

Cuestiones;

- Ejecuta una instrucción docker para visualizar el contenido del fichero `wp-config.php` y verifica que los parámetros de conexión a la base de datos son los mismo que los indicados en las variables de entorno.

$ docker exec -it servidor_wp bash

$ cat wp-config.php



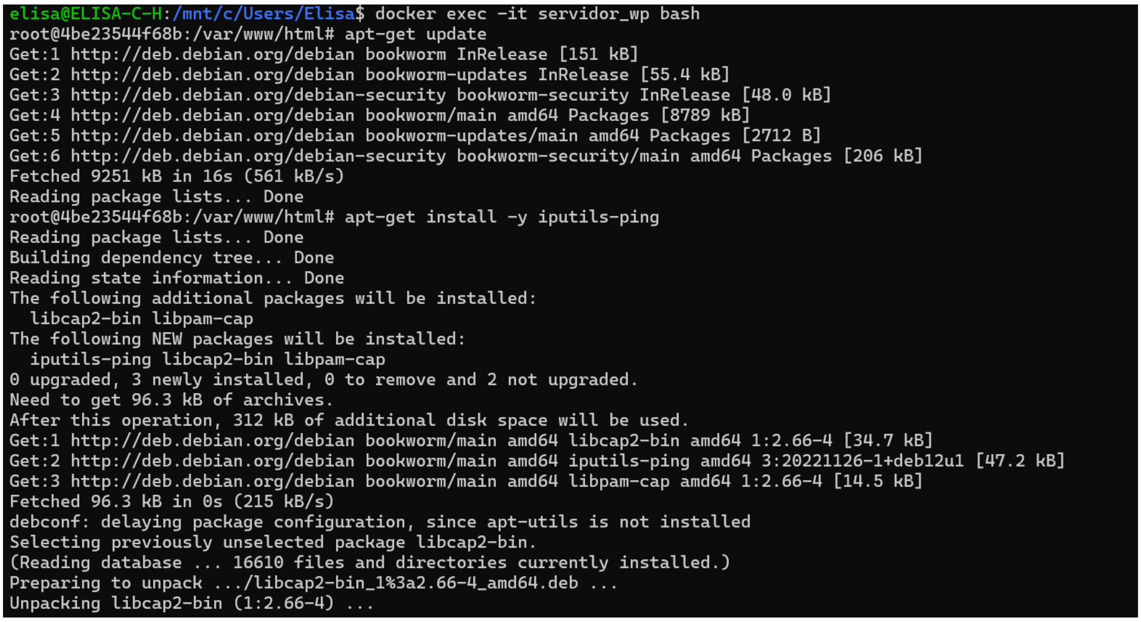
```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wp bash
root@4be23544f68b:/var/www/html# cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * This has been slightly modified (to read environment variables) for use in Docker.
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// IMPORTANT: this file needs to stay in-sync with https://github.com/WordPress/WordPress/blob/master/wp-config-sample.p
hp
// (it gets parsed by the upstream wizard in https://github.com/WordPress/WordPress/blob/f27cb65e1ef25d11b535695a660e728
2b98eb742/wp-admin/setup-config.php#L356-L392)

// a helper function to lookup "env_FILE", "env", then fallback
```

Instalar iputils-ping

Dentro del contenedor servidor_wp hacer

$ apt-get update

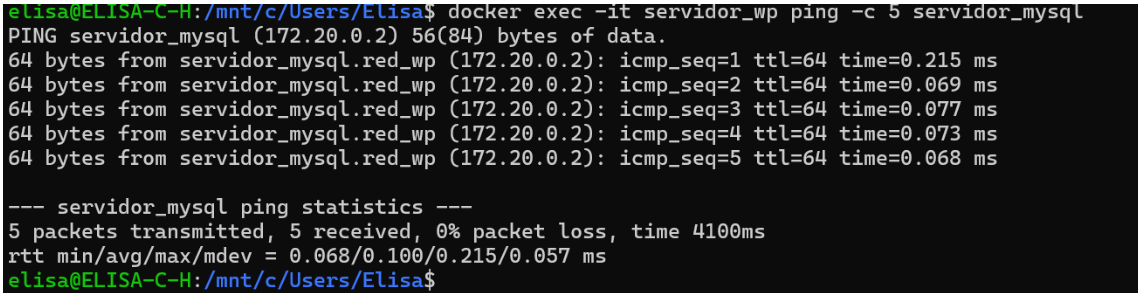
$ apt-get install -y iputils-ping



```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wp bash
root@4be23544f68b:/var/www/html# apt-get update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8789 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [2712 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [206 kB]
Fetched 9251 kB in 16s (561 kB/s)
Reading package lists... Done
root@4be23544f68b:/var/www/html# apt-get install -y iputils-ping
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcap2-bin libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2-bin libpam-cap
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 96.3 kB of archives.
After this operation, 312 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libcap2-bin amd64 1:2.66-4 [34.7 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 iputils-ping amd64 3:20221126-1+deb12u1 [47.2 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libpam-cap amd64 1:2.66-4 [14.5 kB]
Fetched 96.3 kB in 0s (215 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libcap2-bin.
(Reading database ... 16610 files and directories currently installed.)
Preparing to unpack .../libcap2-bin_1%3a2.66-4_amd64.deb ...
Unpacking libcap2-bin (1:2.66-4) ...
```

$ docker exec -it servidor_wp ping -c 4 servidor_mysql

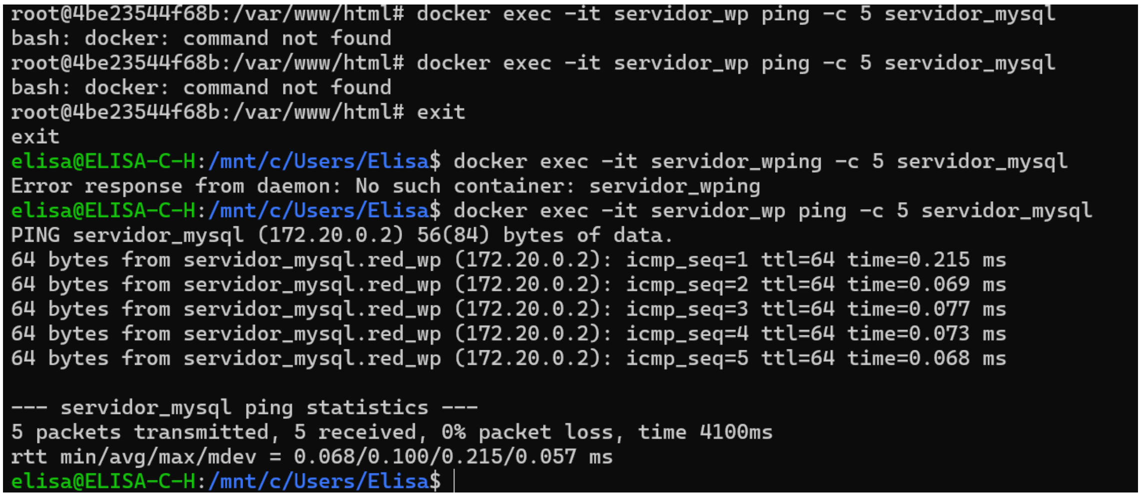
para 5 pings



```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wp ping -c 5 servidor_mysql
PING servidor_mysql (172.20.0.2) 56(84) bytes of data.
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=1 ttl=64 time=0.215 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=5 ttl=64 time=0.068 ms

--- servidor_mysql ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.068/0.100/0.215/0.057 ms
elisa@ELISA-C-H:/mnt/c/Users/Elisa$
```

Troubleshooting:

salirse de root

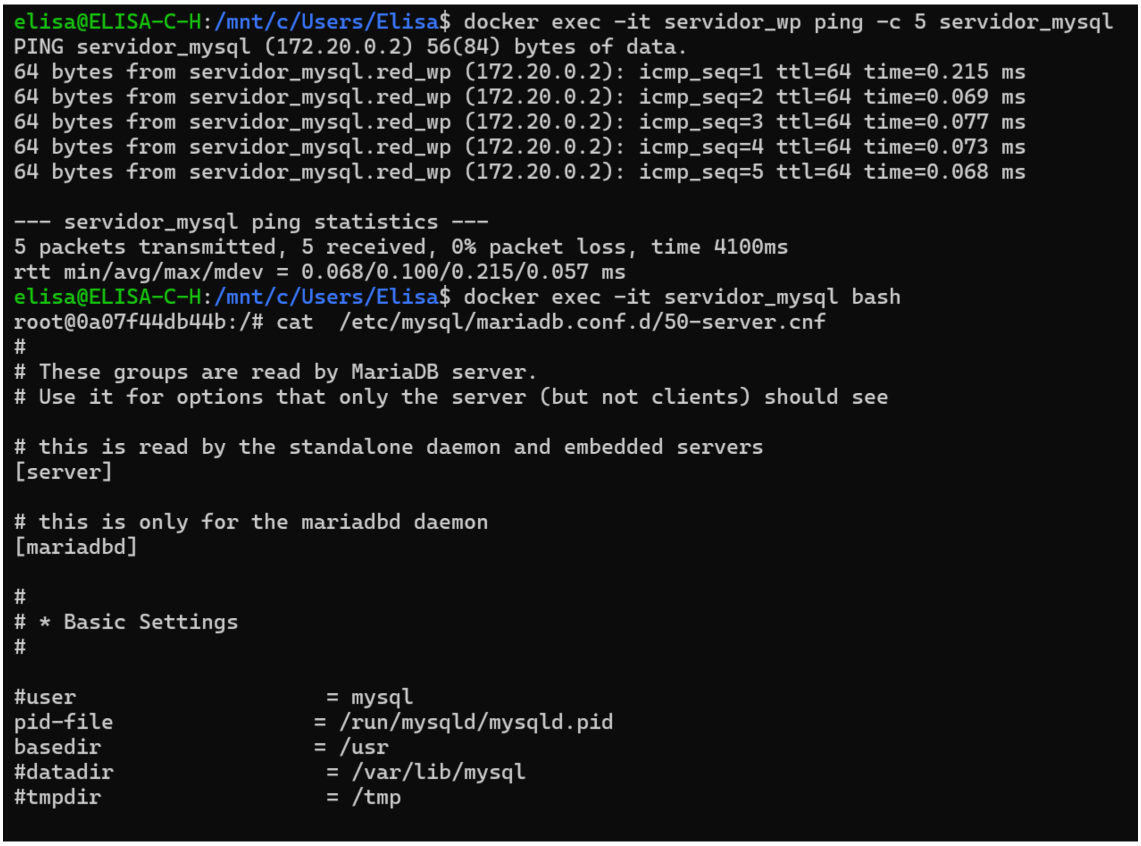


```
root@4be23544f68b:/var/www/html# docker exec -it servidor_wp ping -c 5 servidor_mysql
bash: docker: command not found
root@4be23544f68b:/var/www/html# docker exec -it servidor_wp ping -c 5 servidor_mysql
bash: docker: command not found
root@4be23544f68b:/var/www/html# exit
exit
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wping -c 5 servidor_mysql
Error response from daemon: No such container: servidor_wping
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wp ping -c 5 servidor_mysql
PING servidor_mysql (172.20.0.2) 56(84) bytes of data.
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=1 ttl=64 time=0.215 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=5 ttl=64 time=0.068 ms

--- servidor_mysql ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.068/0.100/0.215/0.057 ms
elisa@ELISA-C-H:/mnt/c/Users/Elisa$
```

- Visualiza el fichero `/etc/mysql/mariadb.conf.d/50-server.cnf` del contenedor con la base de datos y comprueba cómo está configurado el parámetro `bind-address`.

docker exec -it servidor_mysql bash

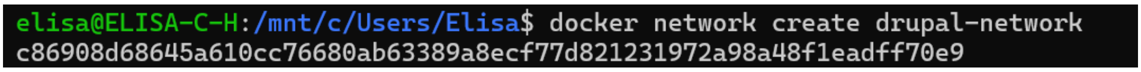
cat /etc/mysql/mariadb.conf.d/50-server.cnf



```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_wp ping -c 5 servidor_mysql
PING servidor_mysql (172.20.0.2) 56(84) bytes of data.
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=1 ttl=64 time=0.215 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=3 ttl=64 time=0.077 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from servidor_mysql.red_wp (172.20.0.2): icmp_seq=5 ttl=64 time=0.068 ms

--- servidor_mysql ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.068/0.100/0.215/0.057 ms
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker exec -it servidor_mysql bash
root@0a07f44db44b:/# cat  /etc/mysql/mariadb.conf.d/50-server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mariadbd daemon
[mariadbd]

#
# * Basic Settings
#

#user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
basedir                 = /usr
#datadir                 = /var/lib/mysql
#tmpdir                  = /tmp
```

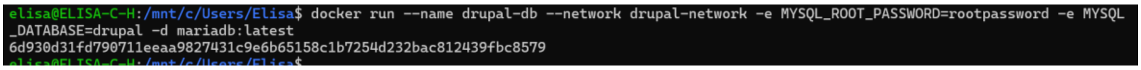
- Instala Drupal (CMS PHP) siguiendo la documentación de Docker Hub de la aplicación seleccionada.

$ docker network create drupal-network



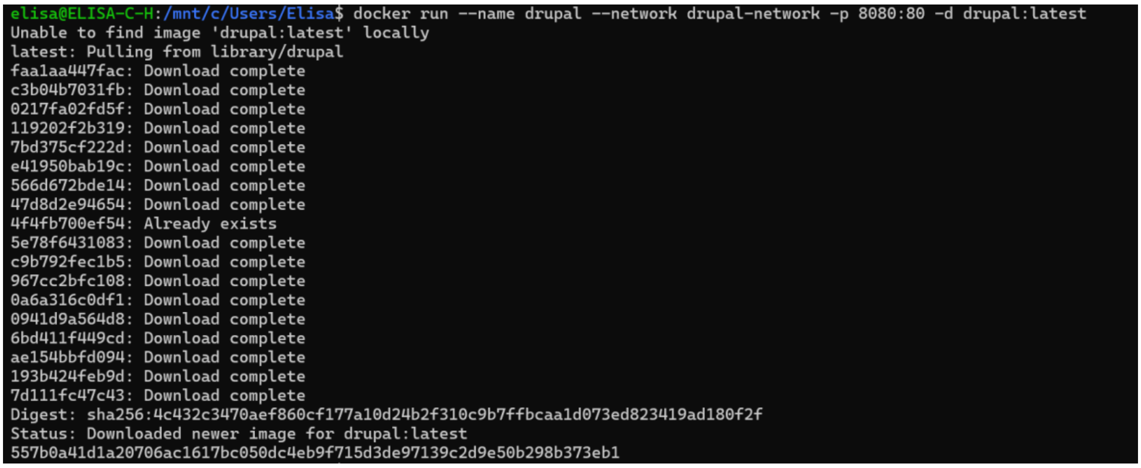
```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker network create drupal-network
c86908d68645a610cc76680ab63389a8ecf77d821231972a98a48f1eadff70e9
```

$ docker run --name drupal-db --network drupal-network -e MYSQL_ROOT_PASSWORD=rootpassword -e MYSQL_DATABASE=drupal -d mariadb:latest



```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker run --name drupal-db --network drupal-network -e MYSQL_ROOT_PASSWORD=rootpassword -e MYSQL
_DATABASE=drupal -d mariadb:latest
6d930d31fd790711eeaa9827431c9e6b65158c1b7254d232bac812439fbc8579
elisa@ELISA-C-H:/mnt/c/Users/Elisa$
```

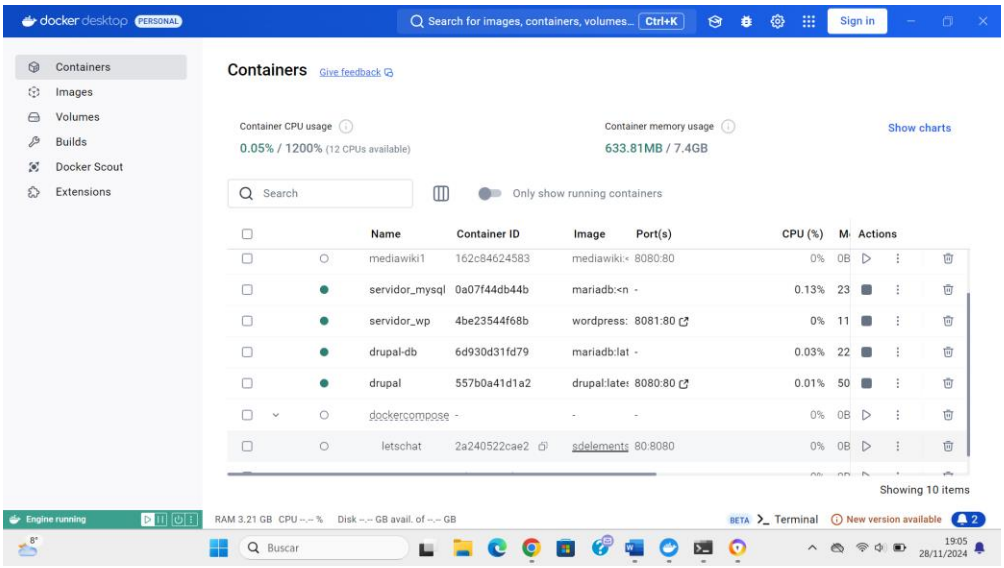
docker run --name drupal --network drupal-network -p 8080:80 -d drupal:latest



```
elisa@ELISA-C-H:/mnt/c/Users/Elisa$ docker run --name drupal --network drupal-network -p 8080:80 -d drupal:latest
Unable to find image 'drupal:latest' locally
latest: Pulling from library/drupal
faa1aa447fac: Download complete
c3b04b7031fb: Download complete
0217fa02fd5f: Download complete
119202f2b319: Download complete
7bd375cf222d: Download complete
e41950bab19c: Download complete
566d672bde14: Download complete
47d8d2e94654: Download complete
4f4fb700ef54: Already exists
5e78f6431083: Download complete
c9b792fec1b5: Download complete
967cc2bfc108: Download complete
0a6a316c0df1: Download complete
0941d9a564d8: Download complete
6bd411f449cd: Download complete
ae154bbfd094: Download complete
193b424feb9d: Download complete
7d111fc47c43: Download complete
Digest: sha256:4c432c3470aef860cf177a10d24b2f310c9b7ffbcaa1d073ed823419ad180f2f
Status: Downloaded newer image for drupal:latest
557b0a41d1a20706ac1617bc050dc4eb9f715d3de97139c2d9e50b298b373eb1
```

Si presenta un error por la puerta, poner 8081:80

Página de los contenedores en DockerDesktop



Containers Give feedback

Container CPU usage
0.05% / 1200% (12 CPUs available)

Container memory usage
633.81MB / 7.4GB

Show charts

Search

Only show running containers

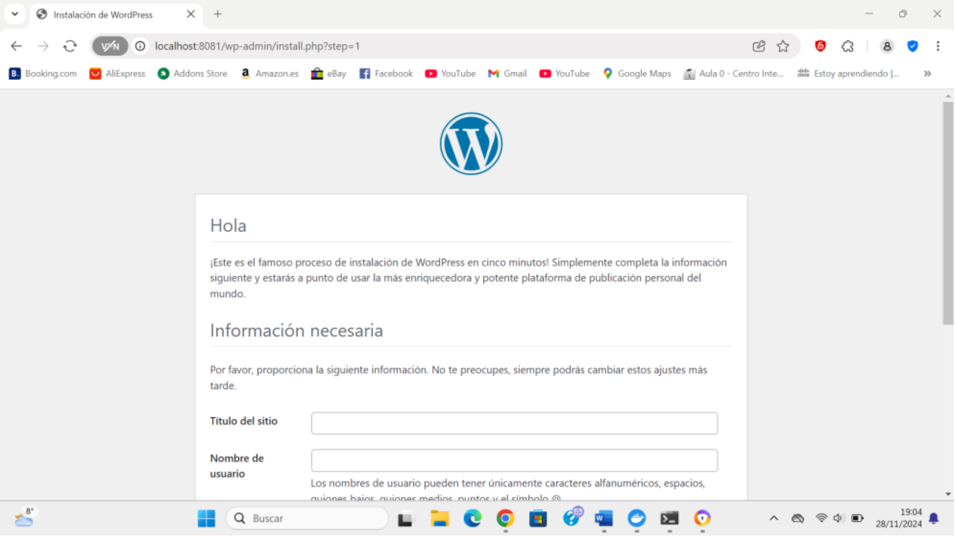
		Name	Container ID	Image	Port(s)	CPU (%)	M	Actions
□	○	mediawiki1	162c84624583	mediawiki:‹	8080:80	0%	0B	▷ ⋮ 🗑
□	●	servidor_mysql	0a07f44db44b	mariadb:‹n	-	0.13%	23	■ ⋮ 🗑
□	●	servidor_wp	4be23544f68b	wordpress:	8081:80	0%	11	■ ⋮ 🗑
□	●	drupal-db	6d930d31fd79	mariadb:lat	-	0.03%	22	■ ⋮ 🗑
□	●	drupal	557b0a41d1a2	drupal:lates	8080:80	0.01%	50	■ ⋮ 🗑
□ ⌄	○	dockercompose	-	-	-	0%	0B	▷ ⋮ 🗑
□	○	letschat	2a240522cae2	sdelements	80:8080	0%	0B	▷ ⋮ 🗑

Showing 10 items

RAM 3.21 GB CPU --.-- % Disk --.-- GB avail. of --.-- GB

BETA Terminal New version available 2

Página web de WordPress



localhost:8081/wp-admin/install.php?step=1

Hola

¡Este es el famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

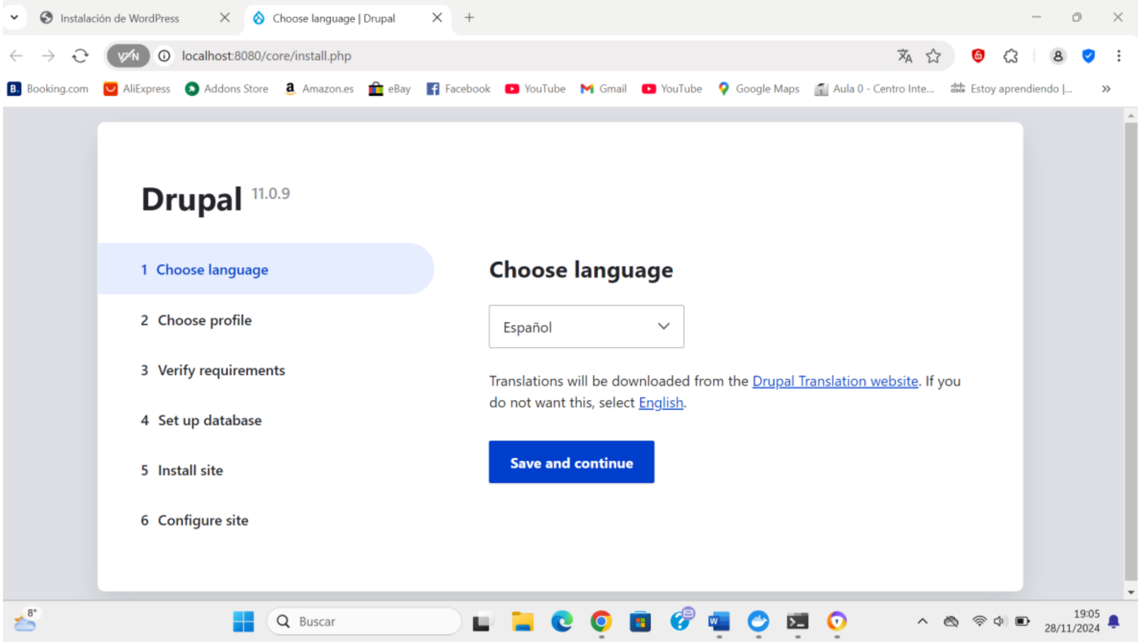
Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Página web de drupal.



Drupal 11.0.9

1 Choose language

2 Choose profile

3 Verify requirements

4 Set up database

5 Install site

6 Configure site

Choose language

Español

Translations will be downloaded from the Drupal Translation website. If you do not want this, select English.

Save and continue