

Introducció a Docker

Part 03. Cas pràctic 02 - Instal·lant LAMP + Wordpress en un contenedor



Autor: Sergi García Barea

Actualitzat setembre 2021

Llicència




Reconeixement - No comercial - CompartirIgual (BY-NC-SA): No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original.

Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per distingir elements importants dins del contingut. Aquests símbols són:

 **Important**

 **atenció**

 **Interessant**

Introducció	3
Preparant el contenidor	3
Instal·lant LAMP i Wordpress	3
Actualitzant repositori i instal·lant LAMP + Wordpress	3
Preparant Apache per a utilitzar Wordpress	4
Preparant MySQL Server	4
Configurant Wordpress	5
Configurant el nostre lloc Wordpress des del navegador	5
Configurant els serveis perquè arrenquen en iniciar el contenidor	5
Comprovar que tot ha funcionat correctament	6
Bibliografia	7

PART 03. CAS PRÀCTIC 02

1. INTRODUCCIÓ

En aquest cas pràctic, utilitzarem com a base de contenidor la imatge oficial de **"ubuntu"**. En aquest contenidor instal·larem Apache, PHP i MySQL (LAMP), al costat del CMS Wordpress.

2. PREPARANT EL CONTENIDOR

Crearem un contenidor amb la imatge base **"ubuntu"**, a la que anomenarem LAMP i en el qual exposarem el seu lloc 80 dins el port 8080 del nostre sistema. A més en crear-ho, deixarem a punt una "shell" per instal·lar els programes pertinents.

Això podem fer-ho amb l'ordre:

```
docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
```

Si tot surt bé ens trobarem a la "shell" del contenidor, de manera similar a:

```
sergi@ubuntu:~$ docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5d3b2c2d21bb: Pull complete
3fc2062ea667: Pull complete
75adf526d75b: Pull complete
Digest: sha256:b4f9e18267eb98998f6130342baacaeb9553f136142d40959a1b46d6401f0f2b
Status: Downloaded newer image for ubuntu:latest
root@904f3a9fd3ff:/#
```

3. INSTAL·LANT LAMP I WORDPRESS

A continuació, instal·larem tot el necessari per a disposar de Wordpress (incloent-hi LAMP) seguint les instruccions d'instal·lació en Ubuntu, disponibles a:

<https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview>

A continuació, detallem els passos a seguir.

3.1 Actualitzant repositori i instal·lant LAMP + Wordpress

En primer lloc, actualitzem la llista de paquets de repositori:

```
apt update
```

Després d'això, vam instal·lar els paquets necessaris per instal·lar LAMP + Wordpress

```
apt install wordpress php libapache2-mod-php mysql-server php-mysql
```

Després d'això podem llançar el servei Apache amb la comanda:

```
service apache2 start
```

Amb això ja tenim tot el programari instal·lat. Podem fer ja una xicoteta prova connectant des de la nostra màquina <http://localhost:8080>

3.2 Preparant Apache per a utilitzar Wordpress

Primer de tot, podem instal·lar un editor de text en mode consola que siga del vostre gust. Per exemple amb aquesta comanda podem instal·lar "nano":

```
apt install nano
```

Un cop instal·lat el vostre editor favorit, procedim a editar la configuració d'Apache per treballar amb Wordpress. Haurem creat el fitxer de configuració del lloc a Apache ***"/etc/apache2/sites-available/wordpress.conf"*** que configurarà l'accés al lloc Wordpress. Aquest fitxer ha de dur el contingut:

```
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

Creat el fitxer, haurem de carregar el lloc, habilitar "URL rewriting" i recarregar Apache amb els següents comandaments:

```
a2ensite wordpress
```

```
a2enmod rewrite
```

```
service apache2 reload
```

Amb això ja podem provar amb l'URL <http://localhost:8080/bloc> tot i que veurem un error perquè encara no hem configurat el fitxer ***"/etc/wordpress/config-localhost.php"***.

3.3 Preparant MySQL Server

Un cop tot instal·lat, hem de preparar el servei MySQL. En primer lloc, haurem de posar en marxa el servei amb la comanda

```
service mysql start
```

Després d'això, haurem d'executar la comanda per generar un password de root de MySQL Server de forma segura (haurem de recordar-ho) i altres opcions. El comandament a executar és:

```
mysql_secure_installation
```

Més informació a <https://dev.mysql.com/doc/refman/5.7/en/mysql-secure-installation.html>

Després d'això, accedirem a la base de dades amb el client MySQL de la següent manera

```
mysql -u root -p
```

Després d'indicar la contrasenya de "root", podrem escriure ordres per a MySQL. Escriurem les següents comandes. En primer lloc vam crear la base de dades "wordpress":

```
CREATE DATABASE wordpress;
```

Després d'ells, vam crear l'usuari "wordpress" (amb contrasenya "MiPass-2021") i li donem permisos totals a la base de dades "wordpress".

```
CREATE USER 'wordpress'@'%' IDENTIFIED BY 'MiPass-2021';
```

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH GRANT OPTION;
```

Finalment, propaguem els privilegis establerts perquè ja estiguen operatius al servidor.

```
FLUSH PRIVILEGES;
```

3.4 Configurant Wordpress

Usant un editor de text de consola, editarem el fitxer de configuració de Wordpress ***"/etc/wordpress/config-localhost.php"***, quedant de la manera que ara indiquem:

```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'MiPass-2021');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

3.5 Configurant el nostre lloc Wordpress des del navegador

Si tot ha anat bé, accedint a l'URL <http://localhost:8080/bloc> podrem configurar el nostre lloc Wordpress, indicant nom de lloc, usuari i email.

3.6 Configurant els serveis perquè arrenquen en iniciar el contenidor

En aquest punt anem a explicar com aconseguir que el nostre contenidor Docker pugui llançar els serveis en iniciar-se. Generalment, pel baix cost de posar en marxa un contenidor, els sistemes Docker estan pensats per executar mico-serveis.

Els contenidors multiservei són l'excepció (aquest cas pràctic ha estat presentat amb una fi didàctica i no d'ús real). Més endavant en el curs, veurem com realitzar aquesta mateixa configuració utilitzant serveis allotjats en diferents contenidors.

En situar-nos en un cas de contenidor multiservei, seguim les premisses que ens indica la documentació de Docker https://docs.docker.com/config/containers/multi-service_container/ que proposa que si necessitem llançar diversos serveis, llancem com a comando l'execució d'un script que llença els serveis que necessitem.

Com en aquest cas pràctic el nostre contenidor està configurat per executar en iniciar-se la shell **"bash"**, **podem** editar el fitxer de configuració **".bashrc"** amb la finalitat que en iniciar la shell, es llancen els serveis Apache i MySQL.

Podem fer-ho seguint els següents passos dins el contenidor:

- Accedim a la carpeta del nostre usuari, **"/root"** escrivint **"cd"**, **"cd ~"** o **"cd /root"**.
- Aquí, amb un editor de text modifiquem el fitxer **".bashrc"** i a la fi de la mateixa afegim les línies següents.

```
service apache2 start
service mysql start
```

Extra: Els contenidors de Docker, al contrari que un sistema Linux real, no s'inicien per defecte llançant un procés que llança a altres processos (substituts de l'antic procés "Init"), com systemd <https://es.wikipedia.org/wiki/systemd> o Upstart <https://es.wikipedia.org/wiki/Upstart>. Per la qual cosa encara que ho configurem, al no iniciar-se juntament amb el contenidor, no arrenquen els processos.

Si en algun context (encara que no és el més habitual), voldríem utilitzar un procés d'aquestes característiques o fins i tot llançar systemd o Upstart, es plantegen les següents solucions.

- Instal·lant els sistemes en el nostre contenidor i crear el contenidor Docker en mode privilegiat. **MOLT D'ULL, problemes de seguretat. No recomanat.** Més informació:
 - https://www.trendmicro.com/en_us/research/19/l/why-running-a-privileged-container-in-docker-is-a-bad-idea.html
- **Una alternativa sense contenidors privilegiats**, és utilitzar una característica recentment afegida a Docker que permet llançar el seu propi procés d'inici en un contenidor no privilegiat amb l'opció **"--init"**, del comandament **"docker run"**, i indicar aquí el procés de systemd o Upstart. Si no s'indica res, usará un procés d'inici propi de Docker anomenat **"tini"**. Més informació:
 - <https://developers.redhat.com/blog/2016/09/13/running-systemd-in>
 - <https://docs.docker.com/engine/reference/run/#specify-an-init-process-a-non-privileged-container/>

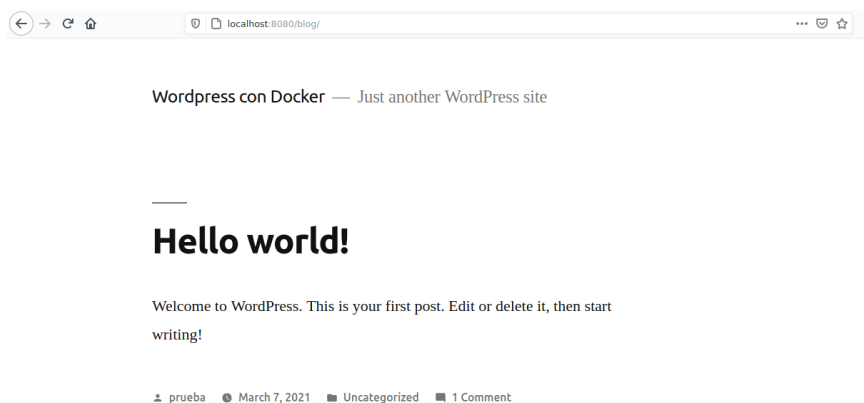
3.7 Comprovar que tot ha funcionat correctament

Finalment per comprovar que tot ha funcionat correctament, provarem de parar al contenidor, posar-lo de nou en marxa i comprovar que podem accedir al nostre Wordpress a <http://localhost:8080/> bloc. Per aturar i llançar el contenidor utilitzarem les ordres:

```
docker stop LAMP
```

```
docker start LAMP
```

Podrem observar alguna cosa similar a:



4. BIBLIOGRAFIA

[1] Install and configure Wordpress on Ubuntu

<https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview>

[2] Create user and grant privileges on MySQL 8

<https://stackoverflow.com/questions/50177216/how-to-grant-all-privileges-to-root-user-in-mysql-8-0>