

Docker en Raspi con Nginx/Drupal

Christian López Rodríguez
Diego Carracedo Conde

¿Quiénes somos?

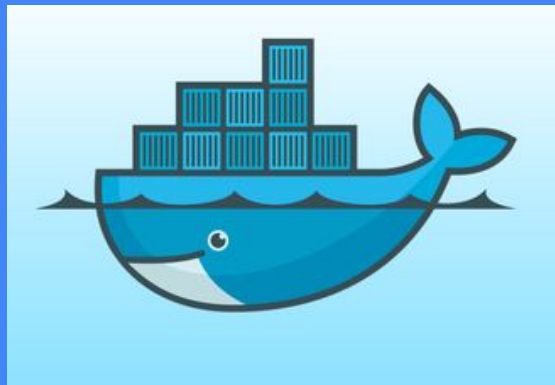
Estudiantes de Ingeniería Informática en la ETSE (USC)

Interesados en la combinación de Electrónica/Hardware/Software

Christian López - clopezrodriguez@hotmail.com

Diego Carracedo - carracedodiego@gmail.com

¿Qué es Docker?



Herramienta que permite una abstracción a la virtualización de una aplicación.

Permite ejecutar cualquier aplicación independiente del SO anfitrión.

Permite la creación de imágenes customizadas y posibilidad de compartir.

¿Qué es Drupal?



Framework basado en PHP.

Orientado a la gestión de contenido dinámico y con bastante tráfico de contenido.

Framework flexible, seguro y con una interfaz de usuario bien formada que hace que la edición sea sencilla.

Difícil de controlar. (twig, yml)

¿Qué es Nginx?

The Nginx logo is displayed in a bold, green, sans-serif font. The letters are slightly stylized, with the 'i' and 'n' having unique shapes. The logo is centered within a black rectangular box, which is itself centered on a blue background.

Servidor web de código abierto.

Similar a Apache pero con una mejor gestión de la concurrencia.

Estructura mejor definida y más fácil de entender que Apache.

Gestión de varios servidores mucho más sencilla.

Instalación Nginx

Drupal basado en php => verificar php

Integración con varias DB. (sqlite, mysql).

Uso de varias librerías: gd, fpm, sqlite/mysql.

Paquete Nginx integrado en los repositorios oficiales.

```
#Actualizamos repositorios y sistema
```

```
sudo apt-get update
```

```
sudo apt-get -y upgrade
```

```
#Instalamos php y librerías importantes
```

```
sudo apt-get -y install php5 php5-fpm php5-cli php5-gd php5-sqlite sqlite
```

```
#Instalamos Nginx
```

```
sudo apt-get -y install nginx nginx-common nginx-full
```

```
#Arrancamos el servicio de Nginx
```

```
sudo /etc/init.d/nginx start
```

Instalación Nginx

Archivo de configuración en /etc/nginx.

Modificaciones intuitivas en archivo.

Cambio de puerto, redirecciones de páginas de error, permitir uso de php.

```
#Definimos el puerto de escucha (también disponible en ipv6)
listen 9999;
#Path hasta el lugar de servidor. Modo de index
root /var/www/html;
index index.php index.html index.htm;

.
|.
#Permitir PHP por medio de fastcgi y fpm
location ~ .php$ {
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
```

Instalación Nginx

Modificación configuración php-fpm.

Archivo en /etc/php5/fpm/php.ini.

Deshabilitar auto-completado de path.

Cambiar parámetro pathinfo=0.

Reiniciar ambos servicios.

```
; this to 1 will cause PHP CGI to fix its paths to conform to the
spec.  A setting
; of zero causes PHP to behave as before.  Default is 1.  You
should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0
```

```
#Reiniciamos servicios de php5-fpm y Nginx
sudo /etc/init.d/nginx restart
sudo /etc/init.d/php5-fpm restart
```


Instalación Drupal

Mediante drush:

<http://docs.drush.org/en/master/install/>
<https://drushcommands.com/>

Mediante drupal console:

<https://drupalconsole.com/>

Desde página oficial: www.drupal.org/download.

```
#Descargamos drupal8  
wget https://ftp.drupal.org/files/projects/drupal-8.2.3.tar.gz
```

```
#Llevamos el tar.gz a su destino final (/var/www/)  
sudo mv drupal-8.2.3.tar.gz /var/www/
```

```
#Descomprimos el tar.gz  
sudo tar -xzf /var/www/drupal-8.2.3.tar.gz
```

```
#Renombramos el html a otro para no perder el original  
sudo mv /var/www/html /var/www/original
```

```
#Renombramos la carpeta a html  
sudo mv /var/www/drupal-8.2.3.tar.gz /var/www/html
```

Instalación Docker

<https://docs.docker.com/engine/>

Añadir nuevos repositorios.

Instalar certificación https.

Descargar e instalar.

```
#Instalamos certificados
sudo apt-get install apt-transport-https ca-certificates

#Añadiremos la clave GPG
sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80
--recv-keys 58118E89F3A912897C070ADB76221572C52609D

#Añadimos en /etc/apt/sources.list.d/docker.list el repo
deb https://apt.dockerproject.org/repo raspbian-jessie main

#Cambiamos las prioridades del repositorio y actualizamos los repos
sudo apt-cache policy docker-engine
sudo apt-get update

#Instalamos docker-engine y actualizamos
sudo apt-get install docker-engine
sudo apt-get upgrade docker-engine
```

Instalación Docker

Arrancar el demonio.

```
#Arrancamos el demonio  
sudo service docker start
```

Recomendable crear usuario nuevo con permisos reducidos para evitar que docker abarque de más.

```
#Creamos el grupo docker y añadimos el usuario actual ahí  
sudo groupadd docker  
sudo gpasswd -a ${USER} docker
```

Reiniciar el demonio.

```
#Reiniciamos el demonio  
sudo service docker restart
```

Y AHORA QUÉ?

Tenemos todo funcionando, pero para qué lo podemos usar?

Por qué Docker y no Ansible?

Tanto rollo para instalar una herramienta de generación de contenedores?

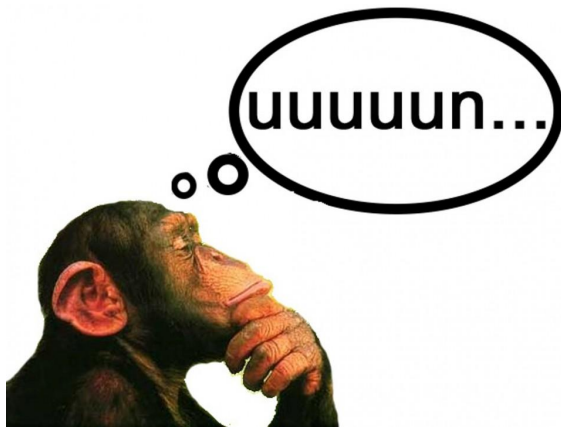


Imagen personalizada Docker

Docker porque permite crear imágenes personalizadas de SOs.

Teniendo como base un Raspbian podemos crear una imagen con Nginx y Drupal por defecto.

Podemos ejecutar esta imagen en cualquier Linux.

Imagen personalizada Docker

La clave es el documento Dockerfile.

Contiene toda la información de la estructura de la nueva imagen.

Definimos la base del SO, los paquetes necesarios, carpetas, etc..

Definimos los **¡¡PUERTOS!!** que queremos habilitar.

Imagen personalizada Docker

Dockerfile muy básico

<https://github.com/reloxo95/dockerNginxDrupal/blob/master/Dockerfile>

```
#Base de arquitectura del sistema
FROM resin/rpi-raspbian

#Quien lo crea/mantiene
MAINTAINER Christian Lopez <clopezrodriguez@hotmail.com>

#Actualizamos los repositorios
RUN sudo apt-get -y update

#Instalamos php y librerias para que funcione con sqlite
RUN sudo apt-get -y install php5 php5-fpm php5-gd php5-cli sqlite php5-sqlite wget

#Instalamos el nginx
RUN sudo apt-get -y install nginx nginx-common nginx-full

#Exponemos el puerto 8080 al exterior
EXPOSE 9999
```

Imagen personalizada Docker

Construir la imagen nueva

`docker build -t <nombre> .`

Indicar nombre de la imagen

No olvidarse del `.`

```
pi@raspi:~/dockerNginxDrupal $ docker build -t drupalbasico/reloxo95 .
Sending build context to Docker daemon 34.14 MB
Step 1 : FROM resin/rpi-raspbian
---> c895a843d727
Step 2 : MAINTAINER Christian Lopez <clopezrodriguez@hotmail.com>
---> Using cache
---> e87ae6e4fdcf
Step 3 : RUN sudo apt-get -y update
```

```
Step 20 : RUN echo "Terminado"
---> Using cache
---> 624b09b81637
Successfully built 624b09b81637
```


Imagen personalizada Docker

Arrancamos la nueva imagen

`docker run -i -p <puertos> <nombre>`

Modo interactivo si/no -i

¡¡MAPEAR PUERTOS!! -p virtual:origen

```
pi@raspi:~/dockerNginxDrupal $ docker run -i -p 9999:9999 drupalbasico/reloxo95
cd /home
ls
cd ..
ls
bin
boot
default
dev
etc
home
lib
media
mnt
opt
```

Imagen personalizada Docker

Tenemos una “máquina virtual” personalizada funcionando en la raspberry.

Arrancar servicio php5-fpm

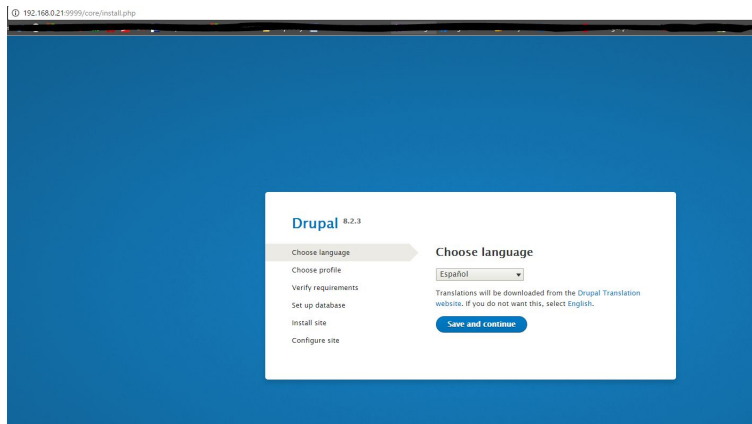
Arrancar servicio de nginx.

Comprobar en localhost.

```
pi@raspi:~/dockerNginxDrupal $ docker run -i -p 9999:9999 drupalbasico/reloxo95  
/etc/init.d/php5-fpm start  
/etc/init.d/nginx start  
█
```

Imagen personalizada Docker

¡HEMOS TERMINADO!



¿Qué hemos hecho?

Hemos testado el funcionamiento de Nginx y Drupal.

Hemos creado una imagen (basada en raspbian) personalizada con Nginx y Drupal preconfigurado.

Hemos arrancado la imagen y hemos arrancado el servidor web.

¿Por qué lo hemos hecho?

La imagen que hemos creado la podemos ejecutar en cualquier sistema Linux.

Se puede ejecutar las veces que queramos.

Muy útil en entornos de testeo y desarrollo (¿qué pasa si..?)

Si queremos modificar la imagen, no se compila de nuevo si no que se modifica solo lo necesario.

Se puede combinar en un “[Docker Swarm](#)” y el rendimiento es mayor.

Gestión de Docker

`docker images` -> muestra las imágenes del sistema

`docker rmi <nombreimagen><nombreimagen>` -> borra varias imágenes del sistema

`docker rm <nombreimagen>` -> borra imagen del sistema

`docker info` -> muestra información de docker y del sistema

Repositorio / Referencias

Repo -> <https://github.com/reloxo95>

<http://jasonwilder.com/blog/2014/10/13/a-simple-way-to-dockerize-applications/>

<https://www.howtoforge.com/tutorial/how-to-install-drupal-with-nginx-and-ssl-on-ubuntu/>

<https://www.digitalocean.com/community/tutorials/docker-explicado-como-crear-contenedores-de-docker-corriendo-en-memcached-es>