



CÓMO LEVANTAR UN SERVICIO WEB CON DOCKER Y DOCKER COMPOSE

escrito por Víctor Rodríguez | abril 17, 2018

Si, como yo, usas Linux en tus desarrollos, te habrás encontrado con que **trabajar con las mismas versiones de software en tu sistema operativo que en producción puede llegar a ser un poco caótico**: dependencias que no puedes cumplir, no dispones de paquetería específica y acabas teniendo que compilar a mano servicios que, en muchas ocasiones, implican horas de cambios y reintento para conseguir que todo funcione medianamente bien.

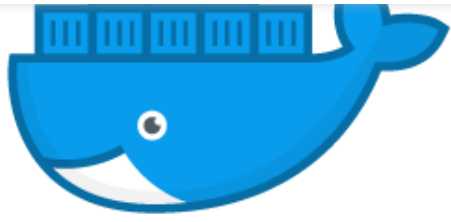
Por suerte, con la llegada de los contenedores y de Docker, muchos de estos problemas pertenecen al pasado. Para que veas lo sencillo que puede ser, preparamos un entorno con Apache, PHP y MySQL que puedes utilizar para probar tu web.

¿Qué es Docker?

Docker es un servicio que **permite ejecutar pequeños contenedores de software sobre el núcleo de nuestro sistema operativo**, por lo que es una opción más rápida y liviana que el uso de máquinas virtuales.

Por regla general, se suele empaquetar un solo servicio por contenedor, por lo que, para nuestro ejemplo, vamos a necesitar un orquestador, es decir, una herramienta que nos permita manejar conjuntamente varios contenedores relacionados entre sí. De ahí el uso de **Docker Compose**.





docker

Hub. Puedes distinguir fácilmente estas imágenes, p
que están marcadas como *official*.

¡A jugar!

Hasta aquí la parte «aburrida». Toca probar el juego
nuevo 😊

Vaya... No lo tengo instalado

Como ya habrás deducido, lo primero que necesitas es tener instalados Docker y Docker Compo
máquina. En muchas distribuciones Linux ya está incluido como parte de la paquetería disponibl
que podrás instalarlo desde tu herramienta de gestión de paquetería preferida. Pero, **si prefiere
a la última o no tienes disponible el paquete, siempre puedes seguir las instrucciones de la
documentación:**

- [Instrucciones para instalar Docker.](#)
- [Instrucciones para instalar Docker Compose.](#)

Si usas Linux, mi consejo es que, a mayores, sigas los pasos de instalación adicionales. Te ayudar
ejemplo, a no depender de ser root, o de utilizar sudo, incluyendo a tu usuario en el grupo de Dc

Ahora que ya tengo Docker, ¿cómo lo configuro?

Para poder levantar los servicios que necesitas, tienes que indicarle a Docker Compose de algun
lo siguiente:

- Contenedores a crear/levantar
- Relación de los contenedores entre sí

Para ello creamos un fichero **YAML**, llamado «docker-compose.yml», donde definimos cada uno c
servicios (puedes ver el fichero completo al final de esta sección).

Comienza tu fichero de configuración indicando la versión de sintaxis a utilizar y abriendo la secc
servicios:

```
1 version: "3"  
2  
3 services:
```

Configuración para MySQL



de acceso. Actualmente, la última versión estable de MySQL es la 5.7, por lo que las líneas a incluir en el fichero son:

```
4  miservicio_mysql:
5    image: mysql:5.7
6    environment:
7      - MYSQL_DATABASE=nombre
8      - MYSQL_ROOT_PASSWORD=claveroot
9      - MYSQL_USER=miusuario
10     - MYSQL_PASSWORD=mipassword
11    volumes:
12      # Montamos un volumen para MySQL para no perder los datos de bd
13      - ./volumenes/mysql:/var/lib/mysql
14    expose:
15      - 3306
16
```

Una pequeña mejora. Si quieres acceder fácilmente al servicio MySQL desde tu aplicación favor puedes hacer que escuche en el puerto 3306 local, o en el que tú prefieras, solamente con añadir:

```
16  ports:
17    - 3306:3306
```

Configuración para el servidor web

Aquí, por ser la opción más sencilla, usamos el contenedor oficial de **PHP** con Apache ya integrado aunque también podríamos combinar un **contenedor oficial de Apache** con la versión FPM de ese contenedor de PHP.

Como puedes ver en la [documentación del contenedor](#), puedes elegir entre varias versiones:

- **php: 7-Apache.** Última versión de PHP 7, actualmente la 7.2
- **php: 7.2-Apache.** Última versión estable de PHP 7.2
- **php: 7.1-Apache.** Última versión estable de PHP 7.1
- **php: 7.0-Apache.** Última versión estable de PHP 7.0
- **php: 5.6-Apache.** Última versión estable de PHP 5.6

Para este ejemplo me he decantado por usar la versión más actual de PHP 7. El servicio se configura así:

```
19  miservicio_php:
20    image: php:7-apache
21    volumes:
22      # Montamos nuestra web desde fuera en el directorio web del contenedor
23      - ./miweb:/var/www/html
24    expose:
25      - 80
26    ports:
```



Con esto queda listo el fichero de configuración, así que ya podrías levantar el fichero. Por prefieres copiar y pegar, aquí tienes la configuración completa de este ejemplo:

```
1 version: "3"
2
3 services:
4   miservicio_mysql:
5     image: mysql:5.7
6     environment:
7       - MYSQL_DATABASE=nombre
8       - MYSQL_ROOT_PASSWORD=claveroot
9       - MYSQL_USER=miusuario
10      - MYSQL_PASSWORD=mipassword
11     volumes:
12       # Montamos un volumen para MySQL para no perder los datos de bd
13       - ./volumenes/mysql:/var/lib/mysql
14     expose:
15       - 3306
16     ports:
17       - 3306:3306
18
19   miservicio_php:
20     image: php:7-apache
21     volumes:
22       # Montamos nuestra web desde fuera en el directorio web del contenedo
23       - ./miweb/:/var/www/html
24     expose:
25       - 80
26     ports:
27       - 80:80
28     links:
29       - miservicio_mysql
```

Añadiendo tu código y levantando Docker

Ya solo falta que incluyas el código en el directorio «miweb», o el que hayas indicado en tu configuración, y ya podrás iniciar todos los servicios.

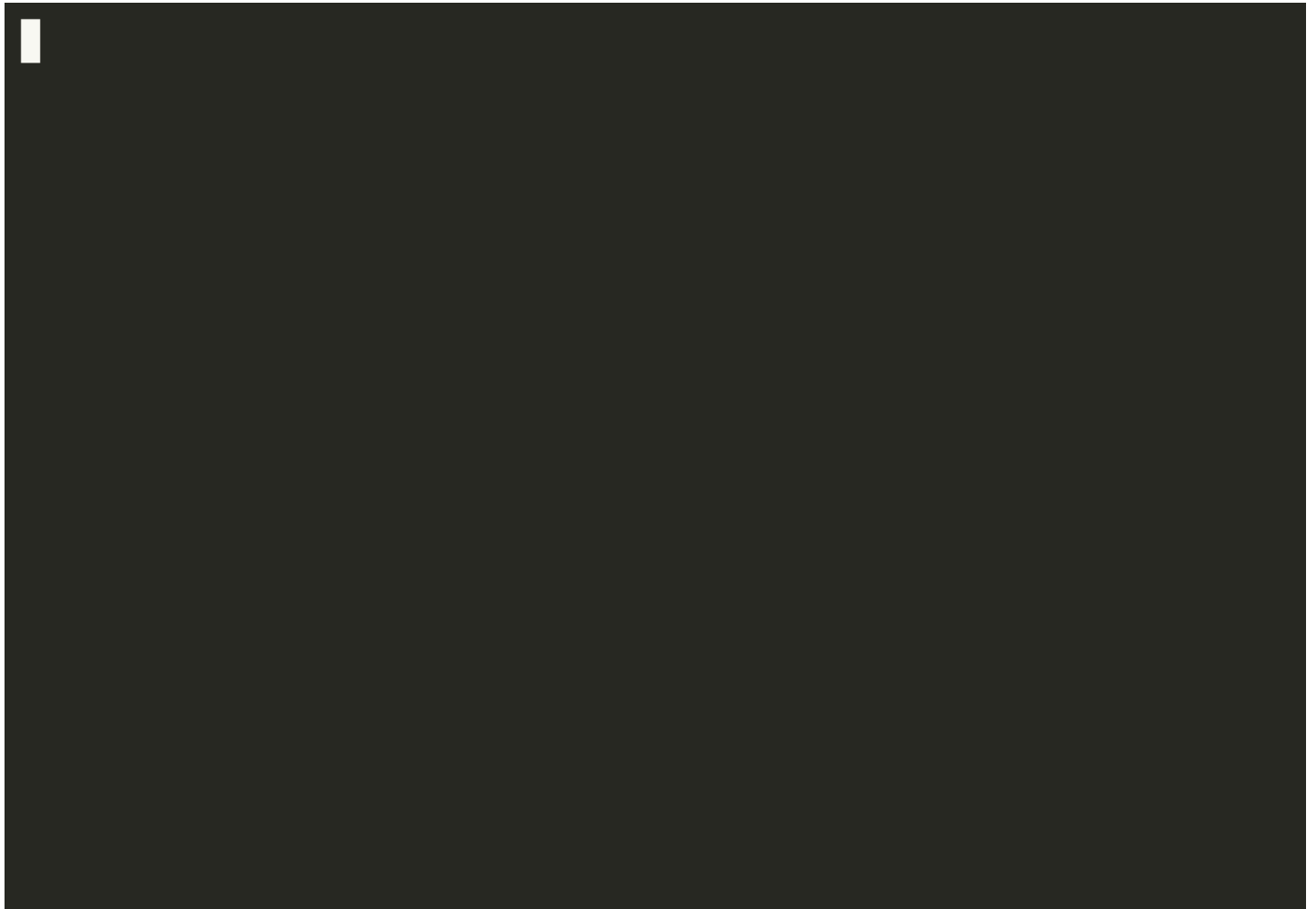
¡Un detalle! Asegúrate de que no tienes nada en tu máquina que esté utilizando los puertos que definido para nuestros servicios (80 y 3306), pues de lo contrario no podrás iniciar los contenedo

Lanzando contenedores en background

Esta es la opción más cómoda a la hora de no cerrar todo accidentalmente ya que, al quedar lan: los contenedores en segundo plano, seguirán funcionando aunque cerremos la terminal. A camk ver los logs tendrás que usar comandos de Docker para cada contenedor por separado. Para inic el contenedor, solamente tendrás que situarte en la misma carpeta donde está el fichero docker compose.yml y ejecutar:



1 | `docker-compose down`



Lanzando contenedores en primer plano

Si, por el contrario, prefieres ver los logs directamente en pantalla y poder parar los contenedores solamente con pulsar Ctrl+C, inicia el servicio en primer plano, ejecutando el de inicio, pero sin «-

1 | `docker-compose up`

¡Y listo! Ya tienes funcionando un servicio web, con PHP 7.2 y MySQL 5.7, con el código de tu web máquina. Solamente tendrás que acceder a <http://localhost/>.

Recuerda que este es un tutorial muy sencillo, con lo básico para arrancar los servicios y que fun pero **los límites los pones tú:**

- Añade extensiones al php y Apache, creando tu propio contenedor
- Cambia rutas, configuraciones, variables de entorno...
- Juega con el mapeo de puertos, por ejemplo:



- Cambia el puerto del apache para mantener levantado tu servicio local
- ...

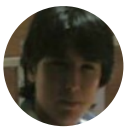
¡Esperamos tus comentarios! Si te ha resultado útil, si quieres que profundicemos más en el tema. Recuerda que también puedes [contactarnos](#) si tienes cualquier duda o sugerencia, ¡que para eso estamos! 😊

💬 8 comentarios

2 ❤️

f 🐦

8 COMENTARIOS



TRUCADOR

🕒 agosto 20, 2018 - 9:28 am

Que terminal es la que usas ? cmdr???

Gracias!

muy buen post!



VÍCTOR RODRÍGUEZ

🕒 agosto 21, 2018 - 7:43 am

Hola!

Me alegra que el post te haya gustado y espero que te resulte útil.

Sobre la terminal, utilizo la que viene por defecto en Ubuntu, pero he añadido [git-prompt](#) para que me muestre datos sobre el repositorio de código, con algún cambio en los temas que trae por defecto. Usando [asciinema](#) he capturado todo que pasaba por la terminal y generado un gif para el post.

Un saludo! 😊



DIEGO

🕒 octubre 21, 2018 - 6:47 pm

como harias para añadir phpmyadmin porque tengo un problema con ello y no consigo
enlace con la base de datos





🕒 octubre 22, 2018 - 11:18 am

Hola Diego!

La forma más sencilla es incluir el contenedor oficial de phpMyAdmin como un componente más, escuchando en el puerto 8080 de tu máquina y enlazado al contenedor de MySQL para que pueda acceder la base de datos. De esta forma podrás acceder al phpMyAdmin con la url <http://localhost:8080/>.

Te dejo un ejemplo en este [fragmento de código](#).

Espero que esto te sirva de ayuda y no dudes en contactar de nuevo con nosotros si lo necesitas 😊



GALLBERS

🕒 enero 11, 2019 - 1:43 am

Hola Víctor!

Muy bueno post! estuve intentando levantar el servidor pero me esta dando problemas con los permisos en el volumen «miweb» me podrías dar algún consejo de que podría hacer?



VÍCTOR RODRÍGUEZ

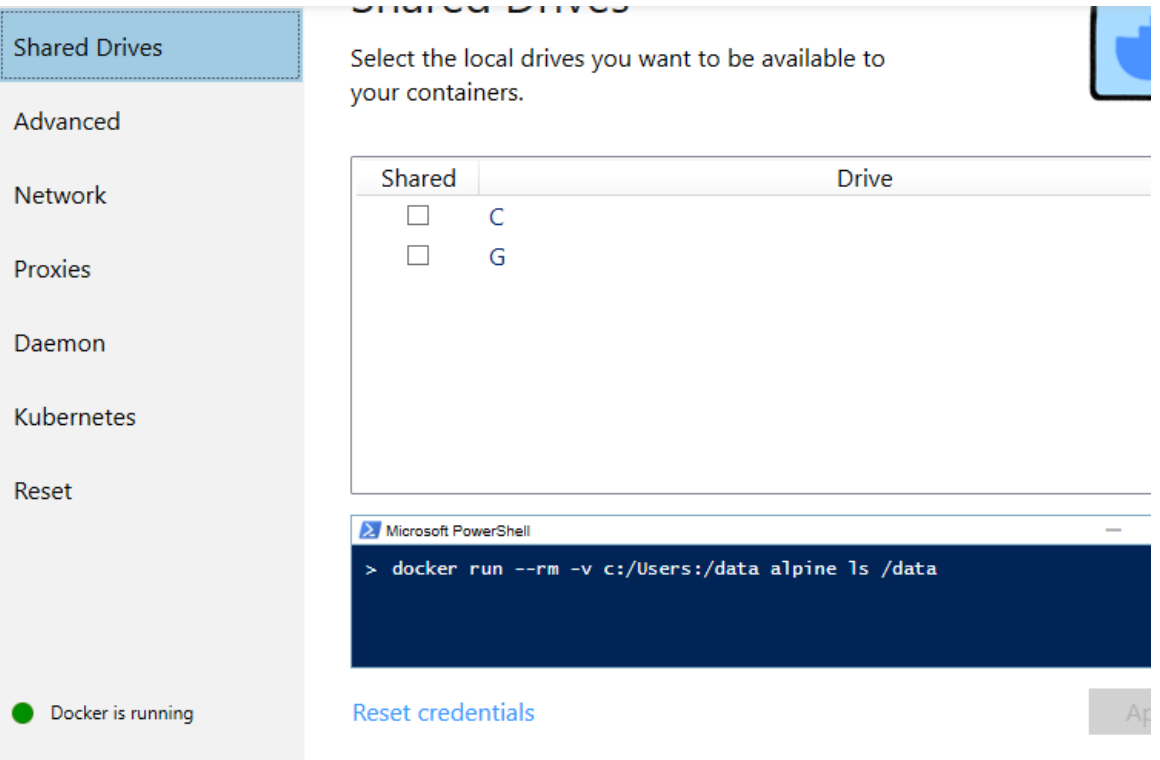
🕒 enero 14, 2019 - 8:35 am

Hola gallbers!

Me alegra que te guste la entrada 😊 Sobre tu consulta:

Si utilizas Linux, no debería ser necesario configurar nada para montar directamente tu máquina en los contenedores, solamente debes recordar lanzar los comandos como root (vale con sudo) o añadir tu usuario al grupo «docker» para no necesitar sudo. En el caso de Windows 10, con Docker for Windows, es posible que tengas que habilitar el uso de las unidades de disco de tu PC dentro de la configuración de Docker. Te dejo una captura:





Shared	Drive
<input type="checkbox"/>	C
<input type="checkbox"/>	G

```
> docker run --rm -v c:/Users:/data alpine ls /data
```

En algunos casos, en Docker for Windows, si la unidad ya estaba compartida, la solución suele ser eliminarla de compartidos y volverla a compartir.

De todas formas, si esto no te soluciona el problema, o utilizas otro sistema operativo, no dudes en contactar de nuevo y estaremos encantados de ayudarte. Un saludo!

**YOLANDA**

🕒 julio 30, 2019 - 11:02 pm

Hola Víctor me gusto muchoo este post gracias.

Sabrías como poner una IP al contenedor para que no tenga que escribir «localhost», una IP propia del contenedor?

**VÍCTOR RODRÍGUEZ**

🕒 julio 31, 2019 - 10:50 am

Hola Yolanda!

Muchas gracias por tu comentario, siempre es agradable saber que a la gente gusta lo que publicamos 😊

Sobre tu pregunta, tienes varias opciones a la hora de acceder por IP:



- Obtener, mediante comandos de Docker, la IP que Docker ha asignado al contenedor
- Definir una red la configuración de Docker Compose y asignar direcciones d red a los contenedores

Te explico las dos últimas con un poco más de detalle:

Usar la IP asignada por Docker al contenedor

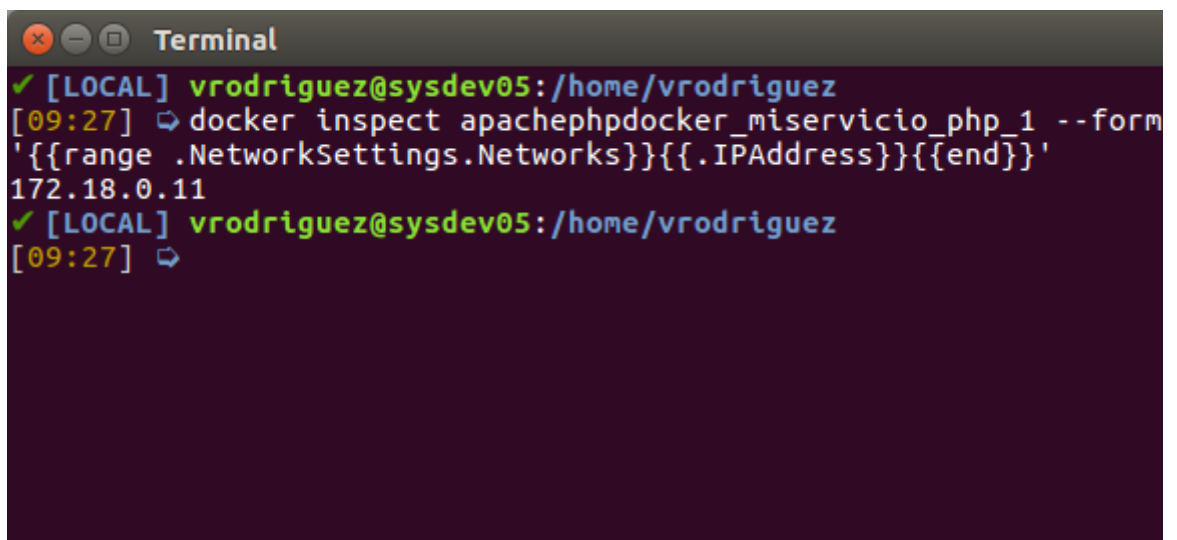
De las dos opciones que voy a explicar esta es la más sencilla, aunque tiene el inconveniente de que la dirección IP puede cambiar al detener y volver a llevar contenedores.

El comando *docker inspect* nos da todos los detalles sobre un contenedor, pero podemos limitar su respuesta especificando un formato para la misma, así ha nos devuelva solamente el dato que nos interesa (por ejemplo, la dirección IP)

En mi caso, mi contenedor se llama «*apachephpdocker_miservicio_php_1*» (para obtener un listado de los contenedores activos ejecutamos el comando «*docker* por lo que ejecuto el comando:

```
docker inspect apachephpdocker_miservicio_php_1 --format '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
```

Con esto me devuelve el valor de la IP del contenedor:



```
Terminal
✓ [LOCAL] vrodriguez@sysdev05:/home/vrodriguez
[09:27] ➔ docker inspect apachephpdocker_miservicio_php_1 --form
'{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
172.18.0.11
✓ [LOCAL] vrodriguez@sysdev05:/home/vrodriguez
[09:27] ➔
```

Usando esta IP en nuestro navegador, siempre que Docker y el navegador est misma máquina, podemos acceder al servicio.

Esta es una opción más avanzada que la anterior, las direcciones IP se mantiene pero tenemos que tener cuidado con las redes que definimos para no tener conflictos con las direcciones.

En este caso, definimos una nueva sección del fichero docker-compose.yml para redes, al mismo nivel que la de servicios y la usamos para asignar IPs a los contenedores.

Para no alargar más el contenido del comentario, he actualizado este [fragmento de código](#), de mi anterior comentario, para incluir también un ejemplo con configuraciones de red.

Si tienes cualquier otra consulta, no dudes en contactar de nuevo, estaremos encantados de ayudarte.

Un saludo!

Los comentarios están bloqueados



LLAMADA GRATIS

900 854 000

24/7

EN DINAHOSTING

Hosting

Dominios

Feedback

Protección de datos

Ayuda

@2019 - Dinahosting



