

Sistemas Informáticos: Actividad Docker 2

En esta actividad expandiremos el trabajo realizado en la anterior:

1. Esta vez descargaremos varias imágenes diferentes, las cuales serán mostradas nuevamente utilizando el comando "**docker images**".
2. Crearemos un contenedor "daemon" en una de las imágenes descargadas.
3. Crearemos un fichero de texto en una de las imágenes descargadas y mostraremos el proceso de copiarlo a otra imagen.
4. Accederemos al fichero desde el navegador web.
5. Mostraremos el espacio disponible en el disco duro tanto antes como después de la creación y copia del fichero.

Descarga y muestra de imágenes

Procederemos a descargar las siguientes imágenes: **ubuntu:18.04**, **httpd**, **tomcat:9.0.39-jdk11**, **jenkins/jenkins:lts** y **php:7.4-apache**. Después las mostraremos.

```
osboxes@osboxes:~$ docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
7c457f213c76: Pull complete
Digest: sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
osboxes@osboxes:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
2d429b9e73a6: Already exists
d675ed392a91: Pull complete
4f4fb700ef54: Pull complete
3ed0d9182dde: Pull complete
0062038102c9: Pull complete
334a67c7f78b: Pull complete
Digest: sha256:6bdbdf5ac16ac3d6ef543a693fd5dfafae2428b4b0cdc52a480166603a069136
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
osboxes@osboxes:~$ docker pull tomcat:9.0.39-jdk11
9.0.39-jdk11: Pulling from library/tomcat
e4c3d3e4f7b0: Pull complete
101c41d0463b: Pull complete
8275efcd805f: Pull complete
751620502a7a: Pull complete
a59da3a7d0e7: Pull complete
9c0f1dffe039: Pull complete
576e3c6f47f8: Pull complete
c7e1b6c3ef84: Pull complete
d1b8a428acdc: Pull complete
7251ae448a6d: Pull complete
Digest: sha256:5b17d5de9c75c9da638c28186c19423b610e7eab3b6f6b975bf47383d12ed0a9
Status: Downloaded newer image for tomcat:9.0.39-jdk11
docker.io/library/tomcat:9.0.39-jdk11
```

```

osboxes@osboxes:~$ docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
7d98d813d54f: Pull complete
a36a1f8447e8: Pull complete
f7ee35b739bf: Pull complete
bb3229cfea8c: Pull complete
d6f97f67d9a1: Pull complete
866fdfadf828: Pull complete
28ae6a41194b: Pull complete
9393a6815458: Pull complete
80c15ff4ebe9: Pull complete
4b1bc6ce70dc: Pull complete
1b47cf50f1d7: Pull complete
089de16bf95f: Pull complete
Digest: sha256:7ea4989040ce0840129937b339bf8c8f878c14b08991def312bdf51ca05aa358
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
osboxes@osboxes:~$ docker pull php:7.4-apache
7.4-apache: Pulling from library/php
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Pull complete
fb5a4c8af82f: Pull complete
25f85b498fd5: Pull complete
9b233e420ac7: Pull complete
fe42347c4ecf: Pull complete
d14eb2ed1e17: Pull complete
66d98f73acb6: Pull complete
d2c43c5efbc8: Pull complete
ab590b48ea47: Pull complete
80692ae2d067: Pull complete
05e465aaa99a: Pull complete
Digest: sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905ff63fae3fe2e0ca6d
Status: Downloaded newer image for php:7.4-apache
docker.io/library/php:7.4-apache

```

Los siguientes comandos sirven para descargar, respectivamente, las imágenes mencionadas:

- 1) **docker pull ubuntu:18.04**
- 2) **docker pull httpd**
- 3) **docker pull tomcat:9.0.39-jdk11**
- 4) **docker pull jenkins/jenkins:lts**
- 5) **docker pull php:7.4 - apache**

```

osboxes@osboxes:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jenkins/jenkins     lts                 58550a76f713       3 weeks ago        472MB
nginx               latest             60c8a892f36f       7 weeks ago        192MB
httpd               latest             dad6ca1caf78       4 months ago       148MB
ubuntu              18.04              f9a80a55f492       18 months ago      63.2MB
php                 7.4-apache         20a3732f422b       2 years ago        453MB
tomcat               9.0.39-jdk11       2703bbe9e9d4       4 years ago        648MB

```

- 6) El comando **docker images** mostrará todas las imágenes creadas. En la captura aparecen todas subrayadas en rojo.

Creación de contenedor y fichero

Ahora que las imágenes han sido creadas, procedemos a crear un contenedor asignado a una de ellas y, dentro de este, un fichero de texto, el cual copiaremos a otra de las imágenes. También mostraremos el tamaño del contenedor dentro del disco duro.

```
osboxes@osboxes:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jenkins/jenkins     lts                 58550a76f713       3 weeks ago        472MB
nginx                latest             60c8a892f36f       7 weeks ago        192MB
httpd               latest             dad6ca1caf78       4 months ago        148MB
ubuntu              18.04             f9a80a55f492       18 months ago      63.2MB
php                 7.4-apache         20a3732f422b       2 years ago        453MB
tomcat              9.0.39-jdk11       2703bbe9e9d4       4 years ago        648MB
osboxes@osboxes:~$ docker run -d --name don_guillermo_server -p 8080:80 php:7.4-apache
4e30cd07ed43776ef2e3e793cde6fb47c2be81fe77031eede63d3926d0ceb8ea
osboxes@osboxes:~$ docker ps -s
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS                               NAMES
4e30cd07ed43   php:7.4-apache "docker-php-entrypoi..." 48 seconds ago     Up 47 seconds      0.0.0.0:8080->80/tcp, :::8080->80/tcp don_guillermo_server
osboxes@osboxes:~$ touch info.php
osboxes@osboxes:~$ nano info.php
```

7) Creamos el contenedor (que llamaremos **don_guillermo_server**) dentro de la imagen **php:7.4-apache**, y asignándole el puerto **8080** (en principio debíamos usar el puerto 8181 como en la actividad anterior, esto se debe a un simple lapsus)

Para ello utilizamos el comando **docker run -d --name don_guillermo_server -p 8080:80 php:7.4-apache**.

8) Utilizamos el comando **docker ps -s**. En la actividad anterior, **docker ps** mostraba la lista de contenedores creados junto con su nombre, ID y puerto asignado. Esta vez, el añadido **-s** ("size", o "tamaño" en inglés) muestra su tamaño en el disco duro como podemos ver en el apartado **SIZE** de la lista.

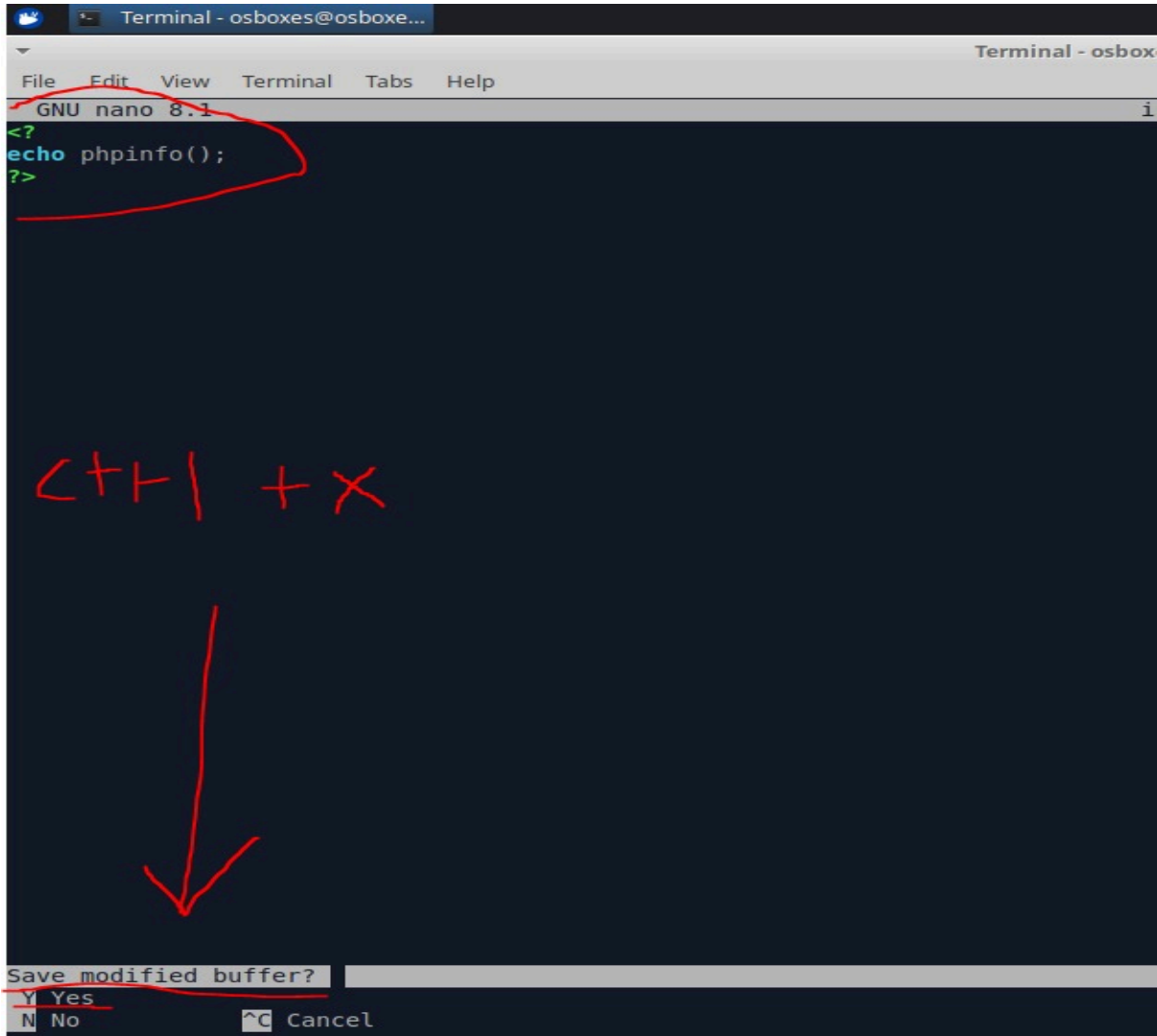
9) Creamos un fichero de texto en la imagen **php:7.4-apache** al cual nombraremos **info.php**. Esto se hace con el comando **touch info.php**:

- **"Info"** determina el nombre del fichero.
- **".php"** determina el formato, por lo que sabemos a qué imagen pertenece

10) Ahora vamos a editar el fichero creado utilizando el comando **nano** seguido del nombre del fichero. Es decir, **nano.info.php**.

Edición del contenido del fichero

Vamos a mostrar brevemente cómo editar y guardar el contenido del fichero con el fin de implementar un comando cuyo uso quedará explicado en breve:



Escribimos en el archivo la información que se nos ha pedido en uno de los enunciados del ejercicio:

```
<?>
echo phpinfo();
?>
```

(Esto es un comando que mostrará la información del fichero cuando accedamos a él desde el navegador web más adelante)

Después, pulsamos **ctrl + x** y aparecerá el mensaje "**Save modified buffer?**". Pulsamos **Y** para "Sí" y damos a enter para guardar los cambios.

Copiar fichero

Ahora copiaremos el fichero creado al directorio **/var/www/html/info.php** tal como se nos indica en uno de los enunciados del ejercicio.

Nota: la siguiente imagen se tomó desde otra consola virtual, pero el funcionamiento es el mismo.

```

osboxes@osboxes:~$ docker cp don_guillermo_server:/var/www/html/info.php ./info.php
osboxes@osboxes:~$ ls -l info.php
-rw-r--r-- 1 osboxes osboxes 22 Nov 25 13:28 info.php
osboxes@osboxes:~$ cat info.php
<? echo phpinfo(); ?>
osboxes@osboxes:~$

```

- 11)** Utilizando el comando **`docker cp don_guillermo_server:/var/www/html/info.php ./info.php`** copiaremos el fichero al directorio indicado.
- 12)** Comprobamos que el fichero existe con **`ls -l info.php`**.
- 13)** Comprobamos el contenido del fichero con **`cat info.php`**. El fichero ha sido copiado con éxito.

Mostrar contenedor y fichero en el navegador web

Por último, mostraremos el contenedor creado así como el fichero desde el navegador web.

De hacer falta, volveremos a comprobar nuestra dirección IP usando el comando **`ifconfig`** igual que hicimos en el ejercicio anterior. Nuestra IP es **10.0.3.109**.

```

osboxes@osboxes:~$ ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:b7ff:feea:b476 prefixlen 64 scopeid 0x20<link>
    ether 02:42:b7:ea:b4:76 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 4459 (4.4 KB)
    TX errors 0 dropped 57 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.109 netmask 255.255.0.0 broadcast 10.0.255.255
    inet6 fe80::60dc:a57e:7f12:6b59 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:89:18:65 txqueuelen 1000 (Ethernet)
    RX packets 685848 bytes 890902908 (890.9 MB)
    RX errors 0 dropped 737 overruns 0 frame 0
    TX packets 240372 bytes 16357736 (16.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Utilizamos el comando **`docker ps -s`** nuevamente para mostrar de nuevo el tamaño dentro del disco duro.

```

osboxes@osboxes:~$ docker ps -s
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS                               NAMES    SIZE
4e38cd07ed43   php:7.4-apache  "docker-php-entrypoi..."  14 minutes ago    Up 14 minutes    0.0.0.0:8080->80/tcp, :::8080->80/tcp    don_guillermo_server    27B


```

Para mostrar el fichero en el navegador, escribimos nuestra dirección IP seguida de dos puntos y el puerto asignado (**8080**) junto con una barra inclinada (**/**) y el nombre de la imagen (**info.php**). Es decir, **10.0.3.109:8080/info.php**.

No es seguro

10.0.3.109:8080/info.php

PHP Version 7.4.33



| | |
|---|---|
| System | Linux 4e30cd07ed43 6.11.0-8-generic #8-Ubuntu SMP PREEMPT_DYNAMIC Mon Sep 16 13:41:20 UTC 2024 x86_64 |
| Build Date | Nov 15 2022 06:03:12 |
| 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-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' '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 | 20190902 |
| PHP Extension | 20190902 |
| Zend Extension | 320190902 |
| Zend Extension Build | API320190902.NTS |
| PHP Extension Build | API20190902.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 |

Activar Windows

Gracias al comando que introducimos a la hora de crear el fichero, el navegador ahora muestra toda la información del mismo al acceder a él. Con esto, el ejercicio ha sido completado.