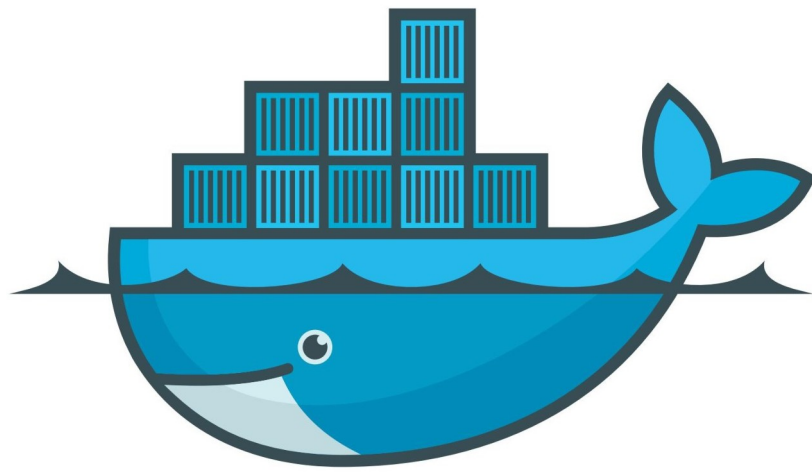


Unidad 09

Introducción a Docker



docker

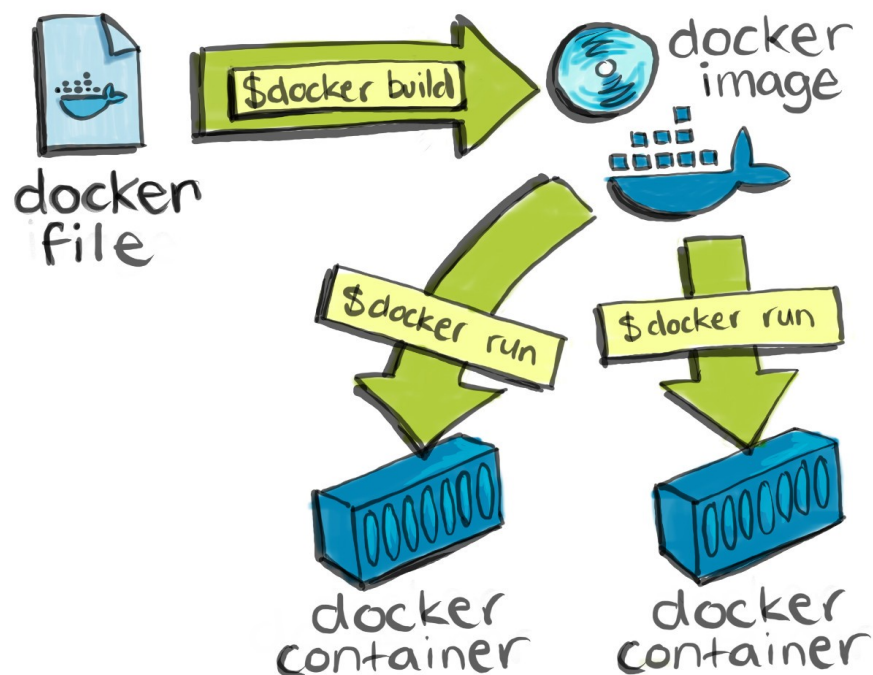
Marcos Zahonero

Índice

Introducción.....	3
Objetivo de la memoria.....	5
Material utilizado.....	6
Desarrollo.....	7
Parte I.....	8
Parte II.....	9
Ejercicio 1.....	9
Ejercicio 2.....	10
Parte III.....	11
Caso práctico 2 – Instalar LAMP + Wordpress.....	11
Caso práctico 3 – Interfaz gráfica con NoVNC.....	12
Parte IV.....	13
Introducción.....	13
Caso práctico 03 – Apache2 con PHP usando Alpine.....	14
Parte V.....	15
Introducción.....	15
Caso práctico 1 – WordPress + MySQL.....	16
Parte VI.....	17
Caso práctico 1 – Docker Compose.....	17
Parte VII.....	18
Caso práctico 1 – Visual Studio con Docker.....	18
Sugerencias, errores y/o problemas.....	20
encontrados en la actividad.....	20
Problema 1# - Versión 22 en Lubuntu.....	20
Sugerencia 1# - Pruebas antes en el instituto.....	20
Conclusión.....	21
Bibliografía/Webgrafía.....	22

① Introducción ①

En esta actividad veremos **Docker**, pero primero de todo [¿Qué es Docker?](#) habiendo ya resuelto la duda habiendo hecho clic si desconocíamos que era sigamos a la actividad, veremos como utilizar de forma básica esta aplicación y sobre todo iremos a cumplir los [Objetivos de la práctica](#) que están especificados ahí mismo.



Docker

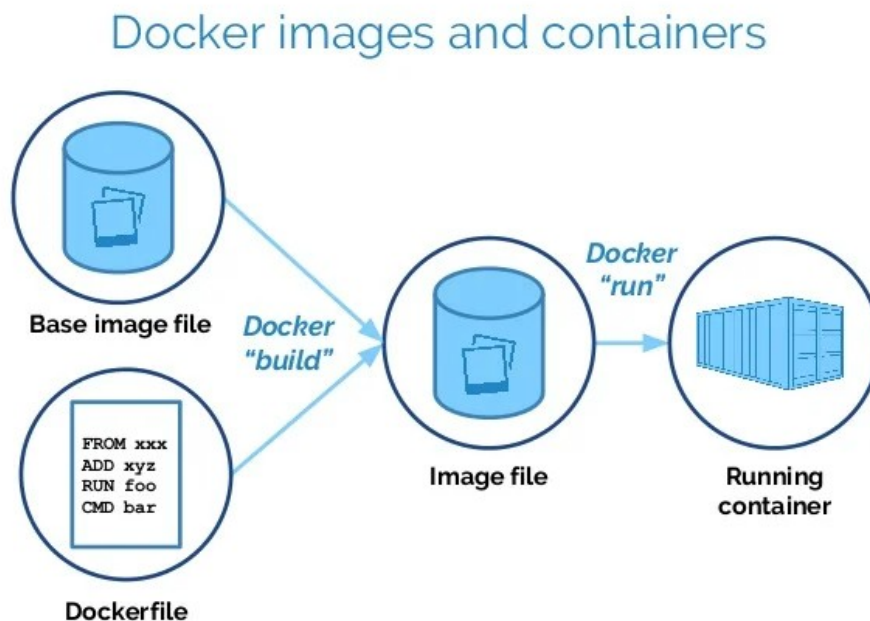
¿Qué es Docker?

Docker es una plataforma de software libre que está diseñada para crear, desplegar y ejecutar aplicaciones en algo llamado **contenedores**, y ahora te preguntarás ¿Qué son los contenedores? Los contenedores son como unas cajas que están aisladas y que tiene incluida en ellas todo lo necesario para que una aplicación pueda funcionar sin problemas (el código, las bibliotecas y las configuraciones)



¿Por qué puede llegar a ser útil Docker?

Buena pregunta, tiene muchas ventajas como por ejemplo la consistencia, los contenedores de los que antes he hablado se aseguran que la aplicación funcione en cualquier entorno posible, a entorno me refiero tu ordenador, un servidor o a la nube y se enlaza bastante también con otra ventaja, que es la portabilidad como no importa el entorno puede estar en cualquier SO que quieras también. Otra ventaja podría ser el rendimiento, ya que no necesita tantos recursos como una máquina virtual así que usan menos recursos.





Objetivo de la memoria

- **Introducirnos** en el mundo de Docker para entender los conceptos.
- **Conocer** las utilidades que contiene Docker y explotarlas.
- **Saber** usar Docker de forma básica



Material utilizado

DATOS	VALOR
Marca (fabricante) placa base	Gigabyte Technology
Modelo placa base	B150M-DS3H-CF
Marca Procesador	Intel
Modelo Procesador	Intel Pentium CPU G4400
Frecuencia Procesador	3.30GHz
Tamaño de cache (L1, L2, ...)	L1d: 32K L1i: 32K L2: 256K L3: 3072K
Marca bios	American Megatrends Inc.
Versión bios	F4
Puente PCH/FCH (chipset)	
Slot de la tarjeta gráfica	
Tipo Memoria RAM (memoria y slots)	16 GB
Dispositivos de almacenamiento (marca, modelo, capacidad, ...)	465 GB

Desarrollo



¡Importante!

Esta actividad está compuesta de 6 partes, es por eso que están divididas así, cada vez que sea una nueva parte estará en una hoja distinta para realizar una lectura más cómoda y más legible.

Las partes constarán de Introducción (a veces) y ejercicios/casos prácticos en los cuales avanzaremos en nuestro conocimiento a Docker haciendo varias cosas con él, de modo introducción únicamente.

Toda esta actividad estará en GitHub subida junto a los apuntes entregados en su momento para que sirva de realización en un futuro por si alguien quiere hacerlo, mediante los apuntes de Docker y esta actividad que servirá como ejemplo se podrá realizar sin ningún problema.

Parte I

Esta parte únicamente se basa en la instalación y configuración de Docker, es por eso que no hay imágenes ya que tampoco se requieren pero resumidamente se realizan las siguientes cosas:

1. Conocer términos como [VIRTUALIZACIÓN](#), [MÁQUINAS VIRTUALES](#), [HIPERVISOR](#), [CONTENEDORES](#)
2. Entender el funcionamiento de Docker
3. Ver la arquitectura de Docker
4. Explicación de Docker entre Sistemas Operativos

Información para realizar esta parte: [Clic para ir a GitHub](#)

Parte II

Ejercicio 1

Ejecuta en tu máquina con la instalación completa de Docker la orden:

> **docker version**

```
marcosz@marcosz-virtualbox:~$ docker version
Client: Docker Engine - Community
Version:      26.1.3
API version:  1.45
Go version:   go1.21.10
Git commit:   b72abbb
Built:        Thu May 16 08:33:29 2024
OS/Arch:      linux/amd64
Context:      default

Server: Docker Engine - Community
Engine:
Version:      26.1.3
API version:  1.45 (minimum version 1.24)
Go version:   go1.21.10
Git commit:   8e96db1
Built:        Thu May 16 08:33:29 2024
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.6.31
GitCommit:    e377cd56a71523140ca6ae87e30244719194a521
runc:
Version:      1.1.12
GitCommit:    v1.1.12-0-g51d5e94
docker-init:
Version:      0.19.0
GitCommit:    de40ad0
marcosz@marcosz-virtualbox:~$
```

Ejercicio 2

Ejecuta en tu máquina con la instalación completa de Docker la orden:

```
> docker run hello-world
```

```
marcosz@marcosz-virtualbox:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Parte III

Caso práctico 2 – Instalar LAMP + Wordpress

La creación de la cuenta y el título de la web:

WordPress - Installation - Mozilla Firefox

WordPress - Installation

localhost:8080/bloc/wp-admin/install.php

Welcome to the famous five-minute WordPress installation process. Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username Marcos Zahonero

Names can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password iNGaXzs%1*pHS@#Vp6 [Hide](#)

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

La web cuando te registras:



Mindblown: a blog about philosophy.

Hello world!

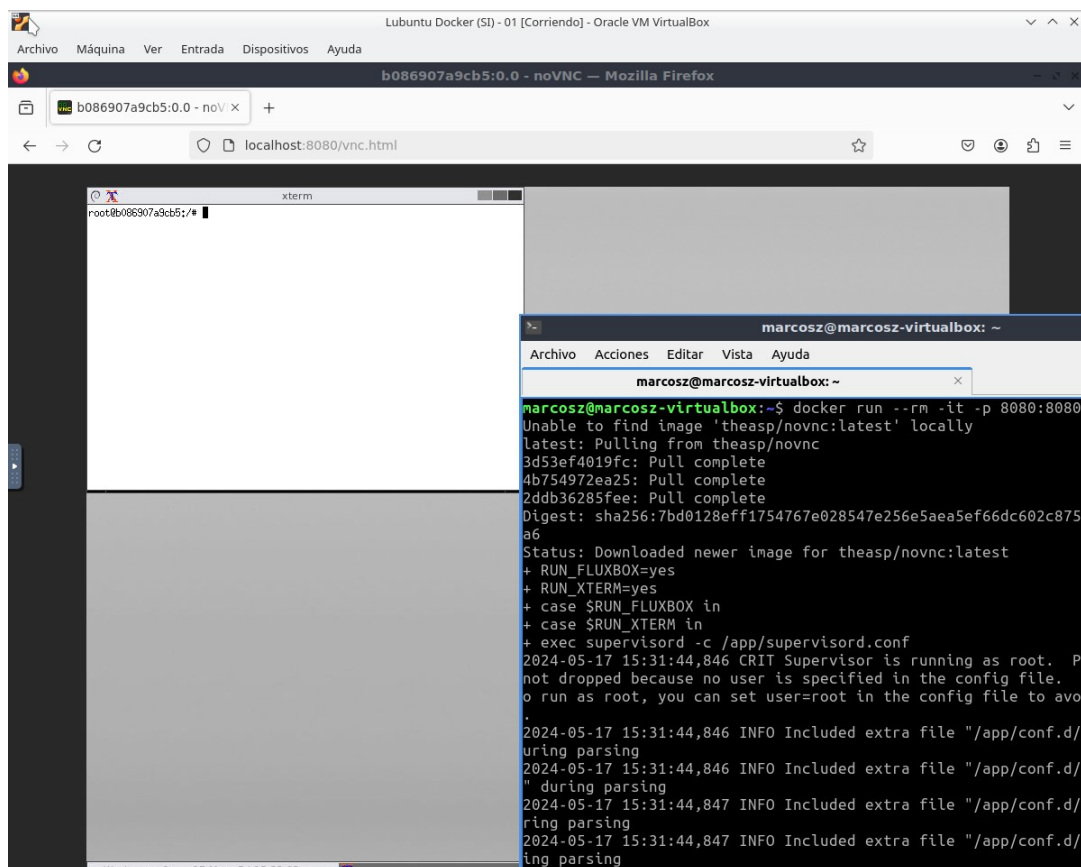
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

May 29, 2024

Caso práctico 3 – Interfaz gráfica con NoVNC

En este caso práctico vemos como tener una interfaz gráfica de nuestro contenedor que podremos visualizar desde nuestro navegador entrando a la dirección web (**localhost:8080/vnc.html**) desde aquí podremos usar la consola del contenedor sin problemas.

Aparecerá de la siguiente manera:



Parte IV

Introducción

En esta unidad tienes que realizar un primer caso práctico (imagen con nano) y subirlo a Docker Hub.

Mi Docker Hub: [maek0s/prueba](https://hub.docker.com/maek0s/prueba)

maek0s/prueba 

Updated 15 minutes ago

This repository does not have a description   INCOMPLETE

This repository does not have a category   INCOMPLETE

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	15 minutes ago

[See all](#)

```

1 FROM ubuntu:latest
2 RUN apt update && apt install -y nano
3 CMD /bin/bash

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● marcosz@marcosz-virtualbox:~/Desktop\$ docker build -t ubuntu nano . /

```

[+] Building 18.5s (6/6) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default
=> => transferring dockerfile: 107B                                              0.1s
=> [internal] load metadata for docker.io/library/ubuntu:latest                 0.0s
=> [internal] load .dockerignore                                                0.1s
=> => transferring context: 2B                                                  0.0s
=> [1/2] FROM docker.io/library/ubuntu:latest                                0.1s
=> [2/2] RUN apt update && apt install -y nano                                16.1s
=> exporting to image                                                          0.7s
=> => exporting layers                                                         0.5s
=> => writing image sha256:7dedd5a7957e080669e5a9914563f83691016600711086ab79cf0292391cdc2e 0.0s
=> => naming to docker.io/library/ubuntu nano                                0.1s
● marcosz@marcosz-virtualbox:~/Desktop$ docker history ubuntu nano
IMAGE          CREATED          CREATED BY          SIZE      COMMENT
7dedd5a7957e  51 seconds ago  CMD ["/bin/sh" "-c" "/bin/bash"]  0B        buildkit.dockerfil
e.v0
<missing>     51 seconds ago  RUN /bin/sh -c apt update && apt install -y ... 37.3MB    buildkit.dockerfil
e.v0
<missing>     5 weeks ago    /bin/sh -c #(nop)  CMD ["/bin/bash"]          0B
<missing>     5 weeks ago    /bin/sh -c #(nop) ADD file:ac9d5a9d5b9b1217a... 76.2MB
<missing>     5 weeks ago    /bin/sh -c #(nop) LABEL org.opencontainers... 0B
<missing>     5 weeks ago    /bin/sh -c #(nop) LABEL org.opencontainers... 0B
<missing>     5 weeks ago    /bin/sh -c #(nop) ARG LAUNCHPAD BUILD ARCH    0B
<missing>     5 weeks ago    /bin/sh -c #(nop) ARG RELEASE                  0B

```

Caso práctico 03 – Apache2 con PHP usando Alpine

He usado el Dockerfile, lo he ejecutado, he parado el puerto 80 para dejar ese puerto libre y al acceder a <http://localhost/index.html> sale que funciona.

```

12 #ADD http://wordpress.org/latest.tar.gz /var/www/localhost/htdocs/wordpress.tar.gz
13 #RUN tar xvfz /var/www/localhost/htdocs/wordpress.tar.gz && rm -rf /var/www/localhost/htdocs/wordpress.tar.gz
14 # Usamos usuario y grupo www-data. El grupo lo crea Apache, pero si quisiéramos crear grupo
15 # Grupo www-data RUN set -x && addgroup -g 82 -S www-data
16 # Creamos usuario www-data y lo añadimos a ese grupo
17 RUN adduser -u 82 -D -S -G www-data www-data
18 # Hacemos todos los ficheros de /var/www propiedad de www-data
19 # Y damos permisos a esos ficheros y a start.sh
20 RUN chown -R www-data:www-data /var/www/ && chmod -R 775 /var/www/ && chmod 755 /start.sh
21 #Indicamos puerto a exponer (para otros contenedores) 80
22 EXPOSE 80
23 #Comando lanzado por defecto al instalar el contenedor

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

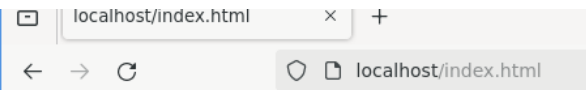
21 | #Indicamos puerto a exponer (para otros contenedores) 80
22 | EXPOSE 80
-----
ERROR: failed to solve: process "/bin/sh -c chown -R www-data:www-data /var/www/ && chmod -R 775 /var/www/ && chmod 755 /start.s
de: 127
marcosz@marcosz-virtualbox:~/Desktop/alpineapache$ docker build -t alpinecache ./
[+] Building 3.5s (12/12) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 1.44kB                             0.0s
=> [internal] load metadata for docker.io/library/alpine:latest   0.6s
=> [internal] load .dockerignore                                  0.1s
=> => transferring context: 2B                                     0.0s
=> [1/6] FROM docker.io/library/alpine:latest@sha256:77726ef6b57dd6f65bb551896826ec38bc3e53f75cdde31354fbff4f25 0.0s
=> [4/6] ADD https://gist.githubusercontent.com/SyntaxC4/5648247/raw/94277156638f9c309f2e36e19bff378ba7364907/i 0.0s
=> [internal] load build context                                  0.1s
=> => transferring context: 29B                                    0.0s
=> CACHED [2/6] RUN apk update && apk add apache2 php php-apache2 openrc tar 0.0s
=> CACHED [3/6] ADD ./start.sh /start.sh                         0.0s
=> CACHED [4/6] ADD https://gist.githubusercontent.com/SyntaxC4/5648247/raw/94277156638f9c309f2e36e19bff378ba73 0.0s
=> CACHED [5/6] RUN adduser -u 82 -D -S -G www-data www-data     1.3s
=> [6/6] RUN chown -R www-data:www-data /var/www/ && chmod -R 775 /var/www/ && chmod 755 /start.sh 0.8s
=> exporting to image                                           0.7s
=> => writing image sha256:44062f8719f1bf196c65cd3a7eaa440b4e2b363e56f6a524677f5d06c2eebaaf 0.0s
=> => naming to docker.io/library/alpinecache                    0.0s
marcosz@marcosz-virtualbox:~/Desktop/alpineapache$

```

```

8aaf268d5f0ef10d45ce0fef78c10d5ec6b3223c0ea15013
sponse from daemon: driver failed programming ext
Error starting userland proxy: listen tcp4 0.0.0.
virtualbox:~$ ^C
virtualbox:~$ ^C
virtualbox:~$ sudo service apache2 stop
a para marcosz:
virtualbox:~$ docker run -dp 80:80 alpinecache
f4865a7175b679e69863c3ed66c5716fbe0b857d1136b5b4
virtualbox:~$

```



It works!

Parte V

Introducción

Enlace 2 contenedores (prueba2 y prueba1) y le hice un ping para que se vea la conexión:

```
marcosz@marcosz-virtualbox:~$ docker start -i prueba2
/ # ping prueba1
PING prueba1 (172.19.0.2): 56 data bytes
64 bytes from 172.19.0.2: seq=0 ttl=64 time=1.040 ms
64 bytes from 172.19.0.2: seq=1 ttl=64 time=0.073 ms
64 bytes from 172.19.0.2: seq=2 ttl=64 time=0.072 ms
64 bytes from 172.19.0.2: seq=3 ttl=64 time=0.075 ms
^C
--- prueba1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.072/0.315/1.040 ms
/ # █
```

Caso práctico 1 – WordPress + MySQL

Esta es la ventana de configuración de WordPress:

WordPress › Instalación del archivo de configuración. — Mozilla Firefox

localhost:8080/wp-admin/setup-config.php?step=1&language=es_ES

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos
El nombre de la base de datos que quieres usar con WordPress.

Nombre de usuario
El nombre de usuario de tu base de datos.

Contraseña [Ocultar](#)
La contraseña de tu base de datos.

Servidor de la base de datos
Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

Prefijo de tabla
Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)

Todo ha salido bien.

¡Muy bien! Ya has terminado esta parte de la instalación. Ahora WordPress puede comunicarse con tu base de datos. Si estás listo, es el momento de...

[Ejecutar la instalación](#)

Comprobamos los contenedores con “docker ps”:

```
marcosz@marcosz-virtualbox:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9d358eb5ed6	wordpress	"docker-entrypoint.s..."	29 minutes ago	Up 3 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp	nuestrowp
90a05f97bf3d	mysql:5.6	"docker-entrypoint.s..."	30 minutes ago	Up 3 minutes	3306/tcp	mariaadb

Parte VI

Caso práctico 1 – Docker Compose

Levantamos y cerramos el servicio así de fácil:

```
marcosz@marcosz-virtualbox:~/Desktop/parte6$ docker-compose up -d
Creating network "parte6_default" with the default driver
Creating volume "parte6_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
20e4dcae4c69: Pull complete
1c56c3d4ce74: Pull complete
e9f03a1c24ce: Pull complete
68c3898c2015: Pull complete
6b95a940e7b6: Pull complete
90986bb8de6e: Pull complete
ae71319cb779: Pull complete
ffc89e9dfd88: Pull complete
43d05e938198: Pull complete
064b2d298fba: Pull complete
df9a4d85569b: Pull complete
Digest: sha256:4bc6bc963e6d8443453676cae56536f4b8156d78bae03c0145cbe47c2aad73bb
Status: Downloaded newer image for mysql:5.7
Creating parte6_db_1 ... done
Creating parte6_wordpress_1 ... done
marcosz@marcosz-virtualbox:~/Desktop/parte6$ docker-compose down
Stopping parte6_wordpress_1 ... done
Stopping parte6_db_1 ... done
Removing parte6_wordpress_1 ... done
Removing parte6_db_1 ... done
Removing network parte6_default
```

El WordPress al entrar sale el proceso de instalación, con el puerto 8000 en este caso:



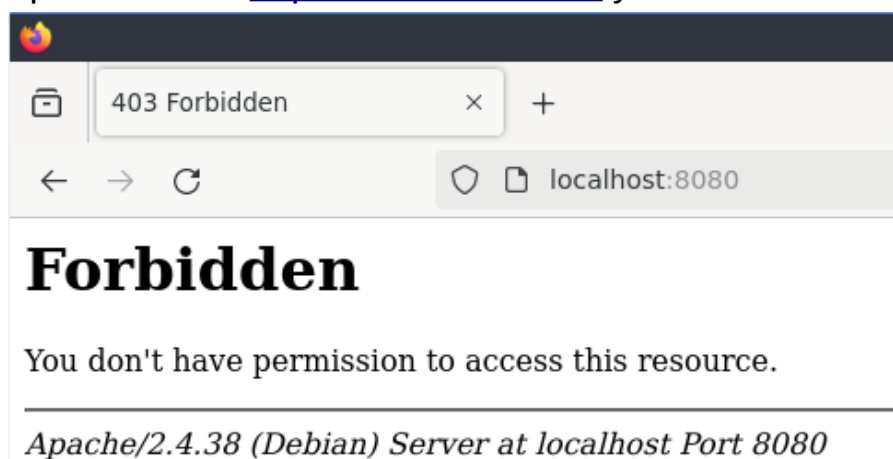
Parte VII

Caso práctico 1 – Visual Studio con Docker

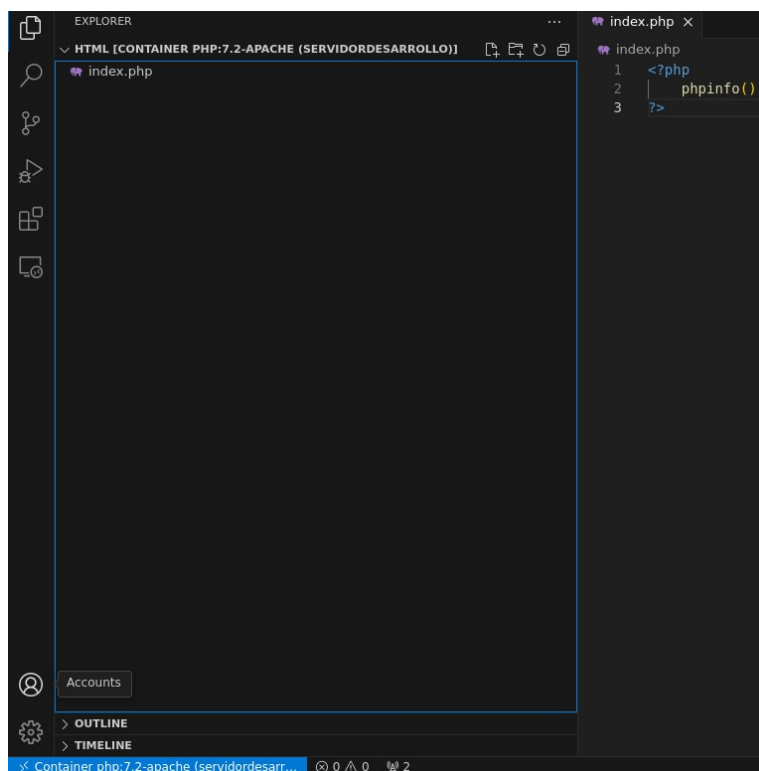
Cuando hayamos ejecutado el:

```
> docker run -d --name servidordesarrollo -p 8080: 80 php: 7.2-apatxe
```

tendremos que entrar en <https://localhost:8080> y nos saldrá lo siguiente:



Instalamos la extensión Docker y abrimos la carpeta **/var/www/html** y creamos un archivo **index.php** con el siguiente texto:



Al poner ese archivo entramos de nuevo a <https://localhost:8080> y nos saldrá la siguiente página web:

<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>16:30 4/6/24</div> <div>marcoszahonero</div> </div> <div>Lubuntu 20 Docker (SI) [Corriendo] - Oracle VM VirtualBox</div>	
phpinfo() — Mozilla Firefox	
<div>PHP Version 7.2.34</div>	
System	Linux 0274b6c20e8d 5.15.0-107-generic #117~20.04.1-Ubuntu SMP Tue Apr 11 16:30:00 UTC 2023; root:x86_64
Build Date	Dec 11 2020 10:50:00
Configure Command	./configure '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--enable-option-checking=fatal' '--with-mcrypt' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' '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	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,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
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

Sugerencias, errores y/o problemas



encontrados en la actividad



Problema 1# - Versión 22 en Lubuntu

Como bien sabemos la versión de 22 para Lubuntu daba incontables errores y instalaciones y por eso mismo lo que hice fue pasar a la versión 20.

Sugerencia 1# - Pruebas antes en el instituto

Sugiero que se prueben las cosas de la actividad cuando toca temas de descargar datos del exterior, tipo los repositorios etc. ya que nos quita mucho tiempo errores los cuales no dependen de nosotros, se que nos enfrentaremos a la vida real en un futuro pero en este caso nos bloquea directamente conselleria y es por eso que estaría bien que se probará antes en algún rato libre.



Imagen 1: Foto realizada con la IA de Microsoft



Conclusión

Hemos concluido con la actividad de Docker, hemos conocido muchas utilidades que conllevan los contenedores y sobre todo en el aspecto web, jugando con el localhost y también el Docker Hub, que es una forma de llevar los contenedores e imágenes a donde quieras.



Bibliografía/Webgrafía

– He utilizado los PDFs guías de Docker entregados por el profesor.

Enlaces que se encuentran en el documento con **hiperenlaces**:

SO: [¿Qué es un Sistema Operativo \(significados.com\)](https://significados.com/que-es-un-sistema-operativo/)

Virtualización: [¿Qué es la Virtualización? \(ibm.com\)](https://ibm.com/que-es-la-virtualizacion/)

MV: [¿Qué es una máquina virtual \(VM\)? \(redhat.com\)](https://redhat.com/que-es-una-maquina-virtual-vm/)

Hipervisor: [¿Qué es un Hipervisor? \(redhat.com\)](https://redhat.com/que-es-un-hipervisor/)

Contenedores: [¿Qué es un contenedor en Docker? \(KeepCoding\)](https://keepcoding.com/que-es-un-contenedor-en-docker/)