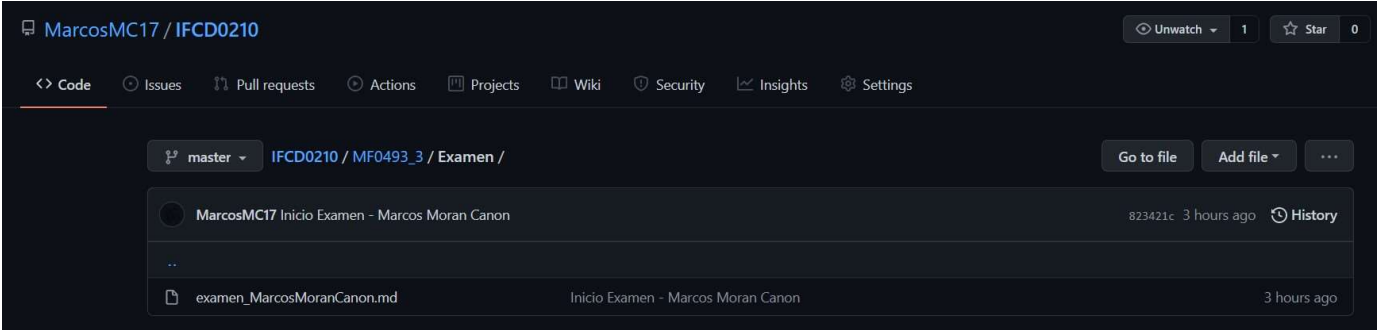


Examen

[Jump to bottom](#)

MarcosMC17 edited this page on 5 Aug · 17 revisions

Estructura de mi repositorio IFCD0210



Creación de una maquina virtual (tipo Windows)

Al inicio de la creación de la maquina, nos solicitará el el tipo de maquina virtual y el nombre de la misma, en mi caso Windows 10.




← Crear máquina virtual

Nombre y sistema operativo


Seleccione un nombre descriptivo y una carpeta destino para la nueva máquina virtual y seleccione el tipo de sistema operativo que tiene intención de instalar en ella. El nombre que seleccione será usado por VirtualBox para identificar esta máquina.

Nombre:

Carpeta de máquina: 

Tipo:

Versión:



Modo experto

Next

Cancelar

Mas adelante nos pedirá establecerle la memoria RAM que queremos asignarle a la maquina virtual, 3 gigabytes.



← Crear máquina virtual

Tamaño de memoria

Seleccione la cantidad de memoria (RAM) en megabytes a ser reservada para la máquina virtual.

El tamaño de memoria recomendado es **2048 MB**.

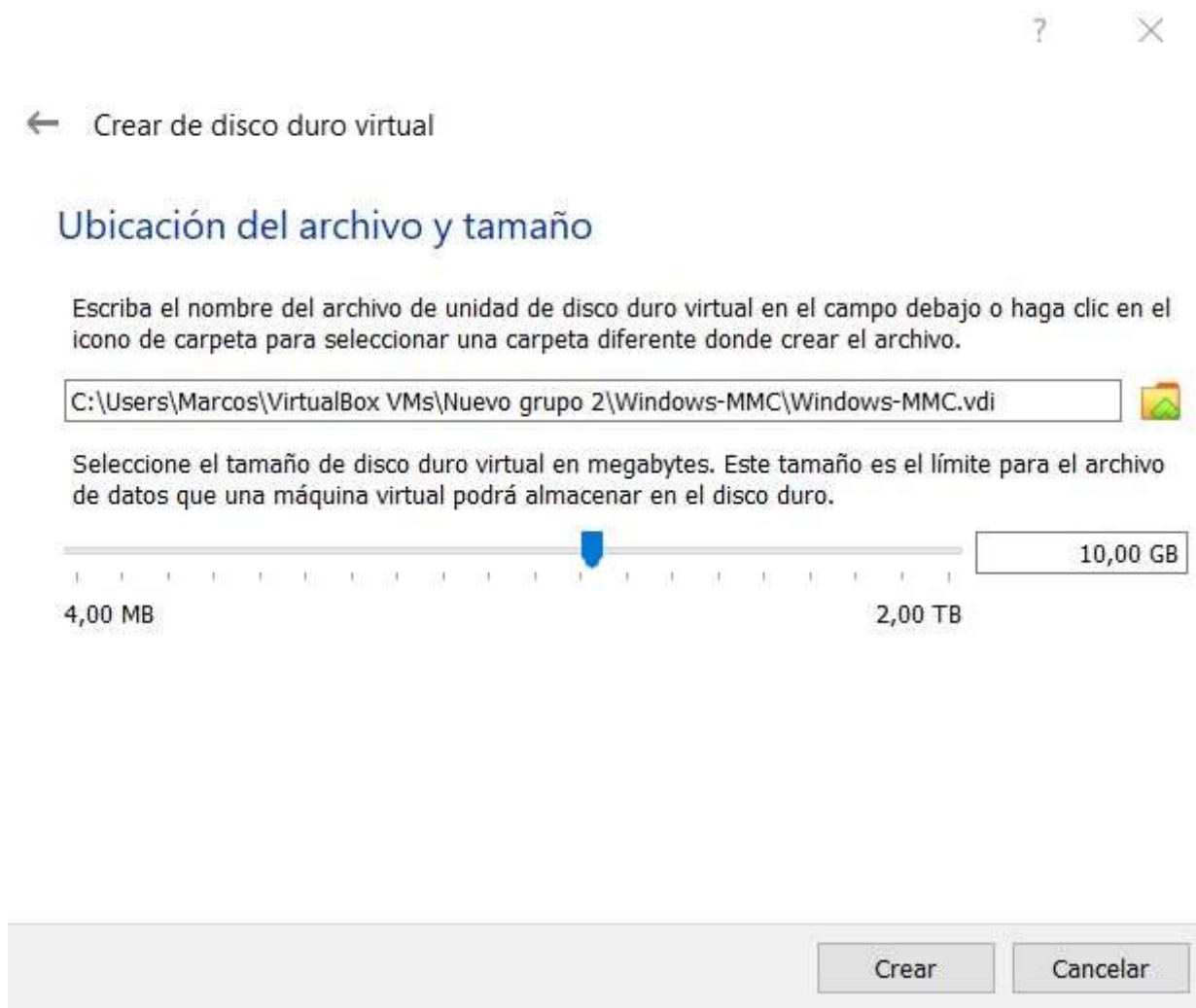
 MB

4 MB 16384 MB

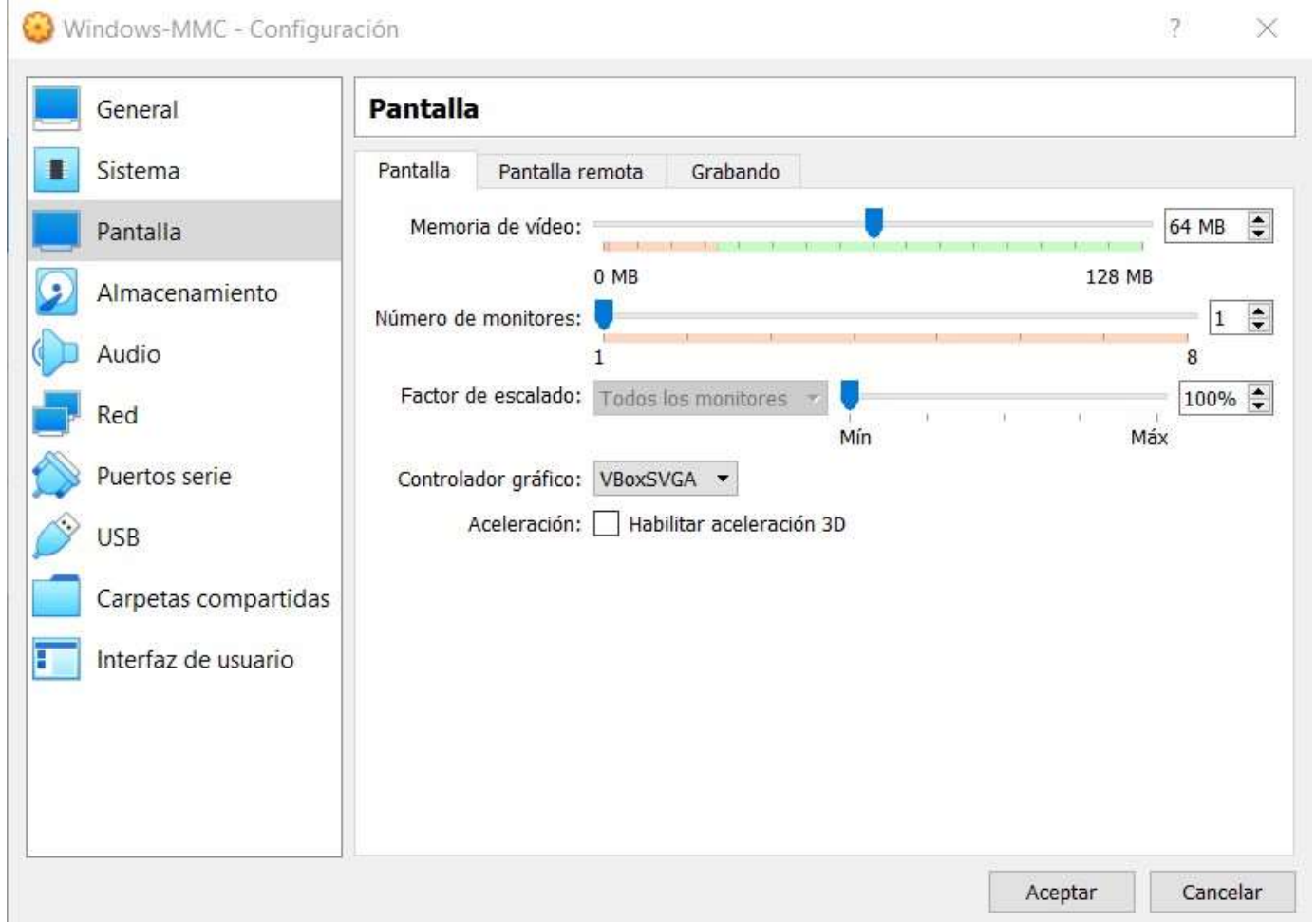
Next

Cancelar

Nos solicitará crear un disco duro virtual (a elegir entre otras opciones) y establecerle el tamaño del mismo, en mi caso 10 gigabytes



Seguiremos los pasos de la creación de la maquina virtual y al terminar, podremos configurar en la opciones > pantalla, la cantidad de memoria de video que queremos dedicar a maquina virtual.



Docker

Dirección de la imagen en Docker Hub:

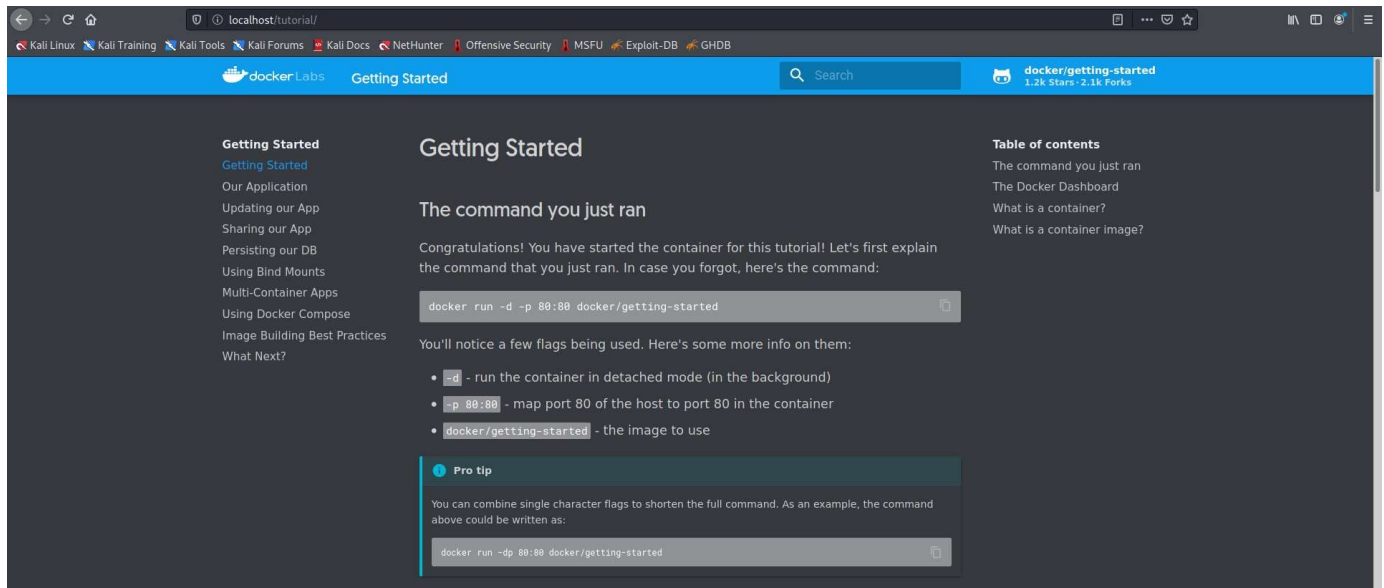
Getting Started

Debemos ejecutar el comando para crear el contenedor `docker run -d -p 80:80 docker/getting-started`

Al no encontrar la imagen, la descargará automáticamente y luego creará el contenedor

```
(kali@kali)-[~]
$ docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
540db60ca938: Pull complete
0ae30075c5da: Pull complete
9da81141e74e: Pull complete
b2e41dd2ded0: Pull complete
7f40e809fb2d: Pull complete
758848c48411: Pull complete
23ded5c3e3fe: Pull complete
38a847d4d941: Pull complete
Digest: sha256:10555bb0c50e13fc4dd965ddb5f00e948ffa53c13ff15dc85b7ab65e1f240b
Status: Downloaded newer image for docker/getting-started:latest
84941601d9ff9de140f32bdbb27eb3a0494fca20585118ecf81476854d2797ff
```

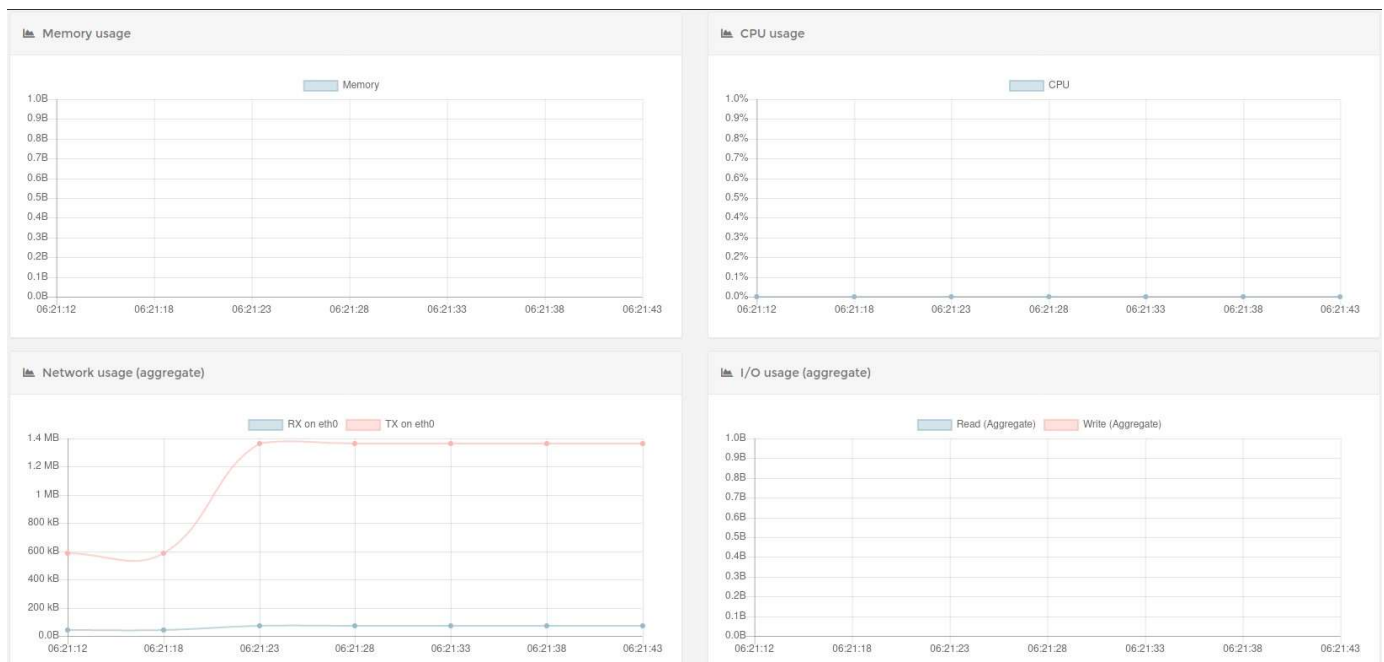
La imagen terminará de descargarse. Una vez descargada y creada, podremos navegar a la URL <http://localhost/> y ver como nos muestra la documentación de Docker.



Si abrimos el Portainer, podremos observar que nuestro contenedor está corriendo:



Y también podremos ver las estadísticas de uso de los recursos del contenedor:



Docker-compose

Para la creación del contenedor con docker-compose, deberemos crear la carpeta donde queremos el archivo YML y seguidamente, crear el archivo `docker-compose.yml`

```
user@user-VirtualBox:~/Desktop/Examen/IFCD0210/MF0493_3/Examen$ mkdir dkComp
user@user-VirtualBox:~/Desktop/Examen/IFCD0210/MF0493_3/Examen$ cd dkComp/
```

El fichero `docker-compose.yml`, podemos crearlo con `touch` y editarlo con `nano` o `vi`. A este fichero le daremos el siguiente contenido.

```
GNU nano 4.8 docker-compose.yml
version: '2'
services:
  db:
    image: mysql:latest
    volumes:
      - "./.mysql-data/db:/var/lib/mysql"
    restart: always
    ports:
      - 4408:3306
    environment:
      MYSQL_ROOT_PASSWORD: marcos
      MYSQL_DATABASE: focyl
      MYSQL_USER: user
      MYSQL_PASSWORD: marcos
```

Para añadir el fichero a PHPMYAdmin, añadiremos el siguiente trozo de código en el fichero `docker-compose.yml`:

```
# phpmyadmin
phpmyadmin:
  depends_on:
    - db
  image: phpmyadmin/phpmyadmin
  restart: always
  ports:
    - '9898:80'
  environment:
    PMA_HOST: db
    MYSQL_ROOT_PASSWORD: marcos
```

Para ejecutar crear el contenedor ejecutaremos (desde el directorio donde se encuentra el fichero `docker-compose.yml`) el siguiente comando:

```
docker-compose up -d
```

Si ejecutamos el comando `docker ps` podremos ver como el contenedor de `mySql` esta arrancado:

```
0e1464e1239d mysql:latest "docker-entrypoint.s..." 3 minutes ago Up 3 minutes 33060/tcp, 0.0.0.0:4408->3306/tcp, :::4408->3306/tcp dkcomp_db_1
```

Despliegue de CMS

Desplegaré un contenedor de `wordpress`. Para ello crearé un directorio dedicado al `wordpress` y dentro un archivo `docker-compose.yml` y le daré el siguiente contenido:


```

user@user-VirtualBox: ~/Desktop/Examen/IF...
GNU nano 4.8 docker-compose.yml
version: "3.3"

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: wordpress
      MYSQL_USER: user
      MYSQL_PASSWORD: passwd

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    volumes:
      - wordpress_data:/var/www/html
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: user
      WORDPRESS_DB_PASSWORD: passwd
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
  wordpress_data: {}

```

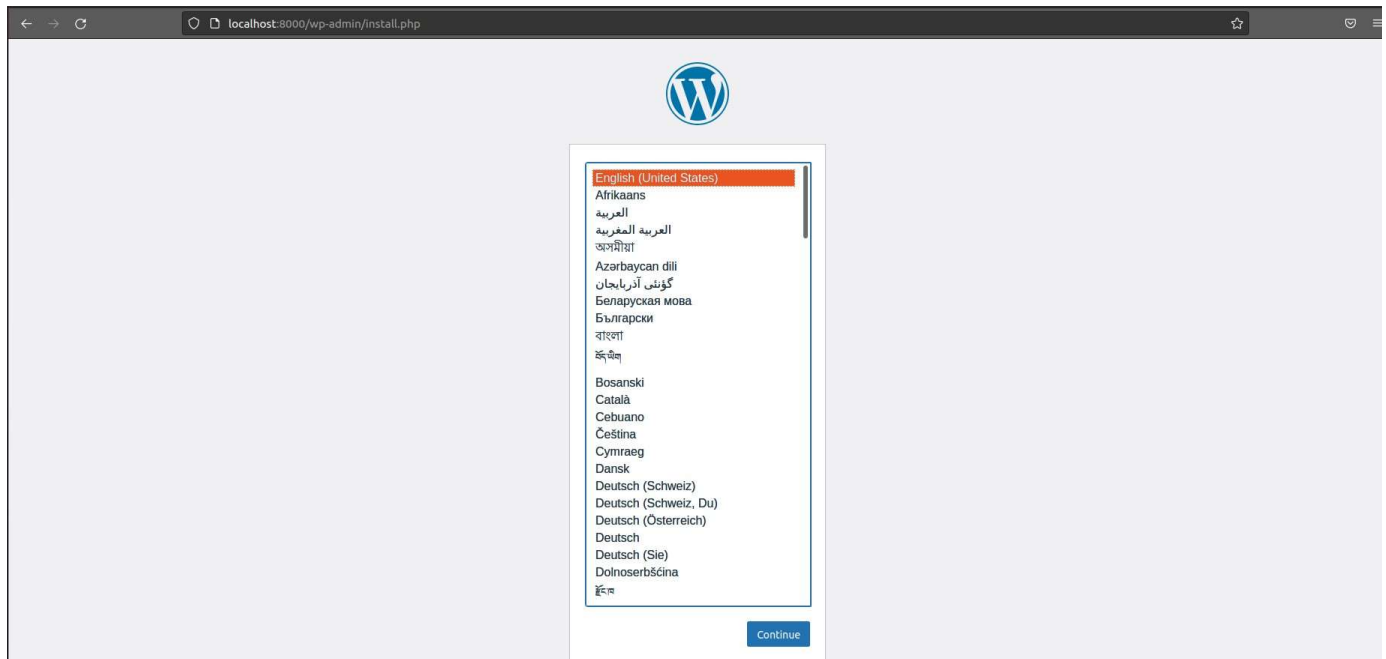
Y a continuación, ejecutaré el comando `docker-compose up -d` para levantar el contenedor definido, cosa que podemos ver ejecutando `docker ps`:

```

user@user-VirtualBox:~/Desktop/Examen/IFCD0210/MF0493_3/Examen/miWordpress$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
99d934e64dbd   wordpress:latest "docker-entrypoint.s..." 46 seconds ago Up 44 seconds 0.0.0.0:8000->80/tcp, :::8000->80/tcp miwordpress_wordpress_1
2e9bc26082bd   mysql:5.7      "docker-entrypoint.s..." 47 seconds ago Up 46 seconds 3306/tcp, 33060/tcp               miwordpress_db_1

```

Si accedemos a `http://localhost:8000`, nos redirigirá a la página de instalación de wordpress, el cual podremos instalar sin problemas como cualquier otro wordpress sin dockerización:



Captura de el estado de los contenedores:

```
user@user-VirtualBox:~/Desktop/Examen/IFCD0210/MF0493_3/Examen/miWordPress$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
99d934e64dbd	wordpress:latest	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes	0.0.0.0:8000->80/tcp, :::8000->80/tcp	miwordpress_wordpress_1
2e9bc26082bd	mysql:5.7	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes	3306/tcp, 33060/tcp	miwordpress_db_1
674bf795bc94	phpmyadmin/phpmyadmin	"/docker-entrypoint.s..."	27 minutes ago	Exited (0) 14 minutes ago		dkcomp_phpmyadmin_1
8e1464e1239d	mysql:latest	"docker-entrypoint.s..."	27 minutes ago	Exited (0) 14 minutes ago		dkcomp_db_1
9b3087766239	getting-started	"docker-entrypoint.s..."	2 hours ago	Exited (0) 14 minutes ago		stolz_herschel
28c61251c7fe	docker/getting-started	"/docker-entrypoint.s..."	2 hours ago	Exited (0) 2 hours ago		angry_jones
1510cf1314ea	getting-started	"docker-entrypoint.s..."	2 hours ago	Exited (0) 2 hours ago		confident_lichterman
ce4c6fe89bb2	node:12-alpine	"docker-entrypoint.s..."	10 days ago	Exited (1) 10 days ago		silly_meninsky
085943e3a5a6	nicolaka/netshoot	"zsh"	10 days ago	Exited (0) 10 days ago		serene_jackson
8e3cb885203	mysql:5.7	"docker-entrypoint.s..."	10 days ago	Exited (0) 10 days ago		jolly_hopper
43448e45a7ea	node:12-alpine	"docker-entrypoint.s..."	10 days ago	Exited (1) 10 days ago		loving_nobel
f9f09c6fb2aa	c84e29ca0626	"docker-entrypoint.s..."	13 days ago	Created		recurring_mendel
85f451b71953	docker/getting-started	"/docker-entrypoint.s..."	13 days ago	Exited (0) 10 days ago		vigorous_lederberg
d25d0857ff40	hello-world	"/hello"	13 days ago	Exited (0) 13 days ago		dazzling_ganguly
819aaa29c79d	docker-lamp-main_www	"docker-php-entrypoi..."	2 weeks ago	Exited (255) 2 weeks ago	0.0.0.0:80->80/tcp, :::80->80/tcp	docker-lamp-main_www_1
86b61992f1a4	phpmyadmin/phpmyadmin	"/docker-entrypoint.s..."	2 weeks ago	Exited (255) 2 weeks ago	0.0.0.0:8000->80/tcp, :::8000->80/tcp	docker-lamp-main_phpmyadmin_1
8d548d64cd5a	mysql:8.0	"docker-entrypoint.s..."	2 weeks ago	Exited (1) 2 weeks ago		docker-lamp-main_db_1
ee074e6af102	ubuntu	"bash"	2 weeks ago	Exited (0) 2 weeks ago		vigilant_wiles

Pages 2

Find a Page...

Home

Examen

Estructura de mi repositorio IFCD0210

Creación de una maquina virtual (tipo Windows)

Docker

Dirección de la imagen en Docker Hub:

Docker-compose

Despliegue de CMS

Captura de el estado de los contenedores:

Clone this wiki locally

<https://github.com/MarcosMC17/IFCD0210.wiki.git>

