FT_SERVER

Indicaciones previas:

- La primera parte muestra comandos básicos de docker, mientras que la segunda se centra en el desarrollo del proyecto ft_server.
- Todo lo contenido entre corchetes se deberá poner sin los mismos.
- Existen varias combinaciones para todos los comandos de ejecución, por lo que se pueden combinar.

Instalación de Docker:

https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-18-04-1-es

Imágenes y contenedores:

docker search [nombre de la imagen] te permite buscar una imagen en el repositorio propio de docker.
docker pull [nombre de la imagen] te permite bajar la imagen indicada del repositorio de docker.
docker images lista todas las imágenes descargadas.
docker ps te permite ver los contenedores que se están ejecutando en este momento
docker ps -a te muestra todos los contenedores que tengas creados.
docker rm [ID] borra el contenedor indicado.
docker image rm [ID] borra la imagen indicada.

Creando contenedores:

docker run [nombre de la Imagen] Crea un contenedor con la imagen indicada.
docker run --name [nombre que queremos dar] [nombre de la Imagen] Permite dar un nombre a nuestro contenedor.
docker run -p 80:80 -p 443:443 -d [nombre la imagen]
-p 80:80 y -p 443:443 = Exponer los puertos que vamos a utilizar, de entrada (Local-machine) y salida (Internal container)
-d = Arrancar el servidor en el background.
La parte de la izquierda de los : es la localización en nuestro ordenador.

La parte de la derecha de los : es la localización de nuestro contenedor.

Otro ejemplo parecido:
docker run -p 80:80 -p 443:443 -it [nombre la imagen]

-it quiere decir interactivo, es decir, veremos en el shell una representación de los procesos que se están ejecutando, así como podremos acceder a su contenido.

Por último, para arrancar o detener contenedores:
docker start [ID o nombre del contenedor] Arranca el contenedor indicado.
docker stop [ID o nombre del contenedor] detiene el contenedor indicado.

Construyendo tus propias imágenes docker

Docker permite crear tu propia imagen docker a partir de un dockerfile. Una vez elaborado nuestro archivo seremos capaces de crear la imagen ejecutando lo siguiente:
docker build -t [nombre que queramos dar a la imagen] [Ruta donde se encuentre el dockerfile]
-t Nos permite añadir un nombre a la imagen que vamos a crear.

Más Información

Creando el Dockerfile

Ver los comentarios del dockerfile.
Working Progress.

PhPMyAdmin

Esta herramienta nos permitirá administrar MySQL/MariaDB utilizando un navegador web.
Descargamos PhPMyAdmin de la página oficial y lo descomprimimos en la carpeta src.
Dentro tendremos que modificar el archivo "config.sample.inc.php" y renombrarlo a "config.inc.php"

Para acceder a la configuración de PhPMyAdmin desde el navegador una vez levantado el contenedor:
Usuario = root
Password =

Más información de PhPMyAdmin

Wordpress

Es un sistema de gestión de contenido enfocado a la creación de cualquier página web.
Lo descargamos de su página oficial y lo descomprimimos en la carpeta src.
Dentro modificamos el archivo wp-config.php

Para acceder al Wordpress Admin una vez levantado el docker, y poder así configurar nuestro Wordpress utilizaremos:
Usuario = root
Password = 123456
Nota: Esto es así porque lo he modificado previamente para tener este usuario y este pass.
El archivo wordpress.sql que se encuentra en la carpeta src, lo generaremos exportándolo desde nuestro PhPMyAdmin una vez arrancado el docker.

Más información de Wordpress

Archivos de srcs:

Default: archivo de configuración de Nginx. Más info: 1, 2

Index.html: Configuración en HTML de la página inicial.

Init.sql: Creamos una base de datos llamada wordpress Más info: 1

Self-signed.conf: ver certificado SSL

Ssl-params: Ver certificado SSL

Wordpress.sql: ver Wordpress

Certificado SSL

EL OpenSSL consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía.

Estas herramientas ayudan al sistema a implementar el Secure Sockets Layer (SSL), así como otros protocolos relacionados con la seguridad, como el Transport Layer Security (TLS). OpenSSL también permite crear certificados digitales que pueden aplicarse a un servidor, por ejemplo Apache.

En el siguiente enlace se explica cómo crear una clave ssl para tu servidor Nginx: https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-on-debian-10

El siguiente enlace detalla cada uno de los comandos utilizados (man) : https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html