



Introducción práctica a Docker para desarrollo web

¿Qué es Docker?

Docker es una herramienta que permite crear contenedores, es decir, **entornos de ejecución aislados** que incluyen el sistema operativo, las dependencias, bibliotecas y la aplicación que queremos ejecutar. Imagina que quieres compartir tu proyecto con otra persona, pero requiere una versión específica de Python, ciertos paquetes y configuraciones especiales. En lugar de explicarle todo lo que debe instalar, puedes enviarle un **contenedor Docker** que ya lo incluye todo. Así, funciona igual en cualquier computadora.

¿Qué es una imagen Docker?

Una **imagen** es como una receta. Define todo lo necesario para que una aplicación funcione: sistema base, dependencias, archivos, comandos de inicio.

Una vez que se crea una imagen, se puede ejecutar como un **contenedor**, que es una instancia activa de esa imagen.



¿Qué es un Dockerfile?

Un **Dockerfile** es un archivo de texto donde escribimos **instrucciones paso a paso** para crear la imagen. Es como escribir la receta de instalación de nuestra app.

Estructura típica de un Dockerfile

```
Dockerfile
CopyEdit
# 1. Imagen base (puede ser Ubuntu, Python, Node, etc.)
FROM python:3.11

# 2. Carpeta de trabajo dentro del contenedor
WORKDIR /app

# 3. Copiar archivos locales al contenedor
COPY . .

# 4. Instalar dependencias
RUN pip install flask

# 5. Comando que se ejecuta al iniciar el contenedor
CMD ["python", "app.py"]
```

Explicación de cada línea:

Línea	Significado
FROM	Define la imagen base. En este caso, una imagen oficial de Python 3.11
WORKDIR	Define el directorio de trabajo dentro del contenedor
COPY . .	Copia todos los archivos del directorio actual (proyecto local) al contenedor
RUN	Ejecuta comandos dentro del contenedor (como instalar paquetes)
CMD	Indica el comando que se ejecutará cuando se inicie el contenedor

¿Cómo se construye y se ejecuta una imagen?

Paso 1: Crear el Dockerfile y los archivos de la app

Tu estructura de proyecto debe verse así:

```
CopyEdit
websec-lab/
├── Dockerfile
└── app.py
```

Paso 2: Construir la imagen

```
bash
CopyEdit
docker build -t flaskvuln .
```

- `docker build`: comando para construir la imagen
- `-t flaskvuln`: etiqueta o nombre que le damos a la imagen
- `.` indica que usará el Dockerfile del directorio actual

Paso 3: Ejecutar el contenedor

```
bash
CopyEdit
docker run -d -p 5000:5000 flaskvuln
```

- `-d`: lo ejecuta en segundo plano
 - `-p 5000:5000`: expone el puerto 5000 del contenedor en el puerto 5000 de tu PC
 - `flaskvuln`: nombre de la imagen creada
-

¿Cómo accedo a la aplicación?

Abre tu navegador y visita:

- <http://localhost:5000>

Deberías ver el sitio web que programaste en `app.py`.

Conclusiones

- Docker permite crear entornos listos para usar sin instalar nada más
- Una imagen se construye con un `Dockerfile`
- Puedes levantar un sitio web en segundos y destruirlo con un solo comando