

PRACTICAS DOCKER IMÁGENES Y DOCKERFILE

1) Crear un Dockerfile para que instale nginx

- a. Imagen base ubuntu:20.04
- b. Ejecutar dos comandos:
 - i. apt-get update
 - ii. apt-get install -y nginx
- c. Crear un fichero index.html con un contenido y copiarlo a la carpeta: /var/www/html/index.html
- d. Exponer el puerto 80
- e. Lanzar el comando nginx con los parámetros “-g” y “Daemon off;”
- f. Construir la imagen en el directorio actual.

`docker build -t nombre_img:tag .`

- g. Verificar si se ha creado la imagen. Listar las imágenes
- h. Crear un contenedor, mapear el puerto en el 8080 del host y 80 del contenedor.
- i. Comprobar si arranca el contenedor
- j. <http://localhost:8080>

2) Crear un Dockerfile para que instale node.

- a. Imagen base node:alpine
- b. Establecer un directorio de trabajo, por ejemplo, app
- c. Crear un fichero de javascript que imprima un mensaje por la consola.
- d. Copiar el fichero js al directorio app
- e. Lanzar el comando: node mensaje.js
- f. Construir la imagen en el directorio actual.

`docker build -t nombre_img:tag .`

- g. Verificar si se ha creado la imagen. Listar las imágenes.
- h. Crear un contenedor, y comprobar si muestra el mensaje.

3) Instalar Python, con la librería Flask, desplegar un servicio por el puerto 5000.

- a. Código Python

```
from flask import Flask, request
app = Flask(__name__)
@app.route('/saludo', methods=['GET'])
def saludo():
    nombre = request.args.get('nombre', 'Mundo')
```

```
return f"¡Hola, {nombre}!"  
if __name__ == '__main__':  
    app.run(host='0.0.0.0', port=5000)
```

b. Hace falta otro fichero adicional que tenga el nombre “flask” que serán: las librerías a instalar (en este caso solo una).

c. Pasos:

- i. Imagen base: Python:3.9-slim
- ii. Directorio de trabajo
- iii. Copiar el fichero de requisitos
- iv. Se ejecuta el comando: `pip install --no-cache-dir -r requisitos.txt`
- v. Copiar un fichero de Python con el código adjunto
- vi. Exponer el puerto 5000
- vii. Lanzar el comando `python` y como parámetro el fichero `python`.
- viii. Crear la imagen, contenedor
- ix. <http://localhost:5000>

4) Crear un contenedor basado en la imagen de tomcat y desplegar la aplicacion1.war. Dejar el puerto 8080 para tomcat. La carpeta donde hay que alojar el fichero war es: `/usr/local/tomcat/webapps`.