

Práctica de Docker: Despliegue y Análisis de una Aplicación Flask

Objetivo: Utilizar el `Dockerfile` creado para construir y ejecutar una imagen Docker, probar la aplicación, y analizar las capas de la imagen con `docker history`.

Paso 1: Crear el `Dockerfile`

Usa el siguiente `Dockerfile` para esta práctica:

```
# Usa una imagen base de Python oficial
FROM python:3.9-slim

# Añade metadatos a la imagen con LABEL
LABEL maintainer="tucorreo@example.com"
LABEL version="1.0"
LABEL description="Imagen de ejemplo para un taller Docker con una aplicación Flask"

# Establece el directorio de trabajo en /app
WORKDIR /app

# Copia los archivos de la aplicación a /app
COPY . .

# Instala las dependencias necesarias
RUN pip install --no-cache-dir -r requirements.txt

# Expone el puerto 5000 para la aplicación
EXPOSE 5000

# Comando para ejecutar la aplicación
CMD ["python", "app.py"]
```

Paso 2: Crear el Código de la Aplicación Flask

1. Crea un archivo `app.py` con el siguiente contenido:

```
# app.py
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "¡Hola desde un contenedor Docker!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

2. Crea un archivo `requirements.txt` con el siguiente contenido:

```
Flask==2.0.3
Werkzeug==2.0.3
```

Paso 3: Construir la Imagen Docker

Ejecuta el siguiente comando para construir la imagen:

```
docker build -t taller-python-app .
```

- **Explicación:**

- `-t taller-python-app` : Asigna un nombre a la imagen.
- `.` : Especifica que el contexto de la construcción es el directorio actual.

Paso 4: Ejecutar el Contenedor

Inicia un contenedor usando la imagen que has creado:

```
docker run -d -p 5000:5000 --name mi_taller_app taller-python-app
```

- **Explicación de los parámetros:**

- `-d` : Ejecuta el contenedor en segundo plano.
 - `-p 5000:5000` : Mapea el puerto 5000 del contenedor al puerto 5000 de tu máquina local.
 - `--name mi_taller_app` : Asigna un nombre específico al contenedor.
-

Paso 5: Probar la Aplicación

Abre un navegador e ingresa la siguiente URL:

```
http://tidpnube-b.rediris.es:5000
```

Deberías ver el mensaje: **¡Hola desde un contenedor Docker!**.

////////////////////////////////////

Paso 6: Detener y Eliminar el Contenedor

Cuando hayas terminado de probar la aplicación, detén y elimina el contenedor:

```
docker stop mi_taller_app  
docker rm mi_taller_app
```

////////////////////////////////////

Paso 7: Ver el Historial de la Imagen

Para ver las capas creadas en la imagen, ejecuta el siguiente comando:

```
docker history taller-python-app
```

- **Explicación:** `docker history` muestra las capas de la imagen, incluyendo los comandos usados para crearlas y el tamaño de cada capa.
- ////////////////////////////////////

Conclusión

En esta práctica, has construido y ejecutado una aplicación Flask usando Docker. También has aprendido a inspeccionar las capas de la imagen para entender cómo se construyó. ¡Bien hecho!