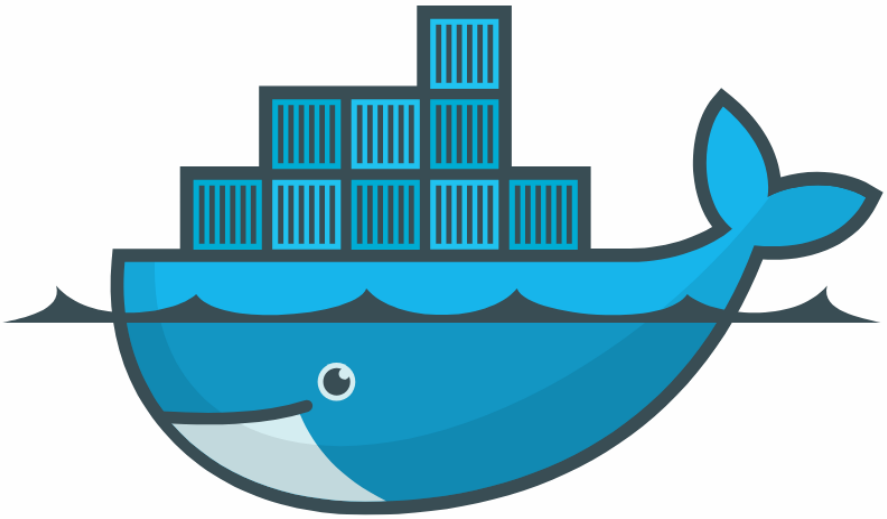
CREANDO DOCKERFILE



Ivana Sánchez Pérez

INTROCUCCIÓN

Docker es una plataforma de código abierto que permite desarrollar, enviar y ejecutar aplicaciones en contenedores. Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias, de modo que la aplicación se ejecute de manera rápida y confiable en diferentes entornos informáticos.

¿Por qué usar Docker?

1. **Consistencia**: Los contenedores garantizan que las aplicaciones se ejecuten de la misma manera en cualquier entorno, ya sea en desarrollo, pruebas o producción.
2. **Aislamiento**: Cada contenedor es independiente, lo que significa que las aplicaciones pueden ejecutarse sin interferir entre sí, incluso si usan versiones diferentes de las mismas bibliotecas.
3. **Escalabilidad**: Docker simplifica el escalado de aplicaciones al permitir el despliegue rápido de contenedores en múltiples instancias.
4. **Eficiencia**: Los contenedores comparten el mismo núcleo del sistema operativo, lo que los hace más ligeros en comparación con las máquinas virtuales, que requieren un sistema operativo completo.

Componentes Clave

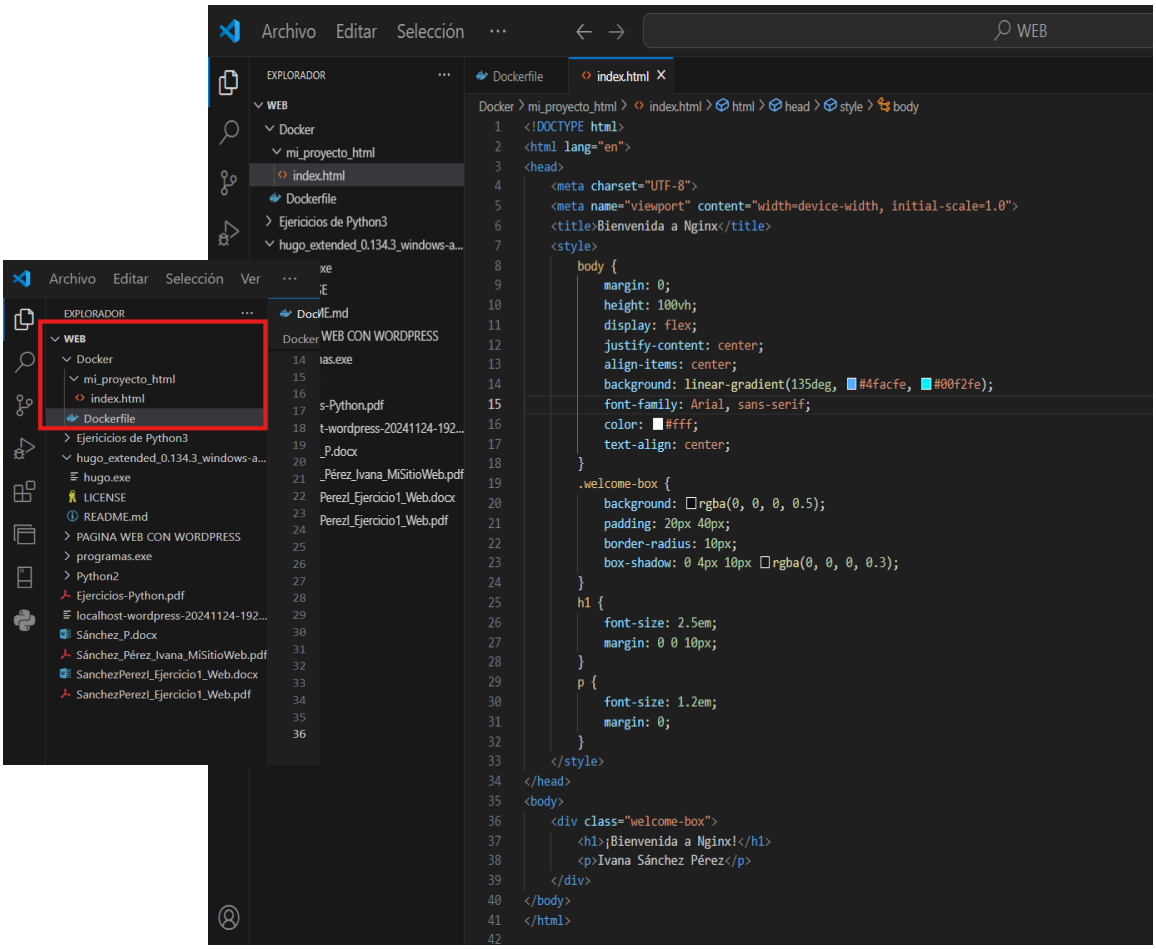
- **Docker Engine**: El núcleo de Docker que permite la creación y gestión de contenedores.
- **Docker Hub**: Un registro en la nube donde se pueden almacenar y compartir imágenes de contenedores.
- **Docker Compose**: Una herramienta para definir y ejecutar aplicaciones multicontenedor mediante un archivo de configuración YAML.

Conclusión

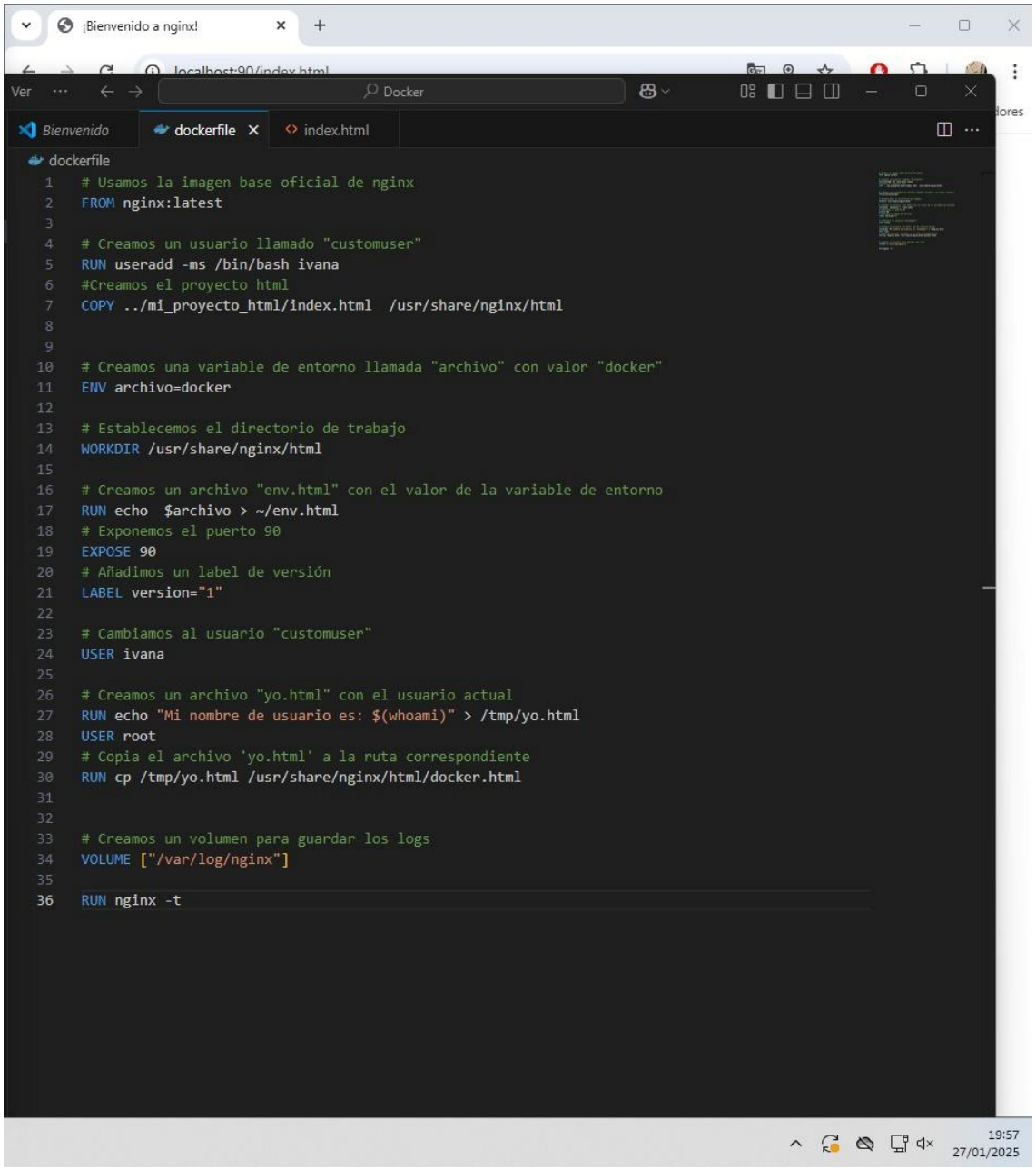
Docker ha revolucionado la forma en que se desarrollan y despliegan aplicaciones, proporcionando una solución eficiente, portátil y escalable para la gestión de software. Con su creciente popularidad, aprender a usar Docker se ha vuelto una habilidad esencial para desarrolladores y administradores de sistemas.

TAREA: Creando Dockerfile completo

Lo primero que he hecho ha sido crear la estructura de carpetas y archivos, que he necesitado para realizar la tarea. He comenzado por la carpeta principal a la que he llamado Docker, y dentro de ella el archivo Dockerfile y la carpeta mi_proyecto_html; en esta última es dónde he ubicado mi archivo index.html, donde he creado un html para el inicio de la página de Nginx.



Y ahora procedo a configurar el archivo Dockerfile tal y como se muestra en la imagen.



EXPLICACIÓN DE LAS INSTRUCCIONES

1. **FROM nginx:latest**
    - Especifica la imagen base que se utilizará para construir la imagen Docker. En este caso, se usa la imagen oficial de Nginx en su versión más reciente.

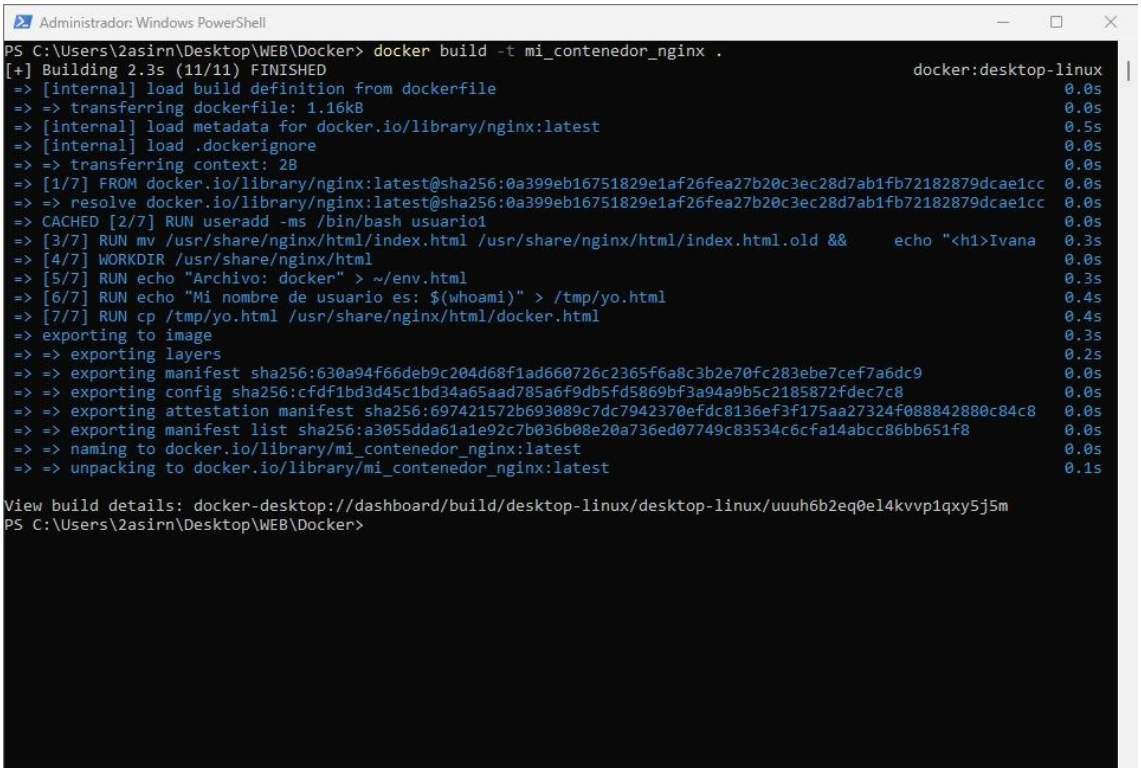
2. **RUN useradd -ms /bin/bash ivana**
    - Crea un usuario llamado "ivana" con un shell Bash como predeterminado.
3. **COPY ./mi_proyecto_html/index.html /usr/share/nginx/html**
    - Copia un archivo HTML desde el sistema anfitrión a la carpeta de Nginx donde se almacenan los archivos estáticos.
4. **ENV archivo docker**
    - Define una variable de entorno llamada "archivo" con el valor "docker".
5. **WORKDIR /usr/share/nginx/html**
    - Cambia el directorio de trabajo dentro de la imagen a "/usr/share/nginx/html".
6. **RUN echo $archivo > ./env.html**
    - Crea un archivo llamado "env.html" en el directorio actual y escribe el valor de la variable de entorno "archivo" en este archivo.
7. **EXPOSE 90**
    - Expone el puerto 90 para permitir el acceso al servidor Nginx desde el exterior del contenedor.
8. **LABEL version="1"**
    - Añade metadatos a la imagen, indicando que esta tiene la versión "1".
9. **USER ivana**
    - Cambia el usuario que ejecutará las siguientes instrucciones al usuario "ivana" creado previamente.
10. **RUN echo "Mi nombre de usuario es: $(whoami)" > /tmp/yo.html**
    - Crea un archivo temporal "yo.html" que contiene el nombre del usuario actual ("ivana").
11. **USER root**
    - Cambia de nuevo al usuario "root" para realizar operaciones que requieren permisos elevados.
12. **RUN cp /tmp/yo.html /usr/share/nginx/html/docker.html**

- Copia el archivo "yo.html" a la carpeta pública de Nginx y lo renombra como "docker.html".

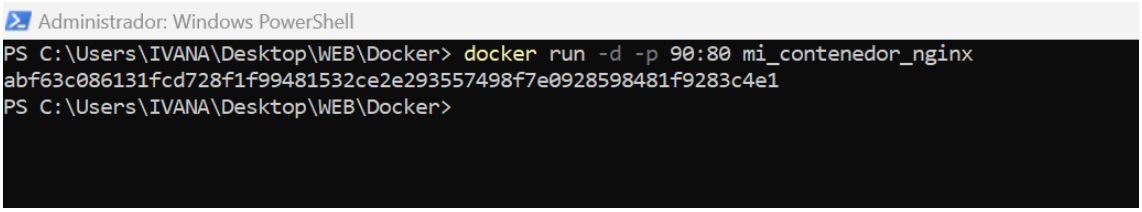
13. **VOLUME ["/var/log/nginx"]**
    - Crea un volumen para almacenar los registros de Nginx fuera del contenedor.
14. **RUN nginx -t**
    - Verifica la configuración de Nginx para asegurarse de que no haya errores.

Demostración del funcionamiento

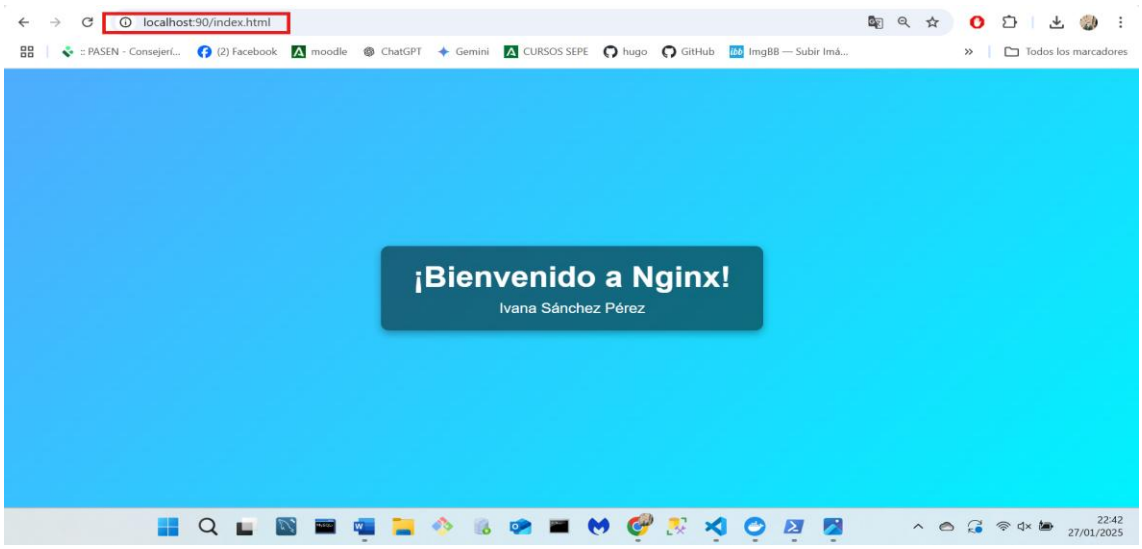
1. **Construcción de la imagen Docker**
    - Abrimos la PowerShell como administrador
    - Hacemos un cd para ubicarnos es la carpeta principal “Docker”
    - Ejecutamos el comando docker build -t mi_contenedor_nginx .



2. **Ejecución del contenedor**
    - Una vez construida la imagen, inicia un contenedor basado en ella:
    - docker run -d -p 90:80 mi_contenedor_nginx
    - Esto iniciará el servidor Nginx y lo expondrá en el puerto 90.



3. **Verificación de los archivos creados**
    - Abrimos un navegador web y accedemos a las siguientes URLs:
        - http://localhost:90/index.html: Debería mostrar el contenido del archivo "index.html" copiado al contenedor.



        - http://localhost:90/docker.html: Mostrará el texto "Mi nombre de usuario es: ivana".