Introducción a Docker  
UD 06. Caso práctico 04 - Whisper AI y Web Whisper

short line

Autor: Sergi García Barea

Actualizado Marzo 2023

Licencia

**Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA)**: No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

📖 **Importante**

❕ **Atención**

💬 **Interesante**

[**1. Introducción 2**](#_dbh0n1vac4c8)

[**2. Pasos a previos para poner en marcha Whisper AI y Whisper Web 3**](#_vyhbfp4t666x)

[**3. Poniendo en marcha Whisper AI y Whisper Web 3**](#_iwctescm4nq0)

[**4. Bibliografía 3**](#_72yv1gty1dke)

UD06. Caso práctico 04

# Introducción

Whisper AI es un software que permite transcribir audio a Texto. Este software fue creado por OpenAI (empresa creadora de ChatGPT) y liberado junto con varios modelos de procesamiento de lenguaje muy precisos. Actualmente, es una de las mejores soluciones “self-hosted” para convertir audio a texto.

Whisper Web es una web, que proporciona una interfaz gráfica web para mandar audios a transcribir a Whisper AI. Su repositorio es <https://codeberg.org/pluja/web-whisper>

En este caso práctico vamos a poner en marcha una versión Dockerizada extraido de su repositorio siguiendo las instrucciones de <https://codeberg.org/pluja/web-whisper/wiki/Self-Hosting>

# Pasos a previos para poner en marcha Whisper AI y Whisper Web

En el propio sitio web, nos piden que clonemos el repositorio con el comando:

| git clone https://codeberg.org/pluja/web-whisper |
| --- |

**Nota:** en el curso os proporcionamos el resultado de clonar este repositorio en un fichero .zip.

Tras ello, copiamos el fichero de ejemplo a un “docker-compose.yml” con un comando similar a:

| cp example.docker-compose.yml docker-compose.yml |
| --- |

Tras ello, también copiaremos y renombraremos el fichero “.env” de forma similar a:

| cp example.env .env |
| --- |

Para adaptar la configuración, deberemos adaptar el fichero “.env”. Como adaptarlo se detalla en

<https://codeberg.org/pluja/web-whisper/wiki/Self-Hosting#user-content-configuration>

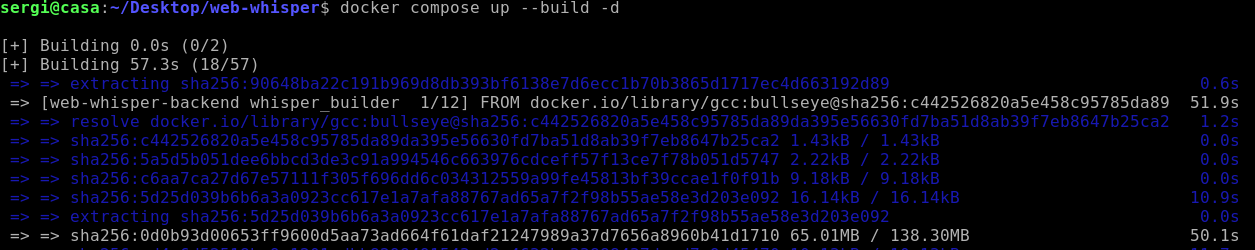
Pero con los parámetros con defecto, podemos tener un funcionamiento correcto.

# Poniendo en marcha Whisper AI y Whisper Web

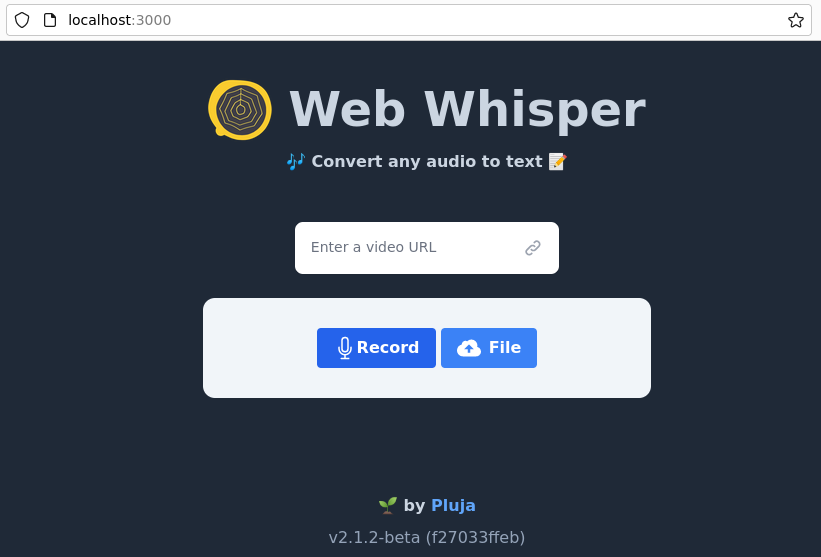
Una vez preparados los ficheros “docker-compose.yml” y “.env”, ejecutaremos:

| docker compose up --build -d |
| --- |

Obteniendo algo similar a:



Cuando haya finalizado, tendremos Whisper Web con Whisper AI funcionando en <http://localhost:3000> obteniendo algo similar a:



# Bibliografía

[1] Docker Docs <https://docs.docker.com/>

[2] Docker Compose Docs <https://docs.docker.com/compose/>