Instrucciones Docker

Planificación y Gestión de Proyectos Informáticos

https://github.com/pabsanper/ACME-Cycling.git

GRUPO 3.8

Alberto Benitez Morales

David Sabugueiro Troya

Pablo Santos Pérez

Francisco Javier Vázquez Monge

Álvaro Paradas Borrego

CLIENTE: José González Enríquez

FECHA:05/12/2022

Control de cambios

Versión	Fecha	Tipo	Responsables	Descripción	Motivación
1.0	05/12/2022	Creación	Álvaro Paradas	Creación y desarrollo del documento	Creación de unas buenas instrucciones para dockerizar el proyecto software

Tabla de contenidos

1. Instrucciones para Docker

En la rama Docker se encuentran los ficheros necesarios para lanzar la aplicación en los contenedores de Docker preparados para ello por primera vez:

Si en el proyecto no se ha creado un espacio virtual con Python, procedemos a crearlo con el siguiente comando: **python3 -m venv venv**
Entramos dentro del espacio virtual: **source ./venv/bin/activate**
Instalamos Django, (en nuestro caso la versión 3), dentro del espacio virtual:
pip install django==3.0.0
Construimos la imagen de Docker: **docker build -t devrrior/docker-acmecycling .**
Ejecutamos el contenedor: **docker run -p 8000:8000 devrrior/docker-acmecycling**

PASO ADICIONAL:
Compartimos el contenido local con el acceso a docker:
docker run -d -v /"directorioindicado en el fichero Dockerfile"/:/"directorio workspace en Dockerfile" -p 8000:8000 devrrior/docker-acmecycling
El comando anterior nos retorna el id del contenedor ya en ejecución, pero no nos muestra los logs de estado de procesos como en el caso de ejecutar el contenedor.

Para parar el contenedor después de usar el comando anterior utilizamos el siguiente comando: **docker stop "idContenedor"**

EJECUTAR MIGRACIONES DE LOS CAMBIOS EN LA BBDD:
A la hora de realizar cambios en la base de datos debemos de indicarle a la base de datos contenida en el contenedor que tiene que hacer un refresco de los datos, para ello:
Dentro del espacio virtual:
docker exec -it "idContenedor" /bin/sh
python manage.py migrate
Para salir del contenedor pulsamos Ctrl+d

Una vez esté todo preparado se ejecutan los siguientes comandos:
docker-compose up -d para levantar el contenedor
docker-compose exec postgres sh para entrar dentro del contenedor
python manage.py migrate para realizar un migrate de la base de datos
docker-compose down para apagar el contenedor

2. Pasos para levantar docker sin la imagen

Tras la explicación de los distintos comandos en el punto anterior, aquí se listan los comandos necesarios para levantar un contenedor con Docker y los ficheros localizados en el proyecto:
Teniendo en cuenta si existe un entorno virtual o no ejecutamos el siguiente comando, (en caso de no tener ninguno):

- python3 -m venv “nombreEntornoVirtual”

A partir de aquí se ejecutan todos los comandos:

- source ./venv/bin/activate
- pip install -r requirements.txt
- docker build -t devrrior/docker-acmecycling .
- docker run -p 8000:8001 devrrior/docker-acmecycling
- ctrl+c
- docker-compose up -d
- docker ps (comprobar que los dos contenedores se están ejecutando correctamente)
- docker-compose exec “nombreContenedor” sh
- python manage.py migrate
- ctrl+d
- docker-compose down

3. Pasos si se tiene la imagen .tar

- docker load -i acme-cycling.tar
- docker run -p 8000:8001 acme-cycling:latest