Demo 1

Hola mundo desde docker usando python

Pre-requisitos

- Instalar docker. Tutorial

Descripción del ejemplo

La idea es ejecutar un "Hola mundo" usando python desde un contenedor docker. No tiene mayor complejidad pero sirve para tener una idea básica de la dinámica de trabajo.

Ejemplo A (el más básico)

Abrir una terminal y escribir (internet):

```
    docker run --rm -it python:3 python
```

Ahora en la misma terminal de antes ejecutar (local):

```
docker run --rm -it 192.168.0.15:5000/python:3 python
```

Qué hicimos ahí?

- `docker run` sirve para ejecutar un contenedor.
- `--rm` indica que se elimina apenas lo terminamos de usar, no "queda guardado" para usarlo después.
- `-it` indica que el contenedor va a "tomar" la terminal en donde estamos ejecutando el comando y vamos a poder interactuar con él.
- `python:3` es el nombre de la imagen con la que queremos generar el contenedor.
- `python` es el comando que queremos ejecutar cuando inicia el contenedor, en este caso el intérprete interactivo de python.

En qué resulta todo eso?

Tenemos python ejecutándose en una versión (*3.x en este caso*) idéntica en todas las estaciones de trabajo de los *devs*. Pero esto es muy básico, veamos algo más elaborado.

Ejemplo B (Una salida más elegante)

Generar un archivo **hola.py** con el siguiente contenido:

```
print('Hola mundo!')
```

Ahora en la misma terminal de antes ejecutar (internet):

```
docker run --rm -it -v $(pwd):/src python:3 python /src/hola.py
```

Ahora en la misma terminal de antes ejecutar (local):

```
docker run --rm -it -v $(pwd):/src 192.168.0.15:5000/python:3 python
/src/hola.py
```

Qué hay de nuevo?

- `-v $(pwd):/src` indica que queremos montar un **volumen** en el contenedor, que es básicamente la forma de dar persistencia a los archivos. En este caso hace que se refleje el contenido de la carpeta actual (`$(pwd)` en la ubicación `/src`).
- Como el archivo **hola.py** ahora está en el contenedor, el final del comando cambia para pedir que el intérprete ejecute ese archivo directamente.

Mismo resultado, diferentes formas:

En este caso, el código a ejecutar está en un archivo, que podrían ser varios, y al momento de construir el contenedor se copia hacia el mismo y se ejecuta con todo el ambiente (librerías, frameworks, configuraciones) que esté definido dentro del contenedor.