Despliegue de servicios con Docker

Componentes:

* **Imagen:** Software disponible para ser usado
* **Contenedor:** Se crea al ejecutar de una imagen (un proceso)
* **Depósito:** ‘Almacén’ de imágenes donde coexisten las imágenes para dejar o obtener.

El objetivo es crear una imagen por cada componente

Ejemplo (Ubuntu como S.O. + Node.js + Zeromq + código para el cliente = Imagen cliente)

Docker utiliza órdenes de consola:

***Docker acción opciones argumentos***

**Lo + importante de todo:**

A partir de una imagen se puede hacer ***docker run -i -t imagenBase sh*** e interactuar con el nuevo contenedor y meterlo cosas seguido de un *exit* y un ***docker commit idContenedor nuevaImagen*** para pasarlo a una imagen nueva (plantilla de una imagen nueva con más cosas) o bien se crea un dockerfile y luego se hace un ***docker build -t nuevaImagen*** para conseguir el mismo resultado (esta alternativa es más automatizada y mejor que hacerlo a mano, sobre todo porque se puede modificar con facilidad y se puede usar para otras imágenes base).

Diagram

Description automatically generated

**El famoso Dockerfile:**  
La 1ª instrucción es una orden ***FROM imagenBase***.

***RUN orden*** ejecuta una instrucción cmd.

***COPY origen destino*** (copia URL/directorios/archivo de A -> B)

***EXPOSE******puerto*** indica el puerto en el cual el contenedor escucha.

***CMD orden arg1 arg2…*** proporciona valores por defecto para la orden.

***ENTRYPOINT orden arg1 arg2…*** es un CMD que se ejecuta como 1ª orden al crear un contenedor.

Nota: Solo se debería de usar como máximo 1 orden CMD o ENTRYPOINT.