Decroly

SANTANDER

SYSTEM

DOCKERS

DOCKERS

A. **Documentación**:

https://docs.docker.com/desktop/setup/install/linux/

Docker es un proyecto open source que ha revolucionado la manera de desarrollar software gracias a la sencillez con la que permite gestionar contenedores. Los contenedores LXC (LinuX Containers) son un concepto relativamente antiguo y utilizado desde hace tiempo por grandes empresas como Amazon o Google, pero cuyo gestión era complicada. Sin embargo, Docker define APIs y herramientas de línea de comandos que hacen casi trivial la creación, distribución y ejecución de contenedores. De ahí que el lema de Docker sea: "Build, Ship and Run. Any application, Anywhere" y se haya convertido en una herramienta fundamental tanto para desarrolladores como para administradores de sistemas.

Podríamos definir un contenedor Docker como una máquina virtual ligera, que corre sobre un sistema operativo Linux pero con su propio sistema de ficheros, su propio espacio de usuarios y procesos, sus propias interfaces de red, … por lo que se dice que son sistemas aislados.

Características de Dockers

1- Portabilidad

Un contenedor Docker es ejecutado por lo que se denomina el Docker Engine, un demonio que es fácilmente instalable en prácticamente todas las distribuciones Linux. Un contenedor ejecuta una imagen de docker, que es una representación del sistema de ficheros y otros metadatos que el contenedor va a utilizar para su ejecución. Una vez que hemos generado una imagen de Docker, ya sea en nuestro ordenador o vía una herramienta externa, esta imagen podrá ser ejecutada por cualquier Docker Engine, independientemente del sistema operativo y la infraestructura que haya por debajo

2- Inmutabilidad

Una aplicación la componen tanto el código fuente como las librerías del sistema operativo y del lenguaje de programación necesarias para la ejecución de dicho código. Estas dependencias dependen a su vez del sistema operativo donde nuestro código va a ser ejecutado, y por esto mismo ocurre muchas veces aquello de que “no sé, en mi máquina funciona”. Sin embargo, el proceso de instalación de dependencias en Docker no depende del sistema operativo, si no que este proceso se realiza cuando se genera una imagen de docker. Es decir, una imagen de docker (también llamada repositorio por su parecido con los repositorios de git) contiene tanto el código de la aplicación como las dependencias que necesita para su ejecución. Una imagen se genera una vez y puede ser ejecutada las veces que sean necesarias, y siempre ejecutará con las misma versión del código fuente y sus dependencias, por lo que se dice que es inmutable. Si unimos inmutabilidad con el hecho de que Docker es portable, decimos que Docker es una herramienta fiable, ya que, una vez generada una imagen, ésta se comporta de la misma manera independientemente del sistema operativo y de la infraestructura donde se esté ejecutando.

3- Ligereza

Los contenedores Docker que corren en la misma máquina comparten entre ellos el sistema operativo, pero cada contenedor es un proceso independiente con su propio sistema de ficheros y su propio espacio de procesos y usuarios (para este fin Docker utiliza cgroups y namespaces, recursos de aislamiento basados en el kernel de Linux). Esto hace que la ejecución de contenedores sea mucho más ligera que otros mecanismos de virtualización. Comparemos por ejemplo con otra tecnología muy utilizada como es Virtualbox. Virtualbox permite del orden de 4 ó 5 máquinas virtuales en un ordenador convencional, mientras que en el mismo ordenador podremos correr cientos de containers sin mayor problema, además de que su gestión es mucho más sencilla.

Componentes de Dockers

Docker Engine

Es un demonio que corre sobre cualquier distribución de Linux y que expone una API externa para la gestión de imágenes y contenedores (y otras entidades que se van añadiendo en sucesivas distribuciones de docker como volúmenes o redes virtuales). Sus funciones principales son:

- Creación de imágenes docker.
- Publicación de imágenes en un Docker Registry o Registro de Docker (otro componente Docker que se explicará a continuación).
- Descarga de imágenes desde un Registro de Docker.
- Ejecución de contenedores usando imágenes locales.

Docker Client

Es cualquier herramienta que hace uso de la api remota del Docker Engine, pero suele hacer referencia al comando docker que hace las veces de herramienta de línea de comandos (cli) para gestionar un Docker Engine. La cli de docker se puede configurar para hablar con un Docker Engine local o remoto, permitiendo gestionar tanto nuestro entorno de desarrollo local, como nuestros servidores de producción. Los comandos de docker más comunes son:

- docker info: da información acerca de la cantidad de contenedores e imágenes que está gestionando la máquina actual, así como los plugins actualmente instalados.
- docker images: lista información de las imágenes que se encuentran disponibles en la máquina (nombre, id, espacio que ocupa, el tiempo desde que fue creada).
- docker build: crea una imagen desde el fichero Dockerfile del directorio actual.
- docker pull: descarga en la máquina actual la versión de la imagen indicada. En caso de no indicar la versión descarga todas las que estén disponibles.
- docker push: sube la versión de la imagen indicada a un Registro de Docker, permitiendo su distribución a otras máquinas.
- docker rmi: elimina una imagen de la máquina actual.
- docker run: crea un contenedor a partir de una imagen. Este comando permite multitud de parámetros, que son actualizados para cada versión del Docker Engine, por lo que para su documentación lo mejor es hacer referencia a la página oficial.
- docker ps: muestra los contenedores que están corriendo en la máquina. Con el flag -a muestra también los contenedores que están parados.

- docker inspect contenedor: muestra información detallada de un contenedor en formato json. Se puede acceder a un campo particular con el comando docker inspect -f '{{.Name}}' contenedor .
- docker stop contenedor: para la ejecución de un contenedor.
- docker start contenedor: reanuda la ejecución de un contenedor.
- docker rm contenedor: elimina un contenedor. Para borrar todos los contenedores de una máquina se puede ejecutar el comando docker rm -fv $(docker ps -aq) .
- docker logs contenedor: muestra los logs de un contenedor.
- docker stats contenedor: muestra las estadísticas de ejecución de un contenedor, como son la memoria utilizada, la CPU, el disco…
- docker exec contenedor comando: ejecuta un comando en un contenedor. Útil para depurar contenedores en ejecución con las opciones docker exec -it contenedor bash .
- docker volume ls: lista los volúmenes existentes en la máquina. Para un listado completo de los comandos relacionados con volúmenes ejecuta docker volume --help .
- docker network ls: lista las redes existentes en la máquina. Para un listado completo de los comandos relacionados con redes ejecuta docker network --help .

Docker Registry

El Registro es otro componente de Docker que suele correr en un servidor independiente y donde se publican las imágenes que generan los Docker Engine de tal manera que estén disponibles para su utilización por cualquier otra máquina. Es un componente fundamental dentro de la arquitectura de Docker ya que permite distribuir nuestras aplicaciones.

El Registro de Docker es un proyecto open sourceque puede ser instalado gratuitamente en cualquier servidor, pero Docker ofrece Dockerhub, un sistema SaaS de pago donde puedes subir tus propias imágenes, acceder a imágenes públicas de otros usuarios, e incluso a imágenes oficiales de las principales aplicaciones como son: MySQL, MongoDB, RabbitMQ, Redis, etc. El registro de Docker funciona de una manera muy parecida a git (de la misma manera que Dockerhub y us métodos de pago funcionan de una manera muy parecida a Github). Cada imagen, también conocida como repositorio, es una sucesión de capas. Es decir, cada vez que hacemos un build en local de nuestra imagen, el Registro de Docker sólo almacena el diff respecto de la versión anterior, haciendo mucho más eficiente el proceso de creación y distribución de imágenes.