

Intrroducción a Docker

Docker es una herramienta que puede empaquetar una aplicación y sus dependencias en un contenedor virtual que se puede ejecutar en cualquier servidor Linux. Esto ayuda a permitir la flexibilidad y portabilidad en donde la aplicación se puede ejecutar, ya sea en las instalaciones físicas, la nube publica, nube privada, etc.

Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de Virtualización a nivel de sistema operativo.

La idea detrás de Docker es crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo, facilitando así también los despliegues

Para aquellos que no lo posean instalado, les dejo la documentación oficial para las distintas plataformas: ¿Cómo instalo Docker?

<https://docs.docker.com/engine/installation/>

Entre las principales ventajas para trabajar con un entorno virtualizado podemos destacar:

- Trabajar con un entorno virtualizado te permite aislar tu proyecto de la infraestructura del SO que tengas por abajo, permitiendo probar varias configuraciones sin tener que desinstalar e instalar paquetes en tu máquina host.
- Te permite tener un entorno que puedas escalar fácilmente.

Comandos útiles de docker

En este apartado se mencionan una serie de comandos de relativa importancia para el uso de Docker. (La mayoría de ellos deben ser precedido por el comando sudo).

\$ docker-compose build: Construye o reconstruye los servicios. Los servicios son contruidos una unica vez y luego son etiquetados como project_service. Si se modifica el Dockerfile de un servicio o el contenido del directorio de construcción se debe ejecutar de nuevo este comando para reconstruirlo.

\$ docker-compose up: Construye, inicia, y agrega los contenedores para un servicio.

\$ docker-compose start: Inicia los contenedores existentes para un servicio.

\$ docker-compose stop: Detiene los contenedores que se están ejecutando sin eliminarlos. Pueden volverse a iniciar con el comando anterior (start).

\$ docker ps -a: Muestra información de todos los contenedores que existen actualmente y en que estado se encuentran, ademas de otra información adicional.

\$ docker rm <Contanider ID>: Este comando permitirá eliminar un contenedor.

\$ docker images: Muestra información acerca de cada una de las imágenes que se encuentran en nuestra maquina (nombre, id, espacio que ocupa, el tiempo que transcurrió desde que fue creada).

\$ docker rmi <Image ID>: Este comando permitirá eliminar una imagen.

Referencias

Documentación oficial Docker.

<https://docs.docker.com/>

Información básica de instalación Docker-engine:

<https://docs.docker.com/engine/installation/linux/ubuntu/linux/>

Información básica de instalacion Docker-compose:

<https://docs.docker.com/compose/install/>