¿Qué es un contenedor?

Un contenedor es una unidad liviana y autónoma que encapsula una aplicación y todas sus dependencias. Piénsalo como una aplicación que se empaqueta y puedes mover/ejecutar en cualquier lugar que tenga Docker.

Estos “paquetes” que llamamos contenedores contienen todo lo necesario para que una aplicación funcione: el código, las bibliotecas y otras dependencias.

¿Qué es Docker?

Docker es una plataforma que simplifica el desarrollo, la creación y ejecución de aplicaciones al utilizar "contenedores".

Es la herramienta de código abierto que nos permite trabajar con contenedores.

Características de los contenedores

- **Portabilidad**: Puedes llevar el contenedor a cualquier lugar y ejecutar la aplicación sin preocuparte por las diferencias en el entorno.

- **Eficiencia**: Los contenedores son eficientes en recursos porque comparten el mismo sistema operativo subyacente.

- **Consistencia**: Al encapsular todo, los contenedores aseguran que la aplicación funcione de manera consistente en cualquier lugar.

Cómo funciona realmente docker

Imagina que eres un programador y estás desarrollando una aplicación genial en tu computadora. Esta aplicación necesita ciertas bibliotecas y herramientas especiales para funcionar correctamente.

Ahora, aquí está el truco: lo que funciona en tu computadora no siempre funciona igual en la computadora de otra persona o en un servidor en línea. Puede haber diferencias en las versiones de las bibliotecas, en la configuración del sistema, etc.

Es ahí donde entra Docker. Con Docker, puedes "empaquetar" tu aplicación junto con todas las bibliotecas y herramientas que necesita en lo que llamamos un "contenedor". Este contenedor es como una caja mágica que contiene todo lo que tu aplicación necesita para funcionar, y lo hace de manera consistente en cualquier lugar donde se ejecute Docker.

Entonces, como programador, puedes estar seguro de que si tu aplicación funciona en un contenedor Docker en tu computadora, también funcionará de la misma manera en la

computadora de otro desarrollador o en un servidor en línea. Docker garantiza que todo lo que necesitas esté disponible y funcione de manera consistente, sin importar dónde se ejecute tu aplicación.

Funcionamiento Interno de Docker

Docker Daemon: El Docker Daemon es el motor de Docker que gestiona la creación, ejecución y gestión de contenedores. Escucha comandos de clientes Docker a través de un socket Unix o una interfaz de red.

Cliente Docker: El Cliente Docker es una interfaz de línea de comandos que permite a los usuarios interactuar con el Docker Daemon. Cuando ejecutas comandos como docker run o docker build, el Cliente Docker envía solicitudes al Daemon a través de la API de Docker.

API de Docker: La API de Docker es la interfaz de programación de aplicaciones que permite la comunicación entre el Cliente Docker y el Docker Daemon. Los comandos del Cliente Docker se traducen en solicitudes a la API que el Daemon entiende y procesa.

Contenedores: Los contenedores son instancias aisladas de aplicaciones que se ejecutan en el sistema operativo subyacente. Docker utiliza características de aislamiento del kernel Linux, como cgroups y namespaces, para garantizar la segregación de recursos entre los contenedores.

Imágenes de Docker: Las imágenes de Docker son plantillas que contienen aplicaciones y sus dependencias. Los contenedores se crean a partir de estas imágenes. Las imágenes se almacenan en caché localmente y se descargan de registros (como Docker Hub) cuando sea necesario.

Volúmenes de Docker: Los volúmenes permiten el almacenamiento persistente de datos entre contenedores y a través de reinicios. Los volúmenes son montados en contenedores y pueden contener datos de aplicaciones, bases de datos, archivos de configuración, etc.

Directorios Importantes de Docker

/var/lib/docker: Este directorio es el lugar donde Docker almacena todos los datos relacionados con contenedores e imágenes. Incluye subdirectorios como "containers" (donde se encuentran los datos de los contenedores en ejecución), "images" (donde se almacenan las imágenes descargadas) y "volumes" (para datos persistentes de contenedores).

/etc/docker: En este directorio se encuentran archivos de configuración de Docker. El archivo "daemon.json" es especialmente importante y se utiliza para configurar el comportamiento del Docker Daemon.

/var/run/docker.sock: Este es un socket Unix que permite la comunicación entre el cliente Docker y el Docker Daemon. Los clientes Docker (como la línea de comandos de Docker)

se comunican con el Daemon a través de este socket para enviar comandos y recibir resultados.

/var/run/docker.pid: Aquí se almacena el identificador del proceso del Docker Daemon.

/etc/default/docker (en sistemas basados en Debian/Ubuntu): Este archivo se utiliza para configurar variables de entorno para el Docker Daemon, como opciones de red.

/etc/sysconfig/docker (en sistemas basados en Red Hat/CentOS): Similar a /etc/default/docker, este archivo se usa para configurar variables de entorno para el Docker Daemon.