TIC

Lección 4: Uso de contenedores con Docker.

Uso de contenedores con Docker

Docker representa una tecnología revolucionaria en el ámbito de la virtualización y la gestión de contenedores. Se puede comparar Docker con una especie de "contenedor virtual", una abstracción que encapsula tanto la aplicación como sus dependencias y entorno de ejecución en una unidad autónoma y portable. Esta unidad puede ejecutarse en cualquier sistema operativo resolviendo la problemática de la inconsistencia entre entornos de desarrollo y producción, proporcionando una solución eficiente y estandarizada. Es similar a poder encapsular todas las dependencias en un computador virtualizado, auto conteniendo sus librerías, rutas, entornos, programas instalados y archivos almacenados. Los contenedores generados por Docker comparten el mismo kernel del sistema operativo anfitrión, lo que los hace más eficientes en recursos y rápidos de iniciar en comparación con máquinas virtuales. Con esto Docker se ha convertido en un estándar para el despliegue de aplicaciones y la gestión de recursos virtualizados.

Para citar un ejemplo, cuando se tiene un aplicativo de análisis de datos, que requiere conexión a bases de datos, aplicaciones de visualización y de procesamiento de datos y, cada unidad es desarrollada por una persona o equipo diferente, lo que se puede hacer es emplear Docker para auto contener cada sección de un aplicativo. La facilidad está en que Docker solo requiere de un archivo llamado Dockerfile, el cual indica qué dependencias tendrá el contenedor y permite que, desde la nube o localmente se puedan recrear tantas copias como sea necesario automáticamente. Solucionando la escalabilidad y practicidad.

¿Qué es Docker?

Docker es una plataforma de código abierto para la creación y gestión de contenedores. Aunque los contenedores son un concepto propio de Linux, Docker posibilita su uso en diferentes sistemas operativos (Windows, Mac y Linux).

TIC

¿Cómo se diferencia de las máquinas virtuales?

Los contenedores comparten recursos con el sistema operativo sobre el que se ejecutan, lo cual permite un arranque o detención rápida. En contraste, las máquinas virtuales se aíslan del sistema operativo sobre el que trabajan y se comunican a través de un sistema Hypervisor (que es un programa como VirtualBox, Vmware, Citrix). Los contenedores, a diferencia de las máquinas virtuales son portables facilitando la instalación y despliegue y reduciendo el tamaño del archivo a compartir. Por otra parte, las máquinas virtuales requieren de toda la instalación de dependencias, creación de sistemas de ficheros y consumo de recursos computacionales.

En la figura 1 se aprecian las diferencias en las capas de sofware de las máquinas virtuales y de Docker.

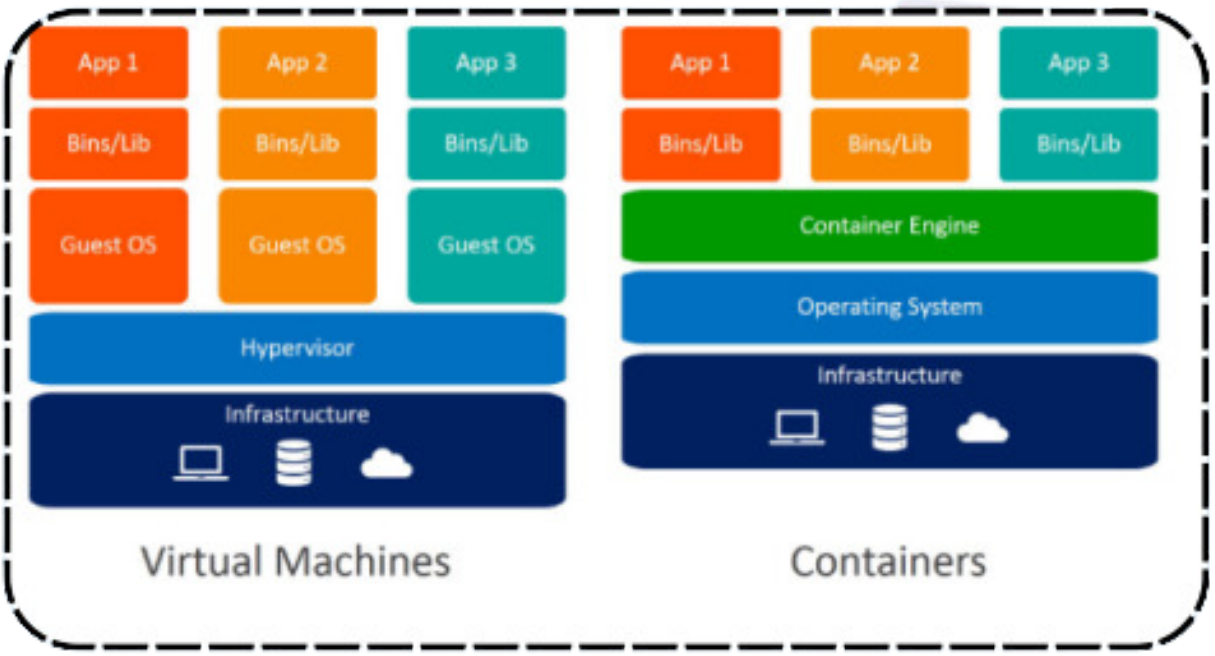

Figura 1: Estructuras de virtualización

El enfoque de Docker sobre la organización en contenedores se centra en la capacidad de separar una parte de un aplicativo.
Además de aprovechar este modelo basado en los microservicios, puede intercambiar procesos entre varias aplicaciones casi de la misma forma en que funciona la arquitectura orientada a los servicios. Veamos los componentes más importantes de Docker

TIC

Imágenes Docker:

Todo comienza con la creación de una "imagen Docker" o Dockerimage. Esta imagen es una plantilla que contiene todos los componentes necesarios para ejecutar una aplicación, como el sistema operativo, bibliotecas y el propio código de la aplicación. Para replicar un sistema solo se requiere de este archivo. Cada instrucción en un archivo de Docker se conoce como una capa, la ventaja de manejar las aplicaciones con capas es que, si una aplicación tiene un contenedor similar a otro, los recursos se reutilizarán, manteniendo el aislamiento entre contenedores y optimizando el uso de memoria. Adicionalmente las capas minimizan el tiempo de creación de los contenedores por medio de la memoria cache. Cuando un dockerfile es ejecutado por Docker, se convierte en una imagen de Docker. Cuando se carga una imagen y se ejecuta se convierte en un contenedor.

Contenedores:

Los contenedores son imágenes que se están ejecutando, o técnicamente, instancias en ejecución de una imagen de Docker. Cada contenedor tendrá un sistema de ficheros aislado, puertos, conexiones dependencias y sistema de manejo independiente, pero compartiendo un mismo kernel que el sistema operativo. A diferencia de las máquinas virtuales que tienen kernels independientes. Esto agiliza el procesamiento y hace que un contenedor no requiera de capacidades de cómputo excepcionales.

Docker Engine:

El motor central que hace posible la creación y gestión de contenedores es el "Docker Engine". Este componente se encarga de construir, ejecutar y gestionar los contenedores, así como de facilitar la comunicación entre los contenedores y el sistema operativo anfitrión. Es el servicio que maneja todos los contenedores, la creación de imágenes y la memoria.

Para crear imágenes personalizadas, se utiliza un archivo llamado "Dockerfile". Este archivo es esencialmente un conjunto de instrucciones que describe cómo debe ser construida la imagen. Especifica qué sistema operativo base utilizar, qué bibliotecas instalar y cómo configurar el entorno de ejecución.

Repositorios y Registro de Docker:

Las imágenes Docker se almacenan y comparten a través de repositorios. Los repositorios pueden ser locales o remotos. Docker Hub es un registro en línea popular donde los desarrolladores pueden compartir y distribuir sus imágenes. Por ejemplo, si se necesita una distribución de Linux con Python, se puede conseguir fácilmente en dockerhub. También se encuentran versiones oficiales de entornos para todo tipo de aplicaciones (javascript, web, servidores, y muchas alternativas más) con todos los paquetes listos y configurados.

Windows subsystem for Linux

Windos subsystem for linux o WSL es una capa de compatibilidad que permite ejecutar un sistema operativo con kernel de Linux dentro de una máquina que tiene el sistema operativo Windows sin necesidad de instalar máquinas virtuales ni reiniciar para tener varios sistemas operativos instalados a la vez. WSL sirve para ejecutar cualquier programa de Linux, desde la terminal con toda la estructura de un sistema Linux hasta programas de ciencia de datos como Python, R, Jupyter notebooks y muchas mas herramientas. La principal ventaja que ofrece es tener un desarrollo consistente entre sistemas Linux sin tener que emplear un equipo diferente o reinstalar software. Para el ejercicio del curso, se empleará la herramienta Docker instalada sobre WSL (Windows subsystem for Linux) ya que es compatible con todas las funciones necesarias del curso. Además, el uso de WSL permite que Docker tenga un mejor rendimiento y se integre con Windows, permitiendo que herramientas como Docker desktop pueda controlar los contenedores y las imágenes existentes. WSL también permite que los contenedores de Docker puedan tener puertos abiertos para ser consumidos desde el sistema Windows, e incluso, permite la creación de redes internas de Docker para que múltiples contenedores en una misma máquina compartan datos entre sí.

Nota: En esta sesión se debe instalar WSL, ya que en el taller es necesario iniciar instalando Docker y es requerido WSL funcionando. Si algún alumno trabaja sobre una máquina con Linux estos pasos se pueden omitir.