**Docker** es una plataforma abierta para que desarrolladores y administradores de sistemas desarrollen, **envíen y ejecuten aplicaciones** distribuidas, ya sea en computadoras portátiles, máquinas virtuales de centros de datos o en la nube.

Docker empaqueta software en “**contenedores**” que incluyen en ellos todo lo necesario para que dicho software se ejecute, incluidas librerías. Con Docker se puede implementar y ajustar la escala de aplicaciones de una forma rápida en cualquier entorno con la garantía de que el código se ejecutará.

**Docker no es una máquina virtual**. En Docker no existe un hypervisor que virtualice hardware sobre el cual corra un sistema operativo completo. Lo que se hace es usar las funcionalidades del **Kernel** para encapsular un sistema, de esta forma el proyecto que corre dentro de él no tendrá conocimiento que está en un contenedor. Los contenedores se encuentran aislados entre sí y se comportaran como máquinas independientes.

Iniciar un contenedor no tiene un gran impacto a diferencia de iniciar una máquina virtual ya que **no tiene que iniciar un sistema operativo completo** (desde cero). **Gracias al uso de contenedores la demanda de recursos baja** limitándose sólo al consumo de la aplicación que contenga. Un contenedor inicia en milisegundos.

**CONTENEDOR**

Algunas de las **características más notables** de un contenedor son:

* Los contenedores son más livianos (ya que trabajan directamente sobre el Kernel) que las máquinas virtuales.
* No es necesario instalar un sistema operativo por contenedor.
* Menor uso de los recursos de la máquina.
* Mayor cantidad de contenedores por equipo físico.
* Mejor portabilidad.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**IMAGEN**

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamenteUna imagen Docker es un **archivo**, compuesto por múltiples capas, que se utiliza para **ejecutar código en un contenedor** Docker. Son las plantillas base desde la que partimos ya sea para crear una nueva imagen o crear nuevos contenedores para ejecutar las aplicaciones.

Interfaz de usuario gráfica

Descripción generada automáticamenteUn **Dockerfile** es un archivo de texto plano que contiene una serie de **instrucciones** necesarias **para crear una imagen** que, posteriormente, se convertirá en una sola aplicación utilizada para un determinado propósito. Similar a lo explicado anteriormente, y la base del funcionamiento de Docker es mediante Dockerfiles.

Es como la receta necesaria para un banquete, en este caso el Dockerfile es necesario para la imagen que queramos construir, el Dockerfile es la receta y el gran banquete será nuestra imagen.

Diagrama

Descripción generada automáticamente

**DOCKER COMPOSE**

Docker Compose es una herramienta desarrollada para ayudar a definir y compartir aplicaciones de varios contenedores. Con Compose, puede crear un archivo YAML para definir los servicios y, con un solo comando, ponerlo todo en marcha o eliminarlo.

La gran ventaja de usar Compose es que puede definir la pila de la aplicación en un archivo, mantenerlo en la raíz del repositorio del proyecto (ahora tendrá control de versiones) y permitir que un tercero contribuya al proyecto. Un usuario solo tendría que clonar el repositorio e iniciar la aplicación Compose.

**KUBERNETES**

Se utiliza para programar y ejecutar los contenedores en clústeres de [máquinas virtuales](https://www.redhat.com/es/topics/virtualization/what-is-a-virtual-machine) o físicas. Su arquitectura divide un clúster en elementos que trabajan en conjunto, para mantener su estado definido.

Un [**clúster de Kubernetes**](https://www.redhat.com/es/topics/containers/what-is-a-kubernetes-cluster) es un conjunto de máquinas de **nodos** que ejecutan aplicaciones en contenedores. Lo puede visualizar en dos partes: el plano de control y las máquinas informáticas o los nodos. Cada nodo es su propio entorno [Linux](https://www.redhat.com/es/topics/linux/what-is-linux), y puede ser una máquina física o virtual. Los nodos se encargan de ejecutar los pods, los cuales están formados por contenedores.

La interfaz de programación de aplicaciones o [**API de Kubernetes**](https://www.redhat.com/es/topics/containers/what-is-the-kubernetes-API) es el frontend del plano de control de este sistema, y se encarga de la interacción de los usuarios con el clúster de Kubernetes. El servidor de la API determina si una solicitud es válida y se encarga de procesarla.

La **API** es la interfaz que se utiliza para gestionar, crear y configurar los clústeres de la plataforma. Es el medio de comunicación entre los usuarios, los elementos externos y las partes del clúster.