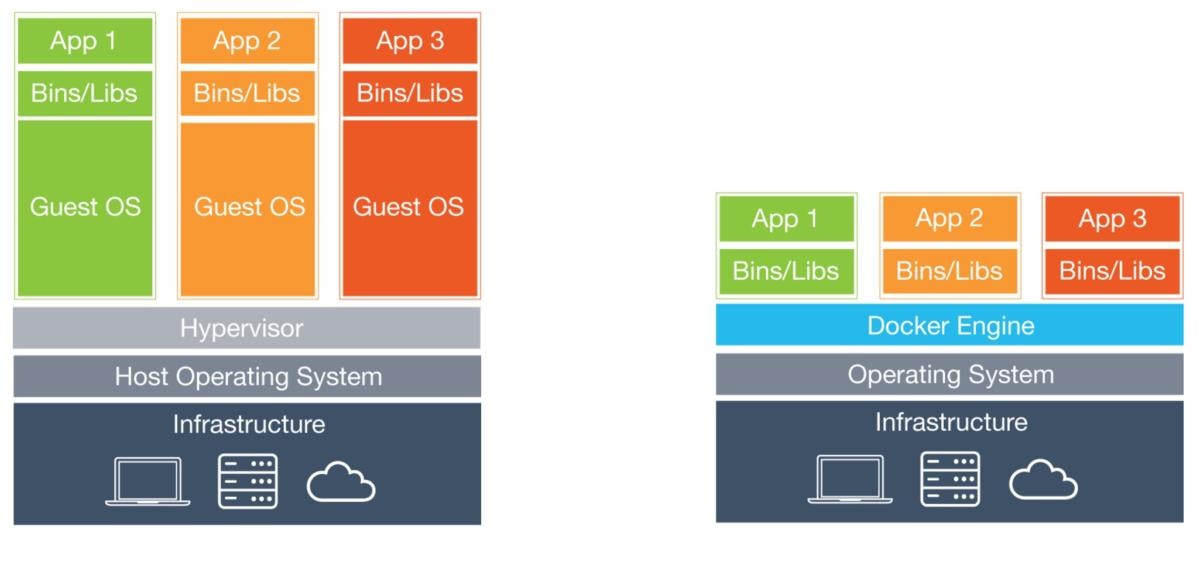
# Qué es un Container?

Un container es una unidad de software que contiene un programa y todas las dependencias necesarias para su funcionamiento. El container introduce una abstracción que garantiza el funcionamiento del programa independientemente de las características del ambiente en el que se encuentre el container.

# Container vs Maquina Virtual



*Figura 1:Imagen de Estructura de máquina virtual vs Docker*

El propósito original de Docker era poder aislar las ejecución de distintos procesos. Para esto es que se creó el concepto de Container.

Para poder aislar las procesos que se ejecutan dentro de una maquina virtual se necesitan virtualizar tantos SO como procesos haya. Esta tarea se lleva a cabo por lo que se conoce como Hypervisor.

Cada sistema operativo necesita su propio set de recursos, lo que implica un alto consumo de los mismos. También hay que tener en cuenta que cada SO viene con un conjunto predeterminado de librerías, independientemente de si el proceso las necesita o no para sus ejecución.

Ahora hablaremos del equivalente en Docker, es decir, utilizando Containers.

El equivalente en Docker al Hypervisor de las máquinas virtuales es el Docker Engine. El Docker Engine en vez de virtualizar un SO para cada proceso, virtualiza únicamente

un SO, y que solamente posee su Kernel y bibliotecas esenciales, y crea un Container para cada proceso, dentro del cual este se ejecuta. Dentro de este Container, además de los procesos, se encuentran las dependencias que este necesita para funcionar.

Esta alternativa, al reducir la cantidad de virtualizaciones a una, y al permitir que se instalen únicamente las dependencias necesarias, permite reducir considerablemente la necesidad de recursos.

# Componentes de docker:

## Volume:

El Volume el es mecanismo preferido dentro de Docker para persistir información generada y usada por los Containers de Docker.

## Volume vs Bind Mounts:

La alternativa a Volume dentro de Docker es lo que se conoce como Bind Mounts. Hacer un Bind Mount consiste en montar un archivo o directorio perteneciene al host, dentro de un container. Este archivo o referencia por su path. Si bien este método provee buen rendimiento, tiene desventajas como el hecho de crear una dependencia con respecto al sistema de archivos del host, y por otra parte el hecho que no puede utilizarse desde la consola de Docker.

Además, el Volume presenta varias ventajas con respecto a los Bind Mounts. **MEJORAR** .

## Container Image:

En la Container Image se almacena la configuración del Container. Podríamos que el Container es la instancia de la Container Image. En esta configuración se establece el proceso a ejecutar y sus dependencias, debiéndose especificar las versiones de las mismas. Esta configuración se almacena en una estructura de árbol.

## Docker File:

Se llama así al archivo que contiene la Container Image. Esta diseñado de forma tal que la Container Image se pueda escribir en un formato fácil de entender.

# Setup basico en Docker

# Kubernetes

Kubernetes agrega una capa de abstraccion al manejo de contenedores, permitiendo manipularlos, agruparlos y utilizarlos en gran escala. Originalmente diseñado por Google. EXPANDIR

# Fuentes:

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

<https://kubernetes.io/es/>

https://www.youtube.com/watch?v=iSkkHdGw-C0

https://www.youtube.com/watch?v=TvnZTi\_gaNc

https://www.youtube.com/watch?v=EnJ7qX9fkcU

https://www.infoworld.com/article/3204171/what-is-docker-docker-containers-explained.html

## 