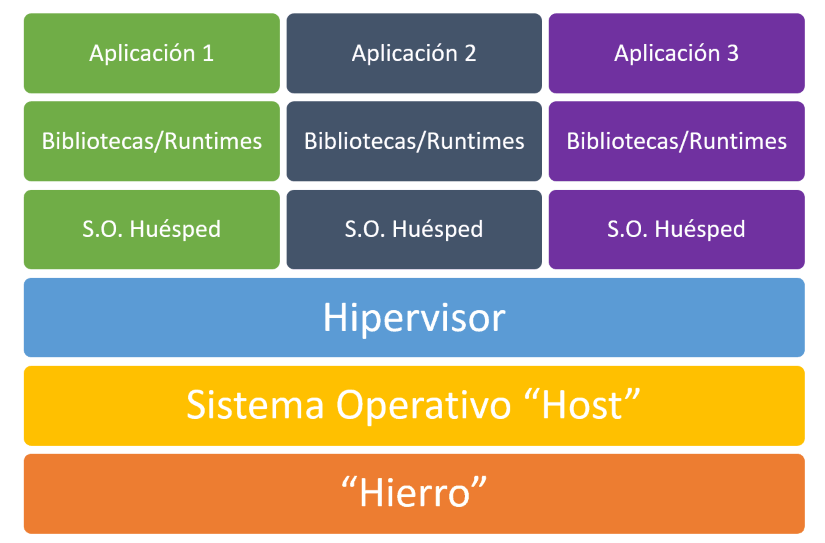
## Docker (Contenedores) y Máquinas virtuales (VMWare, VirtualBox)

## Funcionamiento de las máquinas virtuales

Una máquina virtual (VM, Virtual Machine) consiste en un sistema operativo completo funcionando de manera aislada sobre otro sistema operativo completo.

La tecnología de VMs permite compartir el hardware de modo que lo puedan utilizar varios sistemas operativos al mismo tiempo.

Esquema de su arquitectura



Hardware

En la capa inferior siempre tiene que haber **algún tipo de hardware** que lo sustente todo. Puede ser un ordenador personal o servidores, al final se trata de "**hardware**", máquinas físicas sobre las que se ejecuta todo lo demás. Todo ese poder computacional no vale de mucho si no le añadimos el **sistema operativo**, Windows o Linux.

Para que las máquinas virtuales puedan ejecutarse es necesario instalar otro componente por encima del S.O.: **el hipervisor**. Un hipervisor es un software especializado en exponer los recursos hardware que están debajo del sistema operativo, de modo que puedan ser utilizados por otros sistemas operativos. Esto incluye las CPUs, la memoria y el espacio de almacenamiento además del resto del hardware. Estas VMs "engañan" a un sistema operativo convencional para que crea que se está ejecutando sobre una máquina física. Los hipervisores vienen con productos como Hyper-V (incluido gratuitamente con Windows), VirtualBox o VMWare, entre otros.

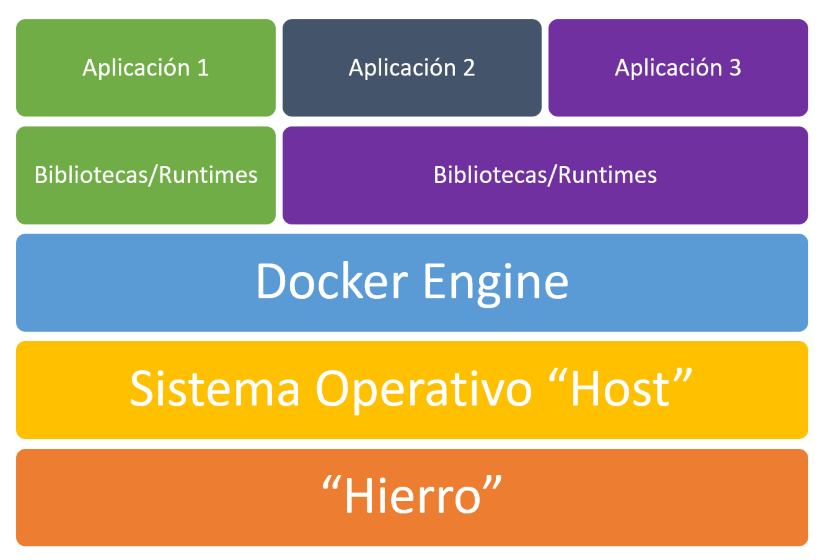
Existen hipervisores de tipo 1 que se instalan directamente sobre el hardware y evitan la necesidad de instalar un sistema operativo convencional en el medio. De hecho, estos son los que se suelen utilizar en los Data Center para aumentar todavía más el rendimiento.

Gracias a todo esto podemos tener diferentes sistemas operativos ejecutándose en paralelo sobre la misma máquina física, cada uno con su memoria y espacio en disco reservados, procesador, y completamente aislados unos de otros. Gracias al avance de los hipervisores en los últimos años, y a las tecnologías orientadas a la virtualización que ofrecen los procesadores modernos, la pérdida de rendimiento es mínima y es una manera muy eficiente de compartir el hardware para sacarle más partido.

## Funcionamiento de los contenedores

La filosofía de los contenedores es totalmente diferente a la de las VMs. En lugar de albergar un sistema operativo completo lo que hacen es compartir los recursos del propio sistema operativo "host" sobre el que se ejecutan.

Esquema contenedor



Hardware

Desaparece la capa del sistema operativo huésped, y se sustituye el hipervisor por el denominado "Docker Engine" (Virtuozzo, LXC/LXD, OpenVZ, DC/OS...).

Docker Engine se encarga de lanzar y gestionar los contenedores con nuestras aplicaciones, pero en lugar de exponer los diferentes recursos de hardware de la máquina para cada aplicación, lo que hace es compartirlos entre todos los contenedores **optimizando su uso** y eliminando la necesidad de tener sistemas operativos separados para conseguir el aislamiento.

Docker funciona a partir de **imágenes que se pueden reutilizar** entre varias aplicaciones (en el esquema anterior, en el que las aplicaciones 2 y 3 comparten el runtime). Cada una de esas imágenes se puede asimilar a una "capa" que se puede superponer a otras para formar un sistema de archivos que es la combinación de todas ellas. Por ejemplo, una capa puede llevar las bibliotecas o runtimes que necesitemos utilizar (como Node.js o PHP), otra con unas bibliotecas determinadas de las que hace uso nuestra aplicación, y otra capa final con el código de nuestra aplicación. La combinación resultante (una nueva imagen, única para nuestra app), es lo que forma la base de nuestro contenedor.

Cuando se lanzan uno o varios contenedores a partir de una imagen, a efectos de nuestra aplicación es como si estuviese ejecutándose en su propio sistema operativo, aislado de cualquier otra aplicación que hubiese en la máquina en ese momento. Pero la realidad es que están compartiendo el sistema operativo "host" que hay por debajo.

Es decir, **Docker aísla aplicaciones, no sistemas operativos completos**.

## Diferencias Docker y Máquinas Virtuales

Debemos tener en cuenta que, en el caso de los contenedores, el hecho de que no necesiten un sistema operativo completo, sino que reutilicen el subyacente **reduce mucho la carga** que debe soportar la máquina física, **el espacio** de almacenamiento utilizado **y el tiempo** necesario para lanzar las aplicaciones. Por lo tanto, **los contenedores son mucho más ligeros que las máquinas virtuales**.

Cuando definimos **una máquina virtual debemos indicar de antemano cuántos recursos físicos le debemos dedicar**. Por ejemplo, podemos decir que procesadores virtuales, RAM y espacio en disco. En el caso de los procesadores, es posible compartirlos entre varias máquinas virtuales (pero no conviene pasarse o bajara el rendimiento), y el espacio en disco se puede hacer que solo ocupe lo que se esté utilizando, de modo que crezca en función de las necesidades.

En el caso de la memoria y otros elementos (acceso a unidades externas o dispositivos USB) la reserva es total. Por eso, aunque nuestra aplicación no haga uso en realidad de los 4GB de RAM reservados da igual: no podrán ser utilizados por otras máquinas virtuales ni por nadie más. **En el caso de los contenedores** esto no es así. De hecho, no indicamos qué recursos vamos a necesitar, sino que es Docker Engine, en función de las necesidades de cada momento, el encargado de **asignar lo que sea necesario para que los contenedores funcionen** adecuadamente.

Una VM puede tardar un minuto o más en arrancar y tener disponible nuestra aplicación, **un contenedor Docker se levanta y responde en unos pocos segundos** (o menos, según la imagen). **El espacio ocupado en disco es muy inferior con Docker** al no necesitar que instalemos el sistema operativo completo.

Docker no permite utilizar en un sistema operativo "host" contenedores/aplicaciones que no sean para ese mismo sistema operativo, no podemos ejecutar un contenedor con una aplicación para Linux en Windows ni al revés.

Resumiendo, los contenedores permiten desplegar aplicaciones más rápido, arrancarlas y pararlas más rápido y aprovechar mejor los recursos de hardware. Las máquinas virtuales nos permiten crear sistemas completos totalmente aislados, con mayor control sobre el entorno y mezclando sistemas operativos host y huésped.

Los contenedores en general y Docker en particular se utilizan para desplegar aplicaciones en producción, sino también para **crear entornos de desarrollo replicables** entre todos los miembros de un equipo, **asegurar que las aplicaciones se van a ejecutar igual en todos los entornos** (desarrollo, pruebas y producción), etc.