

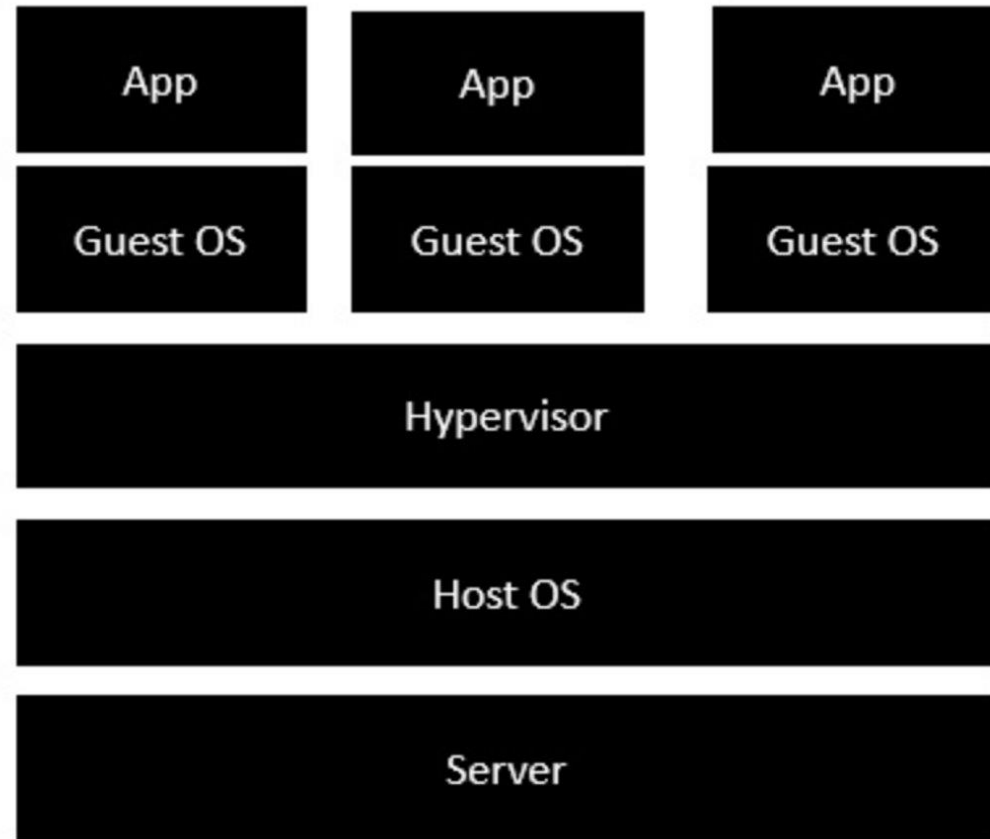
# Virtualización y contenedores

Sistemas informáticos

# Máquinas virtuales

- Llamamos máquina virtual a una computadora no real, instalada y configurada en un sistema informático mediante un software que permite simular su funcionamiento autónomo.
- El sistema informático sobre el que se instala y ejecuta la máquina virtual se denomina host o anfitrión.
- Los sistemas operativos de las máquinas virtuales se denominarán guests o invitados
- Para llevar a cabo la virtualización se necesita un software de abstracción sobre los recursos hardware de la máquina anfitriona denominado **hipervisor** o VMM

# Máquinas virtuales



# Usos de las máquinas virtuales

- Realizar pruebas de sistemas informáticos, software y configuraciones sin que un fallo pueda ocasionar problemas en una máquina real
- Al ser software estos sistemas son fácilmente portables de una máquina host a otra
- Ahorro de costes, al ser posible crear múltiples máquinas virtuales en un solo equipo host
- Copias de seguridad
- Centralización de servicios en distintas máquinas virtuales aisladas

# Ejercicios VirtualBox

1. Crear una máquina virtual Ubuntu
2. Configurar el adaptador de red a modo puente
3. Añadir procesadores a la máquina para que vaya más rápido
4. Realizar una instantánea

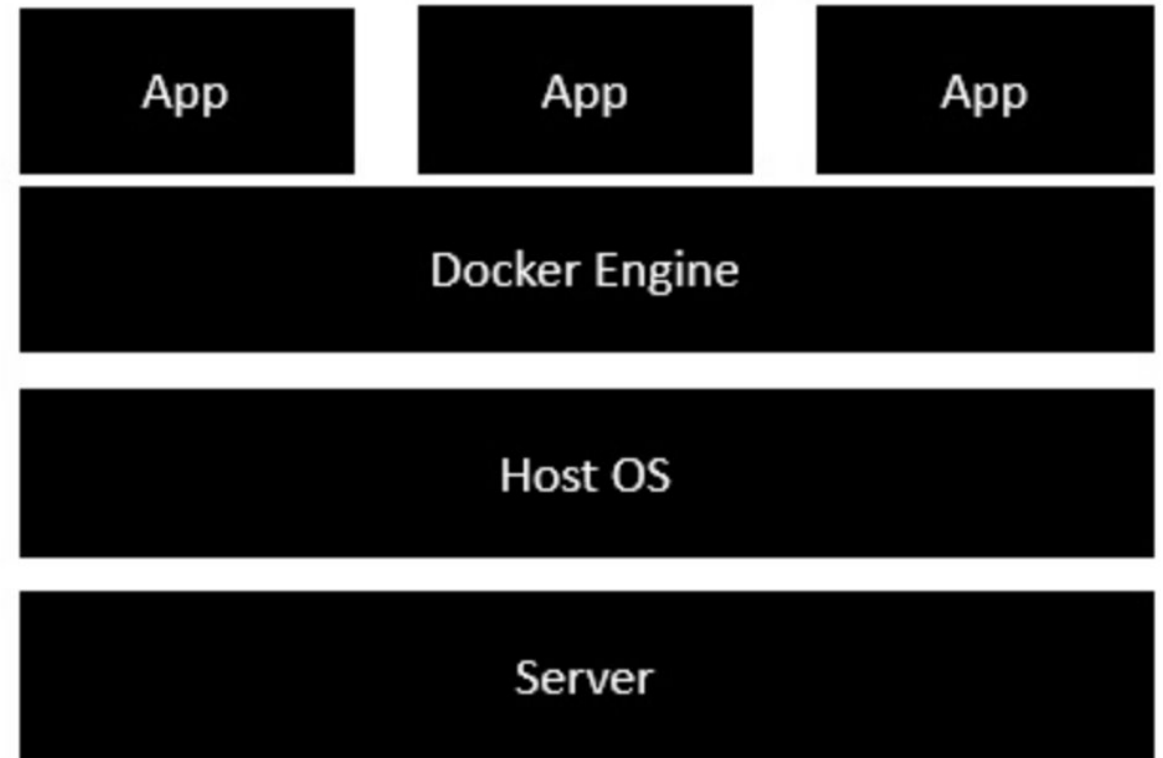
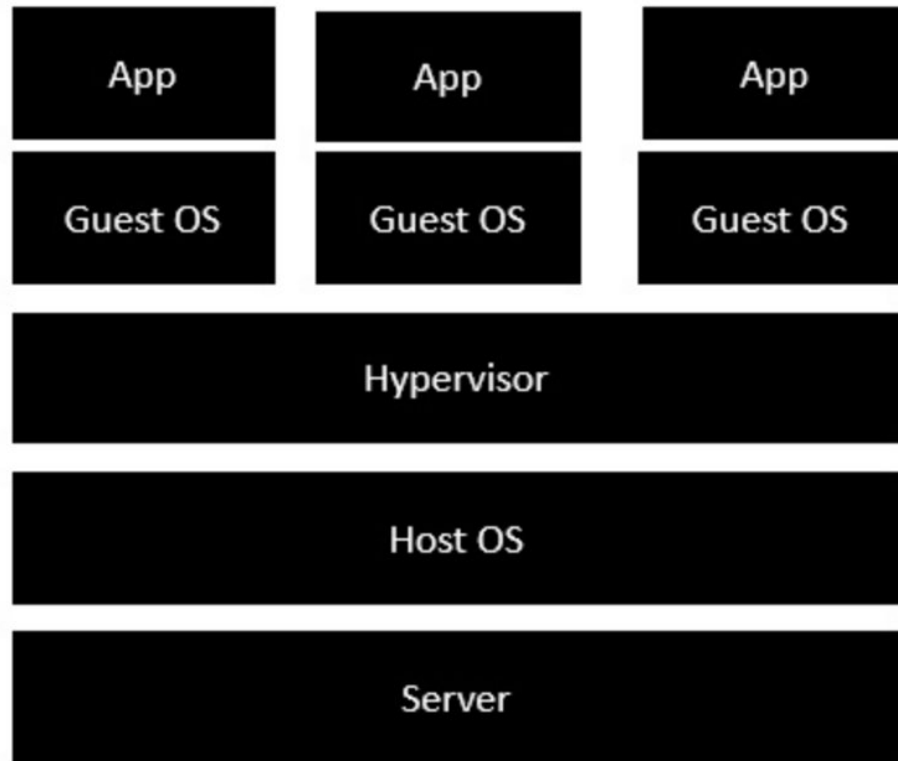
# Contenedores Docker

- Docker es una plataforma para desarrolladores y administradores de sistemas para construir, ejecutar y compartir contenedores de aplicaciones
- El uso de contenedores para el despliegue de aplicaciones se denomina contenedorización
- Un contenedor es un proceso en ejecución en un entorno aislado del host y de otros contenedores.
- Cada contenedor interactúa con su propio sistema de archivos independiente
- Para iniciar un contenedor es necesario partir de una imagen que define todo lo necesario para ejecutar la aplicación: filesystem, código, dependencias...

# Ventajas

- **Flexible:** Incluso las aplicaciones más complejas pueden ejecutarse en contenedores.
- **Lightweight:** Los contenedores aprovechan al máximo los recursos del servidor host, haciéndolos mucho más eficientes que alternativas como la virtualización.
- **Portable:** Puedes construir en local, desplegar en la nube y ejecutar en cualquier parte.
- **Loosely coupled:** Los contenedores son autosuficientes y encapsulados, pudiendo realizar upgrades sobre ellos de manera independiente sin afectar a otros contenedores ni al host.
- **Escalable:** Permiten escalar un sistema de manera sencilla aumentando el número de contenedores de una aplicación, esto se denomina escalado horizontal.
- **Seguros:** Se aplican fuertes restricciones y aislamiento a los procesos sin necesidad de configuración adicional.

# Contenedores Docker vs máquinas virtuales





# Contenedores Docker vs máquinas virtuales

- Virtualización del sistema operativo vs virtualización hardware
- Sistema operativo compartido vs sistema operativo dedicado
- Mismo sistema operativo que el host vs sistemas operativos independientes

# Testeando la instalación de Docker

1. Test that your installation works by running the `hello-world` Docker image:

```
$ docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
...
```

2. Run `docker image ls` to list the `hello-world` image that you downloaded to your machine.
3. List the `hello-world` container (spawned by the image) which exits after displaying its message. If it is still running, you do not need the `--all` option:

```
$ docker ps --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
54f4984ed6a8	hello-world	<code>"/hello"</code>	20 seconds ago	Exited (0) 19 seconds ago

# Imágenes Docker

- Existen una gran variedad de imágenes disponibles en internet. Cada imagen suele incluir unas configuraciones del sistema operativo y alguna aplicación que se va a ejecutar
- Una imagen puede ser instalada en local con el uso del comando `docker pull [url de la imagen]`
- Para mostrar el listado de imágenes descargadas podemos usar el comando `docker image ls`
- Podemos lanzar un contenedor ejecutando el comando `docker run [repositorio imagen imagen]`

# Docker run

- Para darle un nombre al contenedor usaremos la opción `--name=[nombre]`
- Podemos mapear puertos del contenedor al host con el parámetro `-p [puertoHost]:[puertoContenedor]`
- Si queremos montar una carpeta del host dentro del contenedor podemos usar la opción `-v [rutaHost]:[rutaContenedor]`. Esta opción monta un volumen en el contenedor.
- Si queremos que al arrancar el contenedor se abra una terminal del mismo con la que podamos interaccionar usaremos la opción `-it`
- Si queremos que el contenedor se elimine automáticamente al finalizar podemos añadir la opción `-rm`

# Construir una imagen

- Muchas veces vamos a querer crear nuestras propias imágenes Docker con las aplicaciones que vamos desarrollando
- Para crear una imagen debe escribirse su especificación en un fichero `.Dockerfile`
- Este fichero señalará cual es la imagen de la que partimos
- También se podrán ejecutar comandos para instalar software adicional
- Y podremos copiar los ficheros que deseemos dentro del contendor
- Por último, podemos establecer qué debe ejecutar el contendor

# Ejercicios Docker