



# Docker

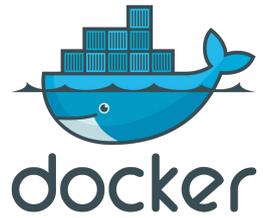
Herramientas de Software

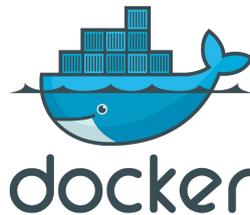


# Docker: Motivación

Al querer ordenar comida por delivery, tenemos la opción de contactarnos directamente con el restaurante. Podemos descargar su aplicación, ir a su página web, pedir por teléfono, etc.

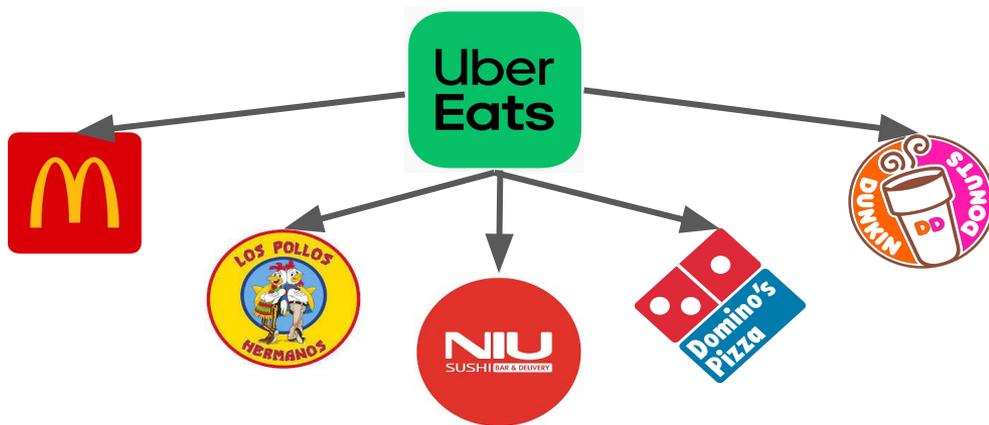
Pero, ¿qué pasa si su aplicación no está disponible para nuestro dispositivo? ¿o si no queremos instalar una aplicación nueva para cada restaurante porque resultaría muy pesado?

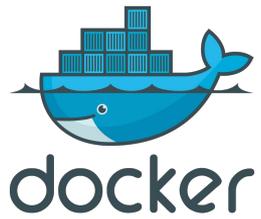




# Docker: Motivación

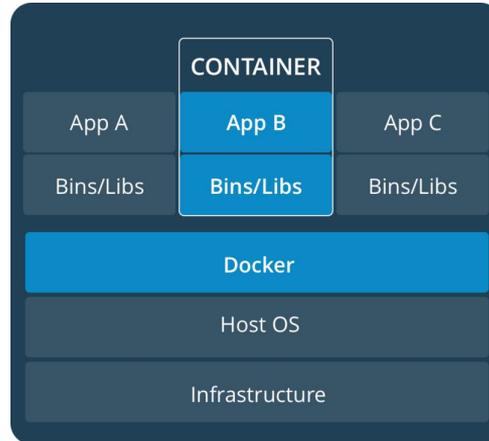
Para no tener que acceder a todas las páginas web una por una, instalar una aplicación para cada restaurante, se puede utilizar una aplicación de *delivery* que se encargue de **contener** los diferentes locales disponibles, a los cuales se podrá acceder de manera fácil y rápida.





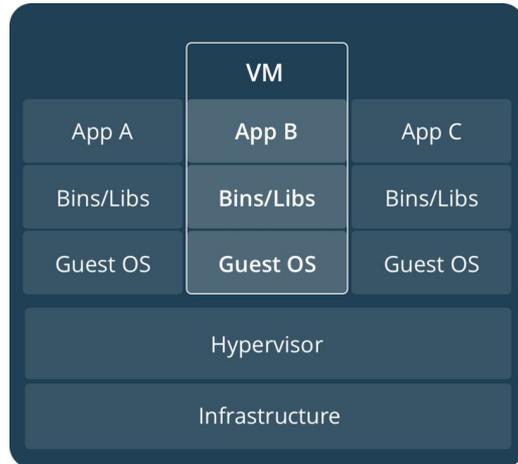
# Docker: ¿Qué es?

Docker es una tecnología que permite crear y ejecutar aplicaciones en entornos aislados llamados '**contenedores**'. Estos contenedores son portátiles y pueden ejecutarse en cualquier computadora con Docker, independientemente del sistema operativo o configuración.



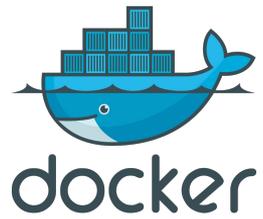
# ¿Qué es una máquina virtual?

Una máquina virtual es un software que crea una computadora virtual dentro de otra, permitiendo ejecutar varios sistemas operativos y aplicaciones en una sola computadora física mediante un hipervisor.



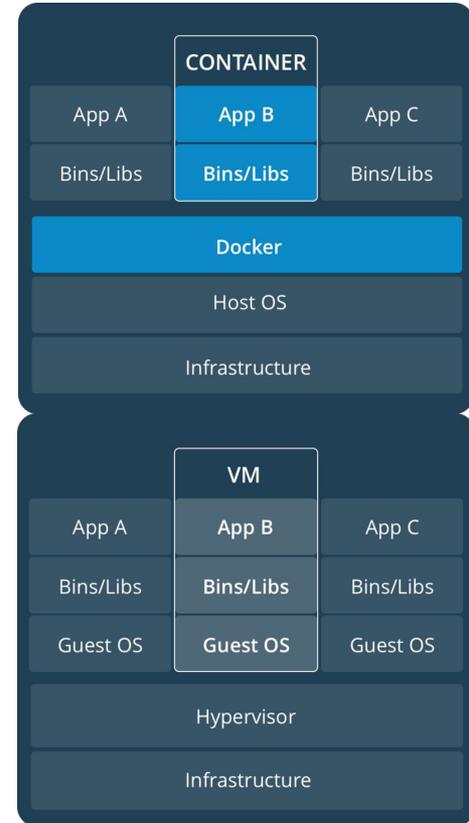
## VirtualBox

VirtualBox, un ejemplo de hipervisor, es un software que permite crear y gestionar máquinas virtuales.

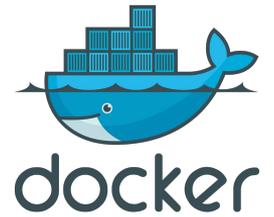


# Docker: ¿Qué es un contenedor?

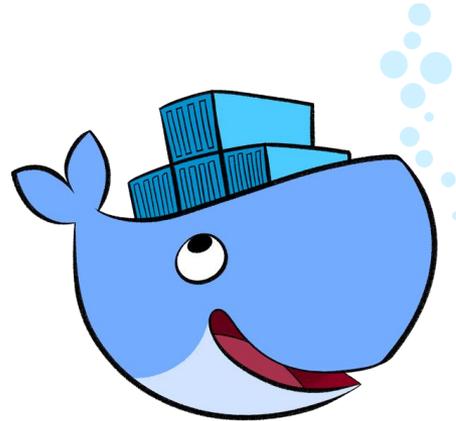
Los contenedores son como pequeñas unidades de aplicaciones que comparten recursos con el sistema operativo anfitrión, lo que los hace más ligeros y eficientes que las máquinas virtuales. En lugar de duplicar todo un sistema operativo, los contenedores solo añaden lo esencial para ejecutar una aplicación específica. Esto facilita su rápida ejecución y portabilidad.

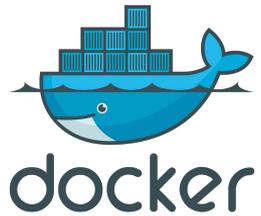


# Docker: ¿Qué es una imagen?



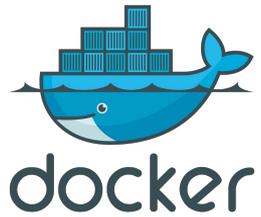
Una imagen es una plantilla de solo lectura con todo lo necesario para ejecutar una aplicación de forma aislada. Las imágenes son la base para crear y ejecutar contenedores, ya que cada contenedor se crea a partir de una imagen específica.





# Docker: Comandos útiles

- **docker pull nombre\_imagen:etiqueta:** Descarga una imagen desde un registro de Docker.
  - Ejemplo: *docker pull library/ubuntu:18.04*
- **docker image nombre\_imagen:etiqueta:** Elimina una imagen específica
  - Ejemplo: *docker image rm ubuntu:18.04*
- **docker run -it nombre\_imagen:** Inicia un nuevo container y permite interactuar con él usando la terminal.
  - Ejemplo: *docker run -it ubuntu*



# Docker: Comandos útiles

- **docker run -it nombre\_imagen -v /home/myuser/data:/data:** Monta una carpeta del sistema anfitrión en el contenedor en la ubicación /data. De esta manera, cualquier archivo o cambio realizado en la carpeta en el sistema anfitrión será reflejado dentro del contenedor.
- Operaciones con containers:
  - **docker container rm ![container name 1] ![container name 2]**
  - **docker container start ![container name]**
  - **docker container stop ![container name]**
  - **docker container restart ![container name]**

# Para profundizar

- Tutoriales de Docker
  - [https://docs.duckietown.com/daffy/opmanual-duckiebot/preliminaries\\_software/docker/index.html#preliminaries-docker-basics](https://docs.duckietown.com/daffy/opmanual-duckiebot/preliminaries_software/docker/index.html#preliminaries-docker-basics)
  - [https://www.youtube.com/playlist?list=PLunhqkrRNRhaqt0UfFxxC\\_oj7jscss2qe](https://www.youtube.com/playlist?list=PLunhqkrRNRhaqt0UfFxxC_oj7jscss2qe)



# Docker

Herramientas de Software

