

Contenedores

Otra ideología de la virtualización

Presentan:

Cárdenas Cárdenas

Jorge

Garrido Sánchez

Samuel Arturo

- En la industria comúnmente se tienen el problema que una aplicación que corre satisfactoriamente en un servidor (virtualizado o no), pero se comporta erróneamente en otro.
- O que el código de un desarrollador, funciona en una computadora, pero luego falla cuando lo mandamos al entorno de producción.

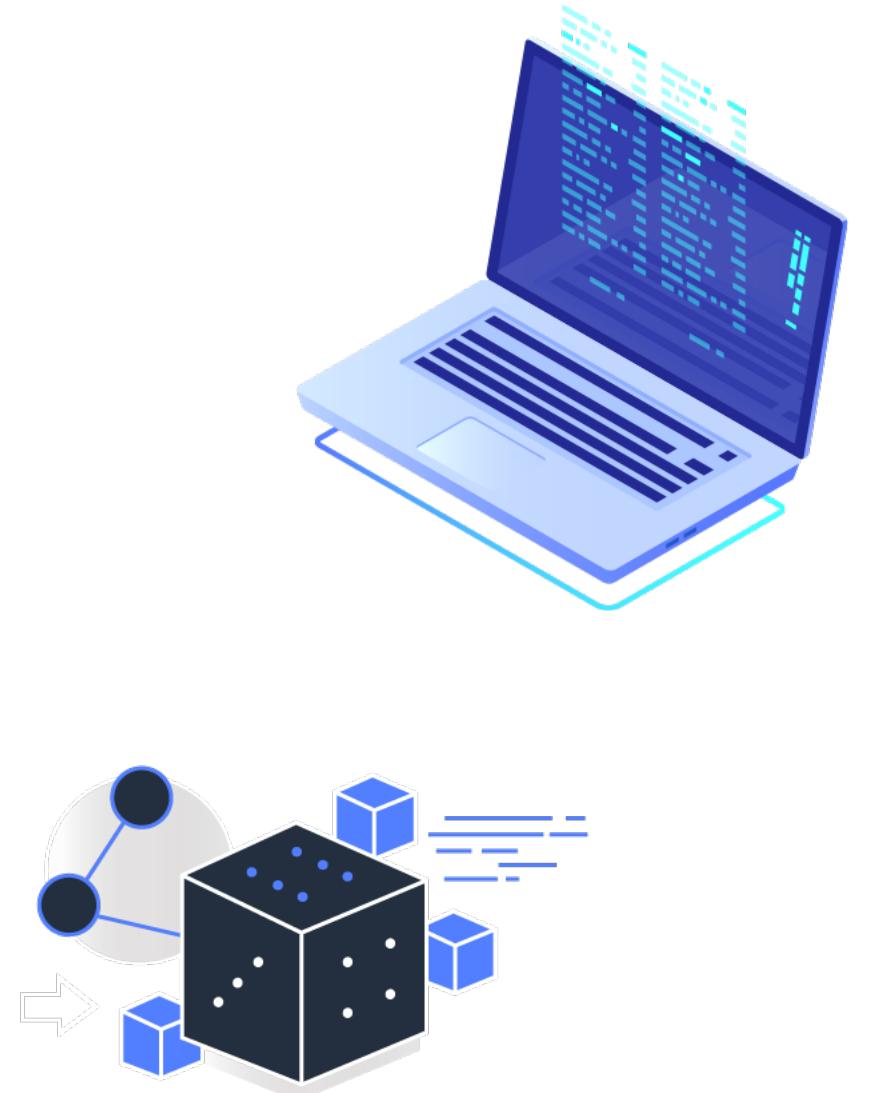
¿A qué se debe?



- Dos copias exactas de una misma aplicación se comportan de manera distinta en dos entornos con configuraciones distintas
- La falta de una dependencia
- Una biblioteca
- Versiones distintas de esta
- Una variable de entorno distinta
- O incluso diferentes versiones del sistema operativo
- Puede ocasionar que nuestra aplicación deje de funcionar correctamente.

En un mundo ideal

- Dado que algunos programas están hechos para ejecutarse en cierto SO y a menudo no todos pueden por X o Y razón poseer ese SO, es necesario una “virtualización”.
- ¿Una máquina virtual puede solucionar el problema?
 - Sí
 - ¿Y a qué va esto de los contenedores?



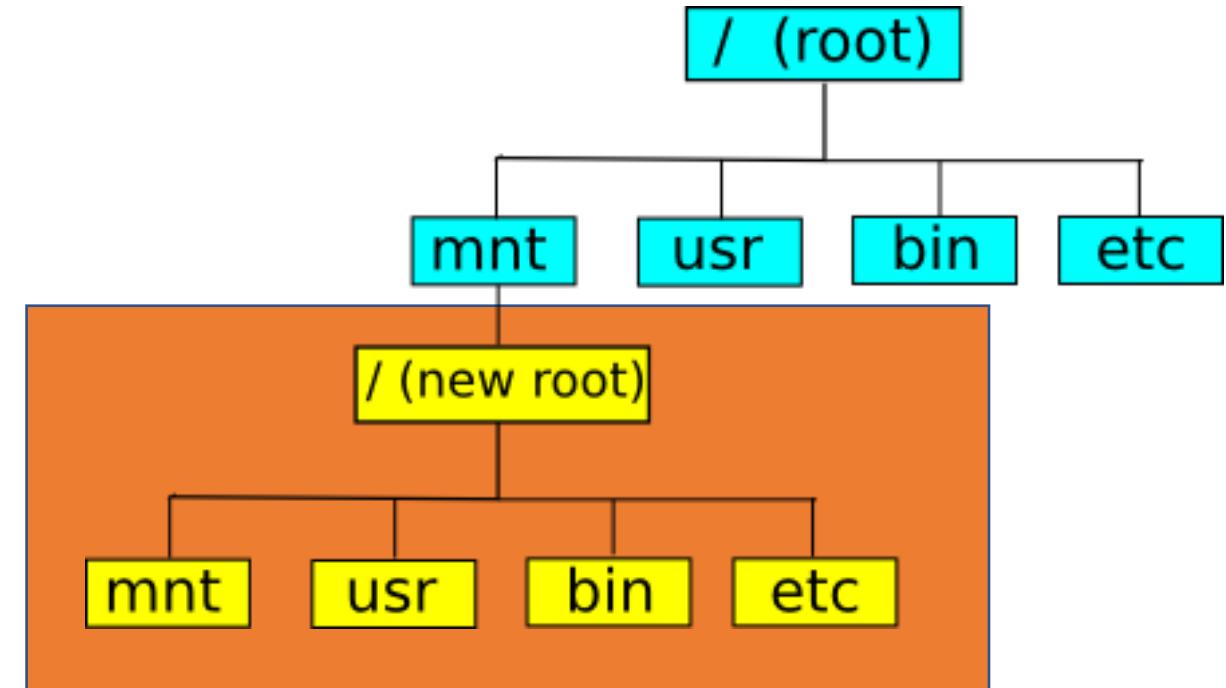
Máquina Virtual

- Pensemos que el sistema operativo que se ejecuta dentro de una máquina virtual está siendo “engañado” para que piense que corre en su propio hardware



Chroot: Antes de un contenedor

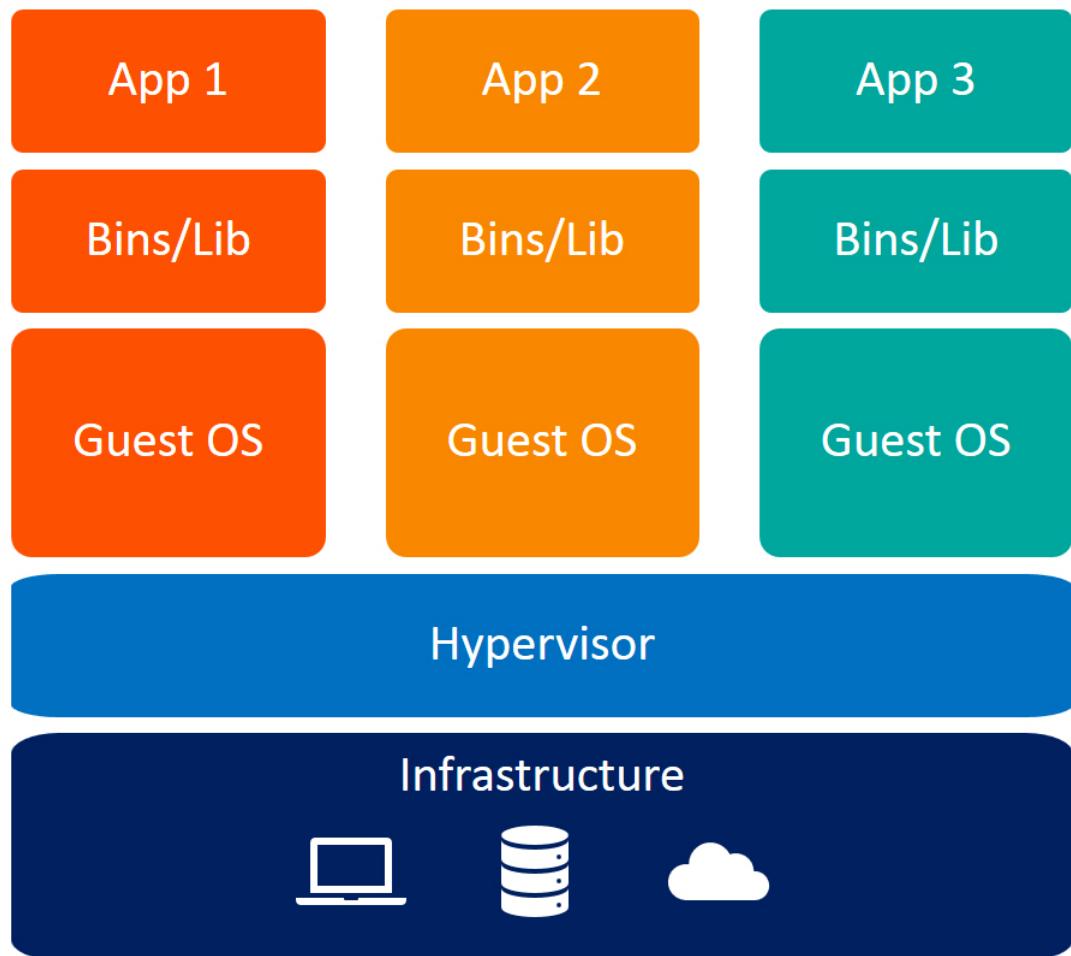
- Bill Joy creó la llamada al sistema chroot para auxiliarse en el desarrollo UNIX
- Concretamente esta llamada se limita a restringir el sistema de archivos



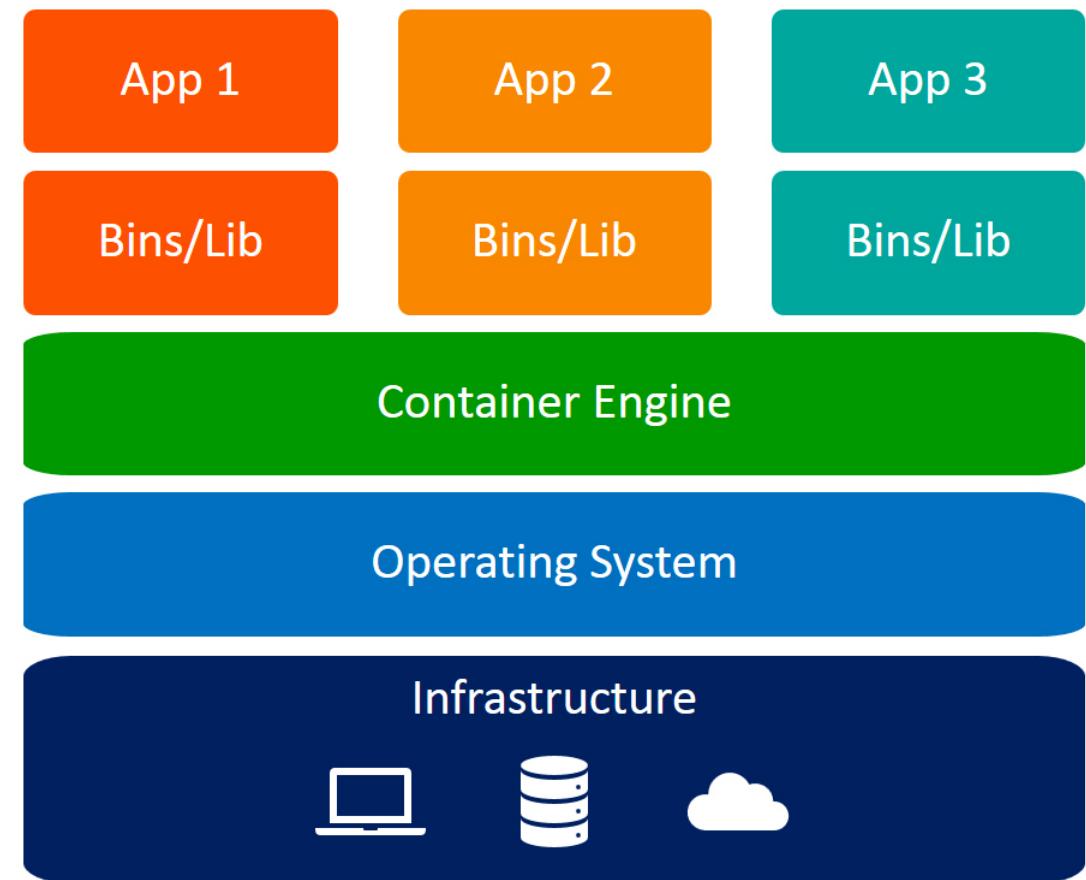
Definición: Contenedor

- Virtualización a nivel Sistema Operativo
- Una aplicación o proceso que corre dentro de un contenedor también está siendo “engañado” para pensar que corre dentro de su propio sistema operativo

- Las máquinas virtuales incluyen la aplicación, las bibliotecas o los archivos binarios necesarios y un sistema operativo invitado completo.
- La virtualización completa requiere más recursos que la inclusión en contenedores.
- Los contenedores incluyen la aplicación y todas sus dependencias.
- Sin embargo, comparten el kernel del sistema operativo con otros contenedores, que se ejecutan como procesos aislados en el espacio de usuario en el sistema operativo host.



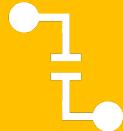
Virtual Machines



Containers



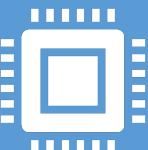
Un contendor comparte recursos con el sistema operativo sobre el que se ejecuta.



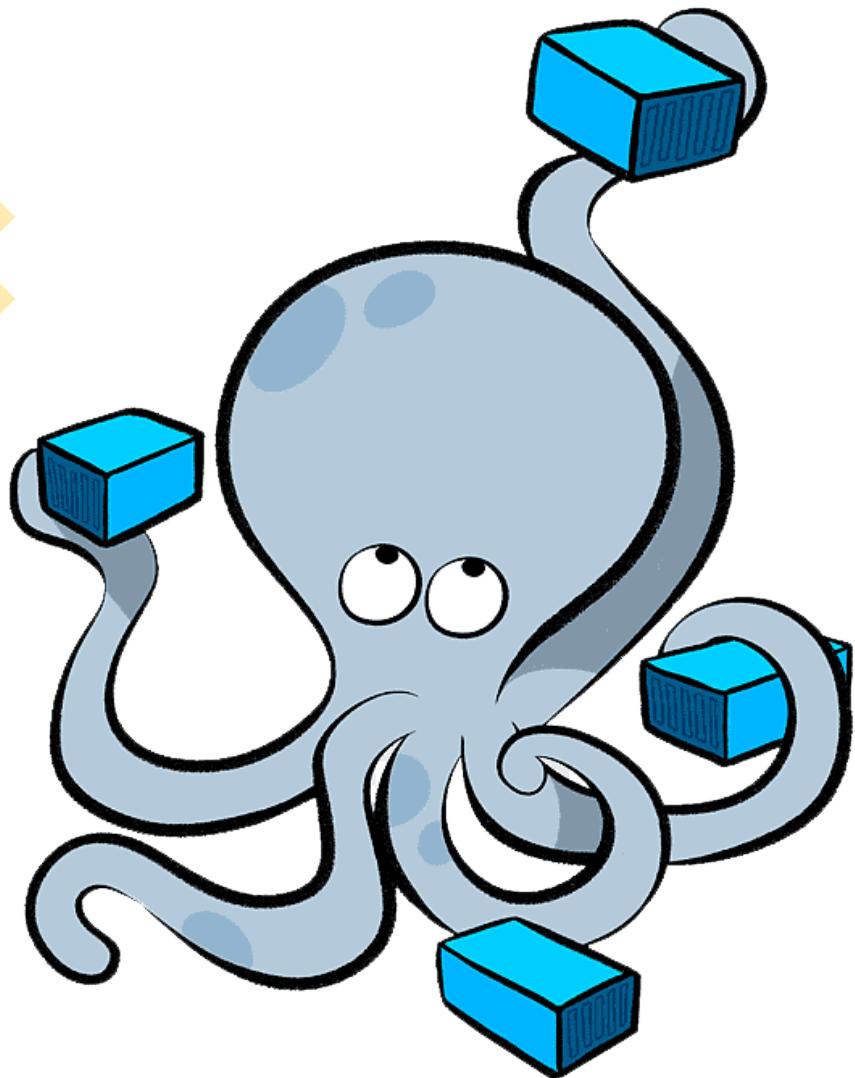
Una máquina virtual se aísla del sistema operativo sobre el que trabaja y se comunican a través del hypervisor.



De esta manera podemos arrancar o parar el contenedor rápidamente



Las máquinas virtuales quieren simular un entorno diferente, el container se centra en crear la aplicación y que se pueda portar todo el contenido de manera sencilla.



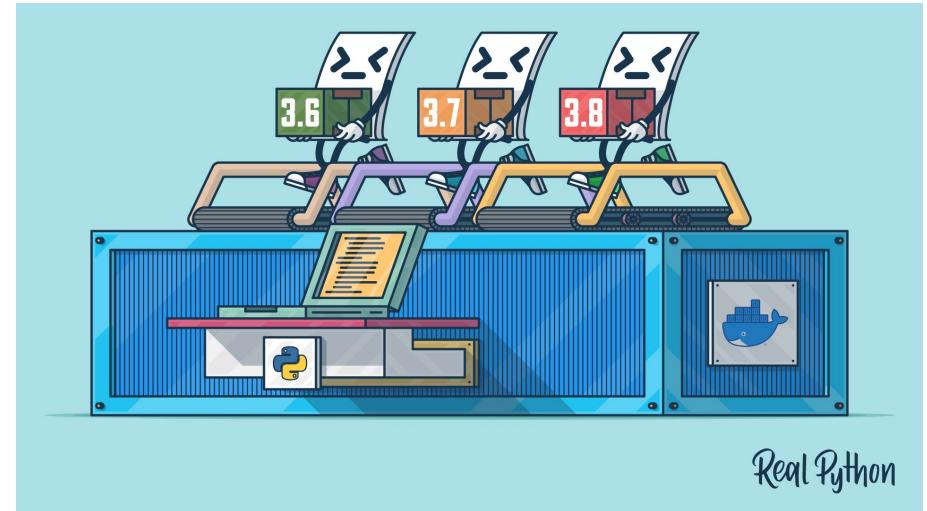
- Cabe destacar que, pese que existen similitudes entre ambos,
 - Un contenedor resuelve cuellos de botella de software
 - Una máquina virtual resuelve cuellos de botella de hardware.
- Ambos son complementarios, y no excluyentes.

Virtualización a nivel SO

- Implica que los contenedores engañarán a los procesos huéspedes, limitándoles:
 - Tablas de procesos
 - Señales y comunicación entre procesos
 - Interfaces de red
 - Dispositivos de Hardware
 - Límite en el consumo de recursos (disco, memoria, ciclos de CPU)
 - Información del sistema

Ejemplo

- Una máquina tiene Python 2.x
- Si se crea un entorno usando chroot con Python 3.x
 - El entorno tiene su propios:
 - Permisos
 - Usuarios
 - Carpetas.
 - Pero no interactúa con los procesos de la máquina, a pesar de que el sistema operativo lo creo.



Docker como principal ofertante

- La gente de Docker dijo entonces “vamos a crear una capa de abstracción para que todo el mundo pueda utilizar los contenedores de forma práctica”
- Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software.



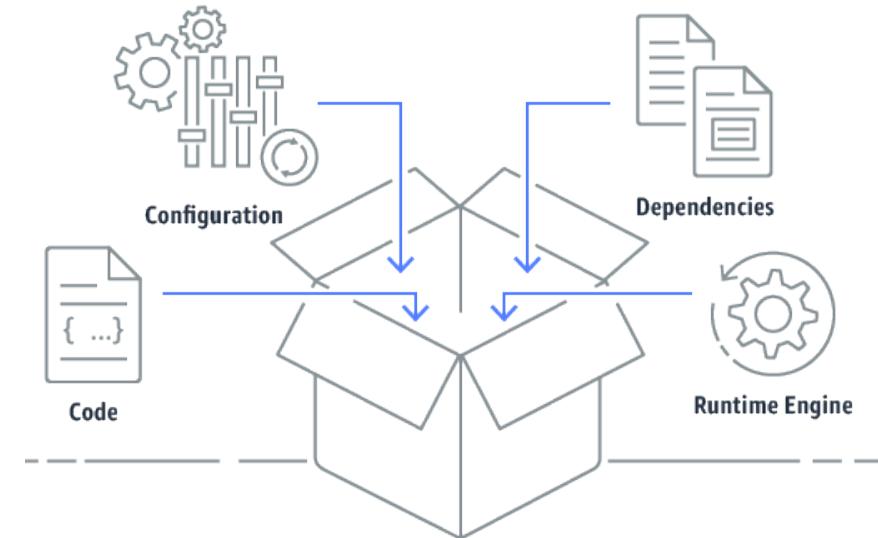
- En Docker, cada capa es el conjunto resultante de los cambios que se producen en el sistema de archivos después de ejecutar un comando, como instalar un programa.
- Por lo tanto, al "Examinar" el sistema de archivos después de que se ha copiado una capa, se verán todos los archivos, incluida la capa cuando se instaló el programa.
- Se puede pensar en una imagen como un disco duro de solo lectura auxiliar listo para instalarse en un "equipo" donde el sistema operativo ya está instalado.

Características ...

- Ofrecen un modo estándar de empaquetar el código, las configuraciones y las dependencias de su aplicación en un único objeto

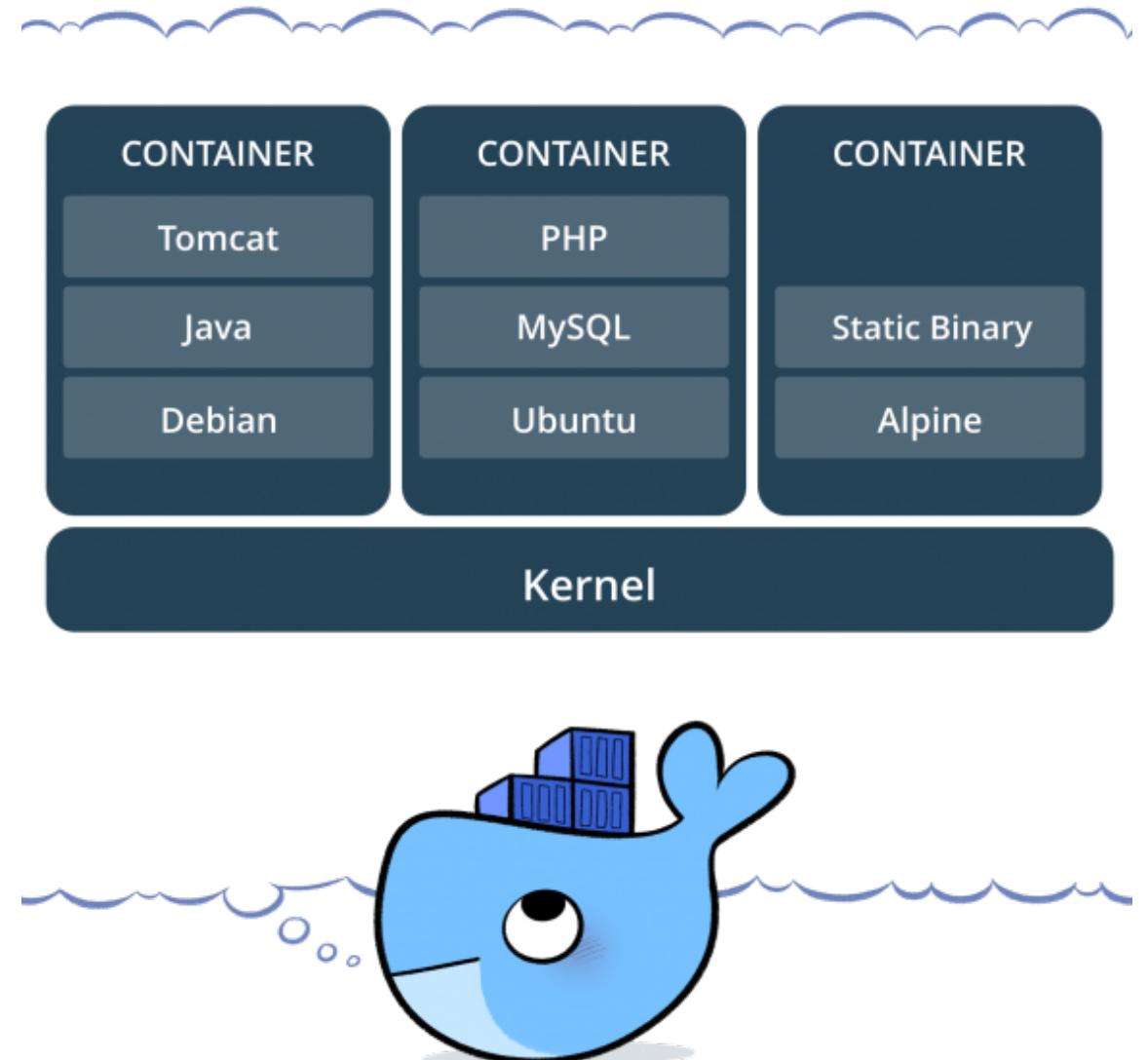
Garantiza:

- Implementaciones rápidas, fiables y consistentes sea cual sea el entorno en el que se realizan.

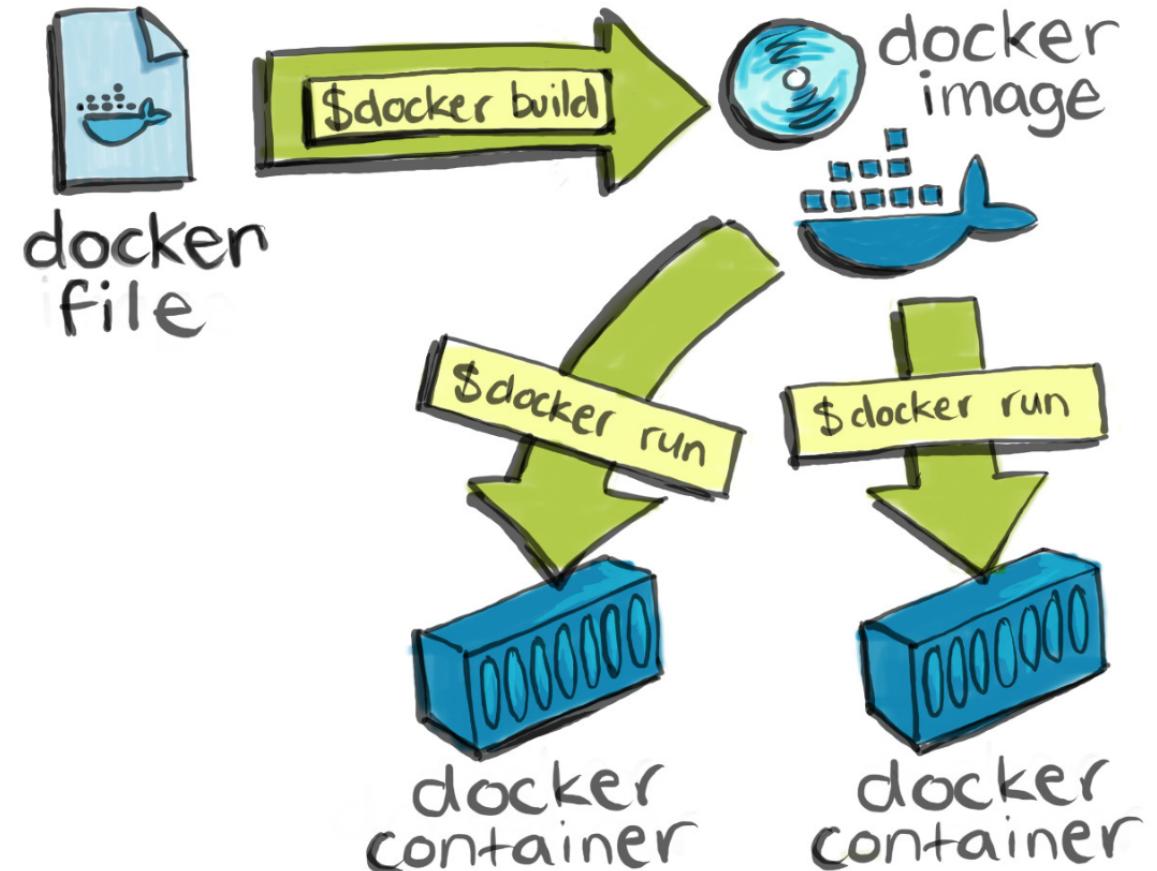


El propósito de Docker es la independencia:

- Capacidad de ejecutar varios procesos y aplicaciones por separado para hacer un mejor uso de su infraestructura.
- Conservar la seguridad que tendría con sistemas separados.



Las herramientas del contenedor, ofrecen un modelo de implementación basado en imágenes.



Servicios

Docker: “The World’s Largest Community of Container Images”

- Docker Hub es un compendio de imágenes que nos permitirán crear nuestro entorno con solo colocar el comando: (una vez instalado en su computadora, claro y el demonio corriendo)
 - docker pull mysql
 - docker pull store/oracle/database-enterprise:12.2.0.1



golang
Official
↓ 10M+



couchbase
Official
↓ 10M+



mysql
Official
↓ 10M+



httpd
Official
↓ 10M+



openjdk
Official
↓ 10M+

Ejemplos

- MySQL
- CowSay - WhaleSay

```
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez — bash — 86x11
[MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez$ cowsay Hola Grupo de SO 2020-2 ]
-----  
< Hola Grupo de SO 2020-2 >
-----  
      \ ^__^
       \  (oo)\_____
          (__)\       )\/\
              ||----w |
              ||     ||  
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez$
```

```
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez — bash — 91x15
< Hola a todos! >
-----
      \
       \
        ##      .
        ## ## ##   ==
        ## ## ##   ==
        /-----\   ==
        ~~~ {~~~ ~~~ ~~~ ~~~ ~~~ ~~~ /   ==- ~~~
        \_\_\_\_ o   /_/
        \_\_\_\_ /_/  
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez$
```

```
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez — docker exec -it mysql-db mysql -p — 88x23
Using default tag: latest
latest: Pulling from library/mysql
619014d83c02: Pull complete
9ced578c3a5f: Pull complete
731f6e13d8ea: Pull complete
3c183de42679: Pull complete
6de69b5c2f3c: Pull complete
122a561a4196: Pull complete
1abf8e9f34f0: Pull complete
1e5887414166: Pull complete
95adaca07078: Pull complete
42c8c6542347: Pull complete
0ae93d9077ae: Pull complete
42131d6ef54e: Pull complete
Digest: sha256:c7c6c5beb312fd2eb21af4f144d14b6ef29c9c2f9c5e1f3f74ffa75e38fad1f4
Status: Downloaded newer image for mysql:latest
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez$ docker run --name EjemploSQL -e MYSQL_ROOT_PASSWORD=proteco -d mysql:8.0
[Unable to find image 'mysql:8.0' locally
8.0: Pulling from library/mysql
Digest: sha256:c7c6c5beb312fd2eb21af4f144d14b6ef29c9c2f9c5e1f3f74ffa75e38fad1f4
Status: Downloaded newer image for mysql:8.0]
```

```
MacBook-Pro-de-Samuel-2:~ samuelarturogarridosanchez — docker exec -it mysql-db mysql -p — 98x15
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.19 MySQL Community Server - GPL

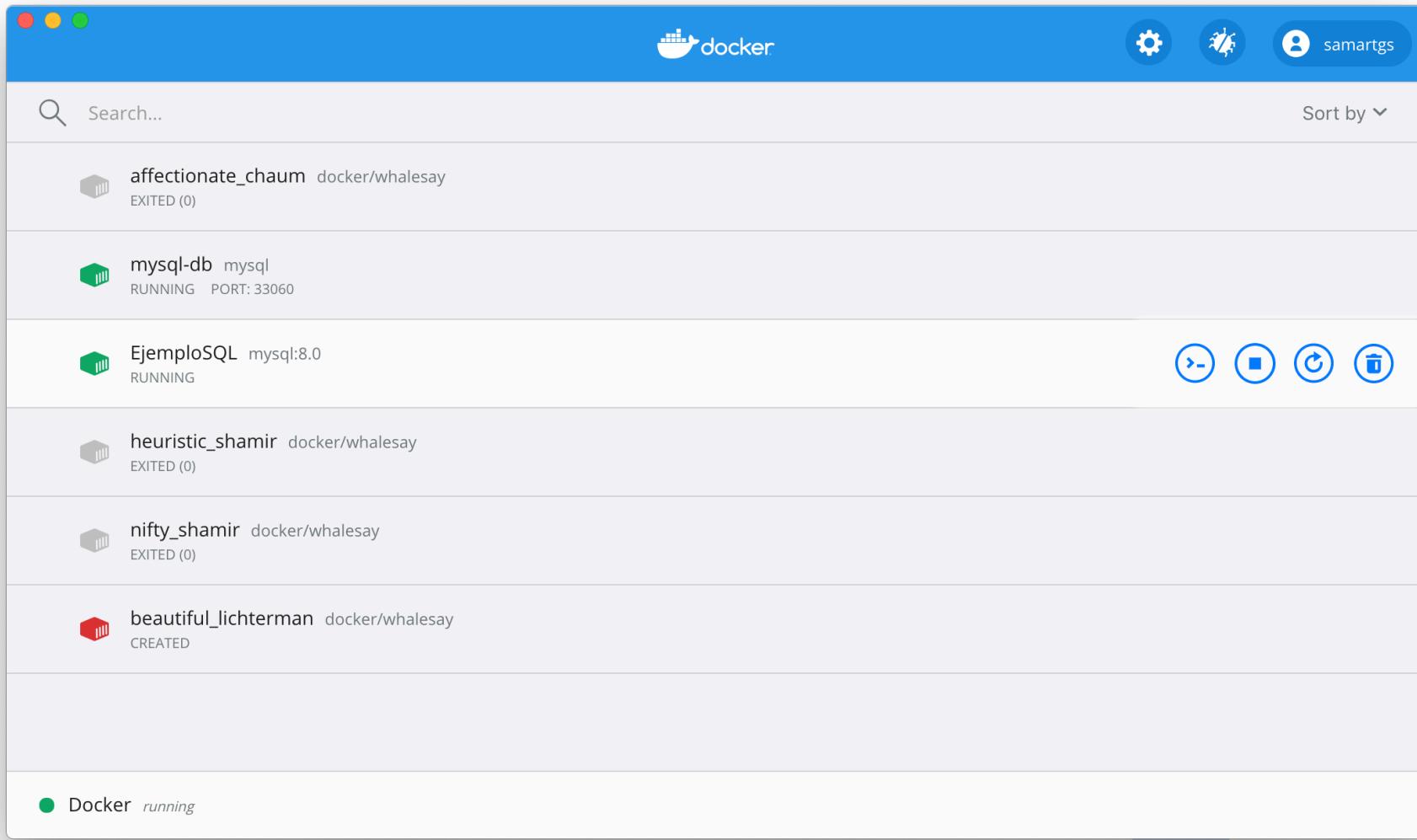
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

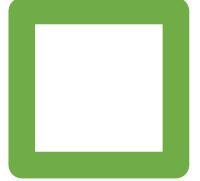
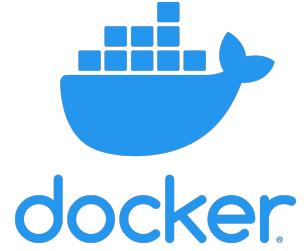
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Dashboard (Antes Kinematic)



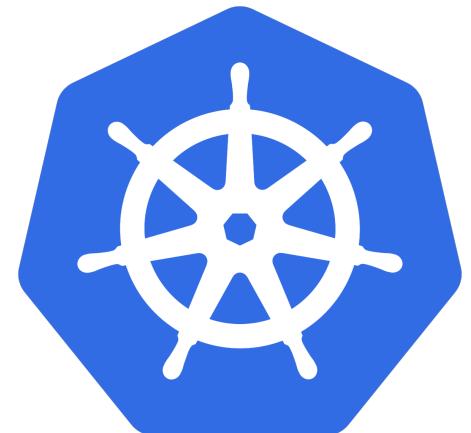


**“Docker no solo es una
tecnología, sino también una
filosofía y un proceso”**



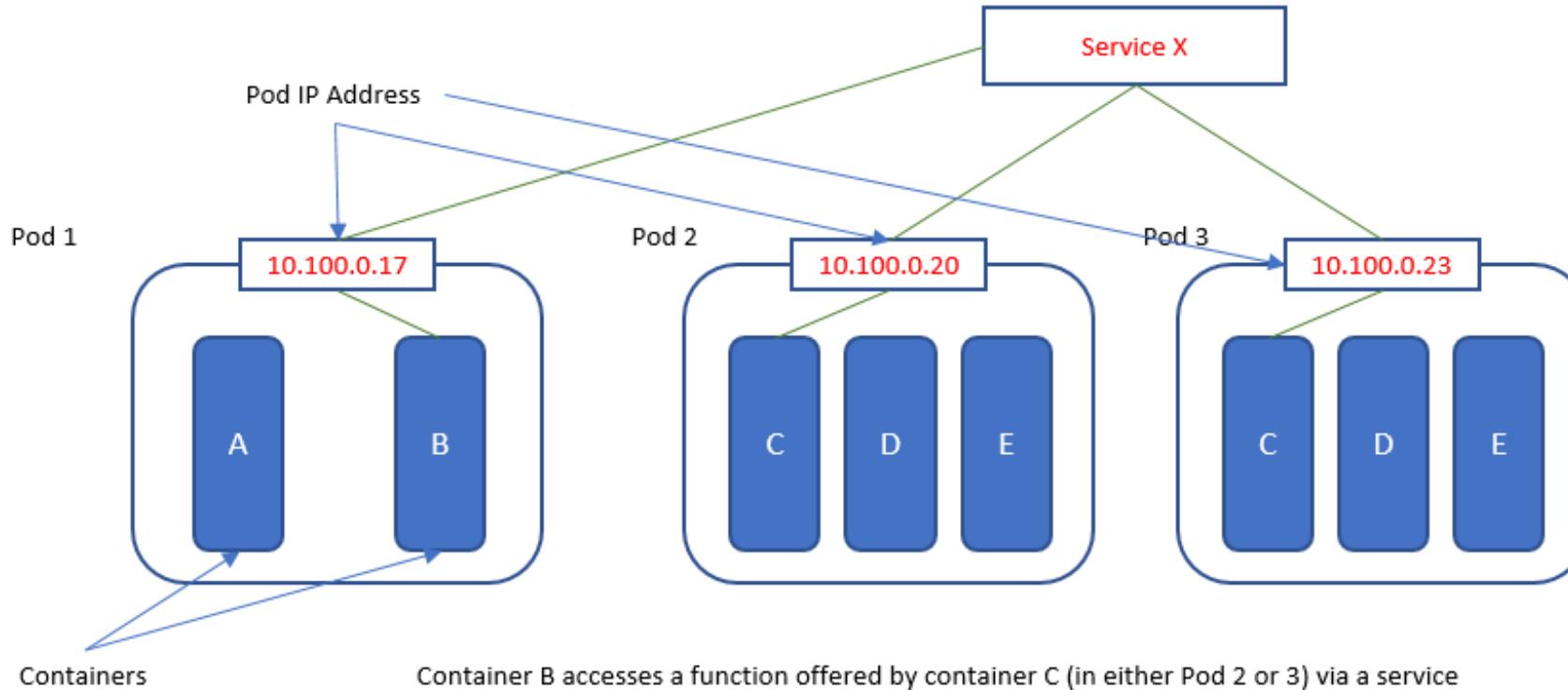
Kubernetes: ¡Que la orquesta comience!

- Kubernetes ofrece un entorno de administración **centrado en contenedores**.
- Kubernetes orquesta la infraestructura de cómputo, redes y almacenamiento para que las cargas de trabajo de los usuarios no tengan que hacerlo.





Funcionamiento básico de Kubernetes



Referencias

- Red Hat. (2020). Contenedores ¿Qué es DOCKER?. 23/02/2020, de Red Hat Sitio web:
<https://www.redhat.com/es/topics/containers/what-is-docker>
- Microsoft. (2018). Introducción a Containers y Docker. 23/02/2020, de Microsoft Sitio web:
<https://docs.microsoft.com/es-es/dotnet/architecture/microservices/container-docker-introduction/>
- Wolf, Gunnar. (octubre de 2016). Contenedores, UN POCO DE HISTORIA. SOFTWARE GURU, 52, 44-45.
- Óscar Villacampa. (2017). Qué es Docker y para qué sirve. 23/02/2020, de Ondho Sitio web: <https://www.ondho.com/que-es-docker-para-que-sirve/>