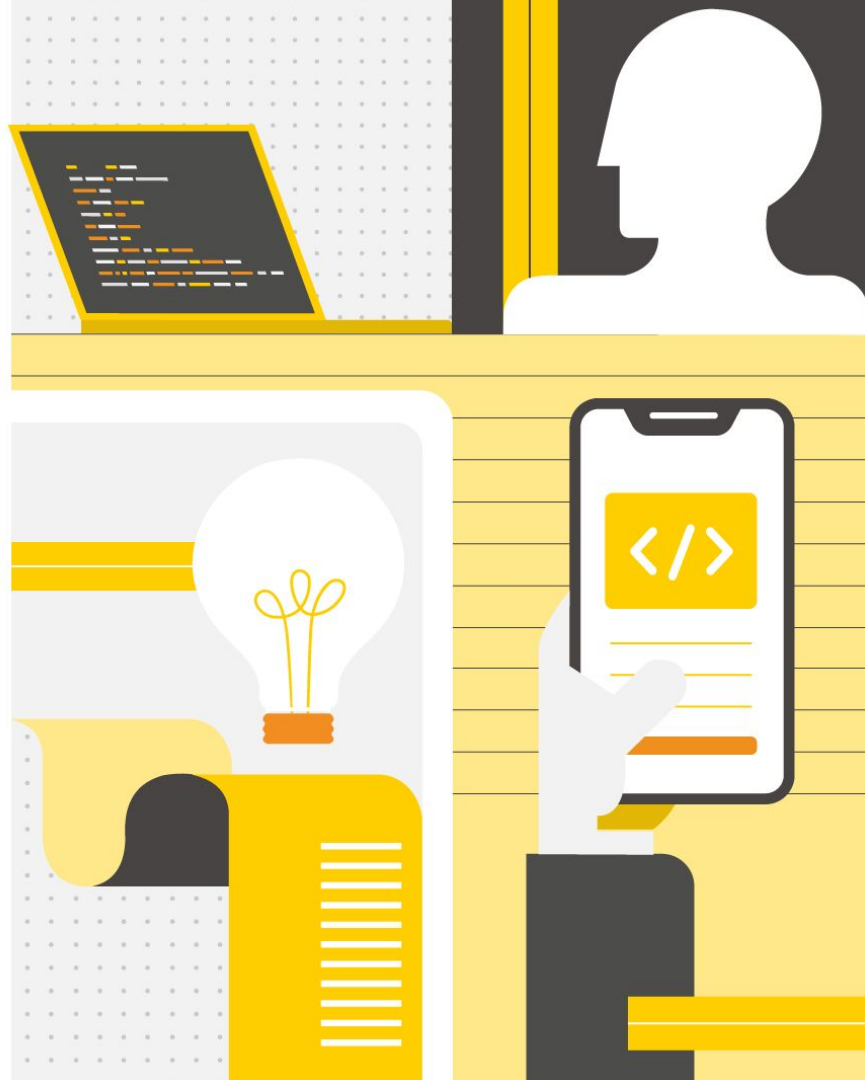




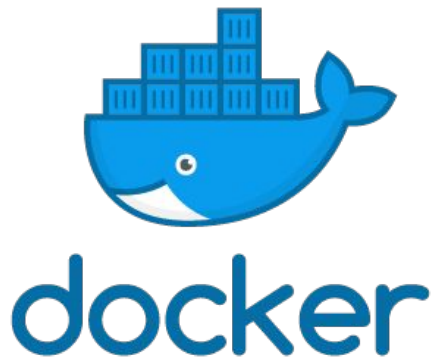
WORKSHOP

DOCKER BÁSICO I

21 de Abril, Santiago



¿QUÉ ES?





SPEAKER

FERNANDO ULLOA
Arquitecto de soluciones



MARTES 21

01

¿QUÉ ES DOCKER?

02

¿CÓMO FUNCIONA?

03

COMPONENTES

04

VENTAJA DE LOS
CONTENEDORES DOCKER

JUEVES 23

05

INSTALACIÓN

06

DOCKER VOLUMES

07

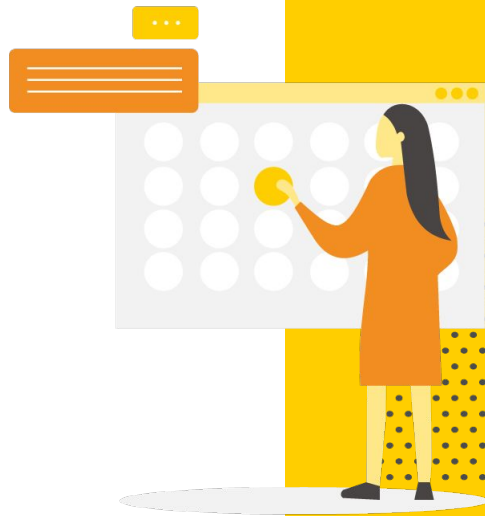
DOCKER COMPOSE

01 I ¿QUÉ ES DOCKER?

La palabra "DOCKER" se refiere a varias cosas. Esto incluye un proyecto de la comunidad open source; las herramientas del proyecto open source; Docker Inc., la empresa que es la principal promotora de ese proyecto; y las herramientas que la empresa admite formalmente. El hecho de que las tecnologías y la empresa compartan el mismo nombre puede ser confuso.

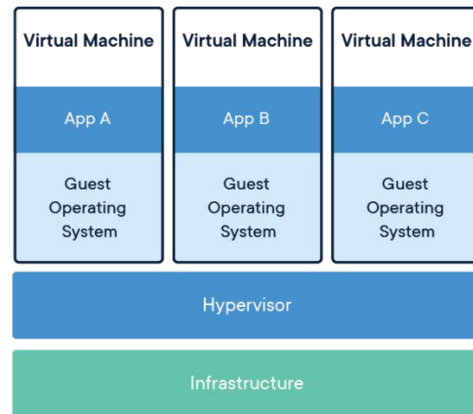
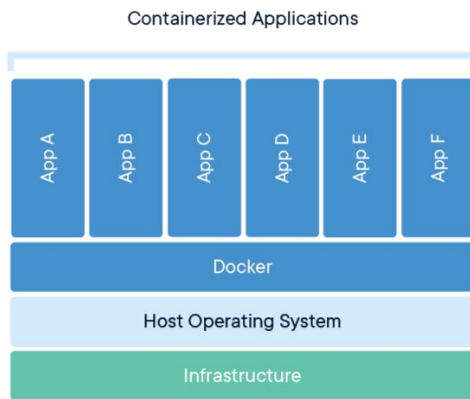
A continuación, una breve explicación:

- "Docker", el software de TI, es una tecnología de creación de contenedores que permite la creación y el uso de contenedores de Linux®.
- La comunidad open source Docker trabaja para mejorar estas tecnologías a fin de beneficiar a todos los usuarios de forma gratuita.
- La empresa, Docker Inc., desarrolla el trabajo de la comunidad Docker, lo hace más seguro y comparte estos avances con el resto de la comunidad. También respalda las tecnologías mejoradas y reforzadas para los clientes empresariales.



¿CÓMO FUNCIONA?

La tecnología Docker usa el kernel de Linux y las funciones de este, para segregar los procesos, de modo que puedan ejecutarse de manera independiente. El propósito de los contenedores es esta independencia: la capacidad de ejecutar varios procesos y aplicaciones por separado para hacer un mejor uso de su infraestructura y, al mismo tiempo, conservar la seguridad que tendría con sistemas separados.

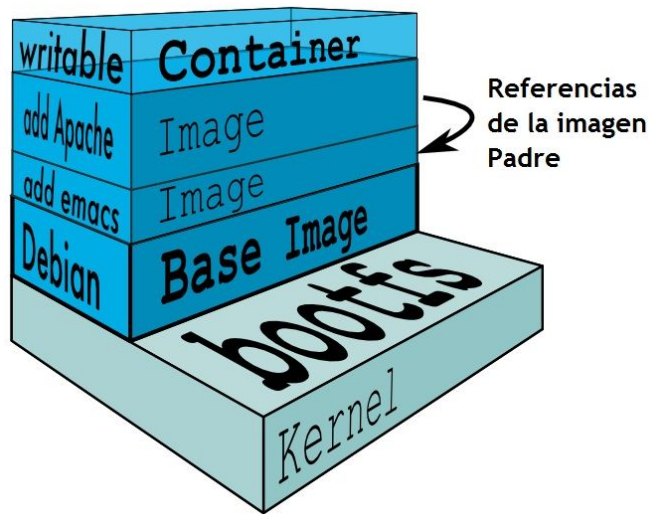


COMPONENTES

Docker tiene una estructura estándar y también ciertos componentes claves, que se repiten siempre y que es bueno tenerlos claros.

Los componentes son los siguientes:

- El cliente y servidor Docker
- Las imágenes
- Los registros
- Los contenedores
- Los volúmenes



VENTAJAS DE LOS CONTENEDORES DOCKER



MODULARIDAD

El enfoque Docker para la creación de contenedores se centra en la capacidad de tomar una parte de una aplicación, para actualizarla o repararla, sin necesidad de tomar la aplicación completa. Además de este enfoque basado en los microservicios, puede compartir procesos entre varias aplicaciones de la misma forma que funciona la arquitectura orientada al servicio (SOA).



CONTROL DE VERSIONES DE IMÁGENES Y CAPAS

Cada archivo de imagen de Docker se compone de una serie de capas. Estas capas se combinan en una sola imagen. Una capa se crea cuando la imagen cambia. Cada vez que un usuario especifica un comando, como *ejecutar* o *copiar*, se crea una nueva capa.

Docker reutiliza estas capas para construir nuevos contenedores, lo cual hace mucho más rápido el proceso de construcción. Los cambios intermedios se comparten entre imágenes, mejorando aún más la velocidad, el tamaño y la eficiencia. El control de versiones es inherente a la creación de capas. Cada vez que se produce un cambio nuevo, básicamente, usted tiene un registro de cambios incorporado: control completo de sus imágenes de contenedor.

VENTAJAS DE LOS CONTENEDORES DOCKER



RESTAURACIÓN

Probablemente la mejor parte de la creación de capas es la capacidad de restaurar. Toda imagen tiene capas. ¿No le gusta la iteración actual de una imagen? Restáurela a la versión anterior. Esto es compatible con un enfoque de desarrollo ágil y permite hacer realidad la integración e implementación continuas (CI/CD) desde una perspectiva de las herramientas.



IMPLEMENTACIÓN RÁPIDA

Solía demorar días desarrollar un nuevo hardware, ejecutarlo, proveerlo y facilitarlo. Y el nivel de esfuerzo y sobrecarga era extenuante. Los contenedores basados en Docker pueden reducir el tiempo de implementación a segundos. Al crear un contenedor para cada proceso, puede compartir rápidamente los procesos similares con nuevas aplicaciones. Y, debido a que un SO no necesita iniciarse para agregar o mover un contenedor, los tiempos de implementación son sustancialmente inferiores. Además, con la velocidad de implementación, puede crear y destruir la información creada por sus contenedores sin preocupación, de forma fácil y rentable.

Por lo tanto, la tecnología Docker es un enfoque más granular y controlable, basado en microservicios, que prioriza la eficiencia.

- **Instalación Virtualbox**
 - **Instalar Centos Linux**
 - **Instalar Docker**
 - **Instalar Portainer**
 - **Crear nuestro primer contenedor**

01

INSTALAR DOCKER

a. Instalar paquetes necesarios

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

b. Configurar Repositorio Docker

```
yum-config-manager --add-repo
```

```
https://download.docker.com/linux/centos/docker-ce.repo
```

c. Instalar Docker

```
yum install docker-ce --nobest -y
```

d. Agregar usuario actual al grupo de docker

```
usermod -aG docker $(whoami)
```

e. Habilitar servicio docker

```
systemctl enable docker.service
```

f. Iniciar Servicio Docker

```
systemctl start docker.service
```

01

INSTALAR DOCKER - COMPOSE

02

INSTALAR PORTAINER

a. **Instalar paquetes extras**

```
yum install epel-release
```

b. **Instalar python pip**

```
yum install -y platform-python-pipplatform-python-devel gcc
```

c. **Instalar Docker Compose**

```
pip3 install docker-compose
```

d. **Actualizar python**

```
yum upgrade python*
```

a. **Crear Volumen Para Portainer**

```
docker volume create portainer_data
```

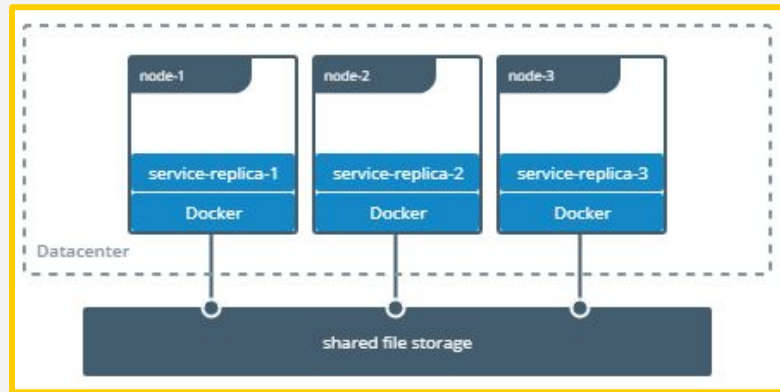
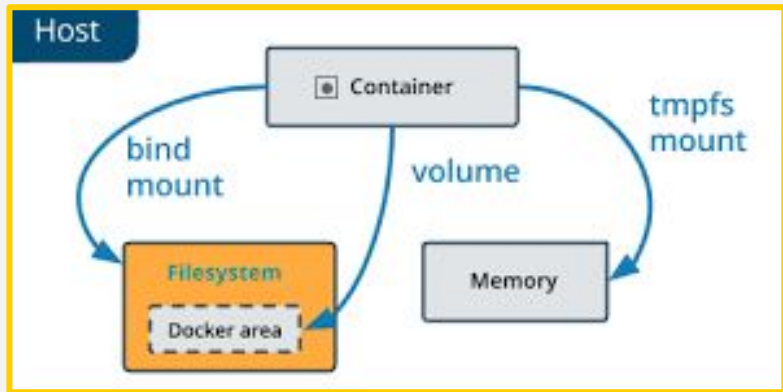
b. **Crear contenedor Portainer**

```
docker run -d -p 9000:9000 -v  
/var/run/docker.sock:/var/run/docker.sock  
-v portainer_data:/data portainer/portainer
```

DOCKER VOLUMES

Los volúmenes en Docker sirven para tener un espacio de disco que no depende directamente del contenedor y que puede ser eventualmente compartido entre distintos contenedores.

Es útil por ejemplo en un servidor web montar un contenedor con el sitio que se despliega, puesto que se podrían montar contenedores que manejan distintas instancias de Apache, pero que muestran un mismo sitio alojado en un volumen.



DOCKER VOLUMES

1. Crear Volumen

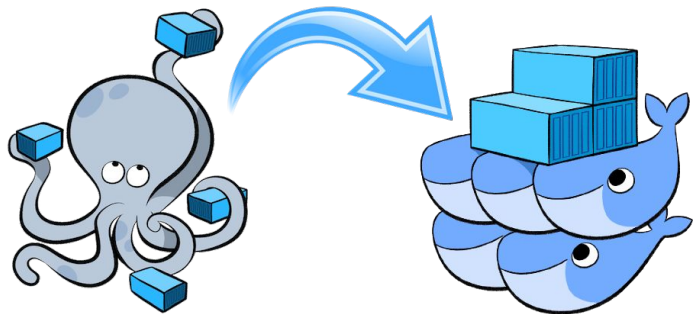
```
docker volume create my-vol
```

2. Listar Volumenenes creados

```
docker volume ls
```

3. Crear contenedor con volumen

```
docker run -d --name prueba -v my-vol:/volumen centos
```



DOCKER COMPOSE

Docker Compose, nos permite crear contenedores a partir de una “receta”. Es distinto a un dockerfile, pues nos permite construir un “stack” que sería en otras palabras, un set de contenedores que cumplen o se apoyan para prestar un servicio

Por ejemplo, para montar un sitio web Wordpress, necesitamos normalmente:

- **Sistema Operativo servidor**
- **Servidor HTTP + php**
- **Servidor de Base de Datos**

En el mundo de docker, sabemos que el sistema operativo del servidor no es “necesario”, y que podemos crear un contenedor para el servidor HTTP, pero necesitaríamos también un servidor para la base de datos, que podría ser otro contenedor.

DOCKER COMPOSE

Con docker compose, escribiremos una receta que incluya dos contenedores, uno para el servidor web y otro para la base de datos.

1. Crear un archivo llamado `stack.yml`
2. Agregar el siguiente contenido

```
version: '3.3'
```

```
services:
```

```
  db:
```

```
    image: mysql:5.7
```

```
    volumes:
```

```
      - db_data:/var/lib/mysql
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: somewordpress
```

```
      MYSQL_DATABASE: wordpress
```

```
      MYSQL_USER: wordpress
```

```
      MYSQL_PASSWORD: wordpress
```

- r

DOCKER COMPOSE

```
wordpress:
  depends_on:
    - db
  image: wordpress:latest
  ports:
    - "8000:80"
  restart: always
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wordpress
    WORDPRESS_DB_PASSWORD: wordpress
    WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
```

1. Correr el siguiente comando luego de haber creado el archivo y haber agregado el texto

```
docker-compose -f stack.yml up
```



WWW.API-UX.COM



Monseñor Sótero Sanz 161, Providencia - Santiago de Chile

Tel. (+56 2) 3244 5004 Correo: marketing@api-ux.com