

Docker

Sistemas y Tecnologías Web (CC3062) - 2026

Docker

Semestre 01, 2026

Virtualización

Tecnología que permite crear múltiples entornos virtuales a partir de un único sistema físico (hardware).

Hipervisor actúa como una capa de abstracción entre el hardware físico y las máquinas virtuales.

Se simula el hardware completo.

Ventajas

- Aislamiento completo
- Seguridad mejorada
- Compatibilidad con múltiples SO
- Flexibilidad
- Escalabilidad

Desventajas

- Alto consumo de recursos
- Arranque lento
- Mayor espacio en disco

Contenerización

La contenerización es el proceso de empaquetar una aplicación junto con sus dependencias en un "contenedor".

El contenedor garantiza que la aplicación funcione de manera consistente en cualquier entorno.

Se comparte el núcleo (kernel) con el sistema operativo.

Ventajas

- Portabilidad
- Reproducibilidad
- Aislamiento
- Escalabilidad
- Agilización de procesos

Desventajas

- Complejidad.
- Riesgos de seguridad (update).
- Almacenamiento persistente.
- Dificultad para depurar problemas.
- Desempeño puede verse afectado.

Virtualización vs Contenedores

VM:

- Hipervisor
- Sistema operativo completo
- Alto consumo

Contenedor:

- Comparte kernel
- Más ligero
- Arranque rápido

☐☐☐ Antes de continuar...

take-this

Host: máquina en donde está corriendo docker.

Contenedor: máquina generada con contenerización.

El comando `docker ps` lista los contenedores que están corriendo.

Docker

Docker es una plataforma de software que utiliza la tecnología de contenedores para crear, desplegar y ejecutar aplicaciones.

Permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores.

Esto garantiza que la aplicación se ejecute de manera consistente en cualquier entorno.

Lanzado en 2013 por la empresa Docker Inc.

Docker es el nombre del producto que utiliza la tecnología de contenedores.

Tupperware || Kleenex || Maseca || Pritt

¿Qué ofrece Docker?

- CLI para gestionar contenedores.
- Herramientas de orquestación y despliegue.
- Repositorios públicos y privados en Docker Hub.

Alternativas

- Podman: Alternativa sin daemon.
- rkt: Solución de contenedores de CoreOS.
- LXC: Contenedores ligeros en Linux.

¿Por qué elegir Docker?

- Amplio ecosistema y documentación.
- Integración continua con servicios como Kubernetes.
- Facilidad de uso para desarrolladores y DevOps.
- Paz Mental

Soluciona

- Problemas de compatibilidad entre entornos (desarrollo/producción).
- Conflictos de dependencias entre proyectos.

- Complejidad en el despliegue de aplicaciones complejas.

"En mi compu si corre pero en el servidor no [] [] [] [] [] []"

Componentes

Docker Engine

Es el núcleo de Docker, responsable de crear y ejecutar contenedores.

- Daemon
- API
- CLI

engine

Dockerfile

Archivo de texto que contiene instrucciones para crear una imagen.

- Define el entorno de la aplicación.
- Especifica dependencias y configuraciones.

```
FROM node:14

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

CMD ["node", "index.js"]
```

Imágenes

Archivos inmutables que contienen el entorno de ejecución y la aplicación.

Se crean a partir de un Dockerfile y sirven como plantilla para crear contenedores.

Capas: Cada instrucción en el Dockerfile genera una capa en la imagen.

layers

Contenedores

Instancias basadas en imágenes.

- Son ligeros y rápidos porque comparten el núcleo del host.
- Cada contenedor es independiente y tiene su propio sistema de archivos.
- Persistencia de datos mediante volúmenes.

flowdoker

Docker Hub

Repositorio público y privado de imágenes Docker.

- Almacén centralizado para compartir y descargar imágenes.
- Imágenes oficiales
- Imágenes de la comunidad

hub

Docker Compose

Herramienta para definir y ejecutar aplicaciones multi-contenedor.

- Usa un archivo docker-compose.yml para configurar servicios.

- Múltiples contenedores, un solo comando.
- Comparte la red.

```
services:
  nginx:
    image: nginx:latest
    container_name: my-nginx
    ports:
      - "8080:80"
```

Puertos

Los contenedores en Docker funcionan en una red interna aislada.

Para que el host pueda acceder a los servicios dentro del contenedor, es necesario mapear los puertos.

Podemos mapear cuantos puertos necesite nuestra aplicación.

Brindan un punto de acceso desde el Host hacia el Contenedor.

- El puerto 8080 se enlaza al 80 (HTTP).
- El puerto 8443 se enlaza al 443 (HTTPS).

Importante

- El puerto del host debe estar libre antes de asignarlo.
- Los puertos en uso no pueden ser mapeados nuevamente.

Volúmenes

Los volúmenes en Docker permiten almacenar datos de manera persistente fuera del contenedor.

Esto asegura que los datos no se pierdan si el contenedor se elimina.

Tipos

- **Con Nombre:** Administrados por Docker.
- **Enlaces (Bind Mounts):** Se conecta un directorio del host al contenedor.

¿Cuándo usar volúmenes?

- Almacenar datos que necesitan persistentes (eliminamos contenedor).
- Compartir archivos entre múltiples contenedores.

Redes

Las redes en Docker permiten que los contenedores se comuniquen entre sí y con el host.

Comandos Básicos

- docker pull
- docker build -t nombre .
- docker run
- docker run -p host:contenedor
- docker ps
- docker stop
- docker rm
- docker compose up
- docker compose down

Resumen

Docker permite el despliegue ágil y la gestión eficiente de aplicaciones en contenedores.

Mejora la portabilidad.

Nos hace la vida más fácil.

📦 📦 **¡Aprendan Docker!** 📦 📦