UD2. Docker

Índice

Introducción....2
¿Qué es un contenedor Docker?....2
¿Cuáles son los beneficios de usar Docker?....3
Instalar docker en linux....3
Crear un contenedor....3
Conectarse a un contenedor....4
listar los contenedores....4
Ejemplo: docker oracle....4
instalamos el contenedor de Oracle....4
arrancar y parar....4
logarse en el docker....4
acceder desde fóra....4

Introducción

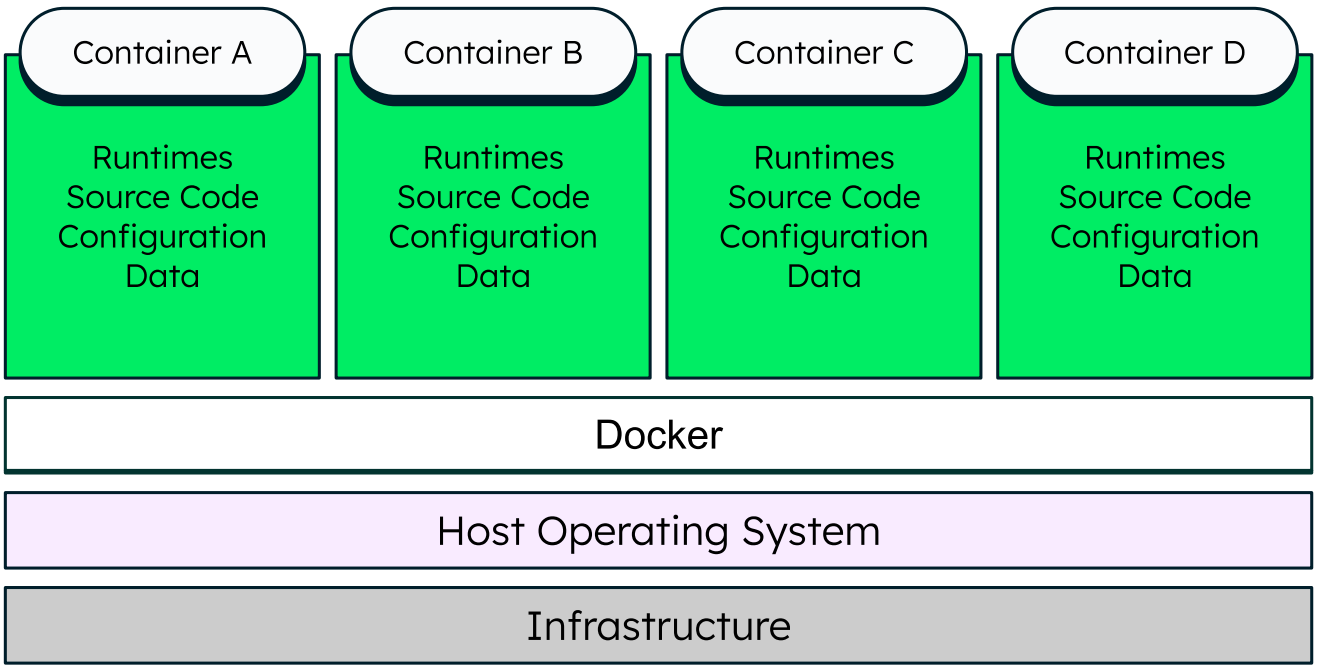
La empresa Docker existe desde 2013 y popularizó el concepto de contenedores. Desde entonces, muchos desarrolladores de software han adoptado los contenedores como parte de su flujo de trabajo diario. Estos contenedores pueden ayudar a los desarrolladores a ejecutar una aplicación en un entorno aislado y consistente.

¿Qué es un contenedor Docker?

Un contenedor Docker es una unidad única que contiene una aplicación y todas sus configuraciones o dependencias necesarias. Imagina una especie de archivo zip grande que incluya todo lo necesario para ejecutar tu aplicación en cualquier sistema operativo o hardware. Docker es una herramienta para ejecutar esos contenedores.

El concepto de contenedores se remonta a los años 70. Tiene sus orígenes en la llegada del sistema operativo Unix. El comando **chroot** se introdujo como parte del sistema operativo y cambia el directorio raíz de un proceso y sus hijos. Este fue el comienzo del aislamiento de procesos.

Los contenedores son una forma de aislar todo lo necesario para ejecutar una aplicación en su entorno. Docker popularizó el concepto de contenedores cuando presentó Docker Engine, que permite que los contenedores se ejecuten en cualquier sistema operativo. Gracias a ese motor, un contenedor puede ejecutarse de manera uniforme en Linux, macOS y Windows.



Los contenedores se ejecutan como procesos individuales en el motor Docker, por tanto se ejecutan en sus propios procesos aislados en un espacio de usuario determinado. Esto tiene la ventaja de reducir significativamente los recursos necesarios para su ejecución en comparación con las máquinas virtuales, que requieren un sistema operativo completo propio. Por lo tanto, el costo total de ejecutar una aplicación como contenedor es mucho menor que ejecutarla en una máquina virtual.

La tecnología de contenedores también está en el corazón de la de microservicios. Una revolución en el desarrollo de software. Su reducido tamaño permite ejecutar múltiples aplicaciones independientes y aisladas. Los contenedores también permiten que pequeños microservicios se ejecuten juntos. Las herramientas de orquestación, como por ejemplo Kubernetes también permite actualizar de forma sencilla microservicios únicos sin tiempo de inactividad.

Otro beneficio de usar contenedores es que son efímeros por naturaleza. Cuando se reinicia un contenedor, siempre será el mismo y cualquier cambio que pudiera haberse producido mientras estaba en ejecución desaparecerá. Esto es excelente para garantizar que la aplicación siempre se ejecute de manera consistente. También es una característica excelente para los desarrolladores de software que ejecutan pruebas en su sistema, ya que cualquier dato ingresado se eliminaría y la ejecución de la prueba comenzaría desde cero cada vez.

Por otro lado, esto puede resultar un tanto complicado cuando es necesario preservar los datos, como sería el caso al ejecutar un contenedor para una base de datos. Si necesita preservar los datos, deberá montar un volumen al que se pueda acceder desde el contenedor.

¿Cuáles son los beneficios de usar Docker?

Existen muchas ventajas de utilizar contenedores como parte de tu vida diaria como desarrollador de software.

- **Coherencia.** Al utilizar la tecnología de contenedores, puede asegurarse de que todos los miembros de su equipo utilicen exactamente los mismos tiempos de ejecución y configuraciones. También reduce significativamente la fricción en la implementación, ya que el entorno de producción será coherente con su entorno de desarrollo.
- **Ligeros.** Los contenedores Docker se inician rápidamente y consumen recursos mínimos en comparación con las máquinas virtuales.
- **Efímero.** Cualquier cambio en el sistema de archivos del contenedor se destruirá al finalizar el proceso. Esta impermanencia garantiza un entorno nuevo en cada inicio.

Instalar docker en linux

Una forma fácil de instalarlo es usar el script que prové el propio docker

root#> apt install curl
root#> curl -fsSL https://get.docker.com -o get-docker.sh
root#> sh get-docker.sh

Crear un contenedor

```
docker run --name nombre -d contenedor:version
```

Si queremos la última versión podemos poner *latest*

-- name le pondremos un nombre al contenedor

Si se necesita acceder al servidor desde otra aplicación que se ejecuta localmente, debemos exponer un puerto utilizando el argumento -p.

```
docker run --name nombre -d -p puerto:puertolocal contenedor:versión
```

Todos los datos creados como parte del ciclo de vida de ese contenedor se destruirán una vez que se elimine el contenedor. Puedes hacerlo con los comandos **stop** y **rm** de Docker .

```
docker stop nombre && docker rm nombre
```

```
Para arrancarlo usaremos start
```

```
docker start nombre
```

Si deseas conservar los datos en tu máquina local, puedes montar un volumen usando -v.

```
docker run --name nombre -d -p puerto:puertolocal -v $(pwd)/datos:/datos/db
contenedor:version
```

Si detienes y reinicias el contenedor, todos los datos seguirán estando allí.

Si tu aplicación se ejecuta dentro de un contenedor, puedes ejecutar un servicio como parte de la misma red Docker que tu aplicación mediante **--network**.

```
docker run --name nombrecontenedor -d --network nombrered contenedor:version
```

Conectarse a un contenedor

Para administrar tu servidor, acceder, importar y exportar tus datos, puedes utilizar el parámetro -it (interactivo) que no sólo ejecuta el comando sinó que espera respuesta:

docker exec -it nombre /*bin/ba*sh

listar los contenedores

listamos los contenedores activos y todos

docker ps

docker ps -a

Ejemplo: docker oracle

instalamos el contenedor de Oracle

docker run -d --name 23ai p 1521:1521 container-registry.oracle.com/database/free:latest

arrancar y parar

docker start 23ai

docker stop 23ai

logarse en el docker

docker exec -it 23ai /bin/bash

acceder desde fuera

sqlplus system/oracle@localhost:1521/FREEPDB1