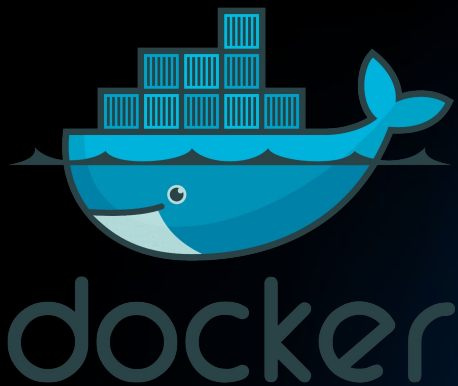


Docker, ECS y EKS



ECS



EKS



Crombie

Virtualización

Es una tecnología que permite ejecutar múltiples sistemas operativos o aplicaciones en un solo hardware físico al dividir los recursos.

Tipos comunes de virtualización:

- **Virtualización con máquinas virtuales (VMs):** Ejecutan un sistema operativo completo en un hipervisor (e.g., VMware, VirtualBox).
- **Virtualización con contenedores:** Ejecutan aplicaciones aisladas dentro de un mismo sistema operativo.

Diferencias entre máquinas virtuales y contenedores

Máquinas virtuales (VMs)	Contenedores
Pesadas: incluyen el SO completo.	Ligeros: comparten el kernel del SO.
Arranque lento (varios minutos).	Arranque rápido (segundos).
Uso alto de recursos.	Uso eficiente de recursos.
Más adecuados para aplicaciones diversas o SO distintos.	Ideales para microservicios.

Contenedores

Son un entorno ligero, portátil y aislado que contiene todo lo necesario para ejecutar una aplicación (código, dependencias, bibliotecas).

Muchas veces, nuestra aplicación necesita de librerías o paquetes y estos se instalan en nuestra PC. Como desarrolladores, esto puede ser un problema al manejar múltiples proyectos, por ejemplo, si trabajáramos con python2 en un proyecto y con python3 en otro, necesitamos cambiar de versión constantemente (instalar y desinstalar constantemente)

→ Asegura que la aplicación funcione de manera consistente en cualquier entorno, ya sea en la computadora del desarrollador, un servidor de pruebas o en producción.

→ Aislamiento: Los problemas de una aplicación no afectan a las demás.

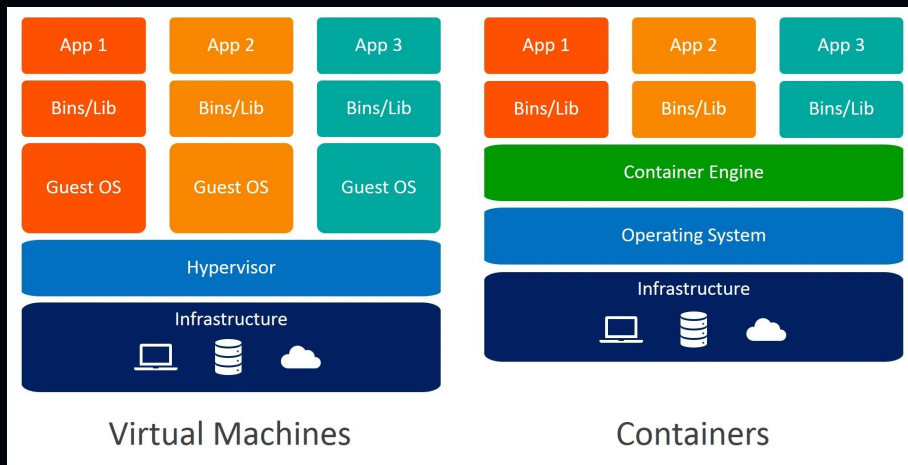
→ Se llaman así porque funcionan de manera muy similar a los contenedores de carga marítima que transportan bienes alrededor del mundo.



Contenedores

A diferencia de las máquinas virtuales, que necesitan un sistema operativo completo, los contenedores comparten el núcleo del sistema operativo del host, lo que los hace más ligeros, rápidos y eficientes.

Es similar a una caja donde guardas tu aplicación con todas sus herramientas. Esta caja puede ser transportada y ejecutada en cualquier lugar que soporte contenedores, como servidores locales, en la nube o en tu laptop. **Docker** es la tecnología más popular para crear y gestionar estos contenedores.



Contenedores: como nacen

Las máquinas virtuales fueron una de las primeras soluciones para aislar aplicaciones. Cada VM incluye un sistema operativo completo, lo que consume muchos recursos. Aunque las VMs resolvían el problema de aislamiento, resultaban poco eficientes y lentas, ya que cada una necesitaba su propio sistema operativo.

Con el auge de las aplicaciones en la nube y el desarrollo de software ágil, surgió la necesidad de soluciones más eficientes y portátiles. Los desarrolladores querían asegurarse de que sus aplicaciones funcionaran de la misma manera en cualquier entorno (local, desarrollo, producción, etc.) sin tener que lidiar con diferencias entre sistemas operativos o configuraciones.

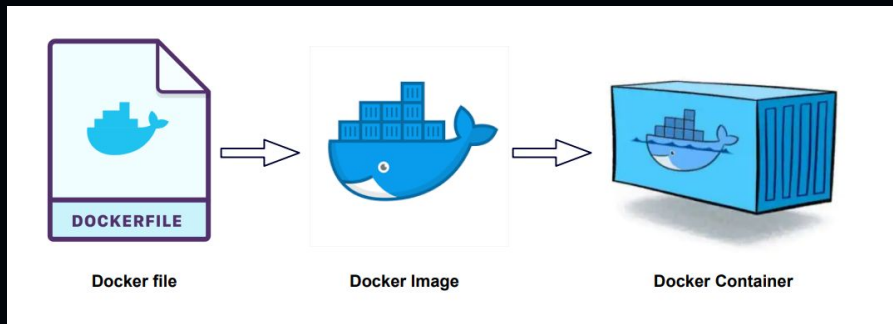
Docker, lanzado en 2013, popularizó el uso de contenedores en el desarrollo de software. Ofreció una forma sencilla de empaquetar aplicaciones junto con sus dependencias en contenedores, lo que resolvía muchos problemas de compatibilidad y despliegue.

Contenedores: Docker para developers

Docker es una herramienta que permite empaquetar aplicaciones y sus dependencias en contenedores, que son entornos ligeros y aislados. Esto significa que puedes desarrollar, probar y ejecutar aplicaciones de manera consistente sin importar en qué sistema o entorno estés trabajando

Documentación y descarga:

<https://www.docker.com/>

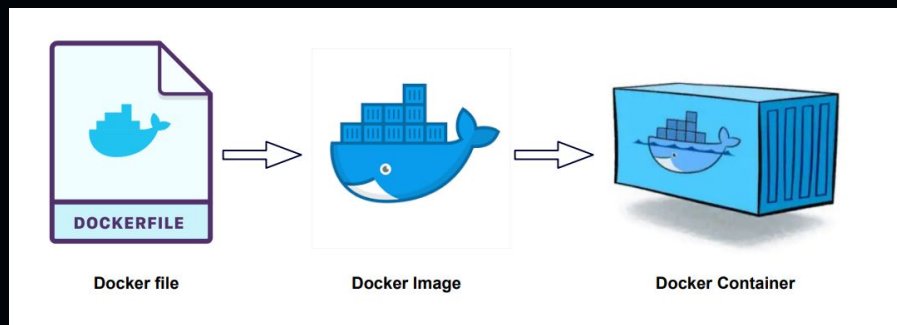


¿Cómo se crean los contenedores?

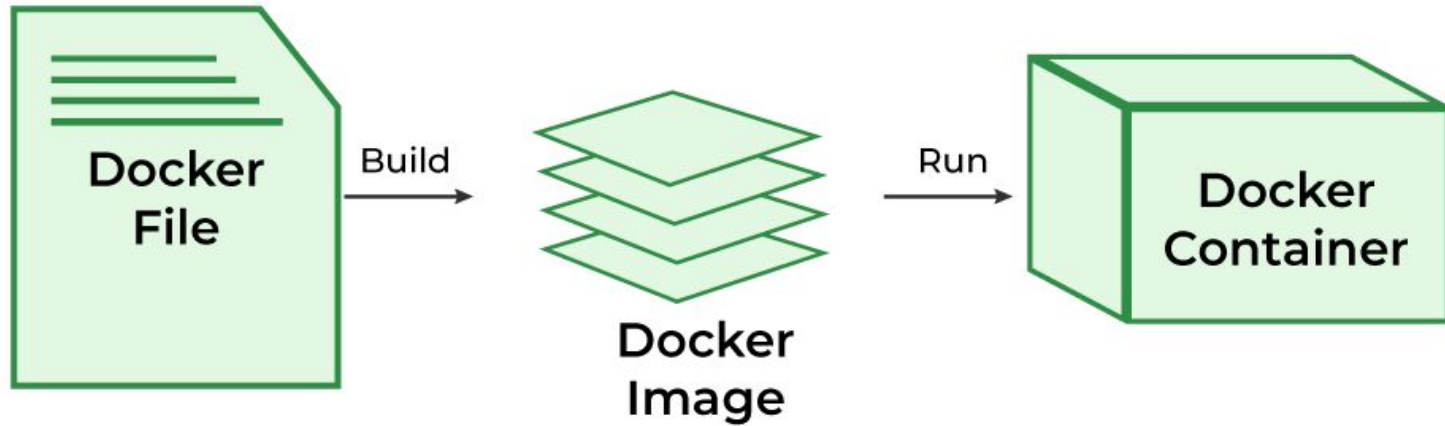
Crear un contenedor es como preparar una receta en tu cocina, pero para una aplicación.

Necesitas un Dockerfile, que es como tu receta escrita: ahí describes qué ingredientes (o dependencias) necesita tu aplicación, cómo configurarla y qué pasos seguir para que esté lista.

- Escribir el Dockerfile: Piensa en esto como la lista de pasos. Empiezas eligiendo una base (por ejemplo, "quiero que esta app use Node.js"), luego dices qué ingredientes (dependencias) necesitas instalar, copias el código de tu aplicación y definir cómo ejecutarla.
- Construir la imagen: Usas el comando **docker build** para que Docker siga tu receta y cree una "imagen", que es como un molde perfecto de tu aplicación con todo listo.
- Ejecutar el contenedor: Con el comando **docker run**, le das vida a esa imagen y ahora tu app está corriendo en un contenedor, lista para usarse en cualquier lugar.



Dockerfile

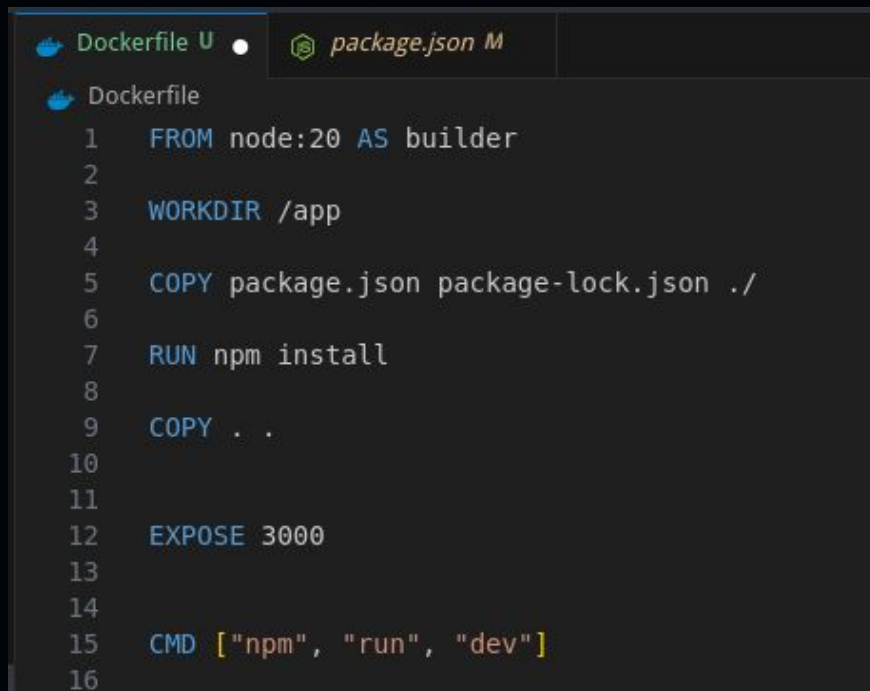


Dockerfile: ejemplo

Una vez definido el dockerfile, podemos ejecutar los siguientes comandos para construir la imagen:

- `docker build -t miapp .`
- `docker run -d -p 8080:8080 miapp`

Una vez ejecutado, nuestra aplicación estará corriendo en un contenedor en nuestra máquina.



The screenshot shows a code editor with two tabs: 'Dockerfile U' and 'package.json M'. The 'Dockerfile' tab is active, displaying a Dockerfile with the following content:

```
1 FROM node:20 AS builder
2
3 WORKDIR /app
4
5 COPY package.json package-lock.json ./
6
7 RUN npm install
8
9 COPY . .
10
11
12 EXPOSE 3000
13
14
15 CMD ["npm", "run", "dev"]
16
```

Amazon ECS

Amazon Elastic Container Service (ECS) es un servicio de orquestación de contenedores completamente administrado que permite ejecutar y escalar aplicaciones en contenedores en AWS.

Casos de uso:

- Ejecución de aplicaciones microservicio.
- Procesamiento por lotes.
- Hosting de aplicaciones basadas en contenedores.



Por qué usar Amazon ECS?

- **Escalabilidad:** Facilita el escalado automático de aplicaciones.
- **Flexibilidad:** Compatible con Docker y otros servicios AWS (e.g., EC2, Fargate).
- **Integración:** Funciona con servicios como CloudWatch, IAM, y ALB.
- **Eliminación de la complejidad:** AWS gestiona la infraestructura de la orquestación.

Amazon EKS

Amazon Elastic Kubernetes Service (EKS) es un servicio totalmente gestionado que permite ejecutar Kubernetes en AWS sin la necesidad de configurar manualmente el clúster.

Que es Kubernetes?:

- Un sistema de orquestación de contenedores que automatiza la implementación, gestión y escalado de aplicaciones en contenedores.
- Proporciona herramientas avanzadas para administrar aplicaciones complejas basadas en microservicios.



Por qué usar Amazon EKS?

- **Servicio gestionado:** AWS administra el plano de control de Kubernetes (control plane), incluidas las actualizaciones, seguridad y disponibilidad.
- **Compatibilidad estándar:** Compatible con cualquier herramienta o configuración que funcione en un clúster de Kubernetes.
- **Integración con AWS:** Funciona de manera nativa con servicios como EC2, Fargate, IAM, y ALB.
- **Escalabilidad:** Admite cargas dinámicas con escalado automático.