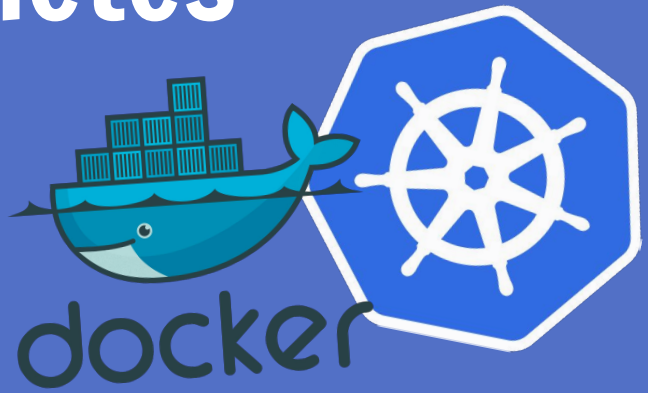
Introducción a microservicios utilizando Docker y Kubernetes

Luis Arana



Docker

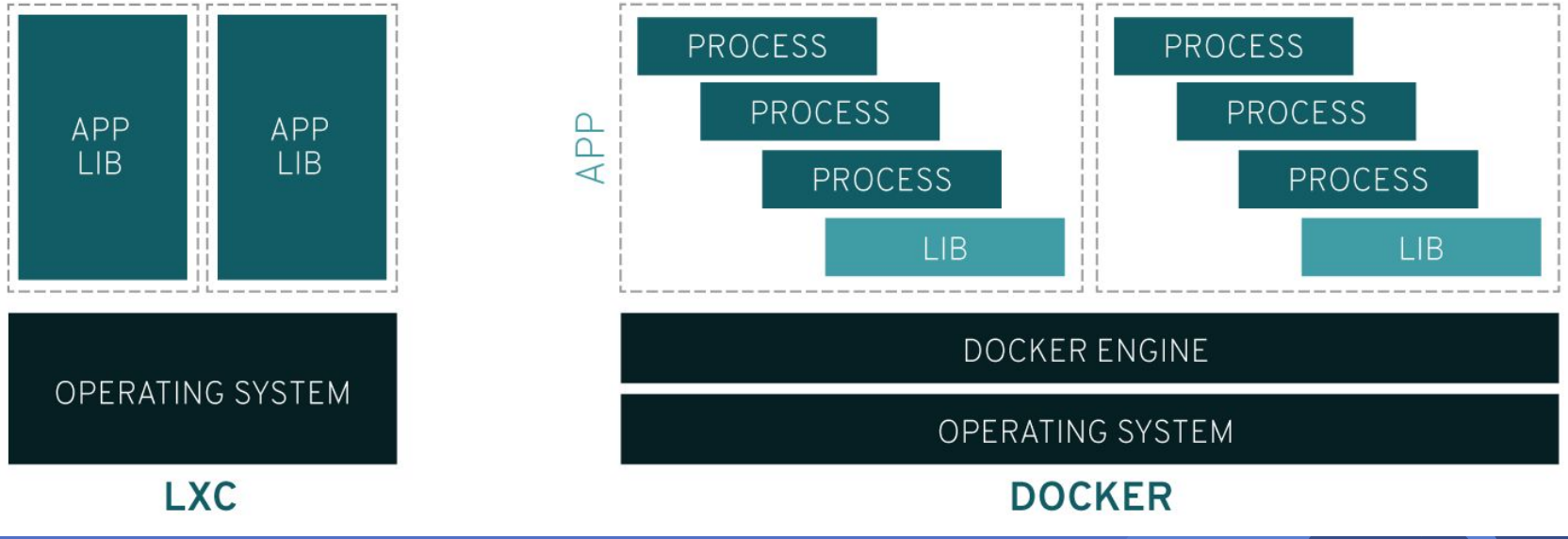
¿Qué es Docker?

Es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente.

¿Cómo funciona Docker?

De manera similar a cómo una máquina virtual, Docker virtualiza el hardware del servidor, los contenedores virtualizan el sistema operativo de un servidor.

Traditional Linux containers vs. Docker



Dockerfile

```
FROM node:14 as build
WORKDIR /usr/local/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3800
CMD ["node", "index.js"]
```

Kubernetes

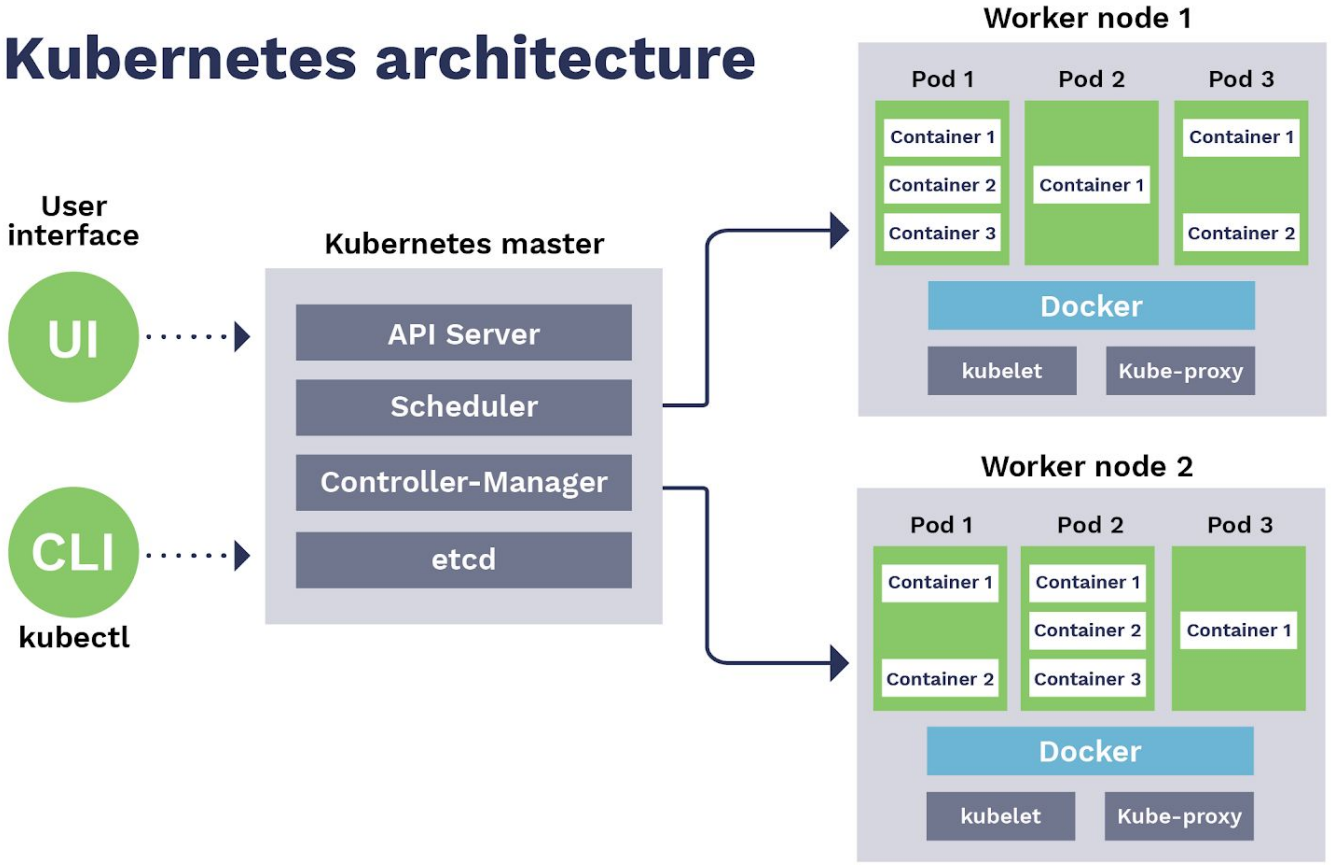
¿Qué es Kubernetes?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios.

¿Por qué es importante conocer Kubernetes?

Kubernetes proporciona toda la infraestructura necesaria para que los desarrolladores o DevOps puedan construir un entorno de desarrollo focalizado en el contenedor, incluyendo funcionalidades tan importantes como el autoescalado, la autorreplicación o el reinicio automático.

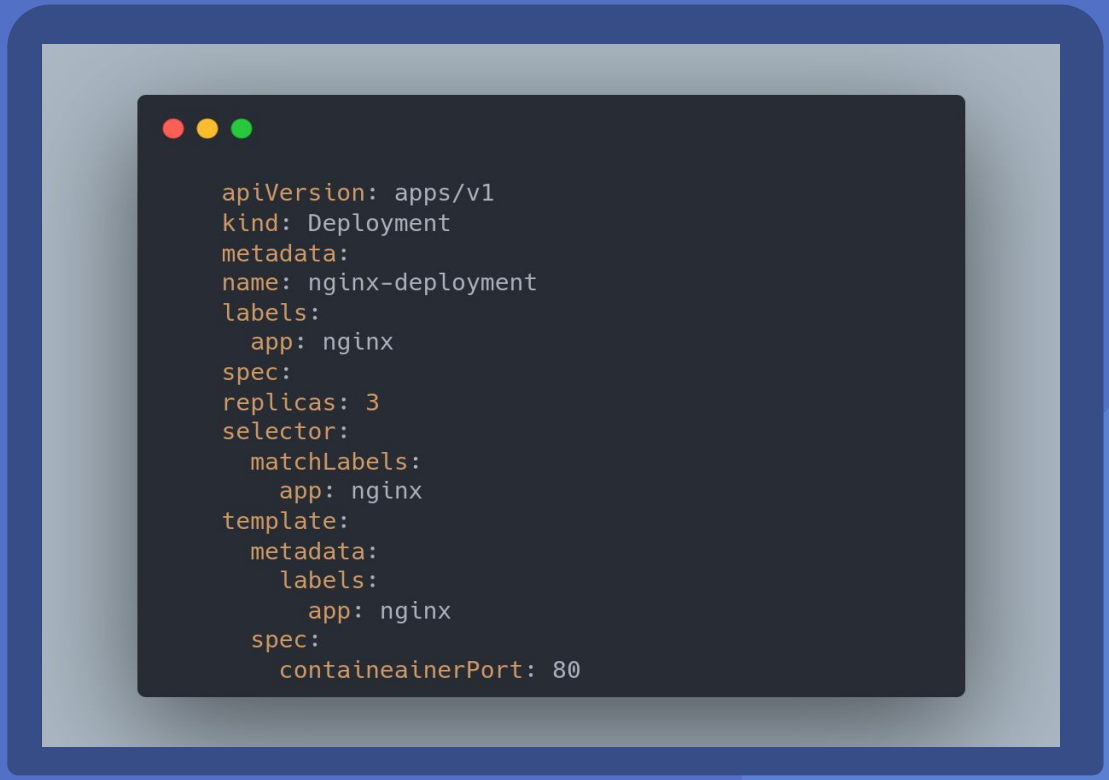
Kubernetes architecture



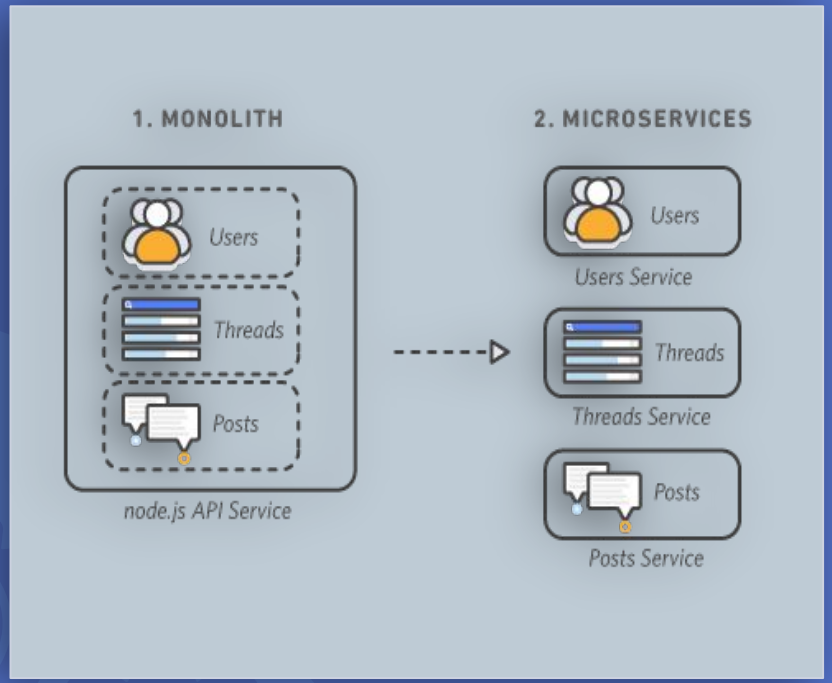
YML - YAML

YAML es un lenguaje de serialización de datos que suele utilizarse en el diseño de archivos de configuración. Para algunas personas, YAML significa otro lenguaje de marcado más; para otras, es un acrónimo recursivo que quiere decir YAML no es un lenguaje de marcado, lo que enfatiza la idea de que se utiliza para los datos, no para los documentos.

Deploy en Kubernetes



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx-deployment
labels:
  app: nginx
spec:
replicas: 3
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containeainerPort: 80
```



Microservicios

Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.

Características de los microservicios

Independencia

Cada módulo es un ejecutable en sí mismo y puede ser desplegado sin afectar al resto.

Granularidad

Cada microservicio se encarga de una única funcionalidad de negocio.

Aplicación modular

Posibilidad de ampliar o reducir en función de las necesidades

Arquitectura distribuida

Cada microservicio dispone de su propia base de datos.

Beneficios de los microservicios

Escalado flexible

Libertad tecnológica

Resistencia

Implementación sencilla

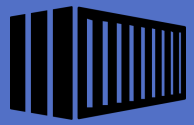
Agilidad

Código reutilizable

Implementaciones de microservicios

Serverless

- Las aplicaciones se dividen en diferentes funcionalidades (o servicios), los cuales son detonados por eventos. El proveedor de la nube se ocupa del resto y asegura que tus funciones estén siempre disponibles y utilizables, no importando que.



Contenedores

- Crear un contenedor el cual tenga todas las dependencias requeridas pre instaladas, poner el código de la aplicación dentro de este y correr donde sea que el runtime del contenedor esté instalado.

Servidor Físico

- Solíamos usar nuestra propia infraestructura en forma de servidores físicos. Configurar esa máquinas, entregamos nuestro código, las escalamos y mantenemos. Todo el proceso era manual y bastante lento para arrancar.

Demo

Ejemplo de microservicios y realizar un despliegue utilizando docker y kubernetes

¡Gracias!

¿Preguntas?

luarana631@gmail.com

https://github.com/LuisArana631/Conferencia-microservicios