

Grado en Ingeniería Informática

Sistemas Industriales

Arquitectura orientada a contenedores

Pablo Casado

pcasado@dtic.ua.es

Departamento de Tecnología Informática y Computación

2022 - 2023

1. Introducción

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

1. Introducción a los servidores de aplicaciones

2. Despliegue en un servidor de aplicaciones

3. Arquitectura orientada a contenedores

4. PaaS y orquestadores de contenedores

1. Introducción

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

¿Qué es un contenedor de software?

Un contenedor es un paquete de software estándar que agrupa el código de una aplicación con las bibliotecas y los archivos de configuración asociados, junto con las dependencias necesarias para que la aplicación se ejecute.



1. Introducción

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

¿Qué beneficios me proporciona la contenerización de software?

El problema de que una aplicación no se ejecute correctamente cuando se cambia de un entorno a otro es tan antiguo como el propio desarrollo de software. Los contenedores solucionan este problema proporcionando una estructura ligera e inmutable para el empaquetado e implementación de aplicaciones.



1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

¿Entonces es una máquina virtual?

¡NO! La filosofía de los contenedores es distinta a la de las máquinas virtuales.

Son tecnologías que persiguen un fin similar pero con enfoques totalmente diferentes.

- Un contenedor funciona a partir de imágenes que se pueden reutilizar.
- Un contenedor aísla una aplicación y no un sistema operativo completo.
- Un contenedor es mucho más ligero que una máquina virtual.

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Diferencias con Java Virtual Machine

- Una aplicación Java que se ejecuta sobre la JVM será un proceso en el kernel del SO.
- La JVM corre dentro del proceso e interpreta o compila el código Java en las clases.
- Una aplicación Java puede usar cualquier cantidad de recursos dentro de los límites de la JVM y usar el sistema operativo de la forma que desee.

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Diferencias con Java Virtual Machine

• Cada aplicación tiene su propio contenedor en lugar de tener múltiples WAR en un proceso compartido como es el servidor de aplicaciones.

 Cuando una aplicación tiene su propia imagen las cosas se vuelven mucho más sencillas: gestión de aplicaciones, descubrimiento de servicios

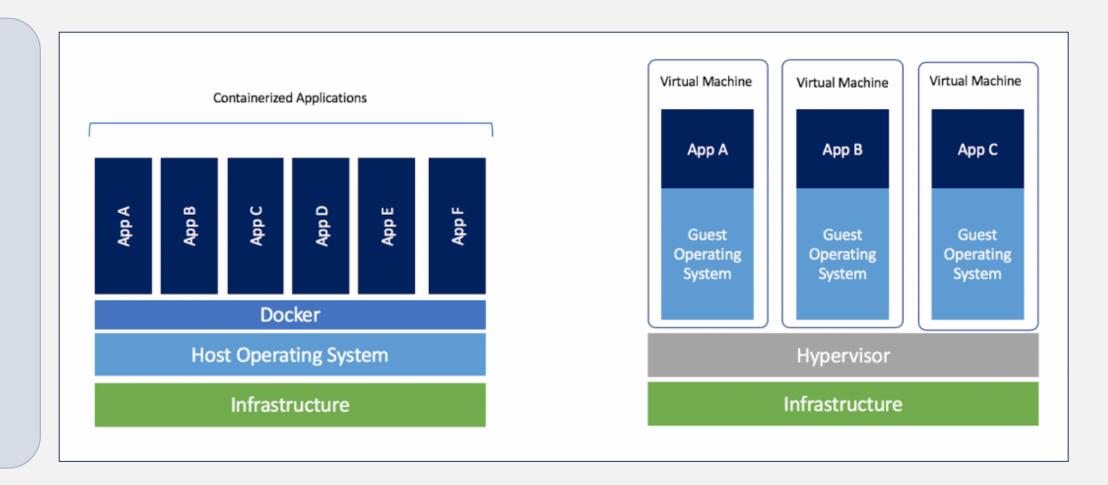
1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica



1. Introducción

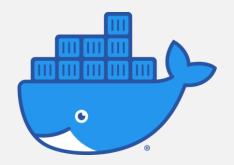
2. Contenedores

3. JVM

4. Docker

5. Práctica

¿Qué es Docker?



Docker es un proyecto de **código abierto** que **automatiza el despliegue** de aplicaciones dentro de **contenedores** de software, proporcionando una capa de **abstracción** y **automatización** de virtualización de aplicaciones es múltiples sistemas operativos.

1. Introducción

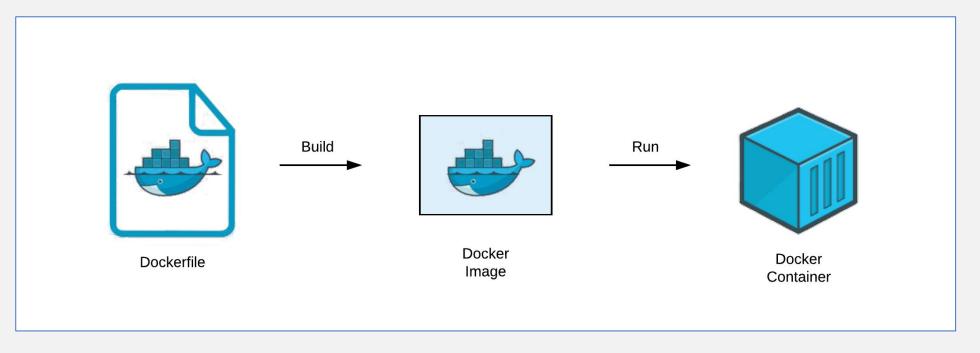
2. Contenedores

3. JVM

4. Docker

5. Práctica

Proceso de construcción de imágenes



1. Introducción

2. Contenedores

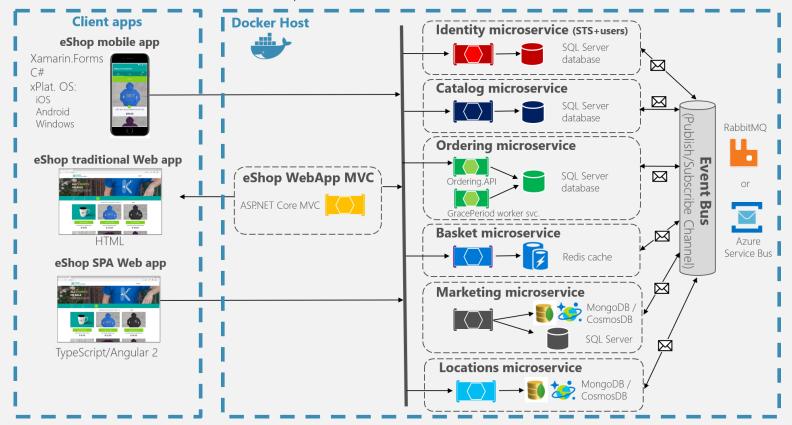
3. JVM

4. Docker

5. Práctica

eShopOnContainers reference application

(Development environment architecture)



1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Dockerfile



El Dockerfile son las instrucciones que Docker lee para construir una imagen automáticamente. Es un documento de texto que contiene todos los comandos que un usuario puede utilizar en la línea de comandos para ensamblar una imagen.

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Dockerfile

```
FROM node:14
WORKDIR /usr/src/app
# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./
RUN npm install
# If you are building your code for production
# RUN npm ci --only=production
COPY . .
EXPOSE 8080
CMD [ "node", "server.js" ]
```

1. Introducción

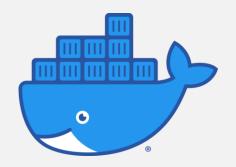
2. Contenedores

3. JVM

4. Docker

5. Práctica

Docker Compose



Compose es una herramienta para componer una aplicación en **múltiples contenedores** de Docker. Se utiliza un **fichero YAML** para **configurar los servicios** de la aplicación.

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Docker Compose

```
version: "3.9"
services:
    web:
    build: .
    ports:
        - "5000:5000"
    volumes:
        - .:/code
    environment:
        FLASK_ENV: development
    redis:
    image: "redis:alpine"
```

5. Práctica

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Persistir en una BBDD los datos obtenidos a través de Node-RED

- Conectar Node-RED a <u>una base de datos</u> en otro contenedor.
- Crear la mínima estructura para almacenar un histórico de los datos recibidos.
- Componer ambos servicios en un Docker Compose.

5. Práctica

1. Introducción

2. Contenedores

3. JVM

4. Docker

5. Práctica

Enlaces de interés

- Contenerizando con Docker una aplicación web en Node.js.
- Referencias de <u>Dockerfile</u>, <u>Docker Compose</u> y <u>Docker Hub</u>.
- Conexión entre contenedores.



Grado en Ingeniería Informática

Sistemas Industriales

Arquitectura orientada a contenedores

Pablo Casado

pcasado@dtic.ua.es

Departamento de Tecnología Informática y Computación

2022 - 2023