

Projecte 2



Departament d'Informàtica
i Comunicació

LA GUINEUETA
I n s t i t u t

Sergi Puerta y Pau Oliver

Tutors: Oriol, Víctor, Michael y Sergio

Projecte 2 de segon d'asix

INDEX:

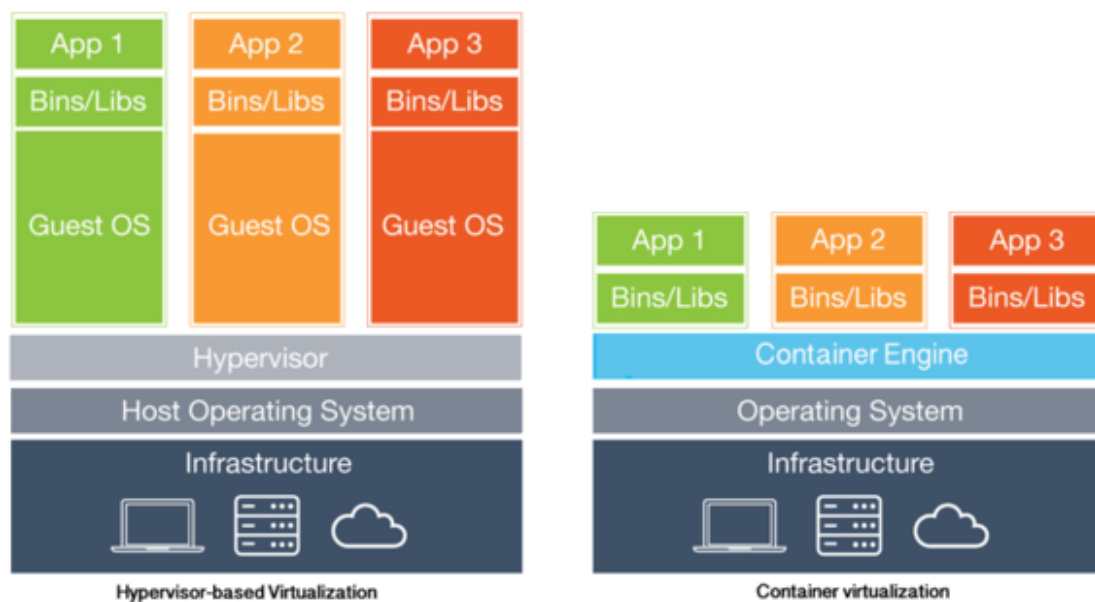
1.1	Introducción.....	Pàg: 2
1.2	Propuesta Técnica.....	Pàg: 6
1.3	Planificación del proyecto.....	Pàg: 6
	1.3.1 Temporal.....	Pàg 6
	1.3.2 Costes.....	Pàg 7
1.4	Desarrollo del proyecto.....	Pàg: 11
1.5	Pruebas del proyecto.....	Pàg: 14
1.6	Conclusión personal.....	Pàg: 19
1.7	Webgrafía.....	Pàg: 20

Que es Docker?

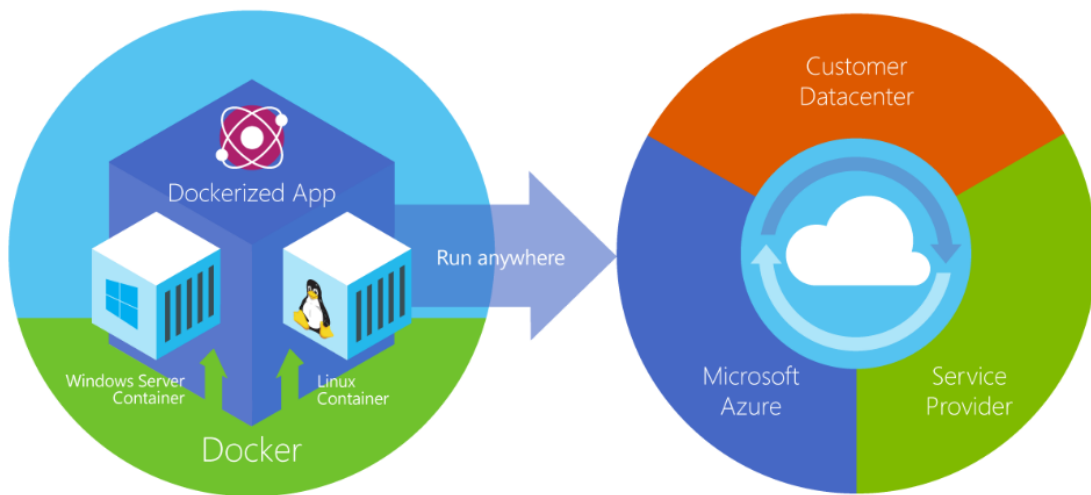
Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución, etc. Los contenedores son fáciles de utilizar gracias a su interfaz de líneas de comandos y permite a los usuarios poder utilizar y controlar varios contenedores a la misma vez gracias a la herramienta Docker Compose.

Funcionamiento de Docker?

Gracias a estos contenedores que hemos comentado anteriormente podemos desplegar la misma versión en distintos sistemas, siempre que tengan soporte de Docker, esto nos permite desarrollarlo por ejemplo en tu portátil Apple y ponerlo a funcionar en un servidor Cloud.



Como podemos observar en la imagen los contenedores adaptan las aplicaciones a nivel de despliegue y homogeneizan su ejecución en diferentes entornos, ya sea en la nube o en las instalaciones. Estas pequeñas piezas de software que componen el equipo de desarrollo son fácilmente ejecutables por el equipo de despliegue y consumen muchos menos recursos que otras opciones como podrían ser las máquinas virtuales.



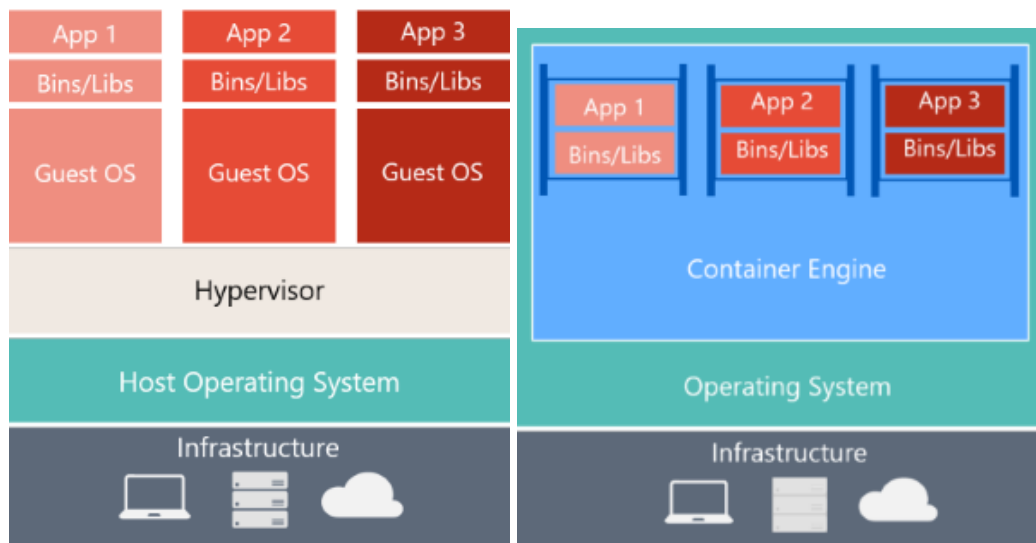
Cómo comenzó?

Docker apareció en escena en 2013 con una interfaz fácil de usar y la capacidad de empaquetar, aprovisionar y ejecutar tecnología de contenedores. Dado que Docker permitió que varias aplicaciones con diferentes requisitos de sistemas operativos se ejecutan en el mismo kernel de sistemas operativos en contenedores, los administradores informáticos y las organizaciones vieron la oportunidad de simplificar y ahorrar recursos. Un mes después de su primer lanzamiento de prueba, Docker fue probado por 10,000 desarrolladores. Cuando se lanzó Docker 1.0 en 2014, el software se había descargado 2,75 millones de veces y solo tendría que pasar un año para de eso para que llegara a la escandalosa cifra de 100 millones de descargas.

Principales Ventajas de Docker

A diferencia de las máquinas virtuales, los contenedores tienen una huella de recursos significativamente menor, requieren menos gastos generales para su administración. Las máquinas virtuales también deben encapsular un sistema operativo completamente independiente y otros recursos, mientras que los contenedores comparten el mismo núcleo del sistema operativo y usan un sistema proxy para conectarse a los recursos que necesitan, dependiendo de dónde se encuentren estos recursos. Como podemos ver en la siguiente imagen las máquinas virtuales incluyen la aplicación, las bibliotecas o los archivos binarios necesarios y un sistema operativo invitado completo. La virtualización completa requiere más recursos que la inclusión en contenedores. En cambio los contenedores incluyen la aplicación y todas sus dependencias. Sin embargo, comparten el kernel del sistema operativo con otros contenedores, que se ejecutan como procesos aislados en el espacio de usuario en el sistema operativo host. Todo esto se traduce en una reducción considerable de costos y un retorno de la inversión mucho más rápido al reducir los recursos materiales necesarios.

Maquina virtual vs Docker



¿Por qué me interesa este proyecto?

Este proyecto me ha llamado la atención porque gracias a ello he podido descubrir que es Docker, cómo funciona y sobre todo lo más importante sus salidas profesionales. Durante estos 2 últimos meses he podido emprender un nuevo reto, la verdad es que al principio me costó bastante entender esta herramienta que nos permite a nosotros como desarrolladores o administradores poder hacer simulaciones en entornos de prueba, y sin la necesidad de tener instalada algún tipo de dependencia. Pero al pasar del tiempo fui comprendiendo cómo funciona esta nueva tecnología, y sus virtudes. La siguiente parte que veremos a continuación ya es la parte práctica del proyecto, en la cual enseñó todos los pasos que he llevado a cabo además de todos los servicios y programas utilizados como apache2, python, mySQL y el SSH.

1.3 Planificación del proyecto

1.3.1 Temporal

Desarrollo del proyecto	
Propuesta técnica	
Definición de objetivos y alcance del sistema	2d

Definición de requisitos	2d
Estimación y planificación	3d
Estimación de costes	1d
Diagrama de contexto	1d
Análisis del sistema	
Modelado de datos	4d
Modelo de procesos	5d
Diseño	
Revisión del análisis	13d
Diseño de BD y ficheros	3d
Diseño de módulos	4d
Diseños de interfaces de usuario	4d
Diseño de procesos	1d
Construcción y puesta en marcha	
Desarrollo del plan de instalación	4d
Preparación del entorno de desarrollo y pruebas	2d
Desarrollo de los componentes de software	
Diseño detallado	5d
Programación	19d
Diseño y realización de pruebas	3d
Memoria final	
Puesta en marcha	3d
Redacción del proyecto	4d
Entrega del producto	1d
Fin del proyecto	

1.3.2 Costes

Recursos	Unidades	Distribuidor
Personal		
Analistas	2	
Programadores	2	
Especialista en estudios de mercado	1	
Administrador de sistemas	1	
Material		
Ordenadores personales	8	Dell
Software edición de video	1	Kodac
Software Web	1	Adobe
Software Ofimática	8	Libre distribución
Máquina fotográfica	2	Microsoft

RECURSOS HUMANOS

RECURSO	CANTIDAD
Especialista en fotografía	2
Especialista en diseño	1
Analistas	1
Programadores	3
Especialista en estudios de mercado	1
Administrador de sistemas	2

RECURSOS MATERIALES	
RECURSO	CANTIDAD
Servidor	2
Pantalla TFT	2
Software	8
Dominio Web	1
Conexión banda ancha	2
Ordenadores personales	8
Máquina fotográfica	2
Software edición de video	2
Software Web	1
Software ofimática	8

RECURSOS HUMANOS

RECURSO	COSTE/HORA
Especialista en fotografía	15 €
Especialista en diseño	20 €
Analistas	20 €
Programadores	30 €
Especialista en estudios de mercado	25 €
Administrador de sistemas	35€

RECURSOS MATERIALES	
RECURSO	PRECIOS
Servidor	3500 €
Pantalla TFT	120 €
Software	70 €
Dominio Web	50 €
Conexión banda ancha	45 €
Ordenadores personales	Propiedad de la empresa
Máquina fotográfica	Propiedad de la empresa
Software edición de video	Propiedad de la empresa
Software Web	Propiedad de la empresa
Software ofimática	Propiedad de la empresa

```
version: '3'
networks:
  app-network:
    driver: bridge
services:
  app:
    build: .
    ports:
      - "3000:3000"
  db:
    image: db
    restart: always
    volumes:
      - ./data:/var/lib/mysql
    ports:
      - 3306:3306
    networks:
      - app-network

phpmyadmin:
  image: phpmyadmin/phpmyadmin
  environment:
    PMA_ARBITRARY: 1
    PMA_HOST: db
    PMA_PORT: 3306
    PMA_USER: root
    PMA_PASSWORD: root
  ports:
    - 8091:80
  links:
    - db:db
  networks:
    - app-network

dev:
  image: dev-docfile
  ports:
    - "8081:8080"
    - "2222:2222"
  volumes:
    - ${HOME}:/home/dev
    - /var/run/docker.sock:/var/run/docker.sock
    - /var/lib/docker:/var/lib/docker
  environment:
    - DOCKER_HOST=unix:///var/run/docker.sock
  networks:
    - app-network
```

Dockerfile mysql

```

FROM ubuntu:22.04

RUN apt update \
    && apt install -y ca-certificates openssh-client \
    wget curl iptables supervisor \
    && rm -rf /var/lib/apt/lists/*
RUN apt install -y mysql-server
RUN apt install -y curl
ENV DOCKER_CHANNEL=stable \
    DOCKER_VERSION=20.10.23 \
    DOCKER_COMPOSE_VERSION=3.3 \
    DEBUG=false

RUN touch /var/lib/mysql/schema.sql
RUN service mysql start && \
    mysql -u root -e "CREATE DATABASE dev;" && \
    mysql -u root -e "CREATE USER 'dev'@'%' IDENTIFIED BY 'dev';" && \
    mysql -u root -e "GRANT ALL PRIVILEGES ON *.* TO 'dev'@'%';" && \
    mysql -u root dev < /var/lib/mysql/schema.sql

COPY my.cnf /etc/mysql/my.cnf

# Docker installation
RUN set -eux; \
    \
    arch="$(uname --m)"; \
    case "$arch" in \
    # amd64
    x86_64) dockerArch='x86_64' ;; \
    # arm32v6
    armhf) dockerArch='armel' ;; \
    # arm32v7
    armv7) dockerArch='armhf' ;; \
    # arm64v8
    aarch64) dockerArch='aarch64' ;; \
    *) echo >&2 "error: unsupported architecture ($arch)"; exit 1 ;; \
    esac; \
    if ! wget -O docker.tgz "https://download.docker.com/linux/static/${DOCKER_CHANNEL}/${dockerArch}/docker-${DOCKER_VERSION}.tgz"; then \
    echo >&2 "error: failed to download 'docker-${DOCKER_VERSION}' from '${DOCKER_CHANNEL}' for '${dockerArch}'"; \
    exit 1; \
    fi; \
    \
    tar --extract \
    --file docker.tgz \
    --strip-components 1 \
    --directory /usr/local/bin/ \
    ; \
    \
    rm docker.tgz; \
    \
    dockerd --version; \
    docker --version

COPY modprobe startup.sh /usr/local/bin/
COPY supervisor/ /etc/supervisor/conf.d/
COPY logger.sh /opt/bash-utils/logger.sh

RUN chmod +x /usr/local/bin/startup.sh /usr/local/bin/modprobe

VOLUME /var/lib/docker

# Docker compose installation
RUN curl -L "https://github.com/docker/compose/releases/download/v2.16.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose \
    && chmod +x /usr/local/bin/docker-compose && docker-compose version

RUN update-alternatives --set iptables /usr/sbin/iptables-legacy && \
    update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy

EXPOSE 3306 3000 8091 2222
ENTRYPOINT ["startup.sh"]
CMD ["bash"]

```

Dockerfile dev

```


GNU nano 6.2 Dockerfile.dev
FROM ubuntu:22.04
RUN apt-get update && \
    apt-get install -y curl git openssh-server software-properties-common supervisor
RUN curl -fsSL https://get.docker.com -o get-docker.sh && \
    sh get-docker.sh
RUN apt-get install unzip
RUN apt-get install zip
RUN curl -s "https://get.sdkman.io" | bash
ENV PATH="/root/.sdkman/candidates/docker/current/bin:/root/.sdkman/candidates/maven/current/bin:/root/.sdkman/candidates/gradle/current/bin:/root/.sdkman/candidates/python/current/bin:$PATH"
RUN bash -c 'source /root/.sdkman/bin/sdkman-init.sh && \
    sdk install java 8 && \
    sdk install maven && \
    sdk install gradle'
RUN apt-get install -y nodejs
RUN apt-get install -y python3 python3-pip
RUN apt-get install -y mysql-client
RUN curl -fsSL https://code-server.dev/install.sh | sh
RUN apt-get install -y python3-pip
RUN curl -fsSL https://cli.github.com/packages/githubcli-archive-keyring.gpg | gpg --dearmor -o /usr/share/keyrings/githubcli-archive-keyring.gpg
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/githubcli-archive-keyring.gpg] https://cli.github.com/packages $(lsb_release -cs) main" | tee /etc/apt/sources.list.d/github-cli.list > /dev/null
RUN apt-get install -y gh
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf
EXPOSE 2222 8081
CMD ["/usr/bin/supervisord"]


```


Conexion al phpmyadmin pero no al mysql

Bienvenido a phpMyAdmin

Error

MySQL ha dicho: 
No se estableció la conexión: los parámetros están incorrectos.

 mysql:real_connect(): (HY000/2002) Connection refused

 phpMyAdmin intentó conectarse con el servidor MySQL, y el servidor rechazó esta conexión. Deberá revisar el host, nombre de usuario y contraseña en config.inc.php y asegurarse que corresponden con la información provista por el administrador del servidor MySQL.

Reintentar conexión