

# Serveis de contenidors: Docker

## Objectius

- Aprendre a desplegar contenidors i interactuar amb ells
- Customitzar imatges a partir de contenidors existents
- Automatitzar la configuració de contenidors pre-desplegament
- Desplegar múltiples contenidors a la vegada
- Utilitzar serveis prèviament vistos a classe en contenidor

## Què és Docker?

És una plataforma per desenvolupar, desplegar i executar aplicacions. Permet empaquetar i executar una aplicació en un entorn aïllat anomenat contenidor. Els contenidors són 'lightweight' i contenen tot lo necessari per executar l'aplicació. El aïllament permet executar varis contenidors en un mateix host, o bé compartir el teu contenidor amb altres persones.

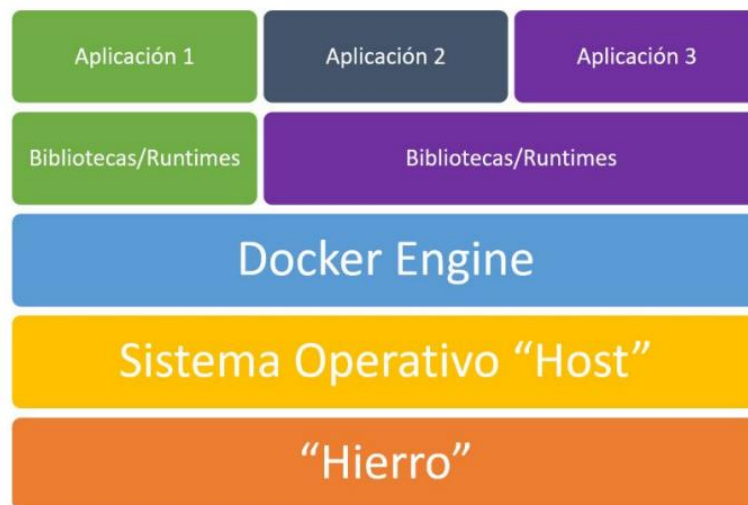
## Contenidor vs Virtualització

Com el seu nom indica, una màquina virtual és un sistema operatiu complet funcionant de forma aïllada en un altre sistema operatiu complet. El *hipervisor* exposa el hardware del host de forma que pugui ser utilitzat per altres sistemes operatius.



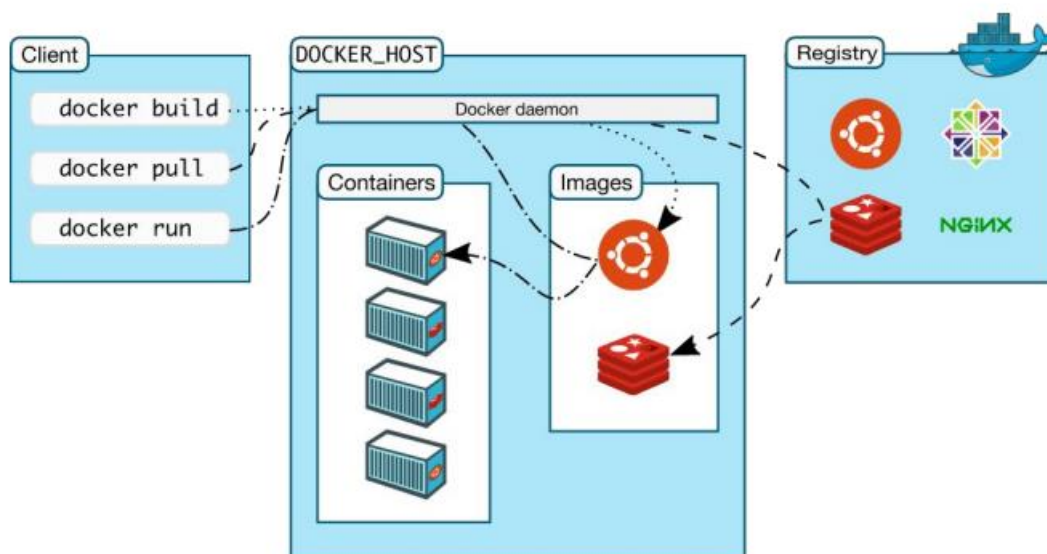
Docker Engine s'encarrega de llançar i gestionar els contenidors amb les nostres aplicacions, però en lloc d'assignar el hardware de forma discreta el comparteix tot entre tots

els contenidors i en funció del moment assigna el que sigui necessari perquè els contenidors funcionin adequadament.



## Arquitectura Docker

Docker utilitza una arquitectura client-servidor. El *daemon* escolta peticions API i gestiona els objectes Docker. El client és la principal forma d'interactuar amb el daemon. Un registre (o repositori) guarda imatges de Docker. Docker hub és un repositori públic i el default.



## Arquitectura Docker: Objectes

**Imatges:** és una plantilla 'read-only' amb instruccions per crear un contenidor Docker. Habitualment una imatge se basa en un altre, amb més customitzacions.

**Contenidor:** és una instància en execució d'una imatge. Pots connectar un contenidor a una o més xarxes, afegir espai d'emmagatzematge o crear una imatge nova a partir d'aquest contenidor. Un contenidor és definit per una imatge i opcions de configuració que donam quan l'executem.

## Exemples

---

Aquesta comanda executa un contenidor ubuntu, s'implementa a la nostra consola i executa `/bin/bash`

```
$ docker run -i -t ubuntu /bin/bash
```

1. Si no tens la imatge, la baixa del *repositori* per defecte
2. Crea un contenidor nou, com si haguessis fet `docker container create`
3. Docker prepara un sistema de fitxers pel contenidor
4. Prepara una *interfície* per connectar el contenidor a la xarxa per defecte. Això inclou assignar una IP al contenidor.
5. Docker encén el contenidor i executa `/bin/bash`. Al utilitzar `-i` i `-t` fa que poguem interactuar amb el contenidor des de la nostra terminal.
6. Al sortir amb èxit el contenidor s'atura però no s'elimina. Se pot tornar a executar o ser eliminat (`docker rm`).

## Comandes bàsiques

---

```
docker (container) run [IMATGE] [COMANDA]
```

Executa un nou contenidor de la imatge proporcionada. Si la imatge no es troba al repositori local, l'agafa del extern. Es pot afegir una comanda per ser executada al obrir-se el contenidor.

```
docker (container) run -it [IMATGE]
```

Executa un nou contenidor en mode interactiu. Per sortir `ctrl+D`

```
docker (container) run -d [IMATGE]
```

Executa un nou contenidor en background.

```
docker (container) run -v [PATH_HOST]:[PATH_CONTAINER] [IMATGE]
```

Executa un nou contenidor amb un volum vinculat.

```
docker (container) start [IDCONTENIDOR]
```

Torna a arrancar un contenidor aturat

```
docker exec [IDCONTENIDOR] [COMANDA]
```

Executa la comanda a un contenidor en execució

```
docker ps (-a)
```

Mostra els contenidors desplegats/existents

```
docker image pull [IMATGE]
```

Baixa la imatge seleccionada al nostre repositori local.

```
docker image ls
```

Mostra totes les imatges que tenim al repositori local

```
docker cp [PATH] [PATH]
```

Copia arxius de host a contenidor o viceversa

```
docker container diff [IDCONTENIDOR]
```

Comprova els canvis que ha hagut en el sistema de fitxers del contenidor desde la seva creació

```
docker container commit [IDCONTENIDOR] [NOM:VERSIO]
```

Crea una imatge a partir d'un contenidor

```
docker image tag [IDIMATGE] [NOM]
```

Canvia el nom a una imatge

## Dockerfile

---

Docker pot crear imatges de forma automàtica a partir d'unes instruccions donades per un arxiu anomenat Dockerfile.

Per exemple:

```
FROM python
RUN mkdir /app
COPY ./hello.py /app
CMD python /app/hello.py
```

FROM crea una capa inicial a partir de la imatge de python (linux)

RUN executa la comanda especificada

COPY fa una còpia del fitxer del client docker dins la carpeta /app de la imatge

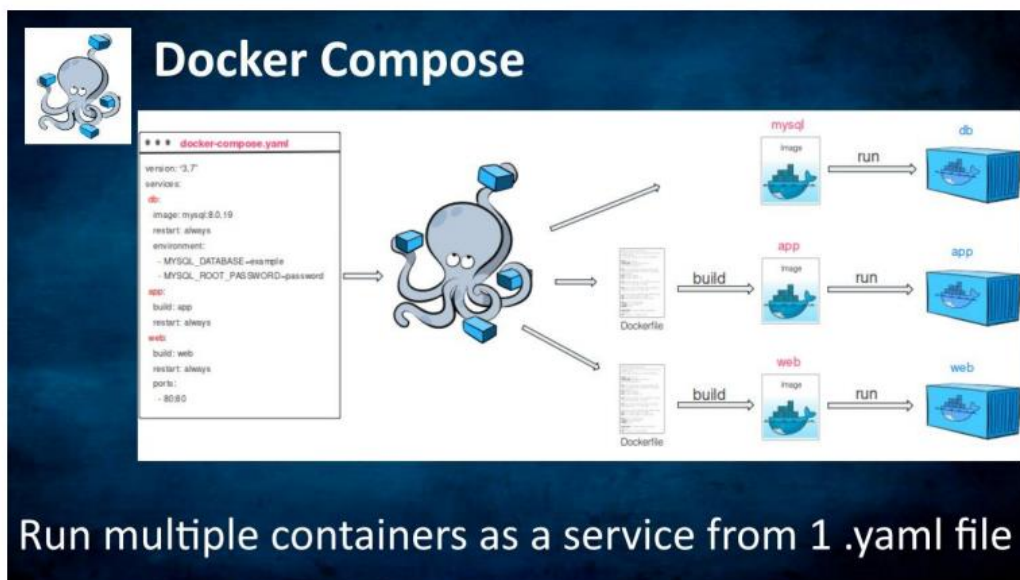
CMD farà que cada vegada que arranqui el contenidor s'executi la nostra aplicació

Per compilar la imatge:

```
docker build -t [NOM:VERSIO] [CARPETA AMB DOCKERFILE]
```

## Docker compose

Compose és una eina per definir i executar dos o més contenidors a la vegada. S'utilitza un fitxer YAML (yet another mark-up language) per configurar els serveis. Així, amb una sola comanda, pots crear i executar tots els serveis de la teva configuració.



Dins el fitxer especifiquem els serveis que volem desplegar i a cada un definim les seves propietats: la seva imatge, volums a utilitzar, els ports, etc. Per arrancar els serveis feim un `docker compose up` en la carpeta que té el `docker-compose.yml`

```
version: "3.9"
services:
  apache:
    image: ubuntu/apache2
    volumes:
      - .\html:/var/www/html
    ports:
      - 80:80
  tomcat:
    image: tomcat
    ports:
      - 8080:8080
```

## Instal·lar Docker damunt Ubuntu

---

Per instal·lar docker damunt un servidor LINUX, ja sigui mitjançant un servei remot o dintre de la nostra màquina si té un S.O Linux, hem de seguir les següents passes:

```
# Eliminamos versiones antiguas
$ sudo apt-get remove docker docker-engine docker.io containerd runc
# Generamos repositorio valido
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg agent software-properties-common
# Añadimos repo docker official
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
# Validamos repo con huella 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
sudo apt-key fingerprint 0EBFCD88
# Añadimos repo docker
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \
stable"
# Instalamos docker
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
# Añadimos permisos sudo al programa docker
$ sudo usermod -aG docker $USER
```

La guia de [Digital Ocean](#) es molt recomanable també. La instal·lació està més explicada i detallada.