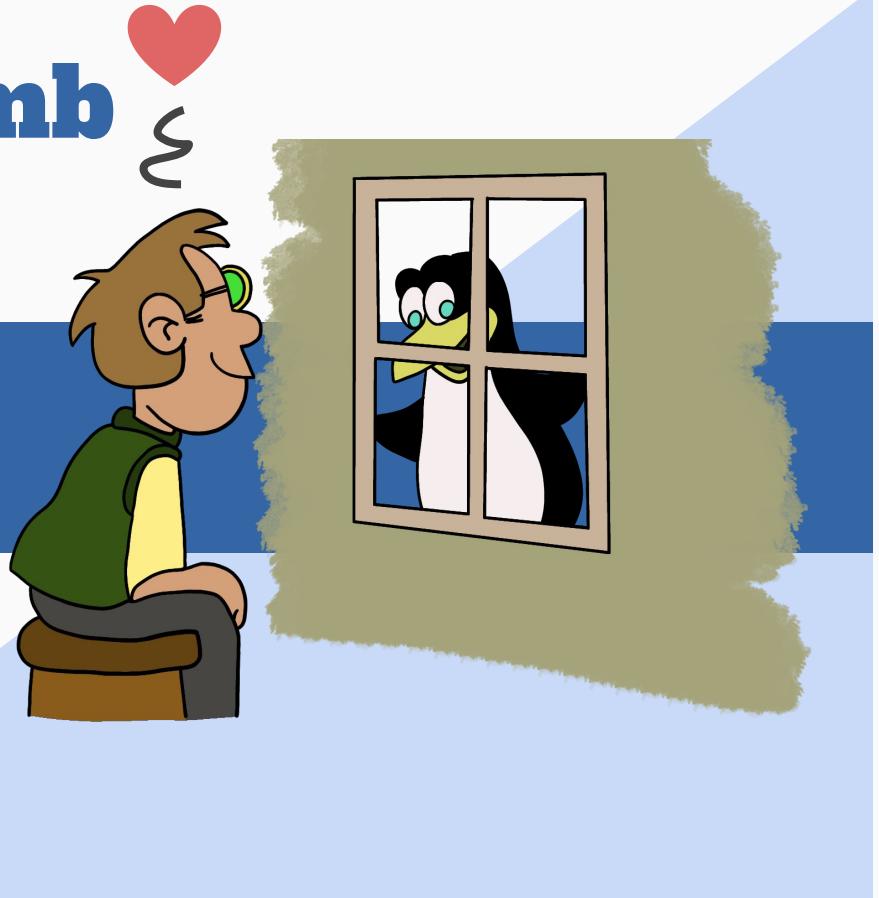


# Desenvolupar amb Docker

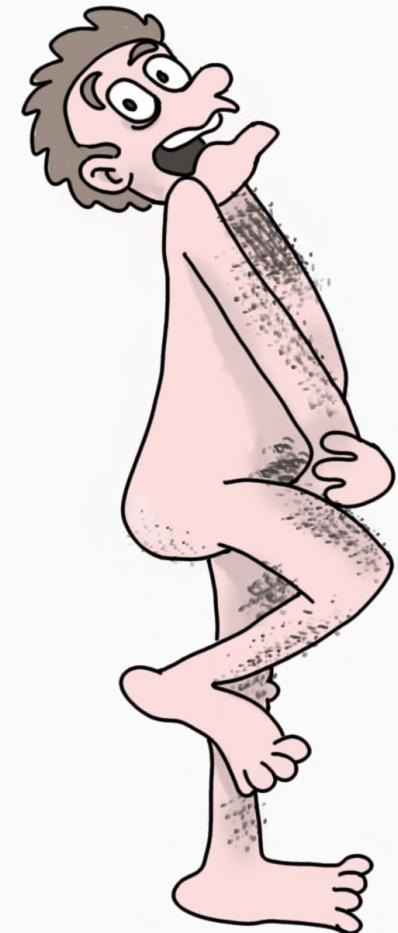
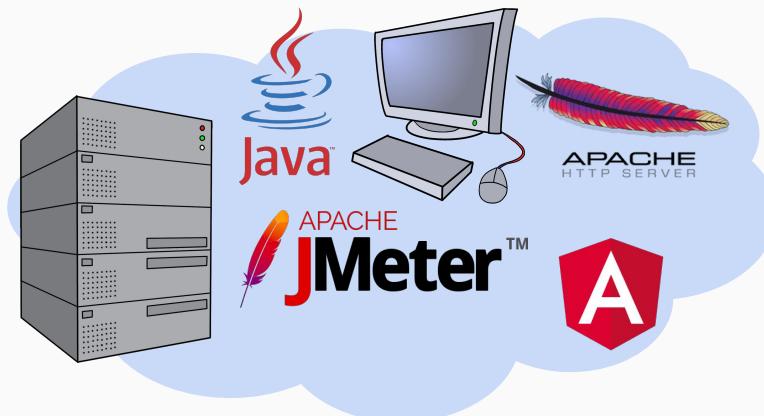
Linux ha enamorat  
Microsoft ☺



# Qui sóc?

Xavier Sala

Professor de Cicles d'Informàtica  
(Institut Cendrassos)





**Contenidors**

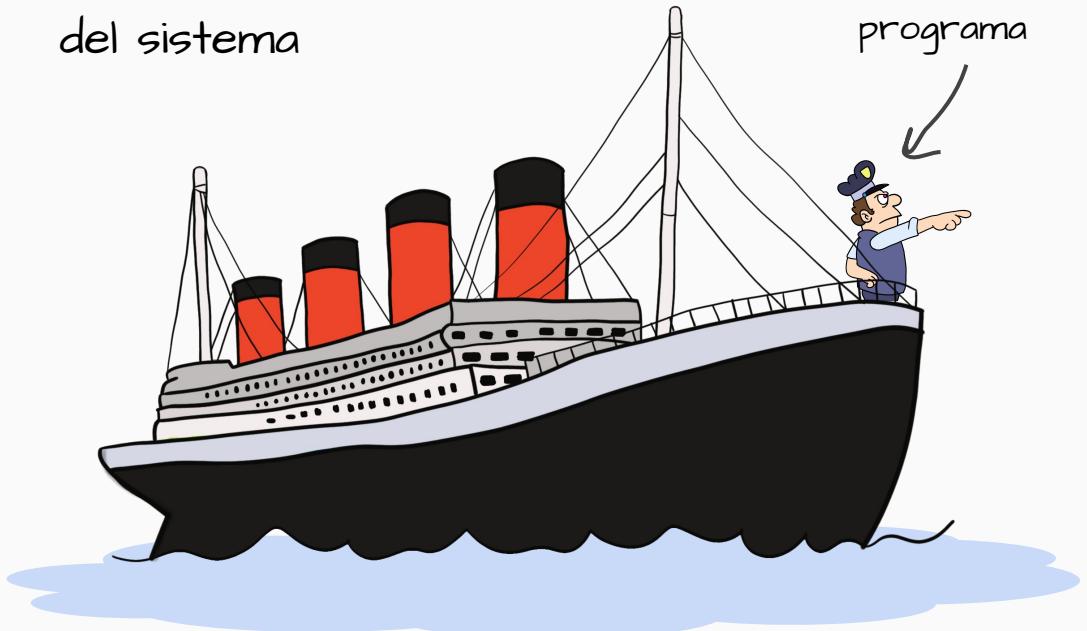
**Ús de Docker**

**Desenvolupar amb Docker**

**Desplegar amb Docker**

**Aplicacions complexes**

Les màquines virtuals permeten  
executar programes de  
**forma aïllada**  
del sistema



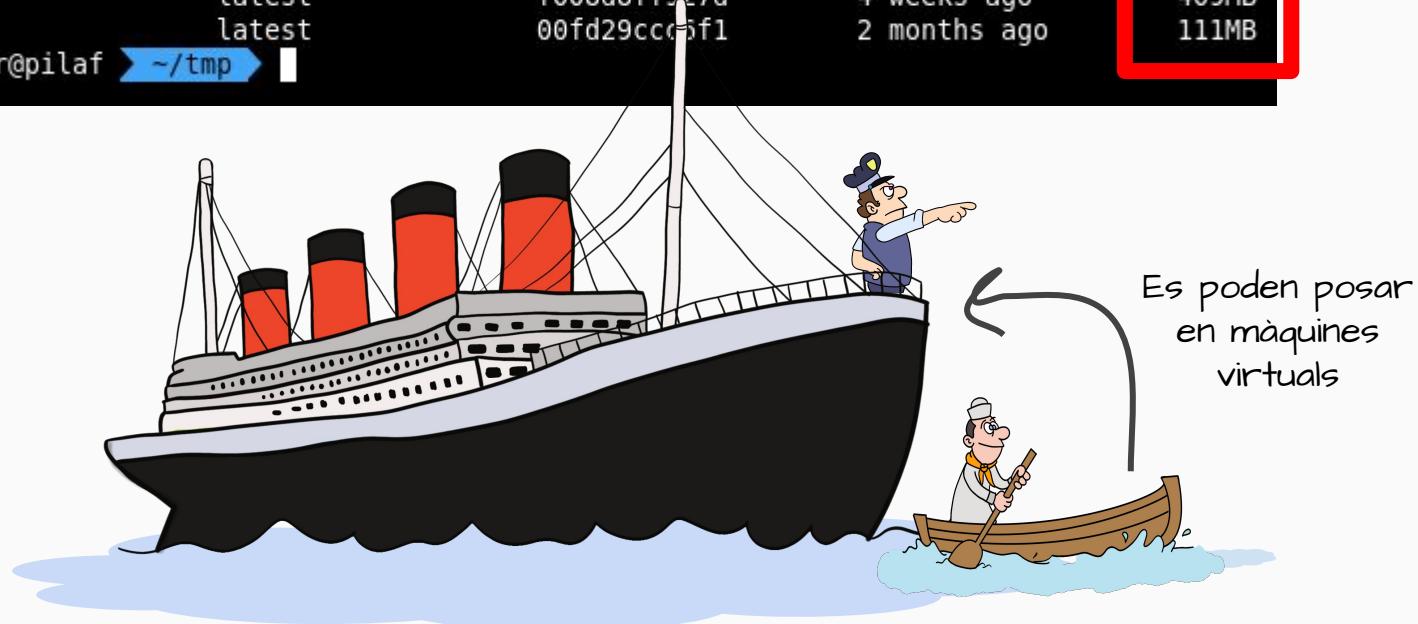


Els **contenidors** permeten al sistema operatiu **executar aplicacions** aïllant-les de les altres

Els contenidors són molt més petits  
que les màquines virtuals

```
xavier@pilaf ~/tmp ➤ docker image list
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest   f008d8ff927d  4 weeks ago  409MB
ubuntu          latest   00fd29cc6f1   2 months ago 111MB
xavier@pilaf ~/tmp ➤
```

SIZE
409MB
111MB



# Sistemes de contenidors

Cada aplicació té  
la seva pròpia  
versió del  
sistema



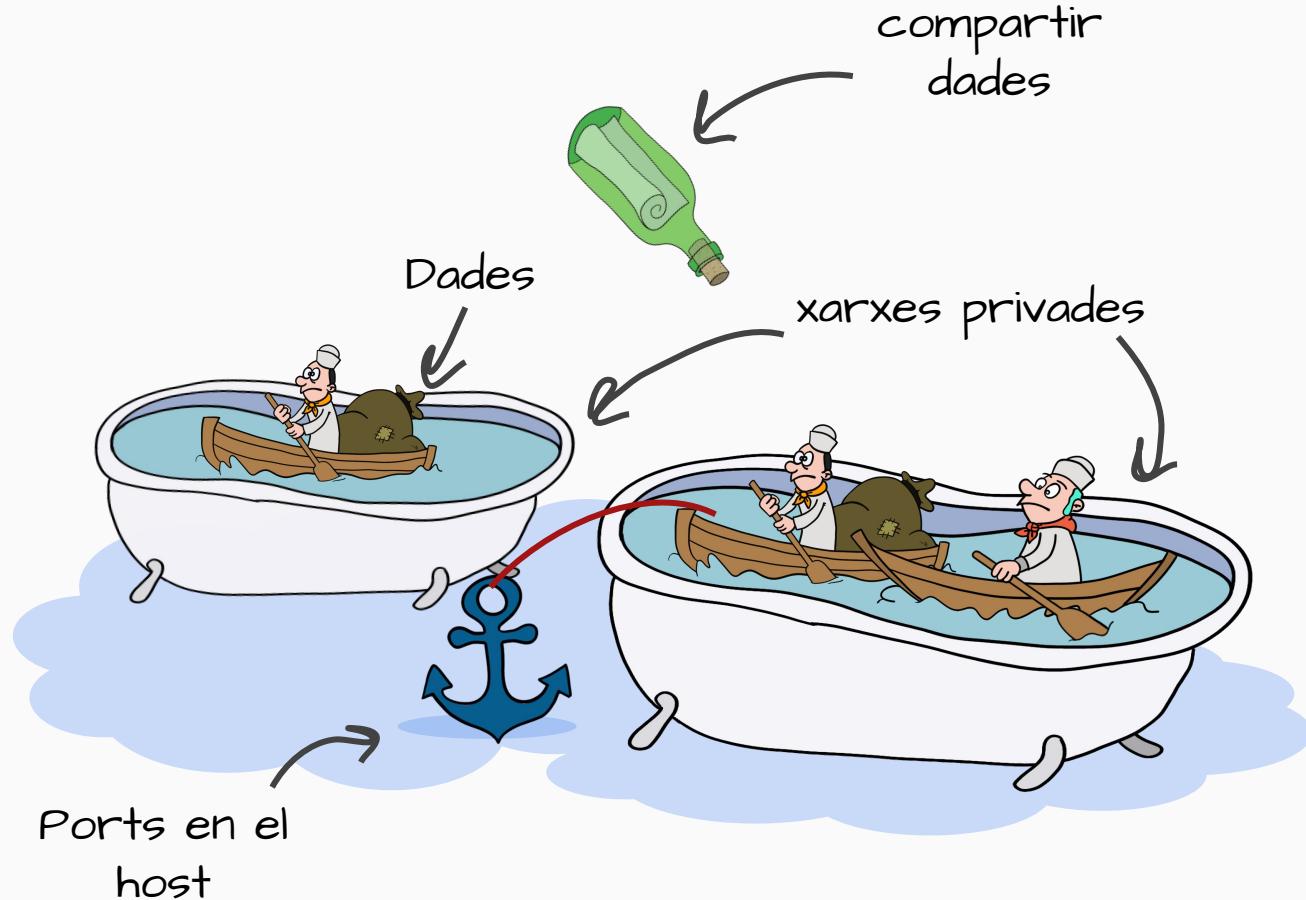
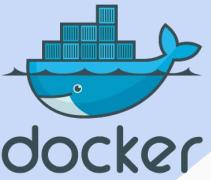
Tenen el mínim  
necessari per  
funcionar

Se'ls hi pot  
limitar l'ús de  
recursos

Fan servir el  
nucli del sistema  
operatiu  
**on corren**

Quan el **programa acaba** el  
contenidor **s'atura!**

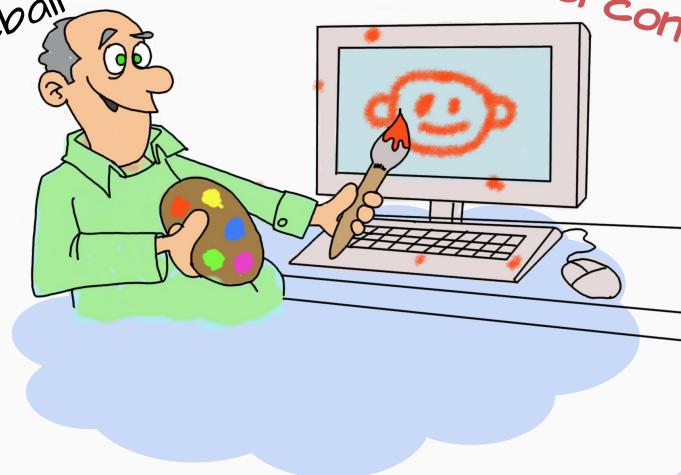




# Faciliten el desenvolupament

És fàcil duplicar l'entorn de treball

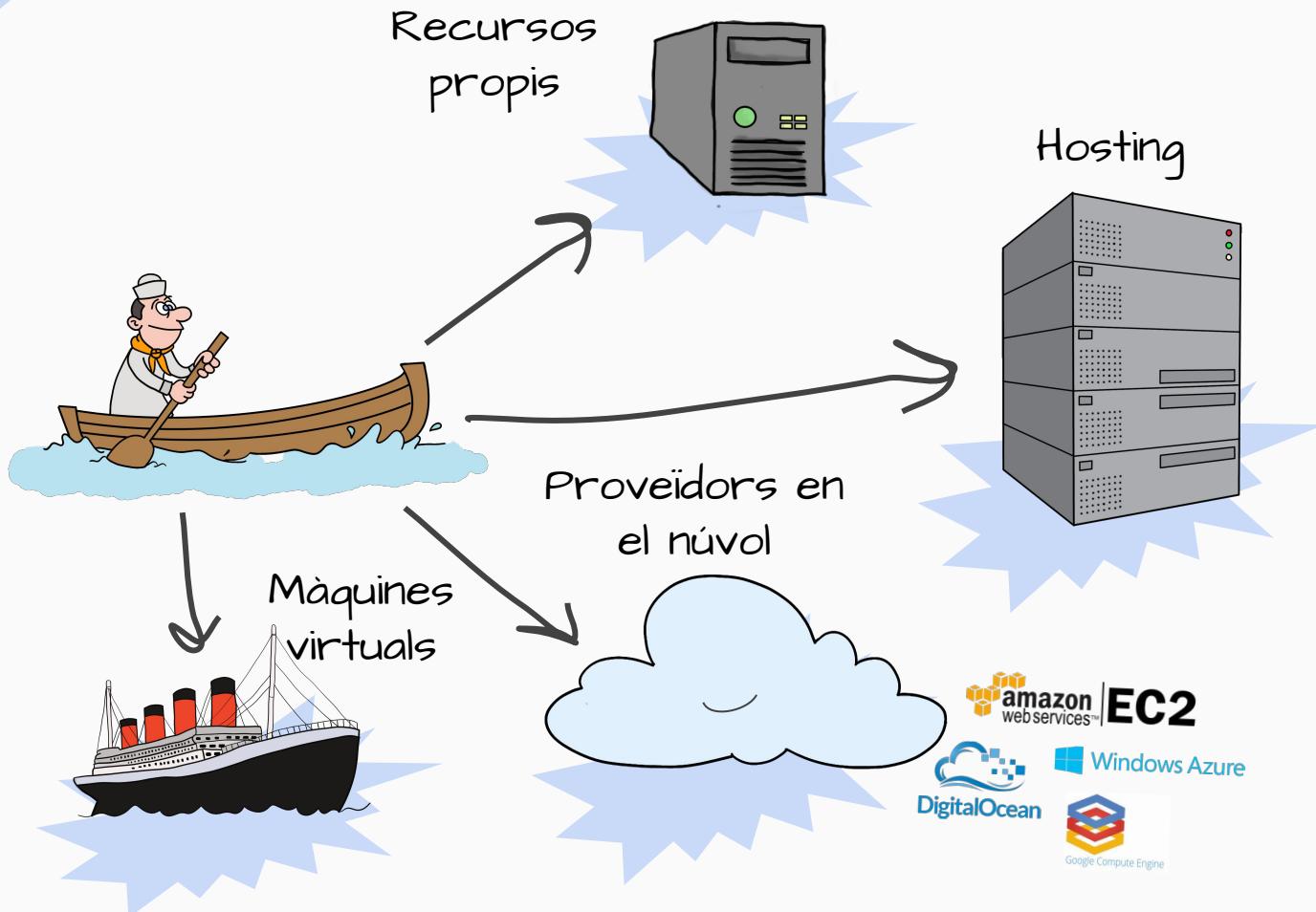
Es pot desenvolupar **alhora** amb servidors incompatibles

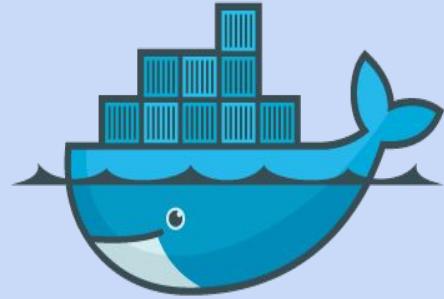


Es desenvolupa en el sistema de **Producció!**

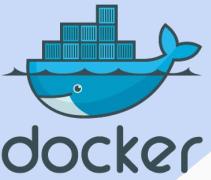
Es pot desplegar **el contingidor**

# Faciliten el desplegament

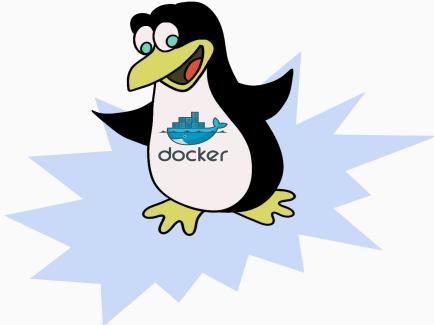




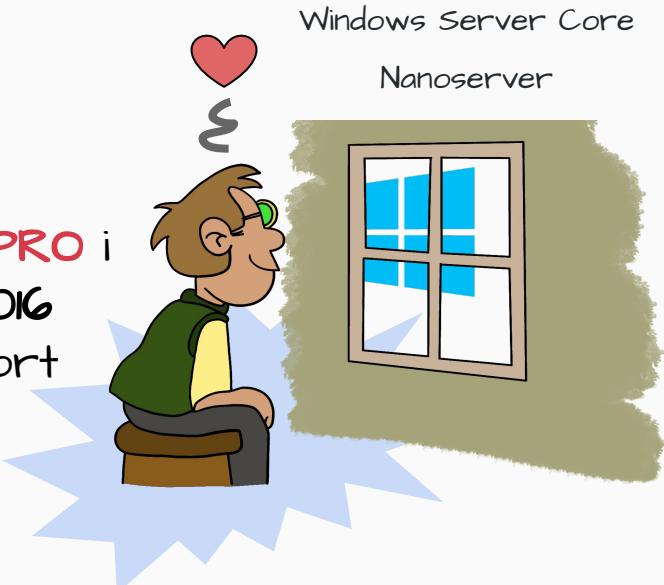
docker

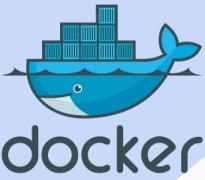


Va néixer en  
Linux

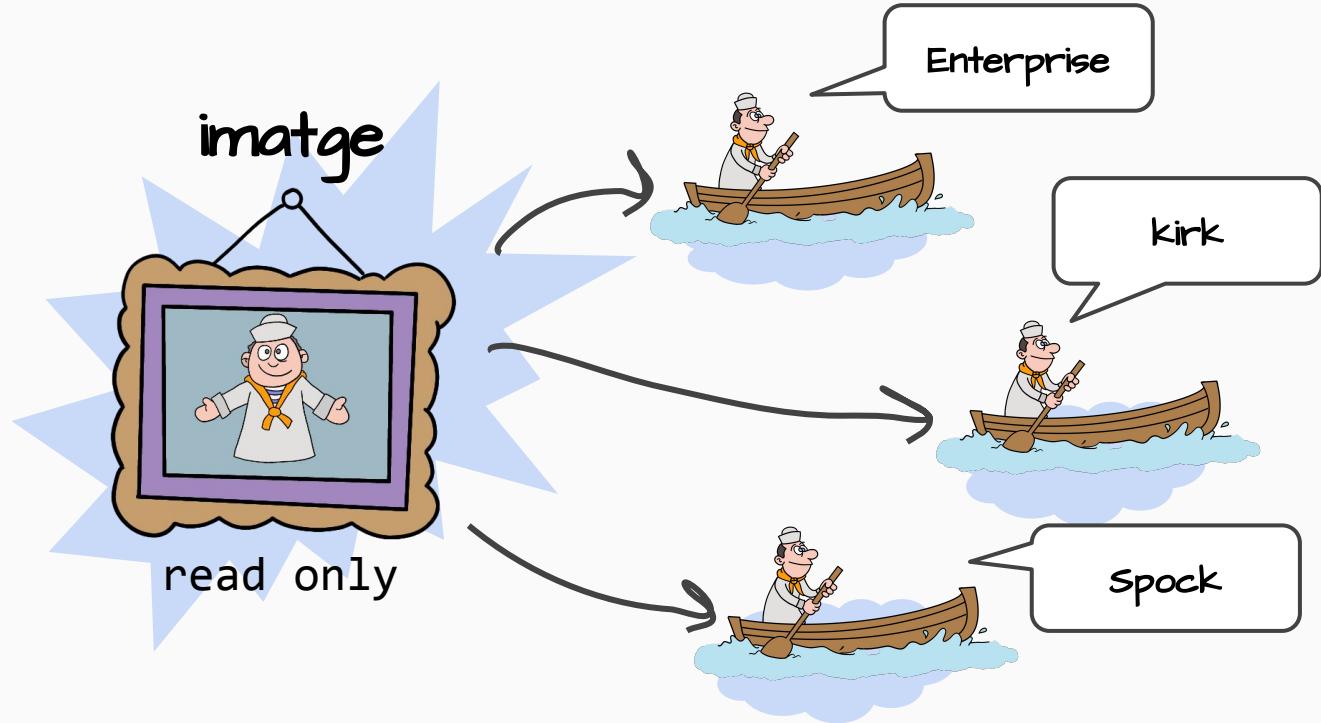


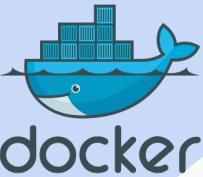
Windows 10 PRO i  
Server 2016  
tenen suport  
nadiu





Els contenidors es creen a partir de les **imatges**

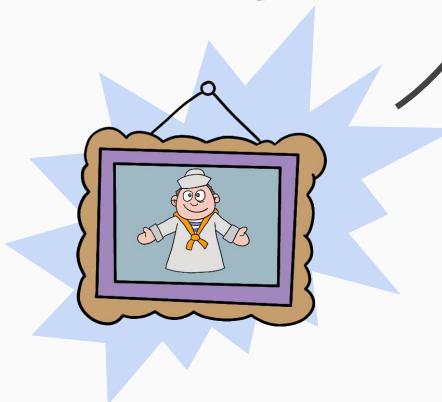




D'on surten  
les imatges?



Es poden **crear**



**obtenir-les** de  
repositoris online

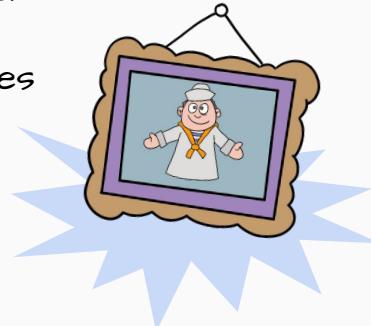


**Modificar** una  
imatge d'existent

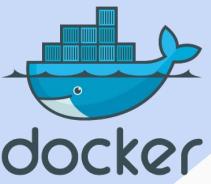
# Repositori oficial



Solen ser la  
base per  
crear imatges  
pròpies



<https://hub.docker.com/explore/>



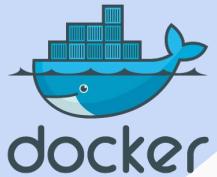
# Executar imatges

Terminal interactiu → Programa a executar

```
$ docker run -ti ubuntu bash
```

→ Nom del contenidor S'executa com a daemon

```
$ docker run --name pirata -d nginx
```



```
$ docker ps
```

```
$ docker images
```

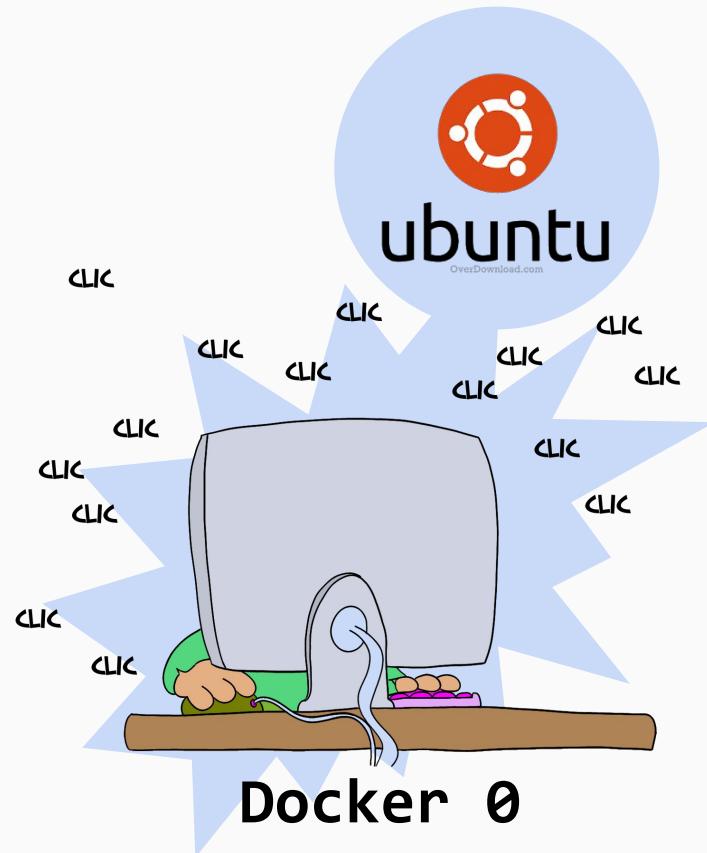
```
$ docker rm contenidor
```

```
$ docker rmi imatge
```

```
$ docker exec contenidor bash
```

```
$ docker inspect contenidor
```

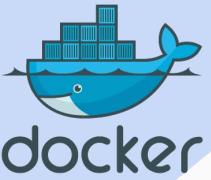
```
$ docker logs contenidor
```



Docker 0

# Desenvolupar amb Docker

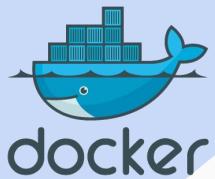




Es poden executar serveis  
**SENSE INSTAL.LAR-LOS** en  
el host

Es pot **DESENVOLUPAR**  
una aplicació web **SENSE**  
instal.lar cap servidor web!





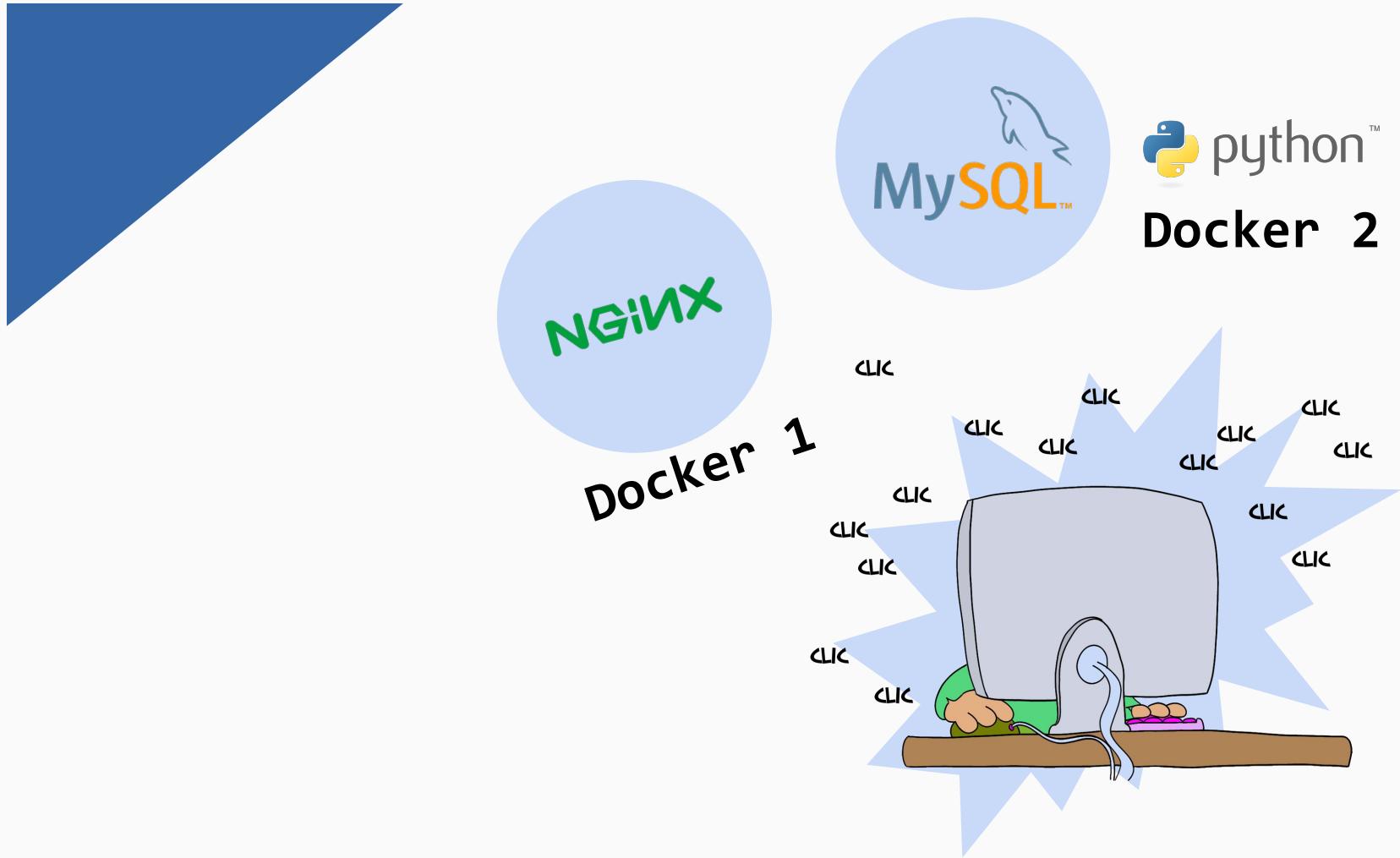
```
$ docker run --name server -d nginx  
-v /home/xx/web:/usr/share/nginx/html:ro  
-p 8080:80
```

Volum  
compartit en  
local

Exposa el port en local  
host : contenidor

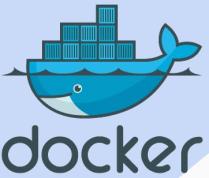
```
$ docker run --name mymysql -d mysql  
-v /home/xx/bdd:/var/lib/mysql  
-e MYSQL_ROOT_PASSWORD=ies2010  
-p 3306:3306
```

Variables  
d'entorn



# Desplegar i empaquetar aplicacions en Docker





imatge original

Afegir-hi  
l'aplicació

comanda a  
executar en el  
contenidor

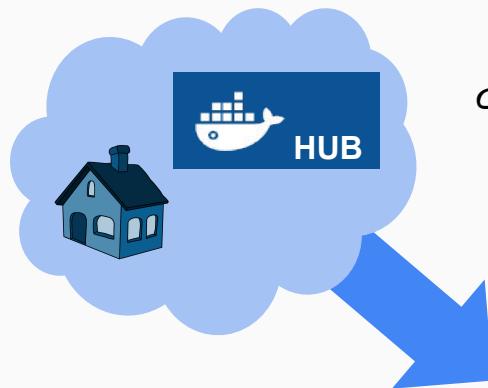
## Dockerfile

```
FROM node:7
```

```
ADD app.js /app.js
```

```
CMD node app.js
```

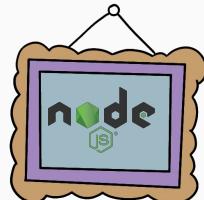
```
$ docker build -t Mariner .
```



FROM node:8

S'obté la imatge original de node.js

node 8



ADD app.js /app.js

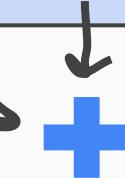
Aplicació node.js

app.js

```
const http = require('http');
const os = require('os');

var handler = function (request, response) {
  response.writeHead(200);
  response.end("Sóc " + os.hostname() + "!\n");
};

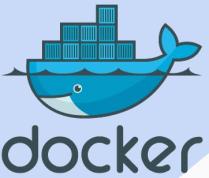
var www = http.createServer(handler);
www.listen(8080);
```



Mariner



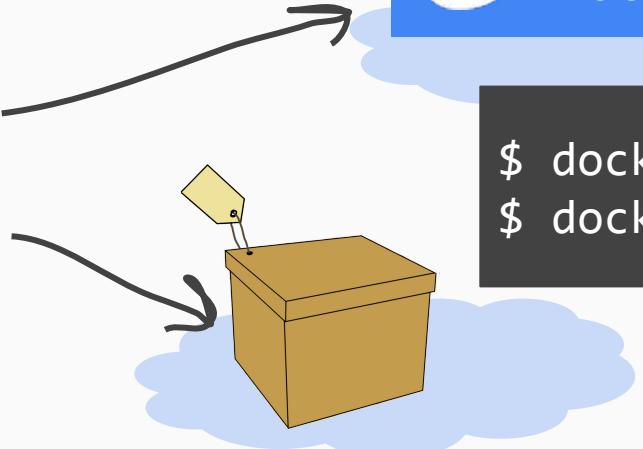
Es guarda la imatge localment



## Distribuir la imatge



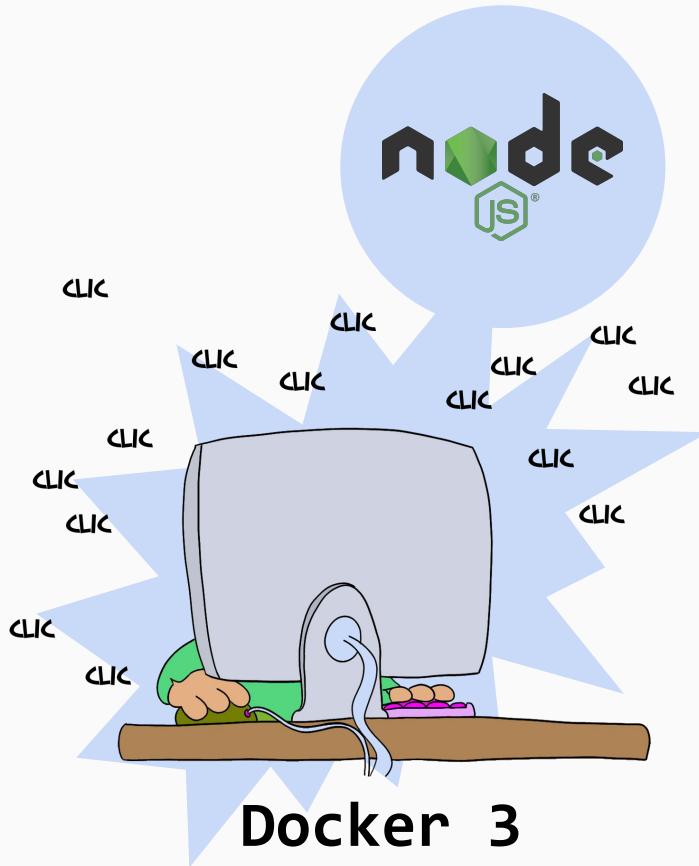
Mariner



```
$ docker login  
$ docker push Mariner
```

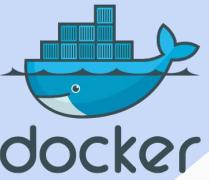
```
$ docker save Mariner > /tmp/mariner.tar
```

```
$ docker load < /tmp/mariner.tar
```

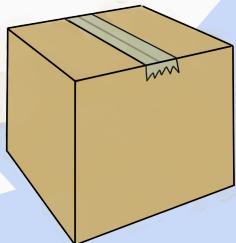


Permet  
desenvolupar  
**APLICACIONS  
COMPLEXES**





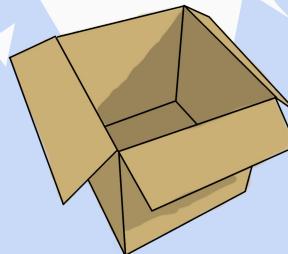
Aplicació  
web



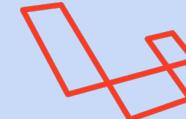
Servidor  
de Base  
de dades



Servidor web



django

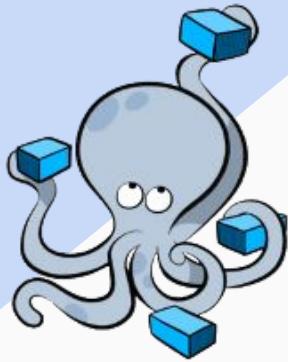


laravel

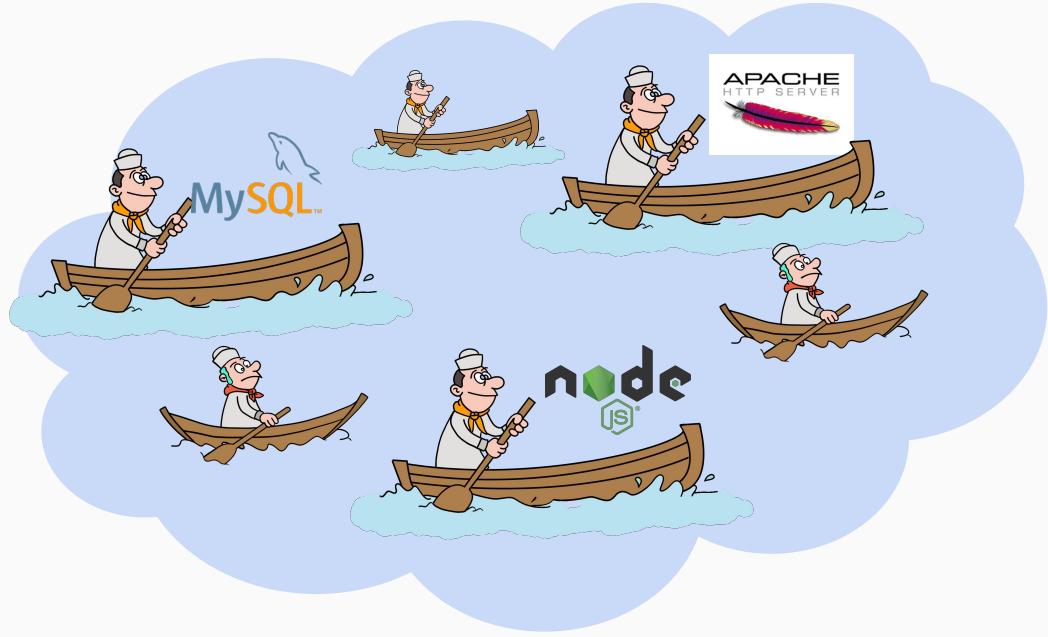
Modul  
de vendes

Frontend

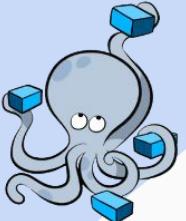
Modul  
d'estoc



**docker-compose**



Facilita desplegar una aplicació  
formada per diferents contenidors



aplicació  
node.js

Base  
dades

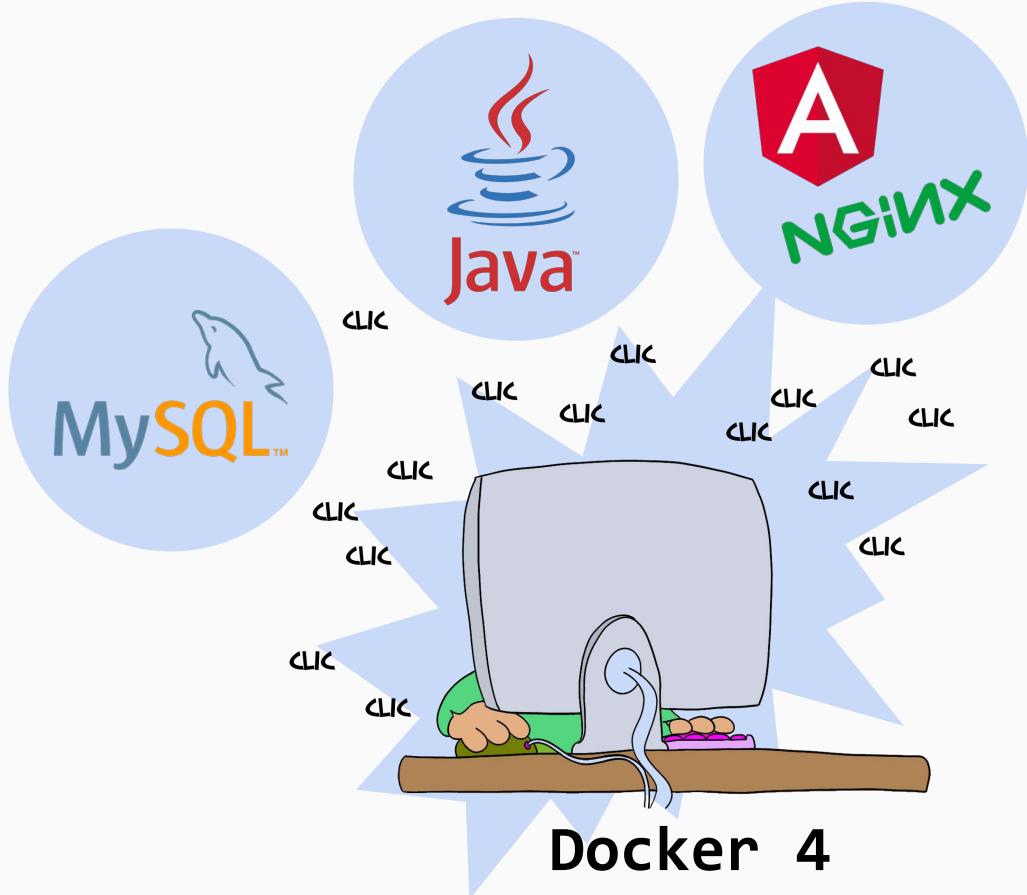
docker-compose.yml

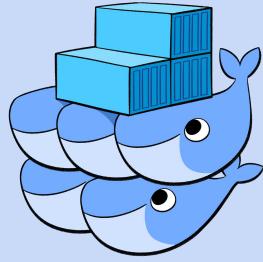
```
version: '3'  
  
services:  
  
  colorsservice:  
    build: ./ColorsService  
    container_name: colorsService  
    ports:  
      - "8123:8123"  
    depends_on:  
      - db  
    environment:  
      - DATABASE_HOST=db  
  
  db:  
    build: ./database
```

Dependències

Variables  
d'entorn

```
$ docker-compose build  
$ docker-compose up
```





Docker swarm



Kubernetes

Executar contenidors en diverses màquines  
sense preocupar-se d'on s'executen



# Informàtics Empordà

27 de febrer de 2018

