

dia 3: Orquestració amb docker-compose

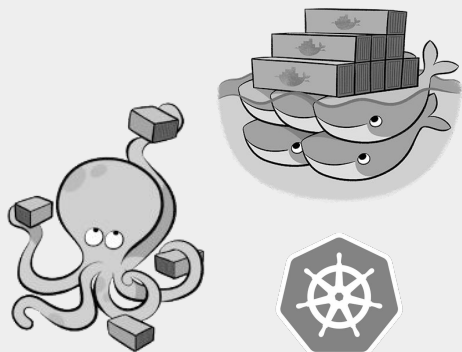
Xavier Sala Pujolar



Universitat
de Girona

Gener 2021

Orquestració

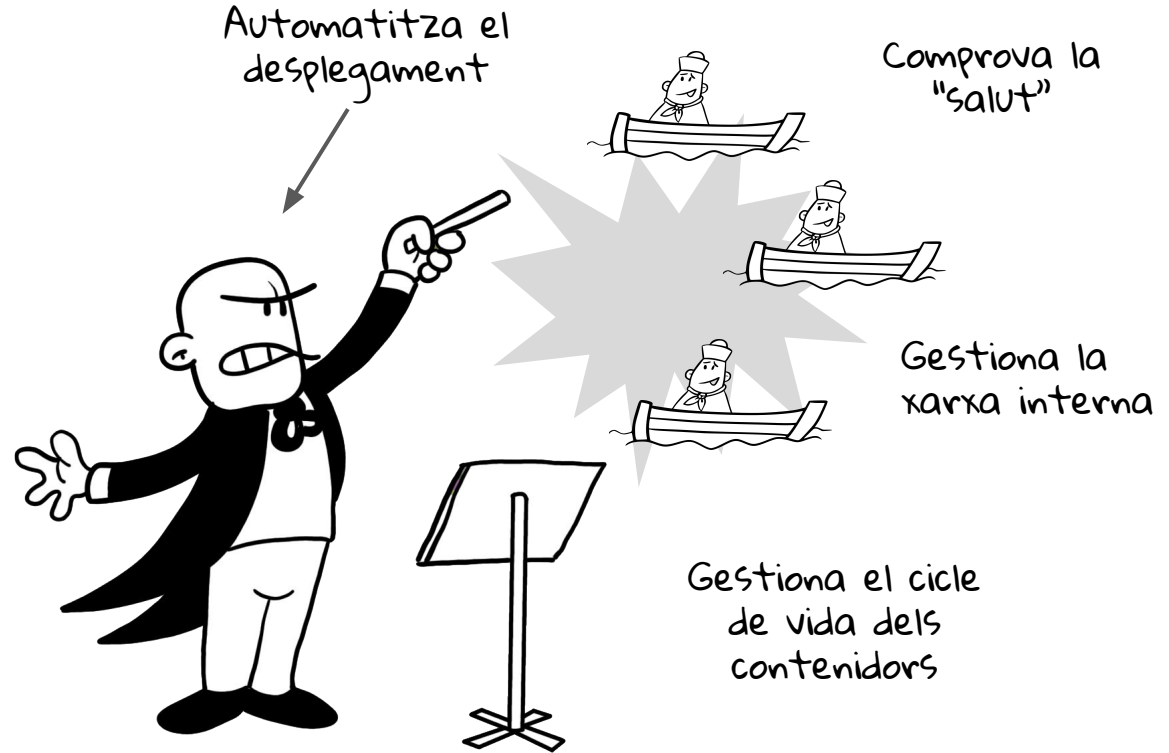


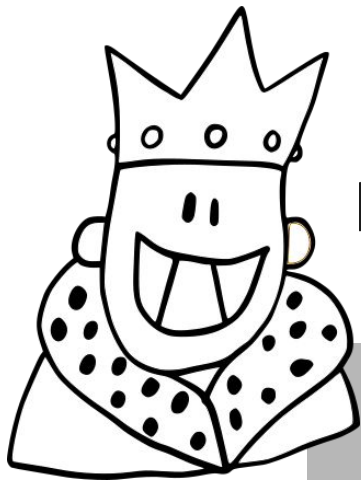
kubernetes

Rarament una aplicació
mínimament complexa funciona
sola.

Normalment conté altres
components com bases de dades,
servidors web, etc...

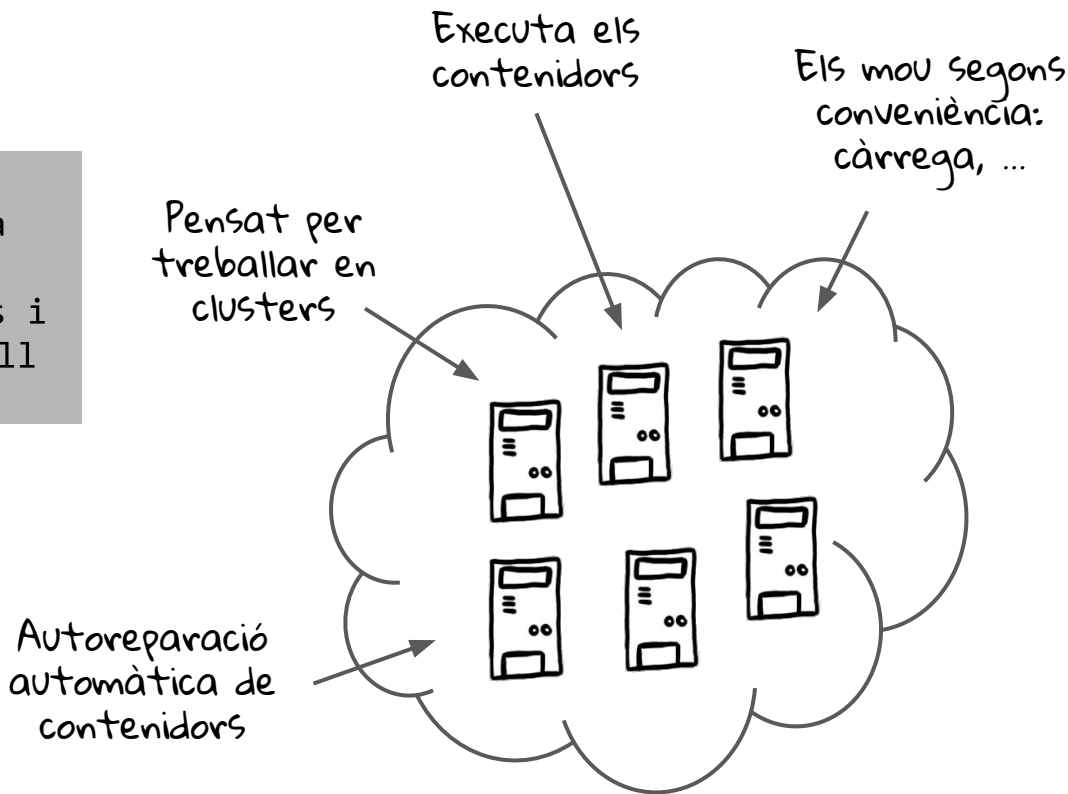
Orquestració de contenidors

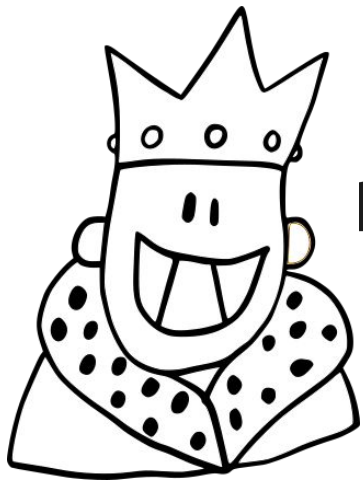




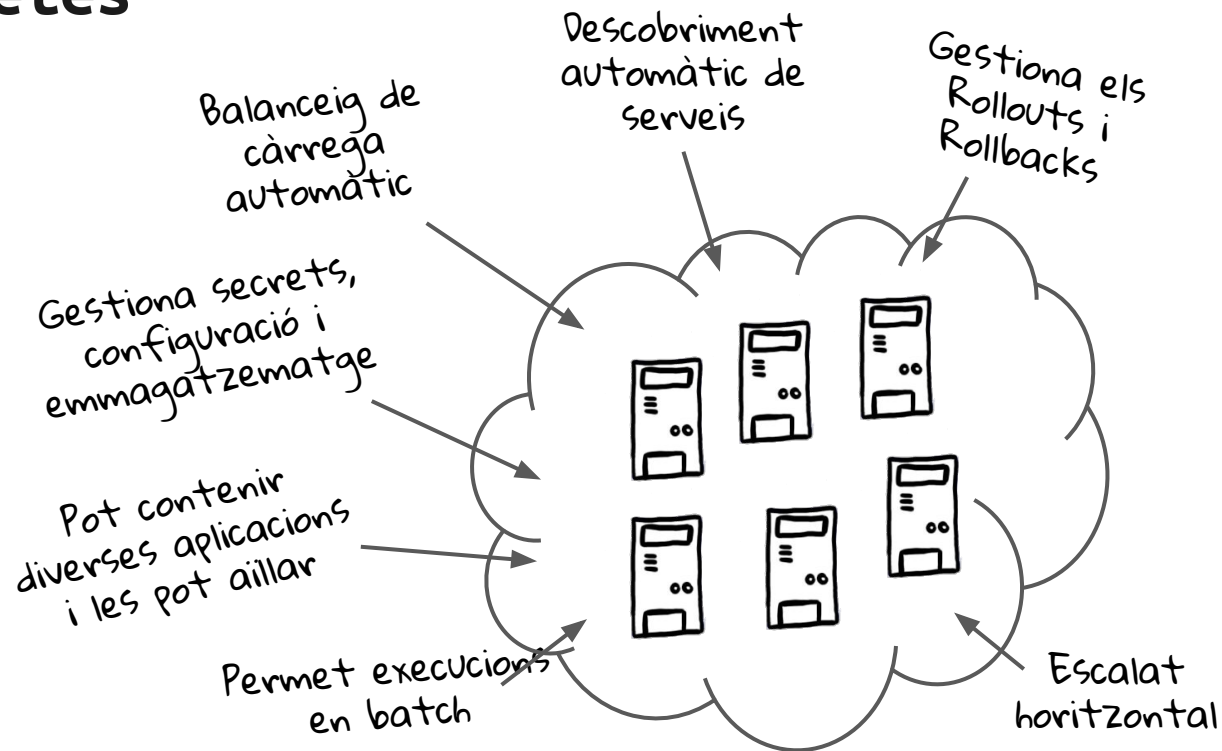
kubernetes

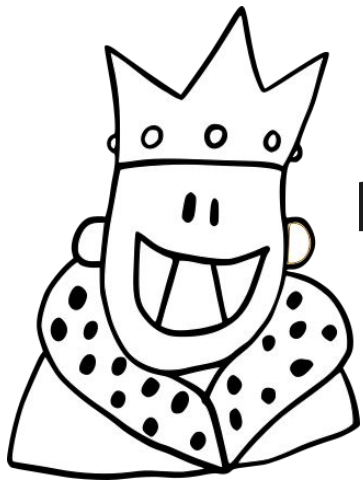
Kubernetes és una
plataforma on
administrar serveis i
càrregues de treball



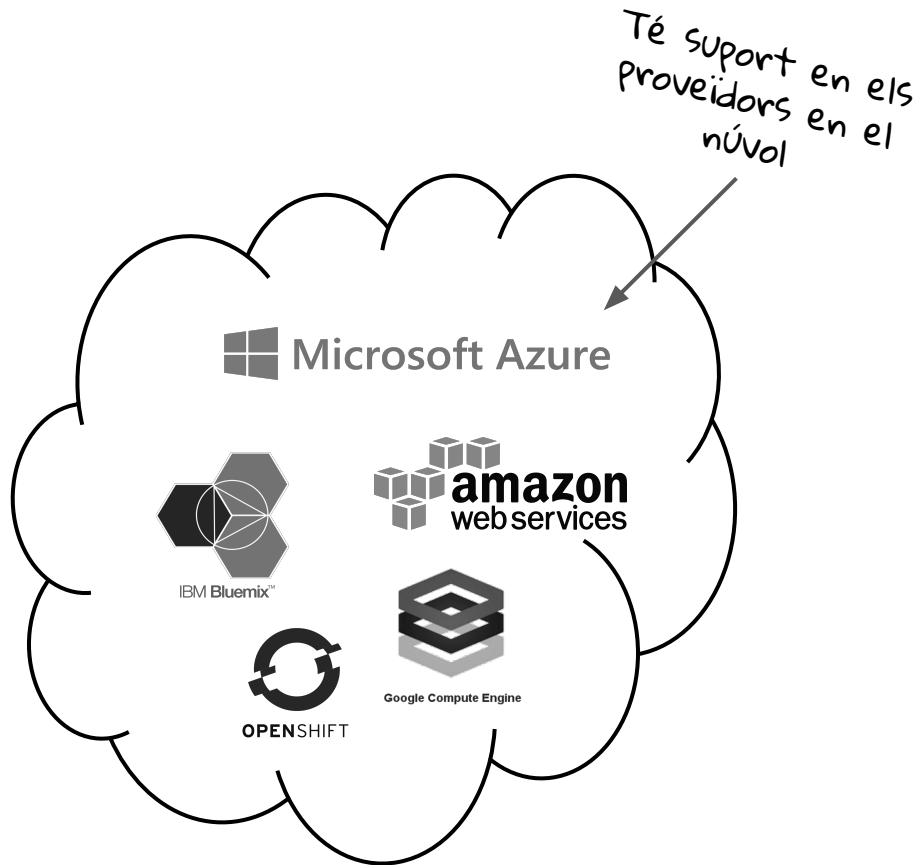


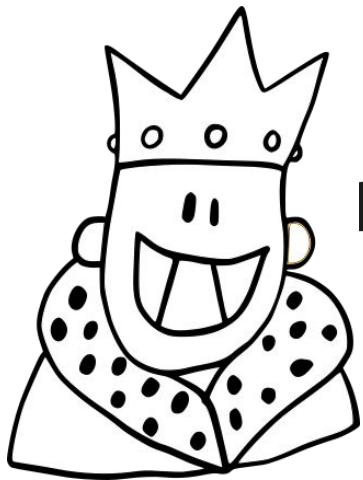
kubernetes





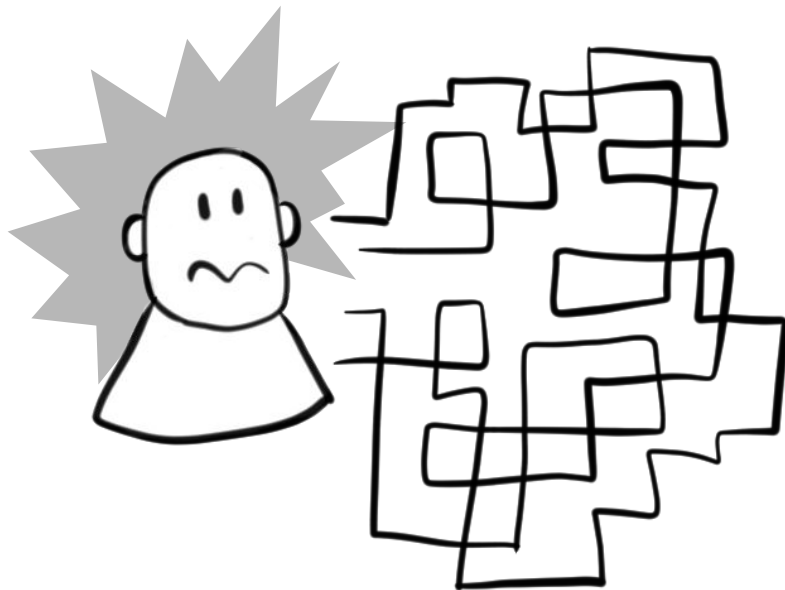
kubernetes





kubernetes

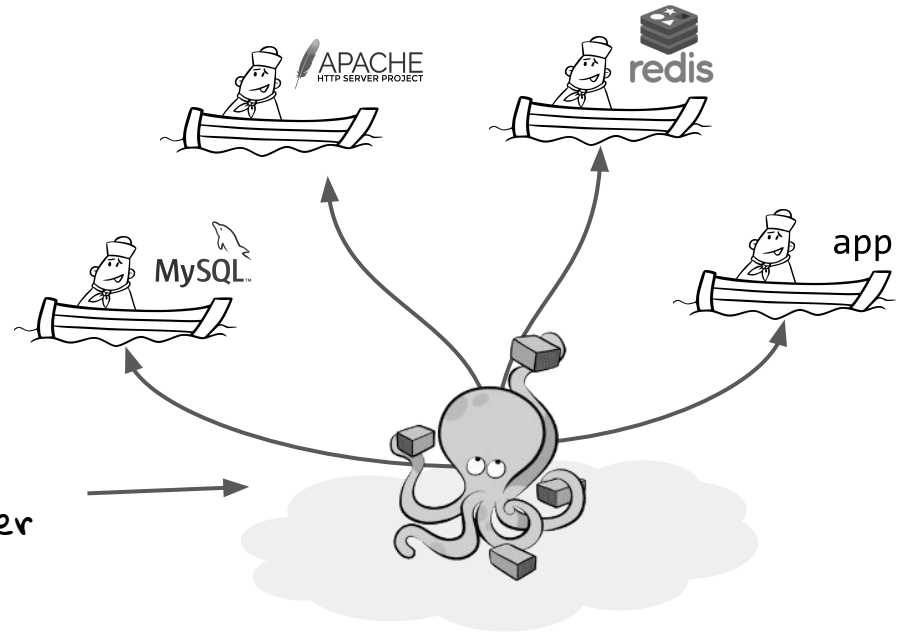
També és famós per la
seva corba
d'aprenentatge ...

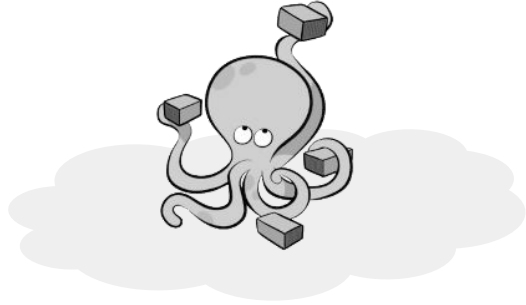




P1: **Docker-compose** - Iniciar contenedors

docker-compose permet
desplegar en una
màquina tots els
contenidors d'una
aplicació





En les versions “desktop” (Windows, MacOS) ja ve instal·lat

En Linux s'ha d'instal·lar

En alguns Linux es pot instal·lar dels repositoris: apt-get, dnf, ...

Install Compose on Linux systems

On Linux, you can download the Docker Compose binary from the Compose repository release page on GitHub. Follow the instructions from the link, which involve running the `curl` command in your terminal to download the binaries. These step-by-step instructions are also included below.

❗ For `alpine`, the following dependency packages are needed: `py-pip`, `python-dev`, `libffi-dev`, `openssl-dev`, `gcc`, `libc-dev`, and `make`.

1. Run this command to download the current stable release of Docker Compose:

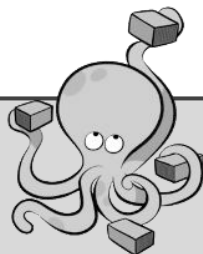
```
sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$"
```

To install a different version of Compose, substitute `1.26.0` with the version of Compose you want to use.

En un fitxer YML
es defineix **què**
necessita
l'aplicació per
construir-se i
iniciar-se

`docker-compose.yml`

```
version '3'
services:
  mydb:
    image: mariadb
    environment:
      MYSQL_ROOT_PASSWORD=pande2020
```



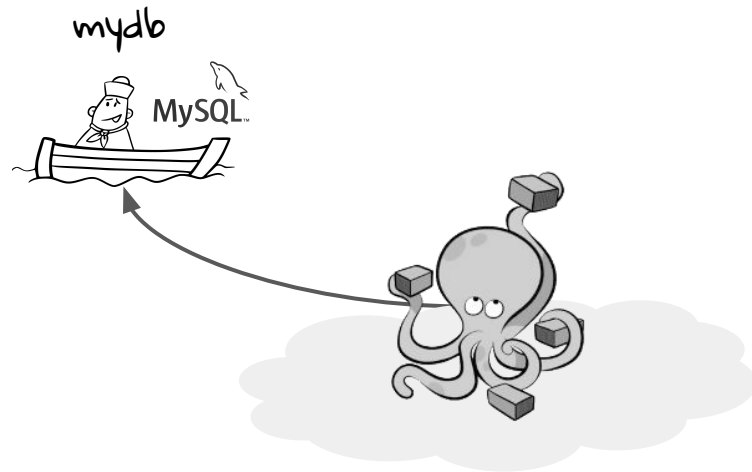
Cada contenidor
serà un 'servei'

El nom serà
reconegut pels altres
contenidors

Inicia la imatge
(refent-la, si cal)

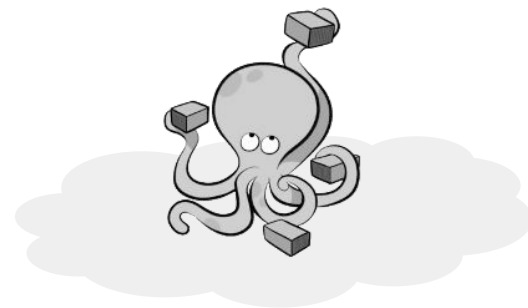


```
$ docker-compose up
```



Elimina els
contenidors

```
$ docker-compose down
```



Atura els
contenidors



```
$ docker-compose down
```

```
$ docker-compose start db
```

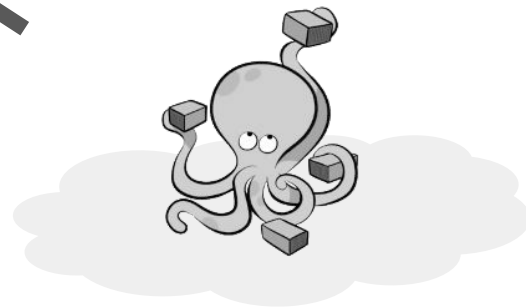
```
$ docker-compose stop db
```

```
$ docker-compose rm db
```

Gestionen els
contenidors
individualment



mydb





```
$ docker-compose build
```

```
$ docker-compose ps
```

```
$ docker-compose logs
```

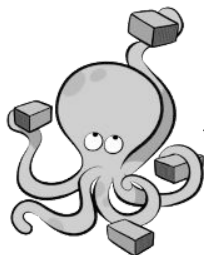
```
$ docker-compose config
```

```
$ docker-compose exec
```



P2: Iniciar diversos contenedors

Es defineixen
serveis



Els noms seran
reconeguts pels
contenidors

```
version: '3'
```

```
services:
```

```
  ubuntet:
```

```
    image: ubuntu
```

```
    tty: true
```

```
  db:
```

```
    image: mysql:5.7
```

```
    environment:
```

```
      MYSQL_DATABASE: 'db'
```

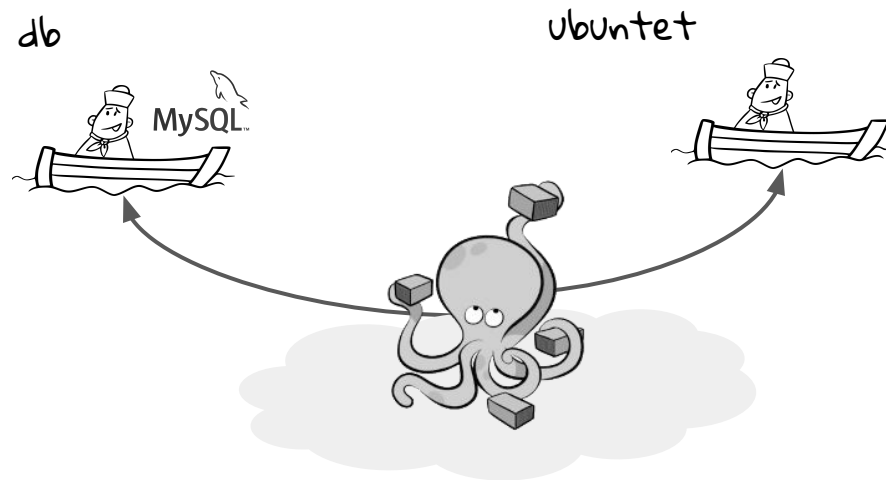
```
      MYSQL_ROOT_PASSWORD: 'password'
```

```
    ports:
```

```
      - '3306:3306'
```

Posa en marxa les
dues imatges

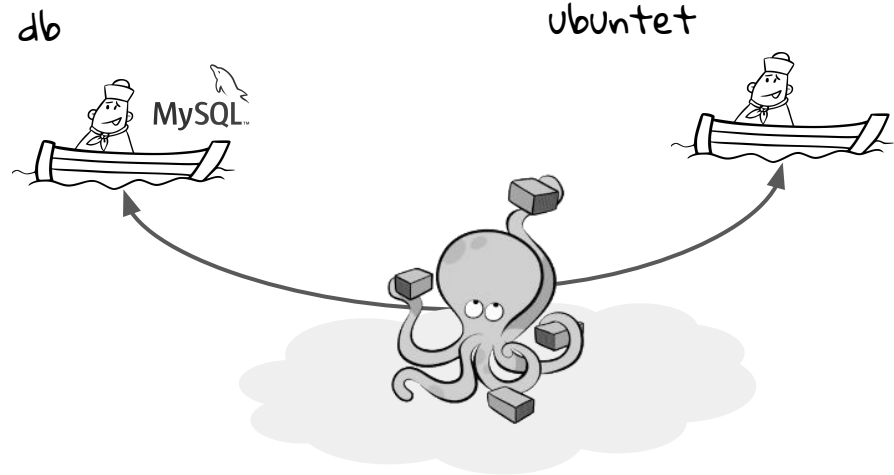
\$ docker-compose up



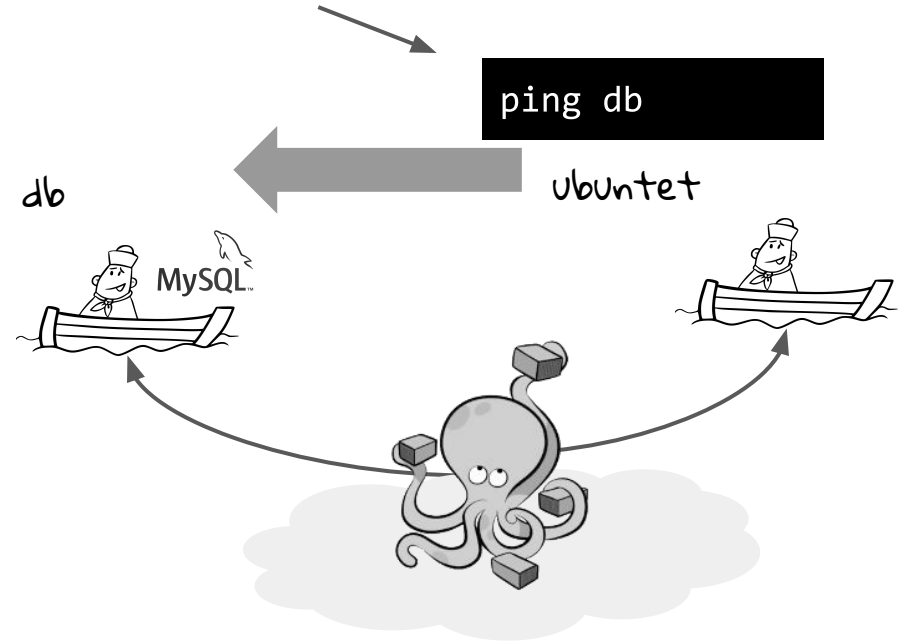
Entrem a
"ubuntet"



```
$ docker-compose exec ubuntet bash
```



Els contenidors es
coneixen pel nom





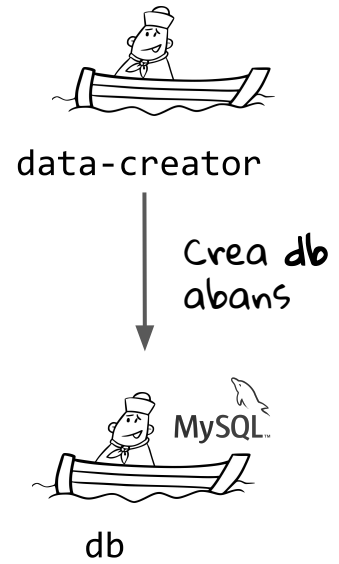
P3: Dependències de contenidors

Definir ordre d'arrencada

Sovint un contenidor
en necessita d'altres
per funcionar



```
version: '3'
services:
  data-creator:
    image: mariner
    ...
    depends-on:
      - db
  db:
    image: mysql:5.7
    ...
```



version: '3.3'

services:

db:

image: mysql:5.7

environment:

MYSQL_ROOT_PASSWORD: somewordpress

MYSQL_DATABASE: wordpress

MYSQL_USER: wordpress

MYSQL_PASSWORD: wordpress

wordpress:

depends_on:

- db

image: wordpress:latest

ports:

- "8000:80"

environment:

WORDPRESS_DB_HOST: db:3306

WORDPRESS_DB_USER: wordpress

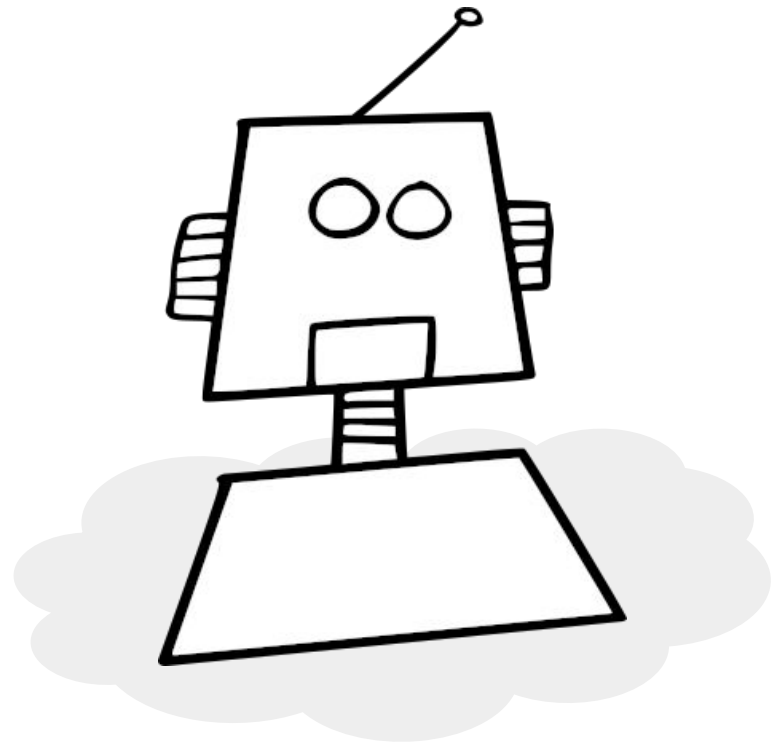
WORDPRESS_DB_PASSWORD: wordpress

WORDPRESS_DB_NAME: wordpress



P4: Construir imatges i executar

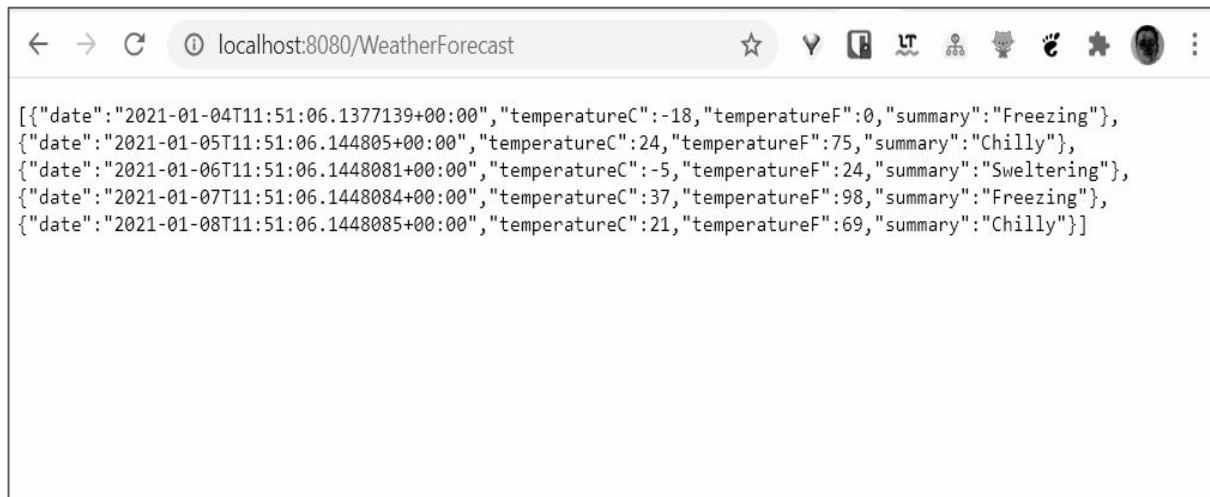
Es habitual
automatitzar la
creació d'imatges
“personalitzades”



En comptes
d'imatges se li pot
dir on hi ha el
Dockerfile

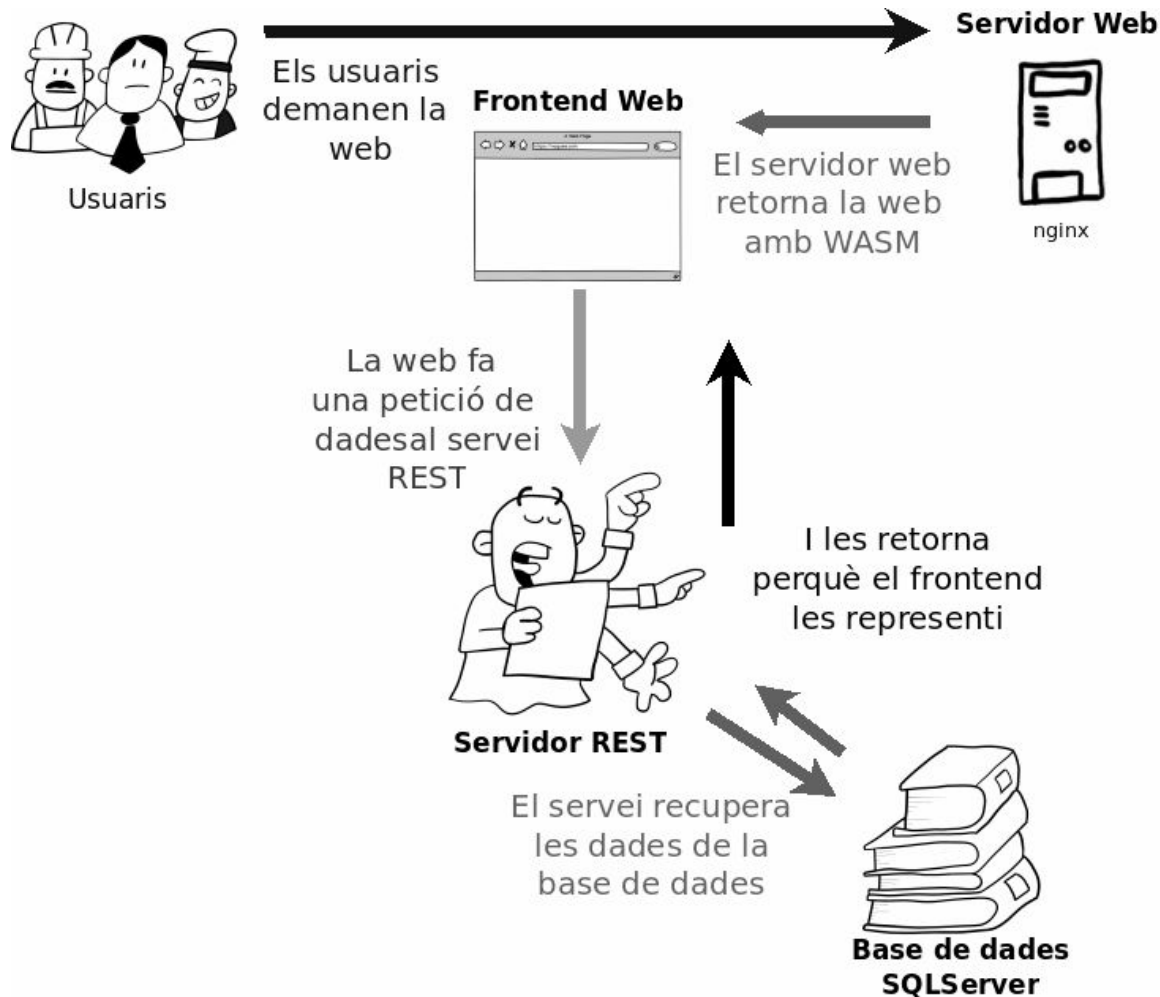
`docker-compose.yml`

```
version: '3'
services:
  weather:
    build:
      context: ./weatherservice
    ports:
      - "8080:80"
```





P5: compilar programa complex





Base de dades

`docker-compose.yml`

```
version: '3'
```

```
services:
```

```
sqlserver:
```

```
  image: mcr.microsoft.com/mssql/server
```

```
  environment:
```

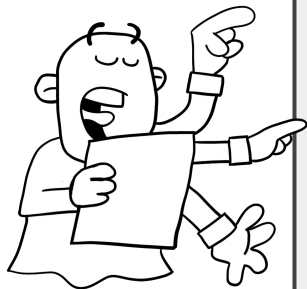
```
    SA_PASSWORD: "cursdocker2021!"
```

```
    ACCEPT_EULA: "Y"
```

```
  ports:
```

```
    - "1433:1433"
```

```
  restart: always
```




**Servei
REST**

`docker-compose.yml`

```
...
colors:
  build:
    context: ./ColorsWebService
  ports:
    - "5000:80"
    - "5001:443"
  environment:
    ConnectionSettings__ConnectionString: ""
  depends_on:
    - sqlserver
```


El connectionString
ha de tenir les
dades definides en
la base de dades



```
Data Source=sqlserver;  
User Id=sa;  
Password=cursdocker2021!;  
Initial Catalog=Colors;  
Trusted_Connection=False;  
MultipleActiveResultSets=true
```

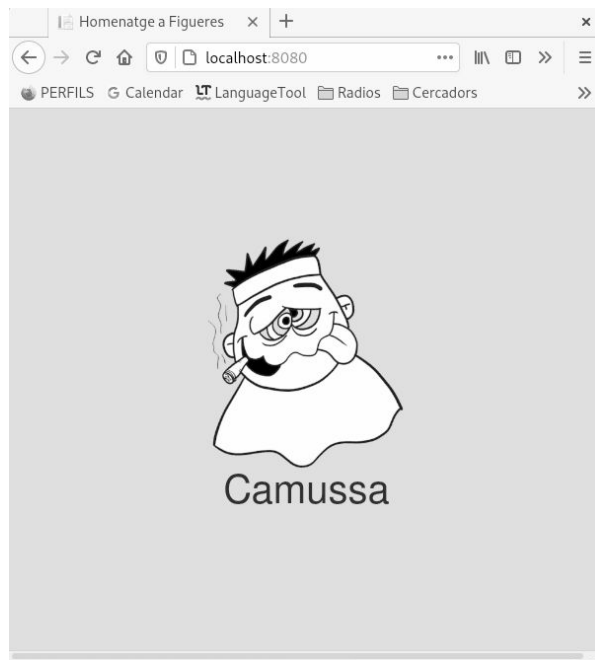


Servidor
web

`docker-compose.yml`

```
...  
web:  
  build:  
    context: ./espantApp  
  ports:  
    - "8080:80"  
  depends_on:  
    - colors  
  restart: always
```

```
$ docker-compose up
```





En vull més!

Més feina?

-



