



1. Docker Básico

Compras una computadora

```
docker create
```

Compras una computadora y la enchufas

```
docker run
```

Haces shutdown a una computadora

```
docker stop
```

Vuelves a encender la computadora

```
docker start
```

La tiras por la ventana con todos los datos que hubiera (Ojo con tus bitcoins)

```
docker rm
```

2. Somos un DataCenter

Vamos a ver a nuestra computadora como un DataCenter que ofrece alojamiento

- ✓ Somos un DataCenter que ofrece infraestructura
- ✓ Damos sevicio a muchos clientes (proyectos)
- ✓ Cada cliente está aislado del resto
- ✓ Un cliente puede cancelar cuando quiera
- ✓ Cada cliente quiere ejecutar diferentes servicios
- ✓ Cada cliente quiere acceder a su infra desde su casa

3. Docker Compose como un DataCenter

Docker compose es un "plugin" de Docker que te simplifica la gestión de coordinar varios containers (red, volumen, variables de entorno, ...)

En un único fichero de texto YAML configuramos todos los servicios relacionados (asi que puedes versionarlo)

- Cada cliente lo alojaremos en un directorio diferente
- Su infra estará en un fichero `docker-compose.yml`
- Los datos están aislados del resto al usar subdirectorios
- Podemos tener varios clientes levantados ... si la computadora tiene recursos
- Podemos compartir puertos de la computadora asignandolos a puertos de las instancias

4. Alta de cliente

Un "cliente" quiere un mysql y un redis ejecutandose

cliente1/docker-compose.yml

```
services:
  mysql:
    image: "mysql:5.7.20"
    environment:
      - MYSQL_ROOT_PASSWORD=password
      - MYSQL_DATABASE=mydatabase
      - MYSQL_PASSWORD=password
      - MYSQL_USER=user
    ports:
      - '3306:3306'
  redis:
    image: "redis"
```

`docker compose up -d`

- ✓ Docker descarga imagenes
- ✓ Crea los containers y su network
- ✓ Abre un puerto del "datacenter" asociandolo al mysql

5. Suspension de cliente

El "cliente" no ha pagado y le suspendemos el servicio (a.k.a. tenemos que trabajar en otro proyecto y no tenemos recursos para todos)

`docker compose stop`

**Los recursos siguen presente pero no están ejecutandose. El puerto 3306 está libre**

6. Baja de cliente

El "cliente" no va a pagar y le damos de baja (a.k.a. hemos entregado el proyecto y ya no lo necesitamos):

`docker compose rm -v`

**Todos los datos se han borrado**

Authors :



<https://www.linkedin.com/in/jagedn>  
Mentors juniors by telling old "war" stories

Ir al repositorio de pildoras