

# Sesión 2: Docker

## info

- <https://es.atlassian.com/software/confluence>
- <https://12factor.net>
- duffle → helm
- istio.io : securiza la conexión entre PODs

## sudo & services

- sudoreplay
  - /var/log/sudo-io
- supervisor daemon (levanta varios servicios)

## datos

- datadog : <https://www.datadoghq.com/>
- influx : <https://www.influxdata.com/>

## Parallel changes

- Avro / Parquet / Hadoop

## docker

- desconfiar (imágenes) de lo que no se puede trazar o comprobar
  - probablemente no se actualicen
  - si no hay Dockerfile o depende de otras imágenes que no son confiables, no es recomendable
- agrupar capas de comandos que las generan siempre que sea posible
  - RUN, COPY, ADD...
- `docker run -ti --entrypoint /bin/bash grafana/grafana`
- VOLUME
  - `docker volume create <nombre>`
    - por defecto en local, al crear se puede especificar otro drivers
  - `docker volume ls`
  - `/var/lib/docker/volumes/...`
- NETWORK
  - bridge = switch software
  - `brctl show`
  - `docker network create --attachable <red>`
    - **--attachable** permite añadir a mano otros contenedores a una red
- REDIS
  - usarlo para guardar sesiones, permite ampliar los frontales

## DOCKER-COMPOSE

- <https://docs.docker.com/compose/install/>
- services : uno o más contenedores haciendo lo mismo

[docker-compose.yml](#)

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
```

/via: <https://docs.docker.com/compose/wordpress/>

## DOCKER-SWARM

- stack: conjunto de servicio+volumenes+redes...
- docker stack ls
- docker node ls
- docker stack servies <STACK>
- docker service scale wordpresito\_wordpress=10 : escala a 10 los wordpress
  - en este caso no coordina las sesiones (para eso el REDIS)
- docker stack deploy -c docker-compose.yml <nombre\_stack>

## recortes

## continous deployment

- canary release : 5% + 95%
  - rolling updates : actualizamos un nodo, si todo correcto, a por el siguiente, si no, atrás
- blue-green (red-black)
- microservicios
  - cada uno con su storage
  - ACID - CAP (2 de 3)

## laboratorio

### fracaso

```
FROM debian:7-slim

ARG WP_FILE=latest.tar.gz

WORKDIR /tmp
#ADD https://wordpress.org/${WP_FILE} .

RUN apt-get -y update
RUN apt-get install -y apache2 wget php7.0
RUN wget --no-check-certificate https://wordpress.org/${WP_FILE}
RUN tar xf ${WP_FILE}
#RUN mkdir -p /var/www/html
RUN mv wordpress/* /var/www
RUN rm ${WP_FILE}
RUN apt-get clean

# apt-get install -y apache2 libapache2-mod-php7.0 php-mysql php-gd php-redis && \

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apache2ctl", "-DFOREGROUND"]
```

From:  
<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:  
<https://miguelangel.torresegea.es/wiki/info:cursos:pue:devops:sesion2?rev=1549796566>

Last update: 10/02/2019 03:02

