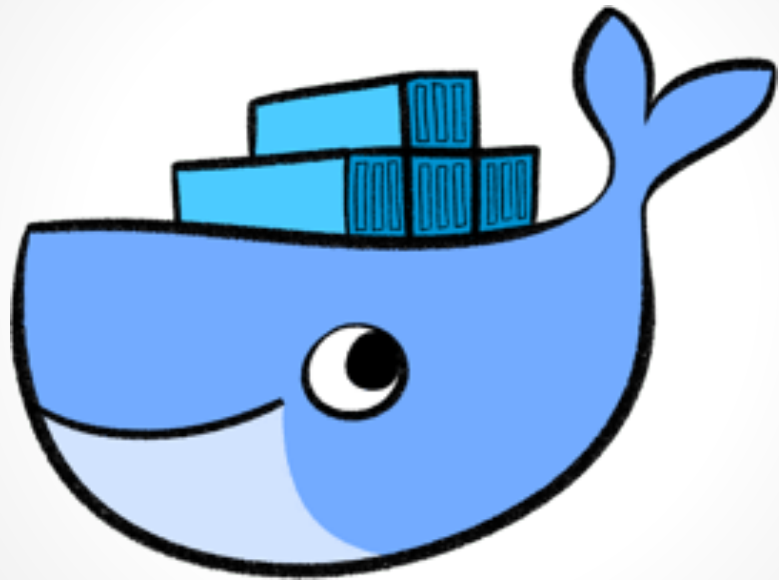


# Curso de Docker

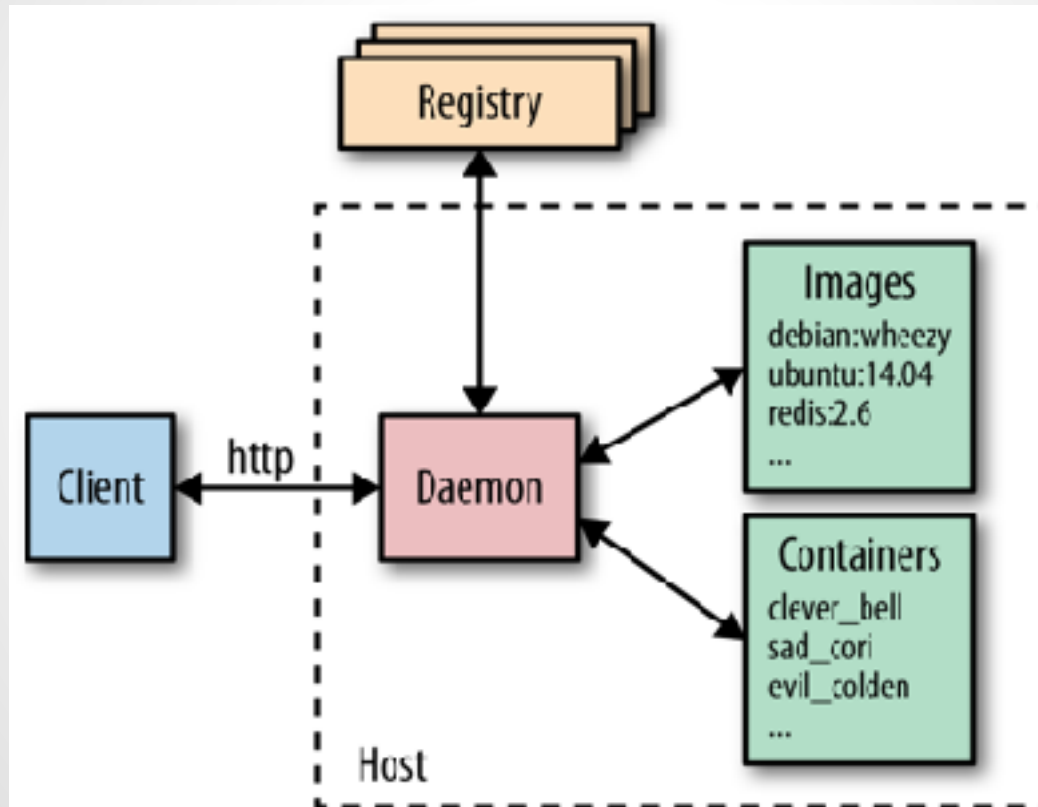


**Comunidad de Castilla la Mancha**

# Guía del Curso

- Introducción a Docker
- Arquitecturas de Microservicios
- Construcción de Imágenes
- Desarrollo con Contenedores
- Integración Continua
- Introducción a Kubernetes

# Componentes Docker



# Comandos Comunes para Contenedores

- docker info
- docker version
- docker run (docker container run)
- docker ps (docker container list)
- docker inspect (docker container inspect)
- docker stop (docker container stop)
- docker start (docker container start)
- docker rm (docker container rm)
- docker cp (docker container cp)
- docker exec (docker container exec)
- docker logs (docker container logs)
- docker stats (docker container stats)
- docker system prune
- \$DOCKER\_HOST

# Docker Build y Dockerfile

## Dockerfile

```
FROM ubuntu # Layer 0  
MAINTAINER Florian Lopes # Layer 1  
RUN mkdir -p /some/dir # Layer 2  
RUN apt-get install -y curl # Layer 3
```

## FINAL IMAGE

Layer 3 (Image 3)

Layer 2 (Image 2)

Layer 1 (Image 1)

Layer 0 (Base image)

# Docker Build y Dockerfile

- FROM (from my image)
- RUN
- CMD ([ "echo", "\$HOME" ])
- ENTRYPOINT
- LABEL key=value
- EXPOSE
- ENV key=value
- ADD src dest
- VOLUME /path
- USER user[:group]
- WORKDIR
- ARG name[=default]  
(FROM ubuntu:\${tag})
- ONBUILD
- STOPSIGNAL
- HEALTHCHECK (see docs)

# Guía del Curso

- Introducción a Docker
- Arquitecturas de Microservicios
- Construcción de Imágenes
- Desarrollo con Contenedores
- Integración Continua
- Introducción a Kubernetes

# Desarrollo con Contenedores

- Docker Compose y docker-compose.yml
- Volúmenes
- Redes
- Casos de Uso Comunes con docker-compose



# **docker-compose.yml**

## **(docker run)**

```
version: "3"
services:
  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: password
  wordpress:
    image: wordpress:latest
    depends_on:
      - db
    ports:
      - "8000:80"
```

# **docker-compose.yml (docker build)**

```
version: "3"
services:
  app:
    image: myapp
    build:
      context: ./dir
      dockerfile: Dockerfile
      args:
        - arg1=val1
        - arg2=val2
```

# Comandos Comunes para Compose

- docker-compose up
- docker-compose pull
- docker-compose build
- docker-compose push
- docker-compose run
- docker-compose rm

# Volúmenes

- Volúmenes de Datos
- Volúmenes de Host
- docker volume (create | ls | inspect | rm)

# Volúmenes

```
services:
  mysql:
    image: mysql
    volumes:
      - data:/var/lib/mysql
      - logs:/var/log/mysql
      - /etc:/etc
  analyzer:
    image: log-analyzer
    volumes:
      - logs:/var/log:ro
volumes:
  data:
  logs:
```

# Ejercicio

- Crear una imagen que escribe a disco en CMD
- Crear docker-compose para lanzar el contenedor
- Modificar el docker-compose y recrear
- Se mantienen los cambios?
- Repetir el ejercicio usando volúmenes

# Redes

- Bridge
- Host
- None
- docker network (create | ls | inspect | rm)

# Redes

```
services:
  proxy:
    image: busybox
    networks:
      - outside
  app:
    image: busybox
    networks:
      - default
      - inside

networks:
  outside:
    external: true
  default:
  inside:
    driver: bridge
    enable_ipv6: true
```



# Ejercicio

- Crear un compose con dos contenedores en una red
- Crear un compose con dos contenedores en otra red
- Son visibles los contenedores en la misma red?
- Y en redes distintas?

# Docker Compose Reference

<https://docs.docker.com/compose/compose-file/>

# Casos de Uso Comunes

Dockerize dependencies

<https://github.com/pchico83/docker-castilla-la-mancha/tree/master/local>

# Casos de Uso Comunes

## Service Discovery

```
version: "3.2"
services:
  web:
    build: .
    depends_on:
      - redis
    ports:
      - 5000:5000
  redis:
    image: redis
```

# Casos de Uso Comunes

## Variables de Entorno

sample:

```
image: golang:${GO_VERSION:-1.9}
```

```
command: go version
```

# Casos de Uso Comunes

Montar Directorio de Trabajo

<https://github.com/pchico83/docker-castilla-la-mancha/tree/master/go-tests>

# Casos de Uso Comunes

## Montar Docker Socket

```
contenedores:  
  image: docker:17.10  
  volumes:  
    - /var/run/docker.sock:/var/run/docker.sock  
  entrypoint: docker  
  command: version
```

# Casos de Uso Comunes

## Aplicación Django, con celery y beat

```
api:  
  image: app  
  build: .  
  command: gunicorn -b 0.0.0.0:80 -w 8 app.wsgi
```

```
worker:  
  image: app  
  entrypoint: celery -A app  
  command: worker -c 8 -P prefork -O fair
```

```
beat:  
  image: app  
  entrypoint: celery -A app  
  command: beat
```



# Casos de Uso Comunes

## Importar Compose File

```
web:
  build: .
  volumes:
    - './code'
  ports:
    - 8883:80
  environment:
    DEBUG: 'true'
```

```
db:
  command: '-d'
  ports:
    - 5432:5432
```

```
web:
  image: app
  depends_on:
    - db
db:
  image: postgres:latest
```

Dev

Prod

```
web:
  ports:
    - 80:80
  environment:
    PRODUCTION: 'true'
```

# Casos de Uso Comunes

Extender otro Servicio

<https://github.com/pchico83/docker-castilla-la-mancha/tree/master/extends>

# Local Environment Sample

<https://github.com/dockersamples/example-voting-app>