



Universidad de Oviedo



Es
Español



ARQUITECTURA
DEL SOFTWARE

Arquitectura del software

Lab. 06

Distribución & Despliegue

2022-23

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Cristian Augusto

GitHub Pages

- GitHub permite crear sitios web
- Útil para información personal
- Despliega lo que se encuentra en la rama de repositorio **gh-pages**

GitHub Pages - ejemplos

Nivel Organizativo

Repositorio:

<https://github.com/Arquisoft/Arquisoft.github.io>

Desplegado:

<https://arquisoft.github.io/>

Muy útil para tener páginas personales

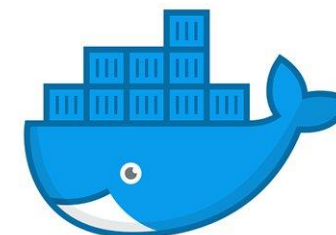
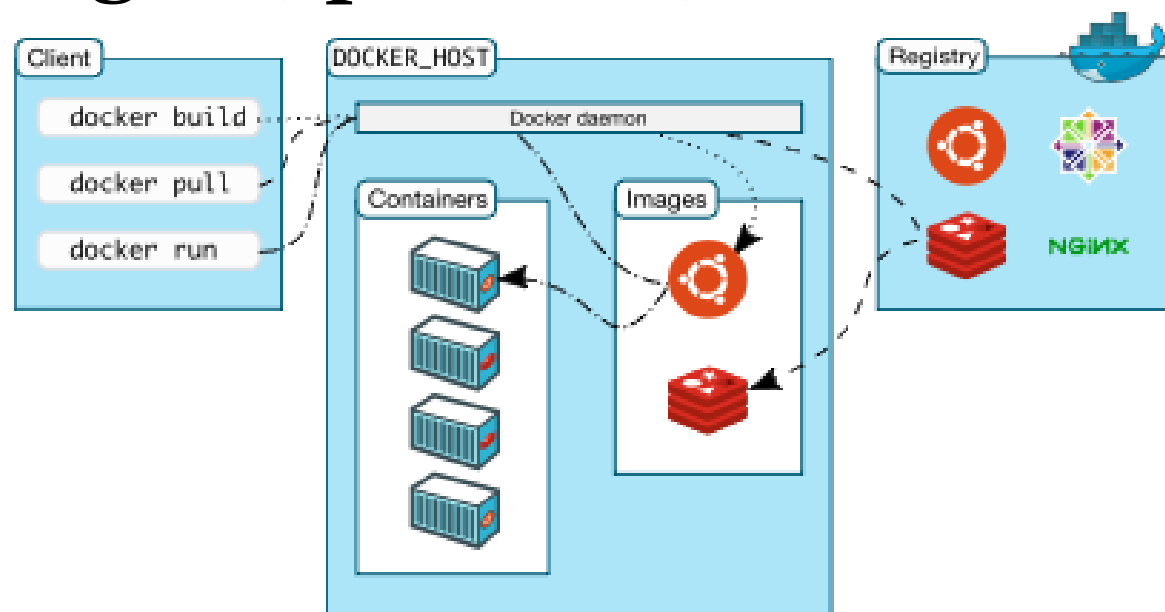
<http://pglez82.github.io>

¿Qué es Docker?

Plataforma para desarrolladores y administradores de sistema

Basada en contenedores

Flexible, ligero, portable, ...



¿Qué es una imagen?

Un fichero que puede ser usado para crear un paquete ejecutable

Incluye todas las cosas necesarias para ejecutar una aplicación:

- Código

- Sistema ejecutable

- Librerías

- Variables de sistema

- Ficheros de configuración

No tiene estado y no cambia

¿Qué es un contenedor?

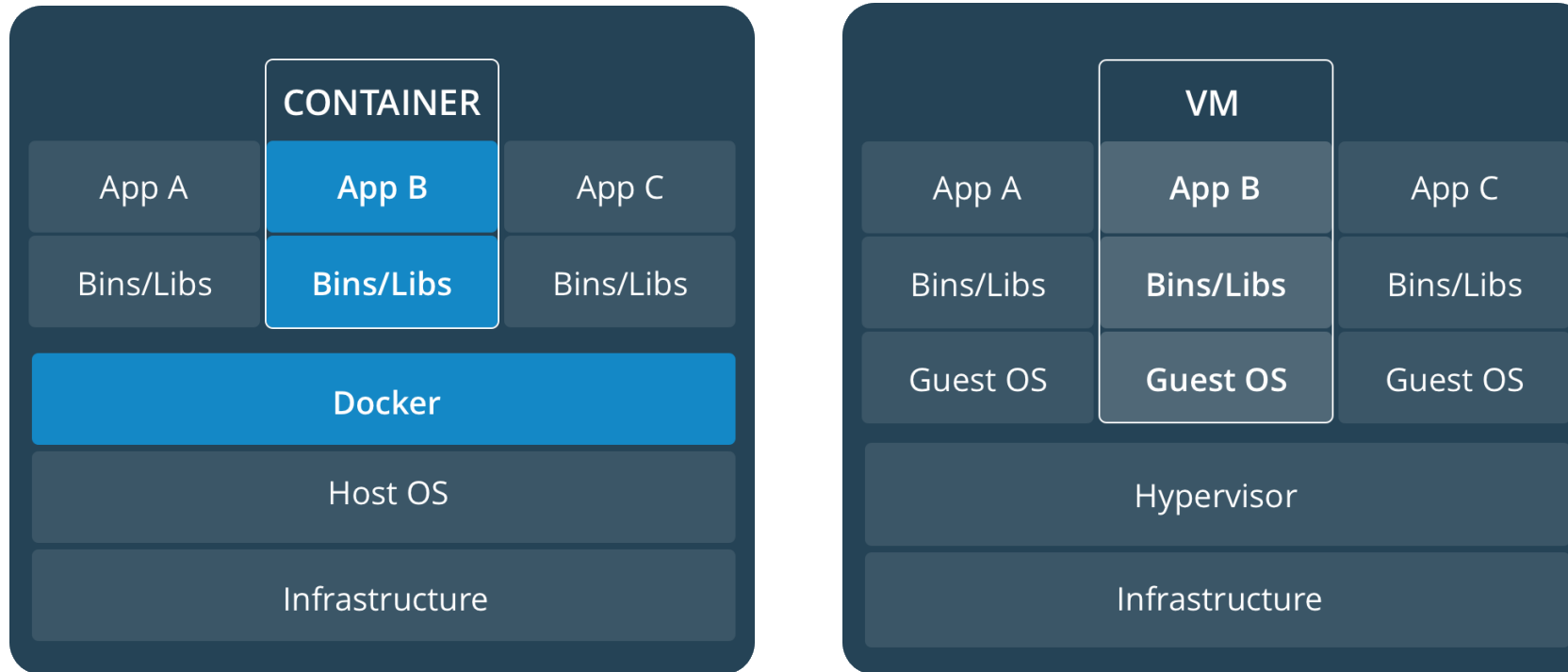
Es una instancia de una imagen

Docker está basada en contenedores que contienen aplicaciones

Docker permite orquestar varios contenedores

Enlazando varios contenedores podemos crear arquitecturas complejas

¿ Es una máquina virtual?



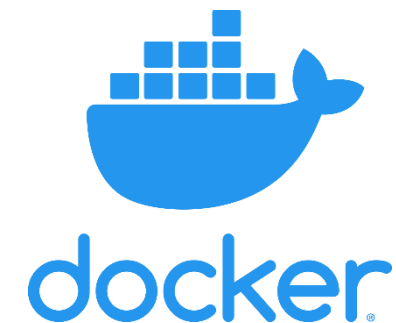
Source: <https://docs.docker.com/get-started/#containers-and-virtual-machines>
<https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-virtual-machine>

Instalando docker

<https://www.docker.com>

Disponibile para GNU/Linux, windows and Mac. Dos versiones:

- Docker desktop (Windows/Mac)
- Docker ToolBox [faq#issue3](#)



Docker Hub

Repositorio de imágenes Docker

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

Más velocidad y modularidad para el desarrollo.

Imágenes testeadas para servicios conocidos

Ejemplo: Necesitamos un servidor web

```
docker pull nginx
```

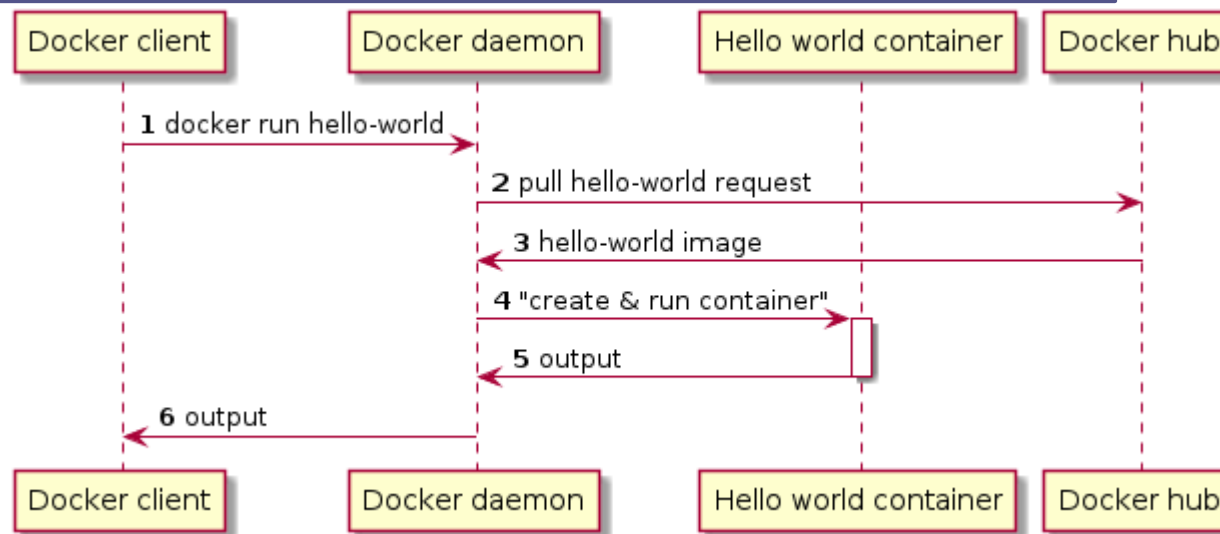
```
docker pull httpd
```

Docker paso a paso

Comprobar Instalación Docker Run “Hello World”

```
$ docker -v
```

```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:f9dfddf63636d84ef479d645ab5885156ae030f...
Status: Downloaded newer image for hello-world:latest
```



Docker para ejecutar Linux

Ejecutar Ubuntu

```
$ docker container run -it ubuntu:latest /bin/bash
. . .
root@813cb77cebb2:/# ls -la
total 72
drwxr-xr-x  1 root root 4096 Mar 30 05:46 .
drwxr-xr-x  1 root root 4096 Mar 30 05:46 ..
-rwxr-xr-x  1 root root    0 Mar 30 05:46 .dockerenv
drwxr-xr-x  2 root root 4096 Mar 11 21:05 bin
drwxr-xr-x  2 root root 4096 Apr 24  2018 boot
drwxr-xr-x  5 root root  360 Mar 30 05:47 dev
drwxr-xr-x  1 root root 4096 Mar 30 05:46 etc
. . .
drwxr-xr-x  1 root root 4096 Mar 11 21:03 usr
drwxr-xr-x  1 root root 4096 Mar 11 21:05 var
root@813cb77cebb2:/#
```

Docker status

Commando para comprobar estados

```
λ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	fce289e99eb9	14 months ago	1.84kB

```
λ docker container ls --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
8b6518da11db	hello-world	"/hello"	9 minutes ago	Exited (0) 9 minutes ago

https://github.com/pglez82/docker_cheatsheet

Sencillo servidor web con Docker

Correr un servidor web con Docker

Ejecutado en background

publish:expose port

```
$ docker run --detach --publish=80:80 --name=webserver nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
68ced04f60ab: Pull complete
28252775b295: Pull complete
a616aa3b0bf2: Pull complete
Digest: sha256:2539d4344dd18e1df02be842ffc435f8e1f699cfc55516e2cf2cb16b7a9aea0b
Status: Downloaded newer image for nginx:latest
b7e9213eb3367cd465b29701a7e6441a7210a46d439196d30e76ddc9c72ee280
```

localhost

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

Otros comandos

```
docker info
```

```
docker ps
```

```
docker image ls
```

```
docker container ls -all
```

```
docker pull
```

```
docker run
```

```
docker stop
```

```
docker rm
```

Ejemplo1: Correr Solid localmente

Node solid server

Imagen de Docker disponible en

<https://hub.docker.com/r/nodesolidserver/node-solid-server>

Traer una imagen

```
$ docker pull nodesolidserver/node-solid-server
```

Ejecutar imagen

```
$ docker run -p 8443:8443 --name solid nodesolidserver/node-solid-server
```

Ver la aplicación en: <https://localhost:8443>

Cómo construir una imagen

DSL para construir imágenes

Necesitamos crear un fichero que se llama **Dockerfile**

Contiene los comandos necesarios para crear la imagen

Keywords: FROM, RUN, ADD, COPY, ENV, EXPOSE, CMD...

Dockerfile

```
FROM ubuntu
```

```
CMD echo "Hi Software architecture students"
```


Leyendo una imagen

1. Crear un directorio para el proyecto
2. Editar un Dockerfile (sin extensión)
3. docker build -t nombre_imagen.
4. docker images (lista de imagenes)
5. docker run nombre_imagen

Dockerfile

```
FROM ubuntu
CMD echo "Hi ASW students"
```

```
λ docker build -t "example1" .
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM ubuntu
latest: Pulling from library/ubuntu
5bed26d33875: Pull complete
...
Digest: sha256:bec5a2727be7fff3d308193cfde3491f8fba1a2...
Status: Downloaded newer image for ubuntu:latest
---> 4e5021d210f6
Step 2/2 : CMD echo "Hi Software architecture students"
---> Running in 9d5516995c2b
Removing intermediate container 9d5516995c2b
---> 41784c740df4
Successfully built 41784c740df4
Successfully tagged example1:latest
```

```
λ docker images
REPOSITORY TAG      IMAGE ID  CREATED        SIZE
example1   latest 41784c740 32 seconds ago 64.2MB
```

```
λ docker run example1
Hi ASW students
```

Ejemplo 2:

Muestra LoMap webapp

https://github.com/Arquisoft/lomap_0/tree/master/webapp

```
1  FROM node:18.13.0
2  LABEL org.opencontainers.image.source https://github.com/arquisoft/lomap_0
3  COPY . /app
4  WORKDIR /app
5  #Install the dependencies
6  RUN npm install
7
8  ARG API_URI="http://localhost:5000/api"
9  ENV REACT_APP_API_URI=$API_URI
10
11 #Create an optimized version of the webapp
12 RUN npm run build
13
14 #Execute npm run prod to run the server
15 CMD [ "npm", "run", "prod" ]
16 #CMD [ "npm", "start" ]
```

Ejemplo 3:

Muestra LoMap restapi

https://github.com/Arquisoft/lomap_0/tree/master/restapi

```
1 FROM node:18.13.0
2 LABEL org.opencontainers.image.source https://github.com/Arquisoft/lomap_0
3 COPY . /app
4 WORKDIR /app
5 #Install the dependencies
6 RUN npm install
7 CMD [ "npm", "start" ]
```

Combinar diferentes contenedores de docker

- Docker compose permite la modularización de una aplicación o arquitectura
- Los servicios definidos pueden comunicarse entre ellos
- Cada servicio se encuentra en un contenedor separado.
- Archivo: docker-compose.yml
- [LoMap docker-compose file](#)

Ejecutar Docker compose

Configuración

- Podemos configurar múltiples servicios
- Los servicios pueden depender de otros
- Por defecto, todos los servicios comparten la misma red y son accesibles a través del nombre del contenedor.

Running

- Para ejecutar (o detener) un archivo docker-compose ejecutamos el comando: `docker-compose (up|down)`

Información adicional

Pequeño repositorio con todos los comandos básicos utilizados en docker:

https://github.com/pglez82/docker_cheatsheet

Tips

Forzar la reconstrucción en docker-compose

```
$ docker-compose up --build --force-recreate
```