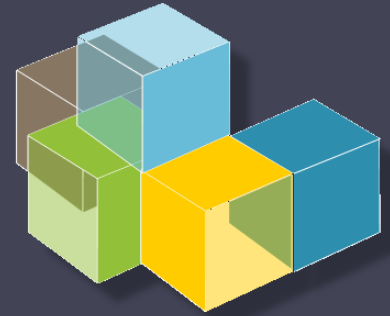


Universidad de Oviedo



Escuela de
Ingeniería
Informática



ARQUITECTURA
DEL SOFTWARE

Arquitectura del Software

Lab. 09 Distribución y despliegue

2019-20

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebrede

GitHub Pages

- Soporte de GitHub para crear websites personales o para un proyecto/repositorio
- Permite crear sitio web en master o en rama **gh-pages**

GitHub Pages - ejemplo

- Página web de cada equipo/usuario:
 - Fuente: https://github.com/arquisoft/viade_o
 - Despliegue: https://arquisoft.github.io/viade_o/
- A nivel organización
 - Repositorio:
 - <https://github.com/Arquisoft/Arquisoft.github.io>
 - Despliegue:
 - <https://arquisoft.github.io/>
- También puede ser útil para páginas personales

Despliegue continuo

- Mediante Travis se despliega automáticamente

```
language: node_js
node_js:
  - 12.14.0
cache:
  directories:
    - node_modules
before_install:
  - sudo apt-get update
  - sudo apt-get -y install ruby openjdk-8-jre
  - sudo gem install asciidoctor asciidoctor-diagram
script:
  - npm install -g codecov
  - npm test && codecov
  - npm run test:e2e
  - npm run build
  - npm run docs
deploy:
  provider: pages
  skip_cleanup: true
  github_token: $github_token
  local_dir: build
on:
  branch: master
```

.travis.yml

github_token variable must be set in Travis settings
It is generated by github

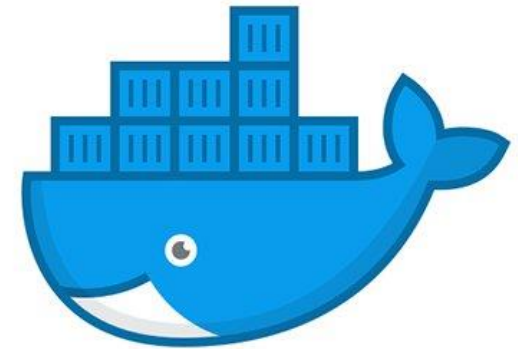
<https://docs.travis-ci.com/user/deployment/pages/>

¿Qué es Docker?

Plataforma para desarrolladores y administradores de sistemas

Basada en contenedores

Flexible, ligero, portable, escalable...



¿Qué es una imagen?

Fichero que permite crear un paquete ejecutable
Incluye todo lo necesario para lanzar una aplicación:

- Código

- Sistema de ejecución

- Librerías

- Variables de entorno

- Ficheros de configuración

No tiene estado y no cambia

¿Qué es un contenedor?

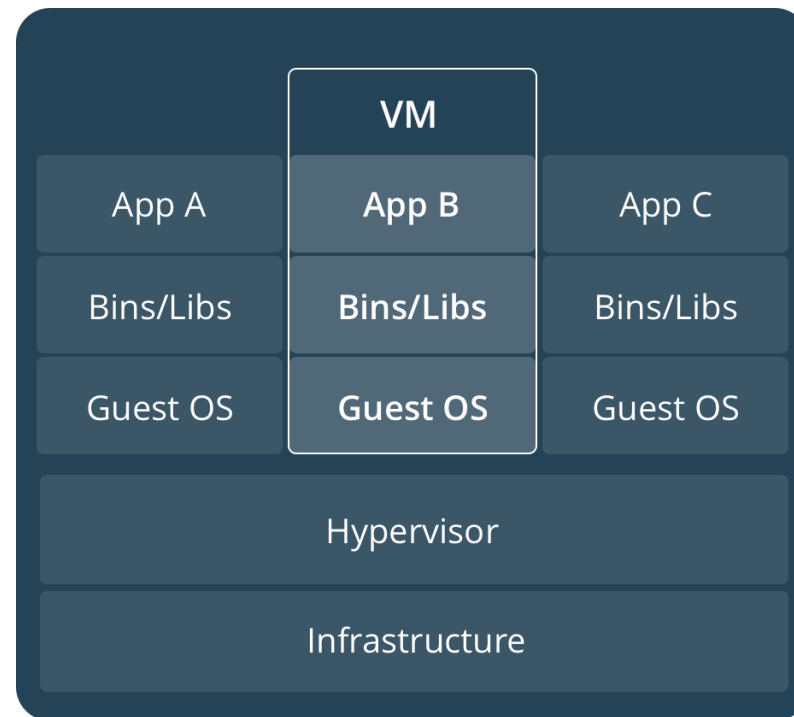
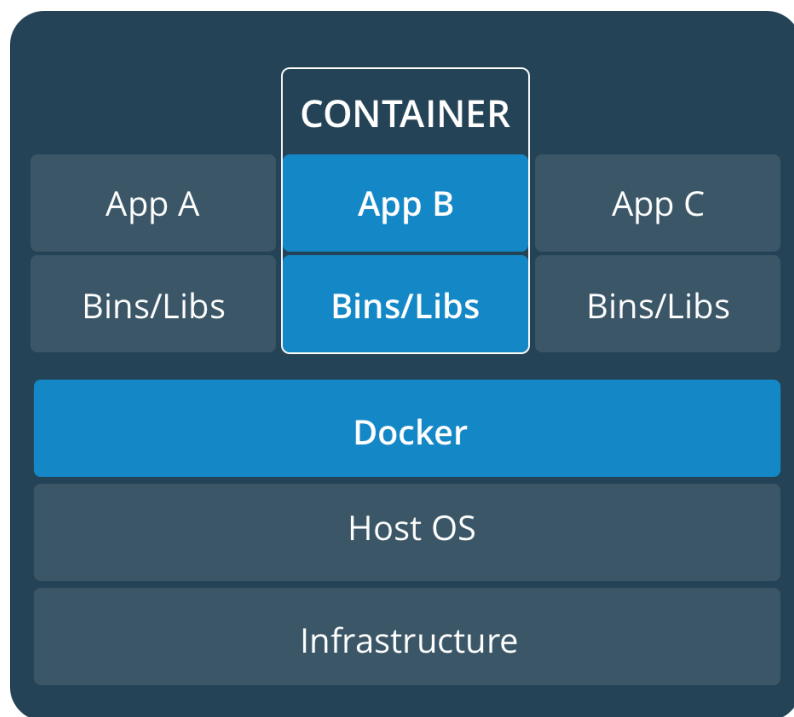
Es una instancia viva de una imagen

Docker se basa en contenedores para envolver las aplicaciones

Docker permite la orquestación entre contenedores

Enlazando varios contenedores podemos crear una arquitectura compleja

¿Y esto no lo hacen las VMs?



<https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-virtual-machine>

<https://docs.docker.com/get-started/#containers-and-virtual-machines>

Instalando Docker

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

Docker desktop (Windows pro/Mac)

Docker toolbox (Windows Home)

NOTA: Para docker toolbox, anotar dirección IP a utilizar en lugar de localhost

Docker Hub

Repositorio de imágenes de Docker

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

Mayor velocidad de desarrollo y modularización

Imágenes bien probadas y servicios conocidos

Ejemplo: Necesito un servidor web para desarrollo

```
Docker pull nginx
```

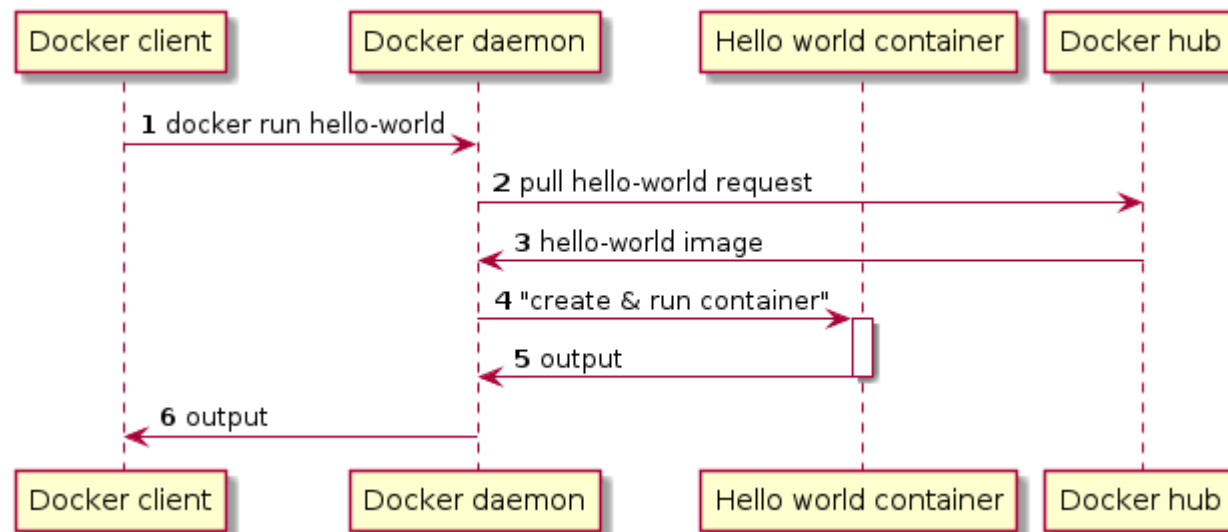
```
Docker pull httpd
```

Docker paso a paso(I)

- Instalar Docker y comprobar versión
- Run Hello World

```
$ docker --version
```

```
$ 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
```



Algunos comandos

docker info

docker ps

docker image ls

docker container ls --all

docker pull

docker run

docker stop

docker rm

Ejemplo ejecutando Linux

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 paso a paso(II)

- Dockerfiles: Creando nuestra primera imagen

Dockerfile

```
FROM ubuntu
```

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

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

Ver contenedores/imágenes

```
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

Servidor web simple con docker

- Ejemplo de servidor web

Ejecutar en background

Puerto publicar:exponer

```
$ 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.

Cómo crear una imagen

DSL propio de Docker para construir la imagen

Fichero de construcción de imagen = **Dockerfile**

Palabras clave:

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

Dockerfile

```
FROM ubuntu
```

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

Construir una imagen

1. Crear directorio para el Proyecto
2. Editar Dockerfile (sin extensión)

Dockerfile

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

3. `docker build -t image_name`

```
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
```

4. `docker images` (ver imágenes)

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

5. `docker run image_name`

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

Docker solid example

Node solid server

Docker image available at

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

Pull image

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

Run image

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

Browse the App at <https://localhost:8443>

Example React App with Solid

- Source code to clone:
- https://github.com/Arquisoft/viade_docker

```
$ docker build -t solidwebapp .
Sending build context to Docker daemon 315.9kB
Step 1/5 : FROM node:12.14.1
---> 839a5e8f04b4
Step 2/5 : COPY . /app
---> 68823456d581
Step 3/5 : WORKDIR /app
---> Running in 9819c3fbda1
Removing intermediate container e44c69532
Step 4/5 : RUN > @ start /app
---> Running in 9819c3fbda1
...
Removing intermediate container 77ced15fe
Step 5/5 : CMD ["npm", "start"]
---> Running in 679e2b77f82e
Removing intermediate container 679e2b77f82e
---> ec54814b5ca6
Successfully built ec54814b5ca6
```

```
$ docker run --name solidwebapp -p 3000:3000 solidwebapp
> @ start /app
> react-scripts start

Starting the development server...
```

Combinando ambos contenedores

- Docker compose permite modularizar una aplicación o arquitectura
- Se definen diferentes servicios que pueden comunicarse entre ellos
- Cada servicio puede estar en un contenedor separado
- Fichero de configuración: `docker-compose.yml`

Ejecutar Docker compose

- Configuración
 - `docker-compose.yml`

```
version: '3'
services:
  solidserver:
    image: nodesolidserver/node-solid-server
    container_name: solidserver
    ports:
      - "8443:8443"
  sampleweb:
    build: .
    ports:
      - "3000:3000"
```

```
$ docker-compose up
```

```
Creating network "viade_docker_default" with the default driver
Building sampleweb
Step 1/5 : FROM node:12.14.1
----> 839a5e8f04b4
Step 2/5 : COPY . /app
----> 9221a1d3d2cf
Step 3/5 : WORKDIR /app
----> Running in 90c4499dc650
Removing intermediate container 90c4499dc650
----> 40afa7189b6e
Step 4/5 : RUN npm install
----> Running in c90224dbb7bc
...
```

Codefresh (<https://codefresh.io/>)

- Es una plataforma para construir y desplegar imágenes Docker.
- Ofrece un conjunto de servicios sobre los contenedores Filosofía SaaS (Software como Servicio)
- En la versión gratuita permite tener un solo proyecto y 120 builds al mes.
- Se puede asociar la cuenta de codefresh con la de Github
(Docker Hub también puede configurar builds automáticos desde Github)

Codefresh: Desplegando desde Github

- Se puede hacer todo desde la consola online
- Directamente desde un repositorio en Github
- Será necesario tener el Dockerfile
 - Este fichero puede estar en el repositorio
 - O lo podemos crear al añadir el repositorio
- <https://codefresh.io/docs/docs/getting-started/create-a-basic-pipeline/>

Otras alternativas

- Heroku (<https://www.heroku.com/>)
- OpenShift (<https://www.openshift.com/>)
- CloudKarafka (<https://www.cloudkarafka.com/>)
- Wercker (<http://www.wercker.com/>)
- Bitrise(<https://www.bitrise.io/>)
- <https://paasfinder.org/vendors>

Heroku

- Servicio de pago “similar” a Docker

<https://tuhrig.de/docker-vs-heroku/>

Docker	Heroku
Dockerfile	BuildPack
Image	Slug 500mb *gratis
Container	Dyno 1 dyno *gratis
Index	Add-Ons
CLI	CLI

- Despliegue y actualización directa desde un repositorio en Github

Enlaces de interés

- **GitHub Pages** <https://pages.github.com/>
 - **A guide to using Github Pages**
<https://www.thinkful.com/learn/a-guide-to-using-github-pages/>
 - **Jekyll** <https://jekyllrb.com/>
- **Docker**
 - **How to Docker (Jonny Langefeld)**
<https://jonnylangefeld.github.io/learning/Docker/How%2Bto%2BDocker.html>
<https://www.youtube.com/watch?v=JprTjTViaEA>
 - **Pushing and Pulling to and from Docker Hub**
<https://ropenscilabs.github.io/r-docker-tutorial/04-Dockerhub.htm>
 - **Dockerizing a Node.js web app**
<https://nodejs.org/es/docs/guides/nodejs-docker-webapp/>
 - **Dockerizing an Angular App**
<https://medium.com/@tupone.mattia/dockerizing-an-angular-app-made-easy-e0e3bb55a39c>
<https://mherman.org/blog/dockerizing-an-angular-app/>
- **Heroku**
 - **How to deploy an Angular application to Heroku**
<https://medium.com/@hellotunmbi/how-to-deploy-angular-application-to-heroku-1d56e09c5147>

Enlaces

- https://github.com/pglez82/docker_cheatsheet
- https://github.com/pglez82/docker_solid_example
- https://github.com/pglez82/travisreact_tut