EN
English

Universidad de Oviedo

Software Architecture

Lab. 09
Distribution & Deployment

2019-20

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebredo

GitHub Pages

GitHub supports creating websites

Useful por personal – project/repository

Branch **gh-pages**

GitHub Pages - examples

Web page:

- Source: https://github.com/arquisoft/viade_0
- Deployed: https://arquisoft.github.io/viade_0/

Organization level

- Repository:
  - https://github.com/Arquisoft/Arquisoft.github.io
- Deployed:
  - https://arquisoft.github.io/

It can be very useful for personal web pages

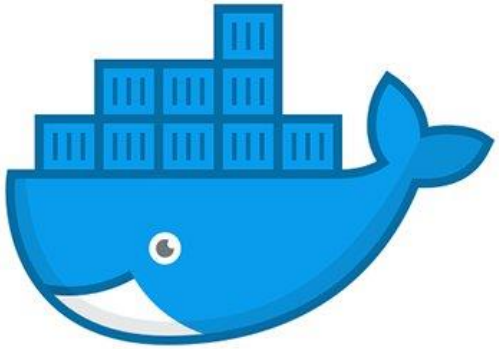
- http://labra.weso.es

What is Docker?

Platform for developers and system administrators

Based on containers

Flexible, light, portable, scalable…



What is an image?

A file that can be used to create a runnable package

Includes all things necessary to run the application:

- Code
- Runtime system
- Libraries
- Runtime variables
- Configuration files

It doesn't have state and never changes

What is a container?

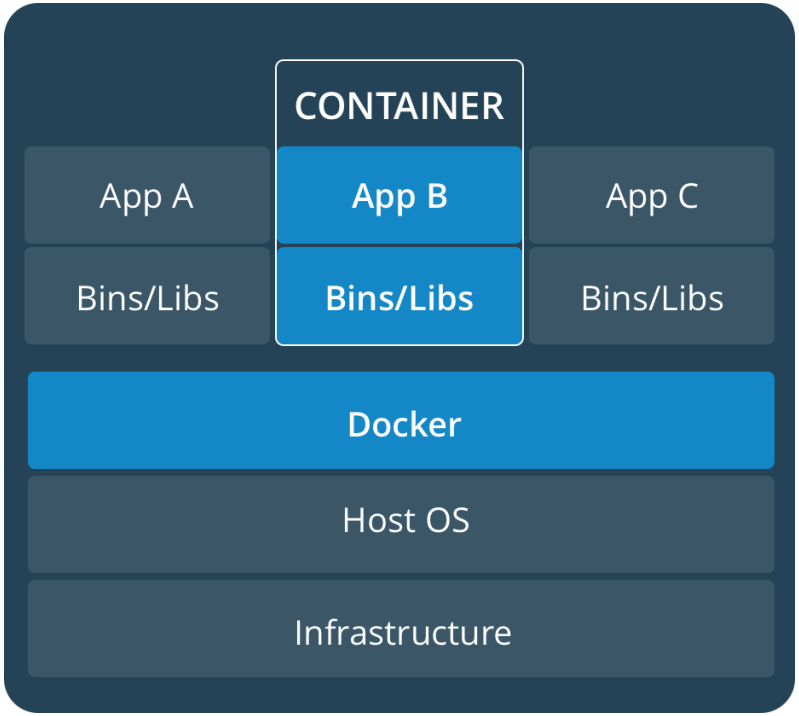
It is a live instance of an image

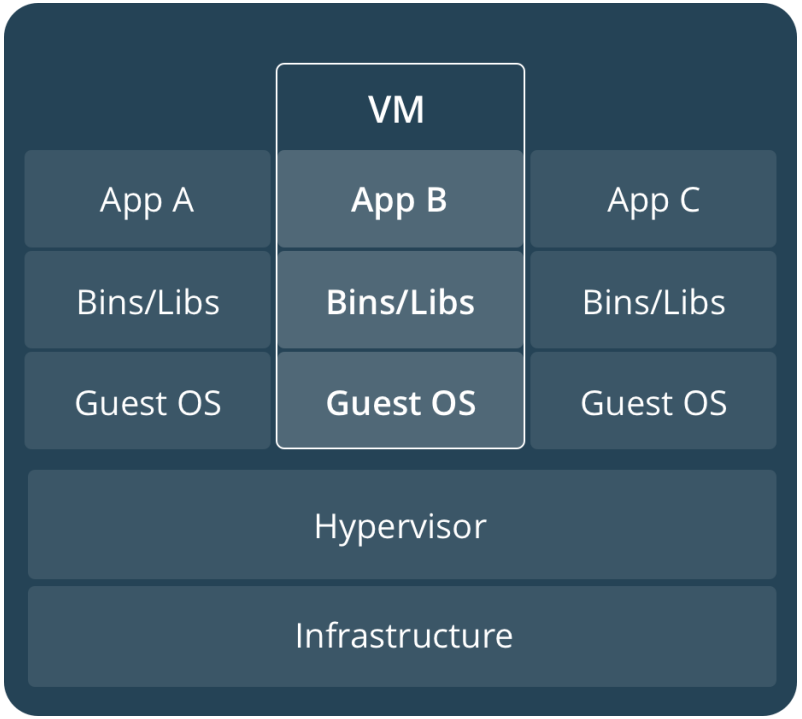
Docker is based on containers that enclose applications

Docker allows orchestration between containers

Linking several containers we can make a complex architecture

Isn't that a VM?





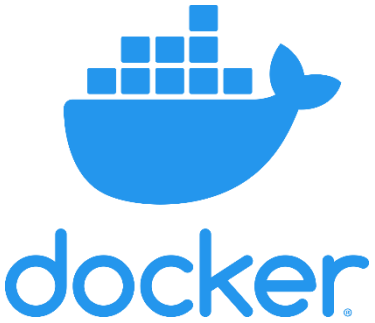
Fuente: https://docs.docker.com/get-started/#containers-and-virtual-machines
https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-virtual-machine

Obtaining docker

https://www.docker.com

Available for linux, windows and Mac

Docker desktop (Windows/Mac)



Docker Hub

Docker image repository

https://hub.docker.com/

Higher speed for development and modularity

Tested images for well-known services

Example: Need a web-server for development

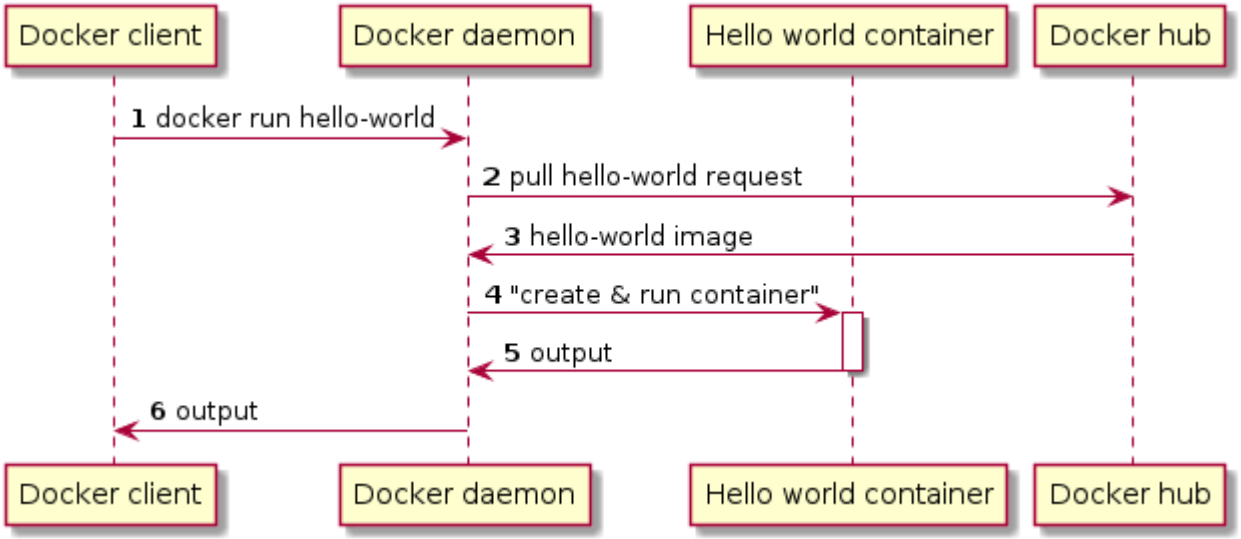
```
docker pull nginx
docker pull httpd
```

Docker step by step

Install Docker `$ docker -v`

Run “Hello World”

```
$ 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 example running Linux

Run 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

Commands to check status

```
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

Docker simple web server

Run a web server with Docker

Run in 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
```

Some commands

```
docker info
docker ps
docker image ls
docker container ls –all
docker pull
docker run
docker stop
docker rm
```

How to build an image

DSL to build images

We need to create a file, called **Dockerfile**

It contains commands necessary to build the image

Keywords: `FROM`, `RUN`, `ADD`, `COPY`, `ENV`, `EXPOSE`, `CMD`…

Dockerfile

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

Building an image

1. Create a folder for the project
2. Edit a Dockerfile (no extension)
3. docker build –t image_name
4. docker images (list images)
5. docker run image_name

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
```

Sharing an image (docker hub)

Sharing == publishing

- Possible to store an image locally as .tar

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 Solid React App

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 9819c3fbeda1
Removing interm
 ---> e44c69532
Step 4/5 : RUN
 ---> Running i
. . .
Removing interm
 ---> 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...
```

Combining both

Docker compose allows modularization of an application or architecture

Different services are defined that communicate among them

Each service is in a separate container

File: docker-compose.yml

Running Docker compose

Configuration

docker-compose.yml

```yaml
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
...
```

Another example

Docker Composer: docker-compose.yml:

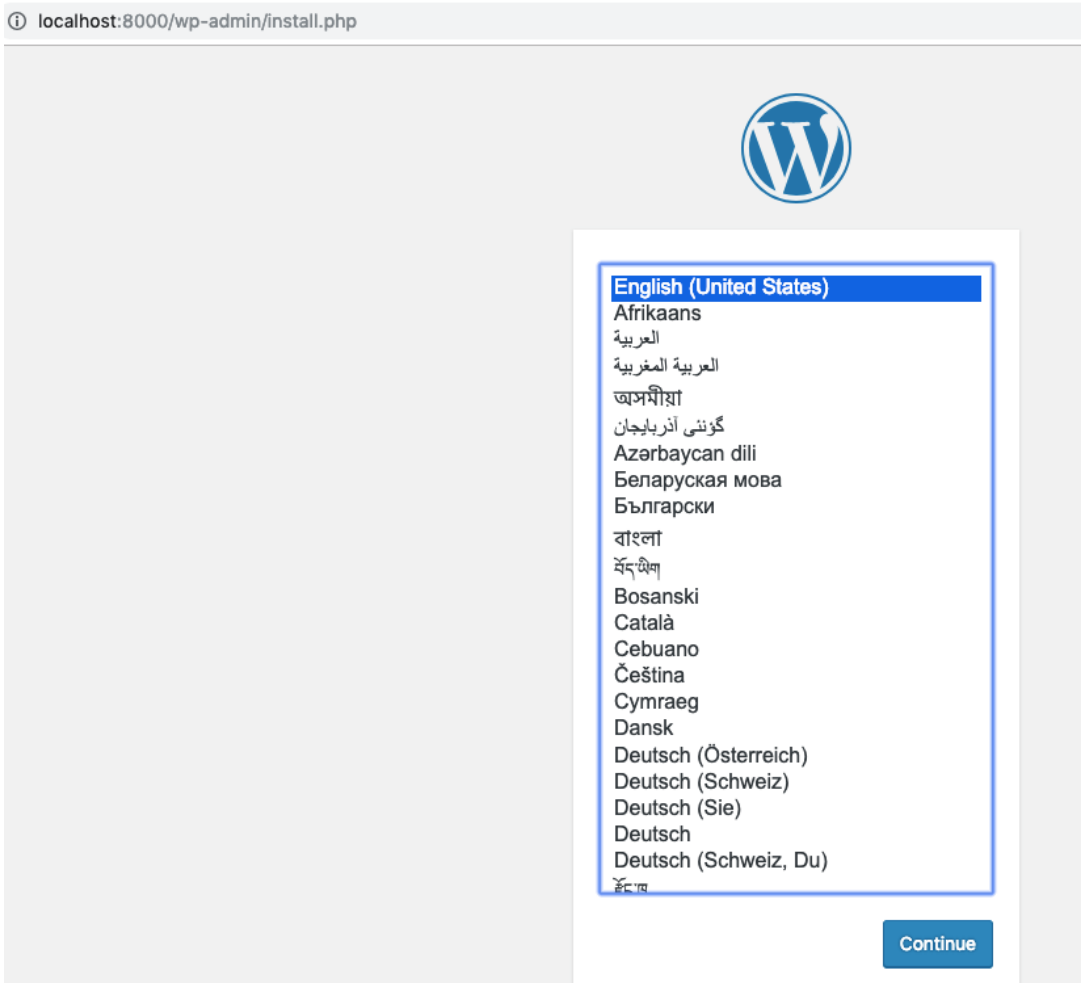
I A docker-compose.yml (yaml)

```yaml
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: {}
```

Composer and WordPress
https://docs.docker.com/compose/wordpress/

Example with Docker Composer and WordPress

```
18:09 $ docker-compose up -d
Creating network "wordpress_default" with the default driver
Creating volume "wordpress_db_data" with default driver
Pulling db (mysql:5.7)...
```



Example repository

https://github.com/Arquisoft/bddExample

Brach master -> Dockerfile

Brach docker-compose -> docker-compose.yml

Codefresh (https://codefresh.io/)

It is a platform to build and deploy Docker images

SaaS (Software as Service)

The free account limited to a single project and 120 builds per month.

It is possible to log in with a Github account.

Codefresh: Deploying from Github

It is possible to do everything from the web console

Directly from a Github repository

We need a Dockerfile

- This file will be in the repository
- Or we can create it when added the repo

https://codefresh.io/docs/docs/getting-started/create-a-basic-pipeline/

Other alternatives

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

Paid service “similar” to Docker

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

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

Deployment and automatic updates from a GitHub repository

Links

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

Tips

Force rebuild in docker-compose

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