



Universidad de Oviedo



SOFTWARE
ARCHITECTURE

Software Architecture

Lab. 06

Distribution & Deployment

2020-21

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Paulino Álvarez

GitHub Pages

GitHub supports creating websites

Useful for personal – project/repository

Branch **gh-pages**

GitHub Pages - examples

Organization level

Repository:

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

Deployed:

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

It can be very useful for personal web pages

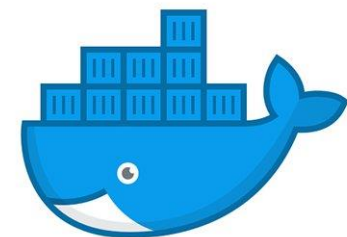
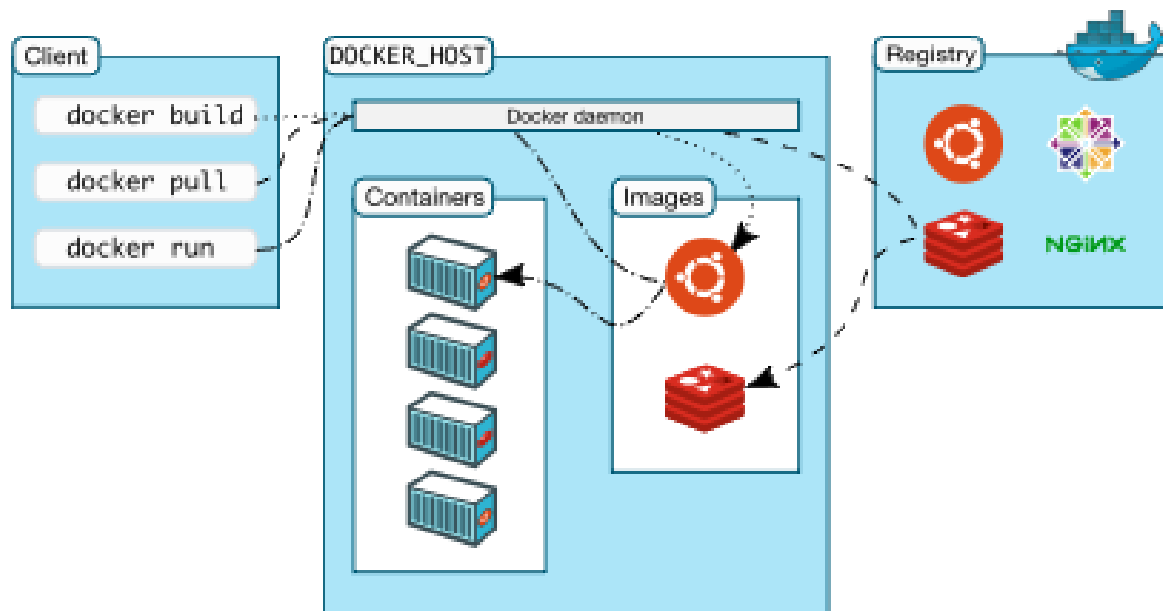
<http://pglez82.github.io>

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?

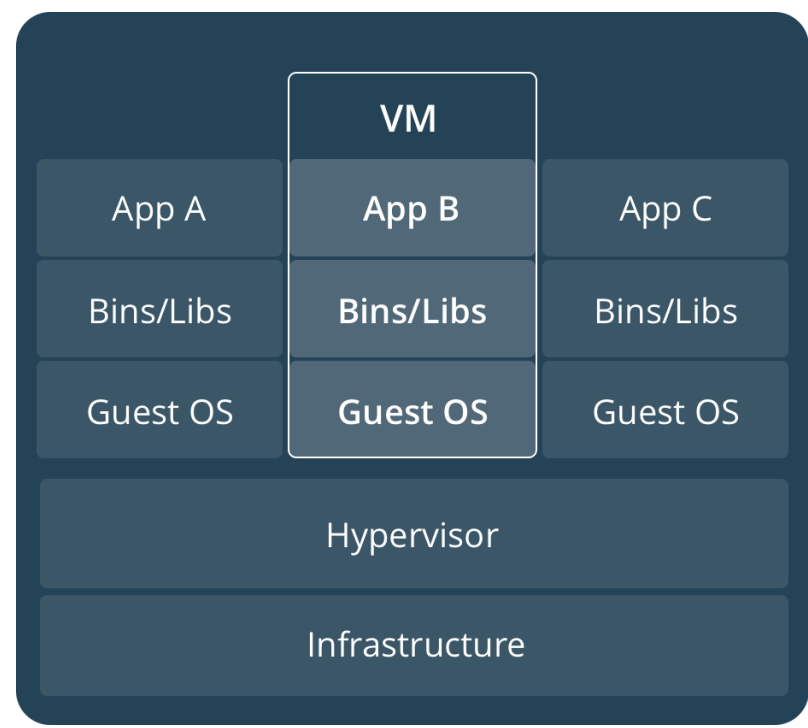
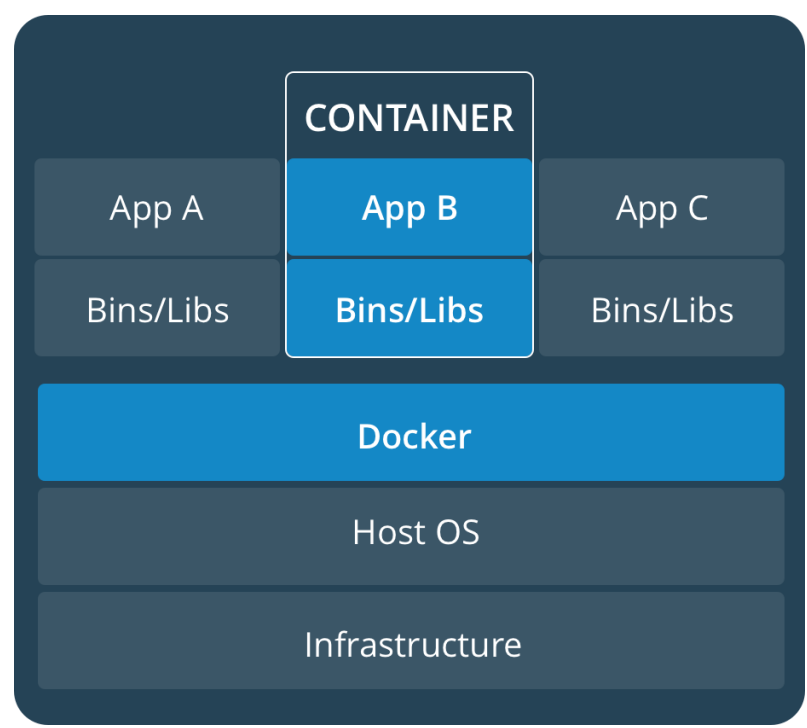
It is a live instance of an image

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>

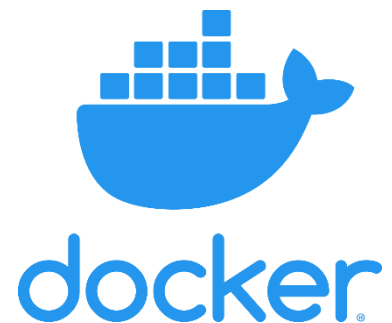
Obtaining docker

<https://www.docker.com>

Available for linux, windows and Mac

Docker desktop (Windows/Mac)

Docker ToolBox [faq#issue3](#)



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
```

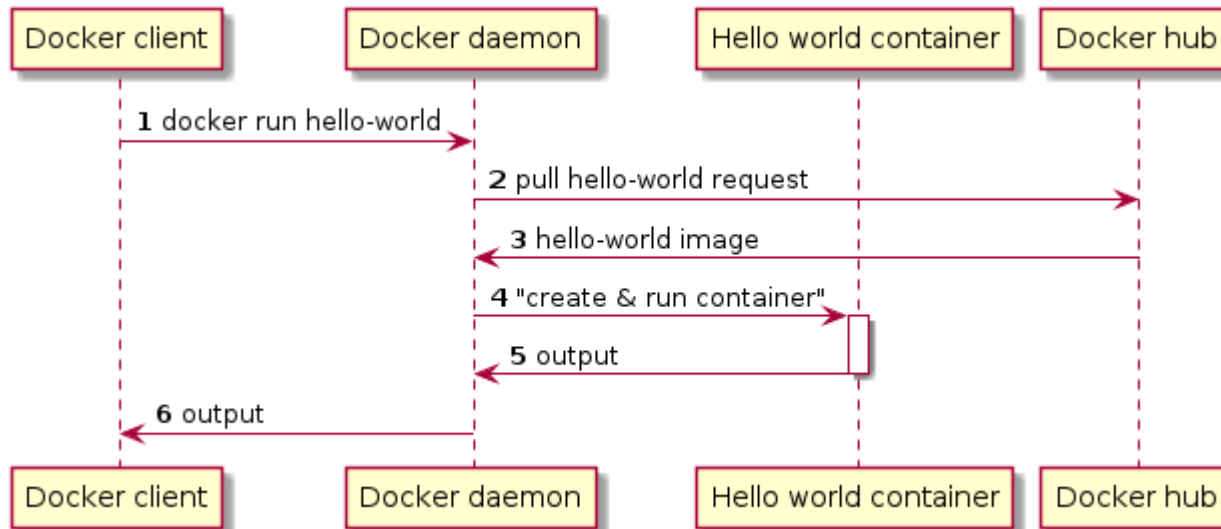
Docker step by step

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
```

localhost

Welcome to nginx!

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

Some commands

```
docker info
```

```
docker ps
```

```
docker image ls
```

```
docker container ls -all
```

```
docker pull
```

```
docker run
```

```
docker stop
```

```
docker rm
```

Example 1: Running solid locally

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>

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
```

Example 2:

Radarin sample webapp

https://github.com/Arquisoft/radarin_0/tree/master/webapp

```
FROM node:12.14.1
COPY . /app
WORKDIR /app
#Install the dependencies
RUN npm install --production
#Create an enviroment variable to set where the api is (check src/api/api.js).
#When we deploy to heroku this will take a different value. Check .github/workflows/radarin.yml
ARG API_URI="http://localhost:5000/api"
ENV REACT_APP_API_URI=$API_URI
#Create an optimized version of the webapp
RUN npm run build
#Install software necessary for generating the doc
RUN apt-get update && apt-get -y install ruby openjdk-8-jre
RUN gem install asciidoctor asciidoctor-diagram
#Generate the doc
RUN npm run docs
CMD [ "node", "server.js" ]
```

Example 3:

Radarin sample restapi

https://github.com/Arquisoft/radarin_0/tree/master/restapi

```
FROM node:12.14.1

# Create app directory
WORKDIR /usr/src/app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
COPY package*.json ./

#In this case, the mongodb in memory dependency is quite heavy so we avoid it for the docker image
RUN npm install --production

#By default this is the address of the mongo container. If we are deploying to heroku we will get
#a mongo_uri here direct from github secrets (check .github/workflow/radarin.yml)
ARG MONGO_URI="mongodb://mongoserver:27017/api"
ENV MONGO_URI=$MONGO_URI

# Bundle app source
COPY server.js api.js ./
ADD models ./models

CMD [ "node", "server.js" ]
```

Combining multiple docker containers

- 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
- [Radarin docker-compose file](#)

Running Docker compose

Configuration

- We can configure multiple services
- Each service can depend on others
- By default, all services share the same network and are accessible through their container name

Running

- For running (or stopping) a docker-compose file we execute: `docker-compose (up|down)`

Extra information

Small repository with all the basic commands used in docker:

https://github.com/pglez82/docker_cheatsheet

Tips

Force rebuild in docker-compose

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