

Creació d'imatges amb Docker

Institut de Ciències de l'Educació Josep Pallach (ICE)

Codi 0007190019

<http://www2.udg.edu/tabid/6126/Default.aspx?ID=1968>

INS Montilivi, Avinguda de Montilivi, 125, 17003 Girona

27, 28, 29 de Juny del 2017

Sessio 1

Introducció

Qui soc?



David Raba
Enginyer Informàtic

Master en Business Innovation
and Technology Management

CTO @ UBIKWA SYSTEMS



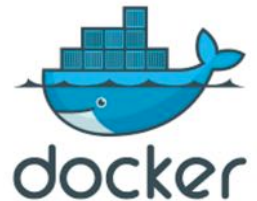
Backend Engineer - R&D - Full stack developer

Fliwer

Feb 2014 – Mar 2015 • 1 yr 2 mos

Barcelona Area, Spain

- Developing for a new start-up project: www.fliwer.com and www.fliwerpro.com
- Backend development: Artificial Intelligence and firmware development
- Data analysis and modeling layers
- Connected devices/Internet of Things/M2M
- C/C++ | Ruby | Apache | GIT | Puppet | Memcached | Gearman | Percona | HAProxy



R&D Analyst

Aplicaciones de Inteligencia Artificial (AIS - Barcelona)

Sep 2011 – Apr 2014 • 2 yrs 8 mos

Barcelona Area, Spain

- Data product creation
- Developing business reports, and data visualization
- Web scraping, Extract, manipulate, and transform data to draw actionable insights by using datamining and statistical modeling tools
- Prepare and deliver presentations to managers and executives
- Participate in ongoing decisions concerning data collection, study design, data analyses
- SAS | R | Ruby | Excel | Java



IT Architect / Developer

Netsuus Internet Intelligence

Jan 2009 – Aug 2011 • 2 yrs 8 mos

- Involved in all aspects of building and maintaining the production infrastructure and services
- Solid understanding of web analytics, e-commerce, and data analysis on clickstream from online traffic collection.
- Core development of an online benchmarking system based on traffic sniffing
- Startup experience
- Ruby on Rails | Ruby | HTML | CSS | C++ | Pound | Apache

Start-ups



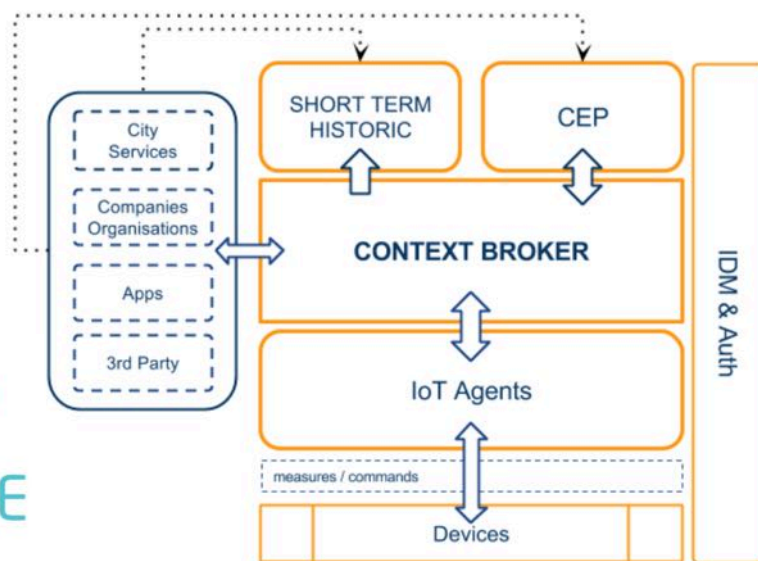
CTO & Development Manager

UBIKWA SYSTEMS SLU

May 2015 – Present • 2 yrs 2 mos

Barcelona Area, Spain

- To get a product out the door
- Deliver results to the market
- To make sure the development team is able to work as efficiently as possible and this means making sure they have clear goals
- From the initial project scope to deploying the product out to customer sites.
- And develop Android SDK | Golang | FIWARE | R | MongoDB | SQLServer | Ruby | C | Docker | Azure | Git.



26.6.2017



Start-ups

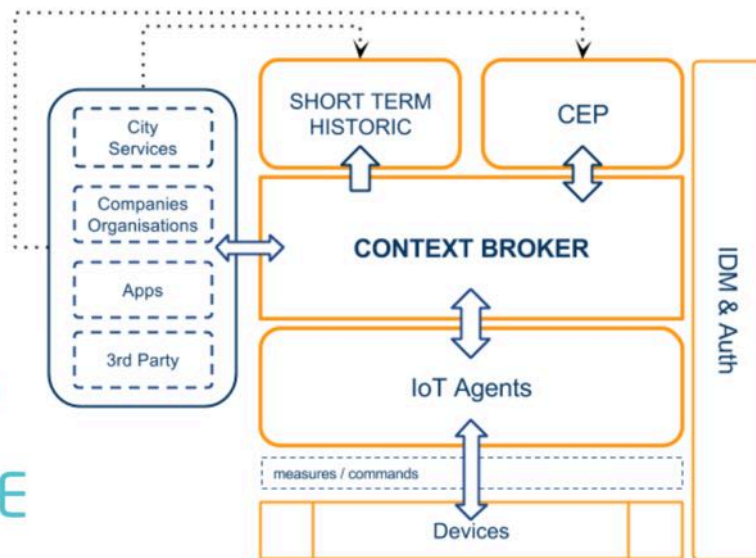


CTO & Development Manager

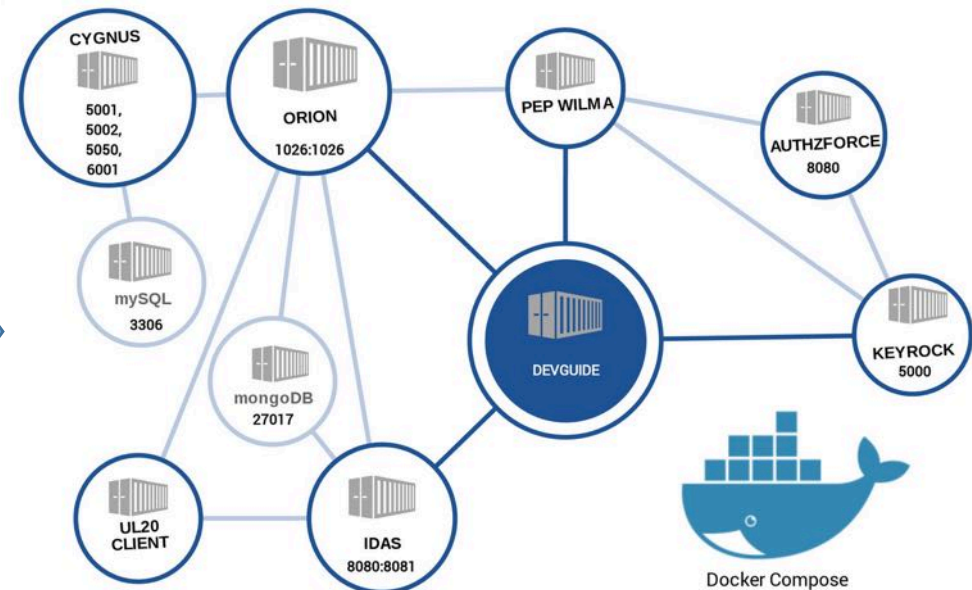
UBIKWA SYSTEMS SLU

May 2015 – Present • 2 yrs 2 mos
Barcelona Area, Spain

- To get a product out the door
- Deliver results to the market
- To make sure the development team is able to work as efficiently as possible and this means making sure they have clear goals
- From the initial project scope to deploying the product out to customer sites.
- And develop Android SDK | Golang | FIWARE | R | MongoDB | SQLServer | Ruby | C | Docker | Azure | Git.



26.6.2017





- Conèixer les principals característiques i funcionalitats de Docker
 - Autogestió dels contenidors.
 - Fiabilitat.
 - Aplicacions lliures de les dependències instal·lades al sistema amfitrió.
 - Capacitat per desplegar multitud de contenidors en un mateix equip físic.
 - Posada en marxa dels serveis amb major rapidesa.
 - Contenidors molt lleugers que faciliten el seu emmagatzematge, transport i desplegament.
 - Capacitat per executar una àmplia gamma d'aplicacions.
 - Compatibilitat Multi-Sistema (podrem desplegar els nostres contenidors en multitud de plataformes).
 - L'aplicació base de Docker gestionarà els recursos existents per assignar-responsablement entre els contenidors desplegats.
 - Podrem establir una base des de la qual començar els nostres projectes, el que ens estalviarà el temps de preparar l'entorn per a cada un d'ells.
 - Podrem compartir els nostres contenidors per augmentar els repositoris de Docker així com beneficiar-nos dels que comparteixin els altres.



- **Sessió 1** Instalar Docker Engine. Crear contenidors Docker. Emmagatzemar i descarregar imatges Docker.
- **Sessió 2** Arrancar contenidors desde imatges. Desplegar aplicacions amb Docker
- **Sessió 3** Interconectar contenidors Docker.
- **Activitat** Deploy de un sistema web amb balanceig de càrrega i base de dades

Agenda del dia... Dilluns 27 Juny



- 09:00 - 09:20 AM – Presentacions
- 09:20 – 11:00 AM - Introducció a Docker
- 11:00 - 11:30 AM – Descans
- 11:15 AM - 2:00 PM - Instalar Docker Engine. Crear contenidors Docker. Emmagatzemar i descarregar imatges Docker

Containers

Què son els containers?



- Abans dels standards dels containers



Què son els containers?



- I es van inventar els containers...

Al 1956 la majoria de vaixells eres carregats i descarregats a ma amb un cost de **\$5.86** la tona.

Malcom McLean va neixer al 1913 i va desenvolupar el sistema de container intermodal, que va revolucionar el transport internacional.

El su principi deia que *“Un vaixell només guanya diners quan està navegant”*

Utilitzant containers, els seus costs eren de de **16 centims la tona**. La containerització va estandaritzar els processos de càrrega, reduint dràsticament el temps i el cost aplicat.

A més, de dotar de fiabilitat al sistema.



https://en.wikipedia.org/wiki/Malcom_McLean

Què son els containers?



- Omplir el container



Què son els containers?



- Allotjat el container





docker

Característiques de Docker



- Light-Weight

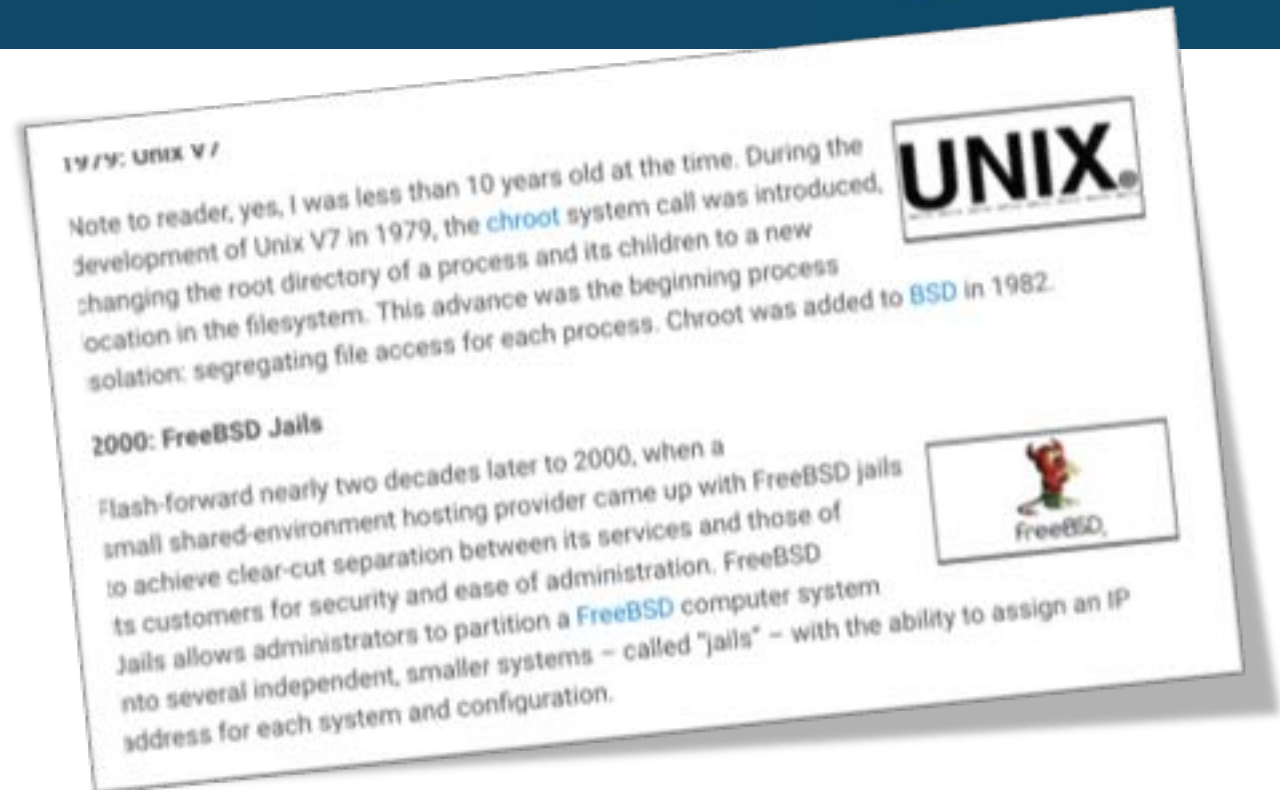
- Mínima empremta (*cpu/io/network*)
- Basats en containers linux
- Utilitza un sistema de fitxers multicapaper aprofitar espai (AUFS/LVM/OverlayFS)
- Utilitza una estratègia copy-on-write per fer seguiment de canvis

- Portable

- Pot funcionar en qualsevol Linux que doni suport a LXC.
- La release 0.7 inclou suport per les distros RedHat/Fedora.
- Raspberry pi support.
- Planificades altres eines de distribució (lmctfy, etc.)
- Planificats suports per altres sistemes operatius (Solaris, OSX, Windows)

- Autocontingut

- Un contaoner Docker conté tot lo necessari per funcionar
- És una base mínima del SO
- Llibreries i frameworks
- Codi d'aplicació
- Un container docker ha de ser capaç de funcionar a qualsevol lloc on Docker pugui.



<http://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>



<http://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>

Historia



1979: UNIX V7

2001: Linux VServer

2008: LXC

LXC (Linux Containers) was the first, most complete implementation of Linux container manager. It was implemented in 2008 using cgroups and Linux namespaces, and it works on a single Linux kernel without requiring any patches.

2011: Warden

CloudFoundry started Warden in 2011, using LXC in the early stages and later replacing it with its own implementation. Warden can isolate environments on any operating system, running as a daemon and providing an API for container management. It developed a client-server model to manage a collection of containers across multiple hosts, and Warden includes a service to manage cgroups, namespaces and the process life cycle.

During the time, the process was introduced, to a new process was added to BSD in 1982.

FreeBSD jails those of the BSD computer system with the ability to assign an IP

The LXC logo, featuring the letters "LXC" in a stylized, blue, diamond-shaped font.The Linux VServer logo, featuring the text "Linux VServer" with a green "V" and a small illustration of people.The ORACLE SOLARIS logo, featuring the word "ORACLE" in red and "SOLARIS" in black.The OpenVZ logo, featuring a green circular icon with a white "V" and the text "OpenVZ".The UNIX logo, featuring the word "UNIX" in a bold, black, sans-serif font.The FreeBSD logo, featuring a red dragon and the text "FreeBSD".

<http://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>



2013: Docker and the Future

That's my (not so brief) summation of the pre-Docker container landscape. All those iterations had their adopters and devotees, but when Docker emerged in 2013, containers exploded in popularity. It's no coincidence the growth of Docker and container use goes hand-in-hand.

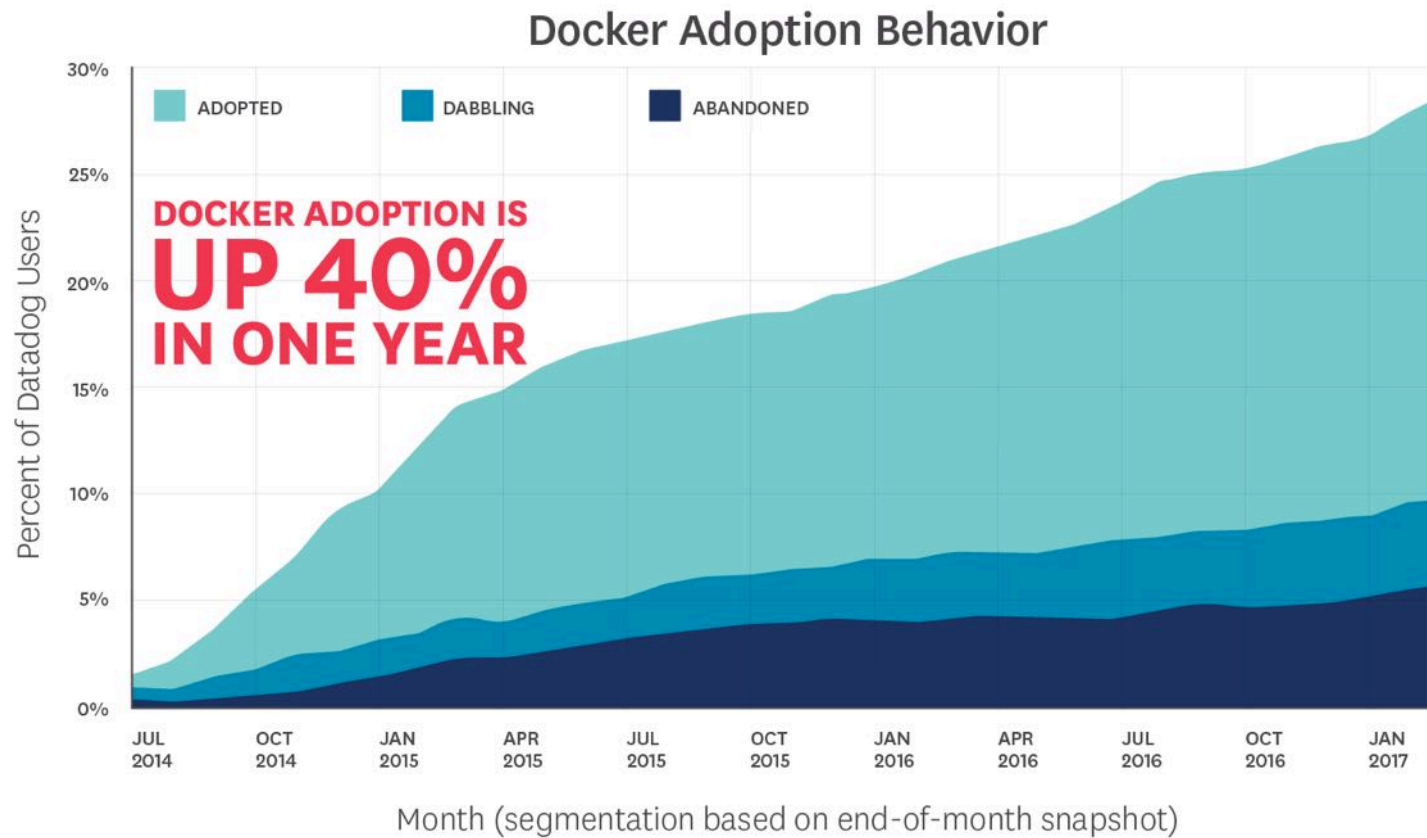
Just as Warden did, Docker also used LXC in its initial stages and later replaced that container manager with its own library, libcontainer. But I've no doubt that Docker separated itself from the pack by offering an entire ecosystem for container management.

With Docker, developers can create and run application containers quickly. And with the release of Docker Hub, developers can download and run application containers even faster.

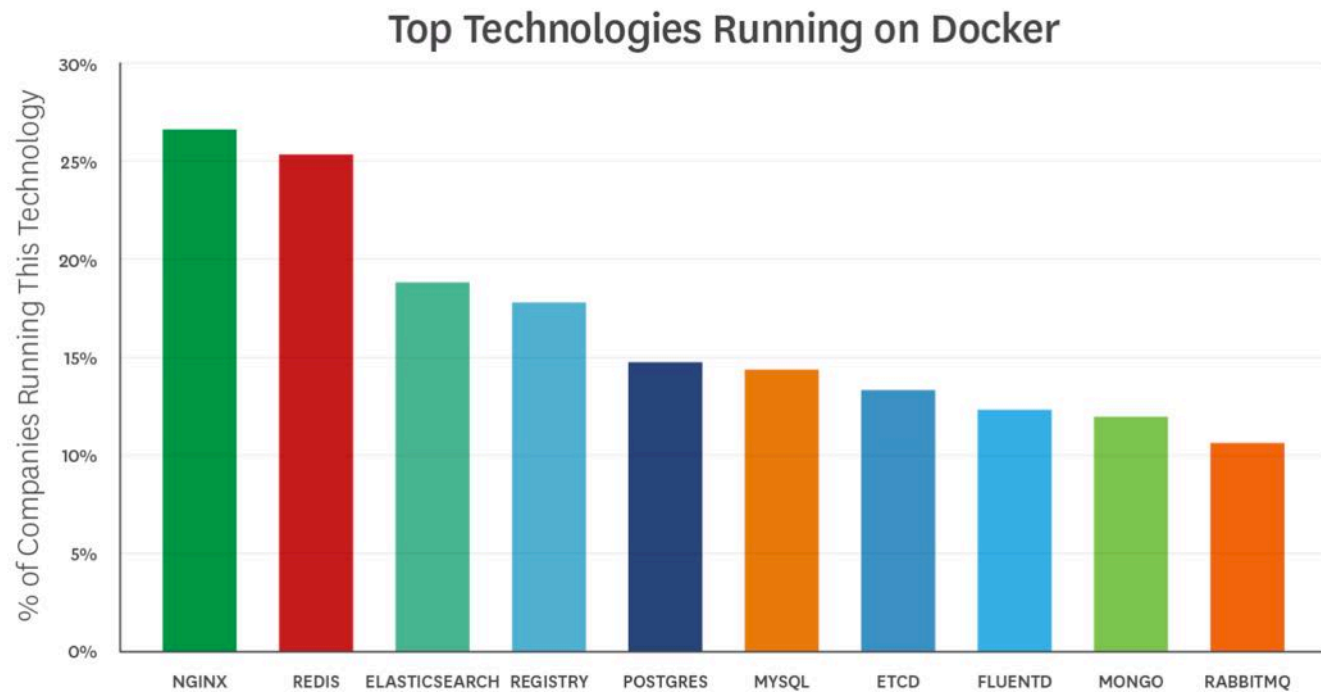
Docker 0.1.0 released March 25, 2013

<http://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>

Utilització de docker

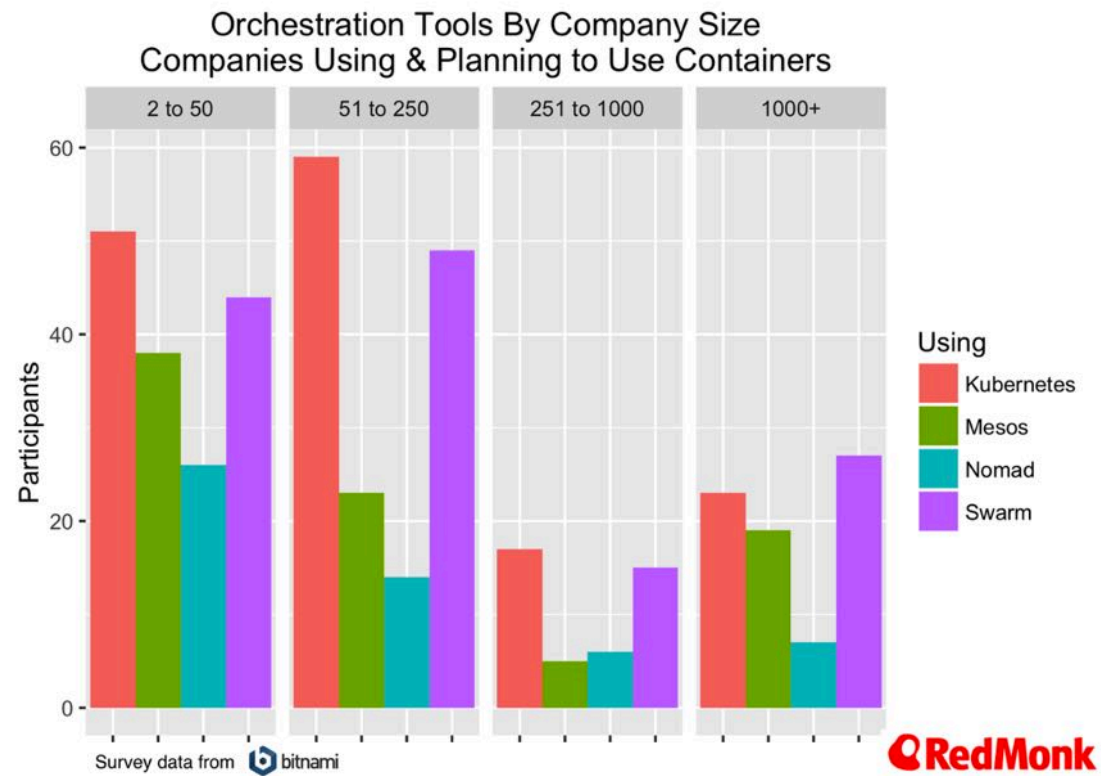


Utilització de docker



Source: Datadog

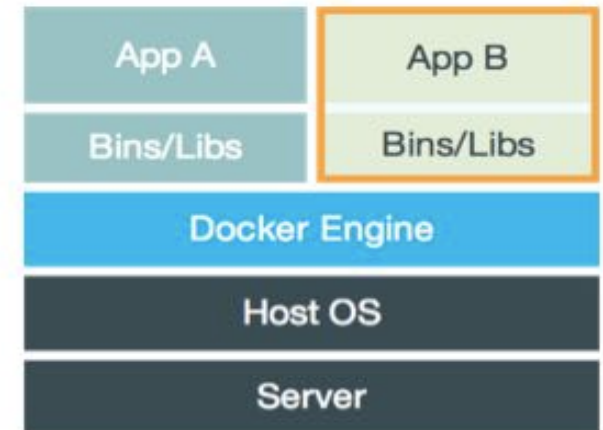
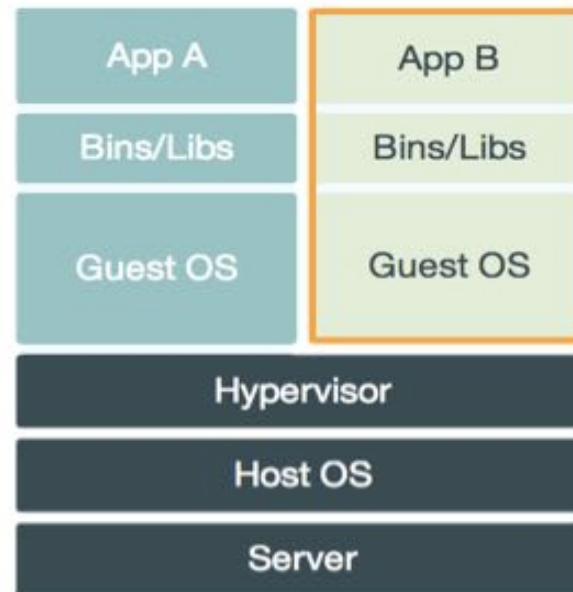
Utilització de docker



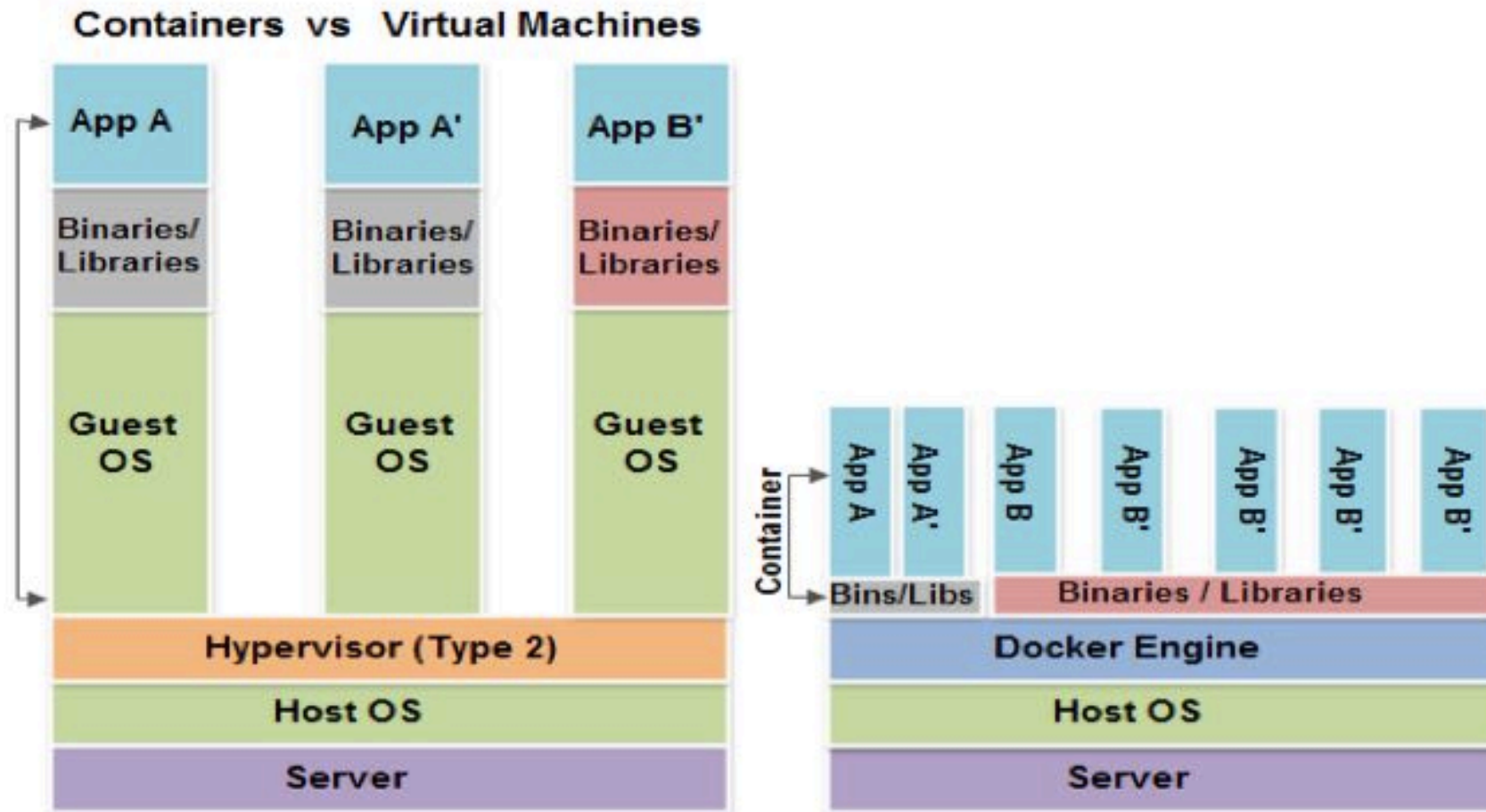
Docker vs Virtual Machines



- VM's
 - Instància completa del sistema Operatiu
 - No és fàcil la multi-instància
- Containers
 - Porcions del SO + components
 - Fàcil de duplicar, iniciar i aturar
 - OS Lleuger (Tiny Core Linux & Windows Server Core)



Docker vs Virtual Machines



Com de petits son els containers?



- ~24MB de descarrega
- S'executa complet a RAM
- Requirements mínims:
 - 46MB RAM
 - i486DX CPU (50MHz, 8KB cache)
- Requirements recomanats:
 - 128MB+ RAM
 - Pentium 2 CPU
- Ho tenim corrent amb una raspberry



Requeriments de host



- Requirements mínims:
 - 256-512MB of RAM
 - 1GHz (x86) or 1.4Ghz (x64)CPU
- Recomanats
 - 512MB+
 - 2GHz+ CPU

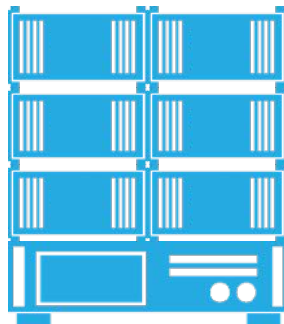
On els podem posar?



Locally with:

- Docker Toolbox (Linux)
- Hyper V (Windows)

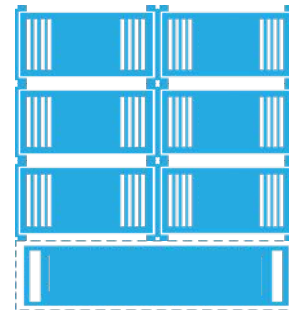
Locally



Physical Servers

- Linux (Linux)
- Windows 2016 TP3 (Windows)

On Premises



Clouds

- Azure (Linux & Windows)
- Digital Ocean (Linux)
- AWS (Linux)
- Google (Linux)
- Rackspace (Linux)
- ...etc.

Azure



Service Provider

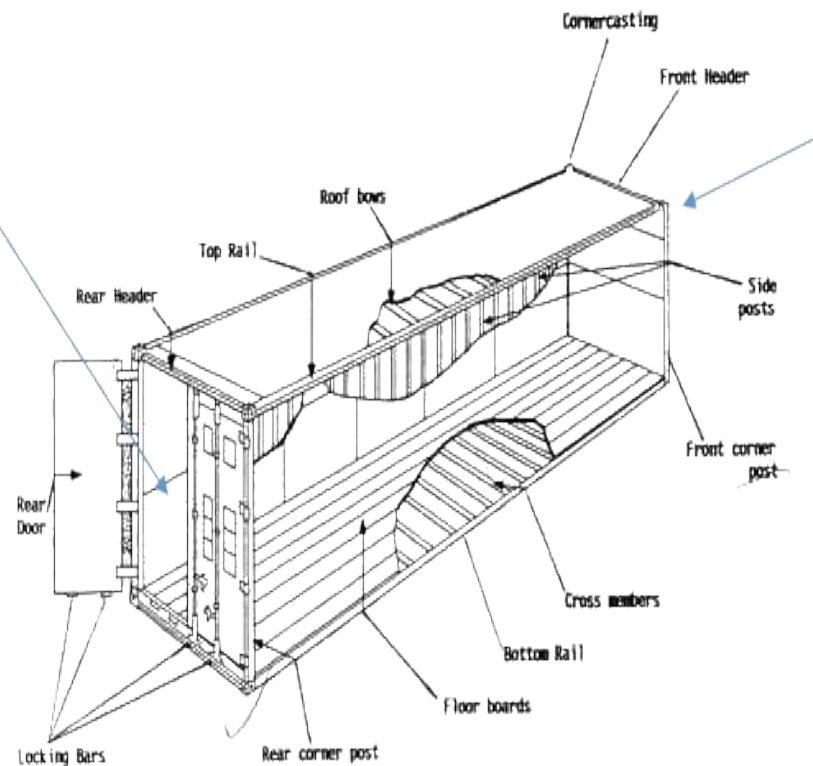


Què en treiem de bo?



• Dan the Developer

- Worries about what's "inside" the container
 - His code
 - His Libraries
 - His Package Manager
 - His Apps
 - His Data
- All Linux servers look the same



Major components of the container:

• Oscar the Ops Guy

- Worries about what's "outside" the container
 - Logging
 - Remote access
 - Monitoring
 - Network config
- All containers start, stop, copy, attach, migrate, etc. the same way

Què en treiem de bo?



Developers

Enable 'write-once, run-anywhere' apps
Enables microservice architectures
Great for dev/test of apps and services
(thousands available from Docker)
Reproducibility



Operations

Portability, Portability, Portability
Standardized development, QA, and prod
environments
Abstract differences in OS distributions and
underlying infrastructure
Easily scale-up and scale-down in response
to changing business needs

DevOps



Modern DevOps

1x every 2 weeks to 100x a day
MTTR 12x faster
Amazon deploys every 11.6 seconds

Què en treiem de bo?

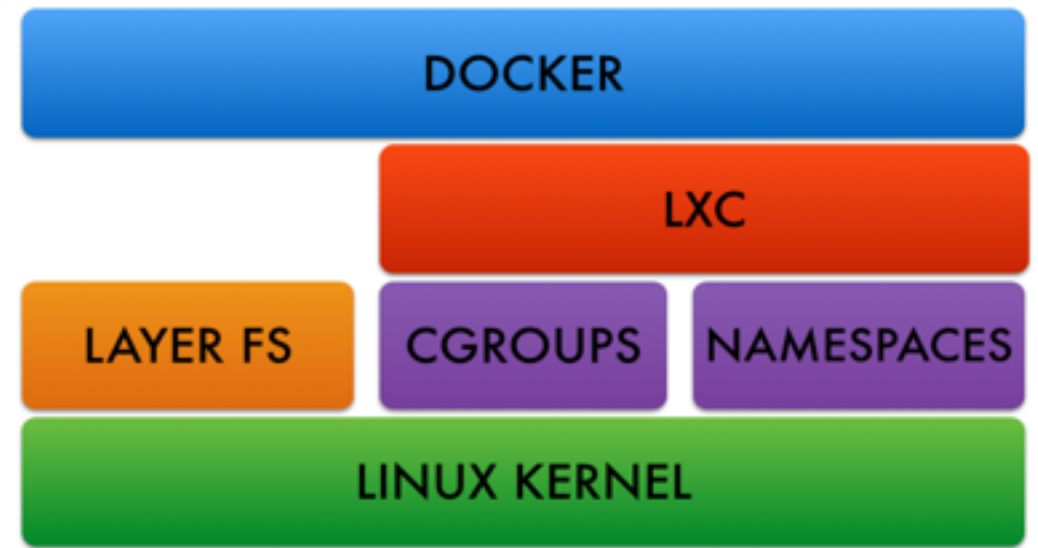


- La infraestructura es converteix en immutable
- Inici ràpid
- Portable & lleuger
- Tenim una unitat de deploy
- Fàcil creació
- Cada container por ser una porció de la gran aplicació
 - Podem tenir múltiples containers que serveixes a més d'una aplicació (microserveis)

Terminologia



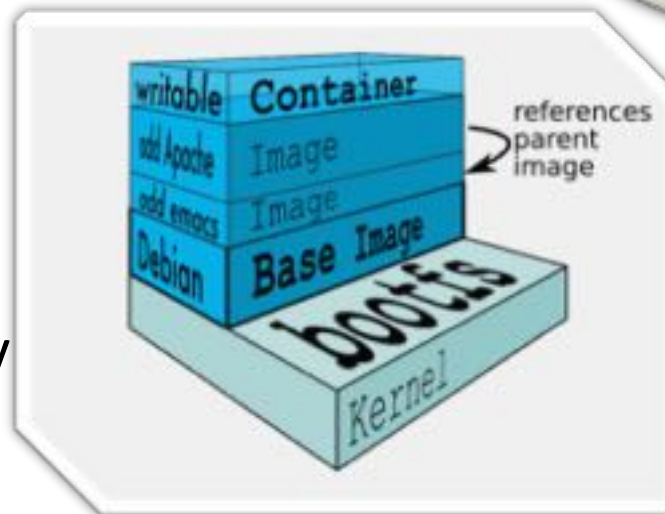
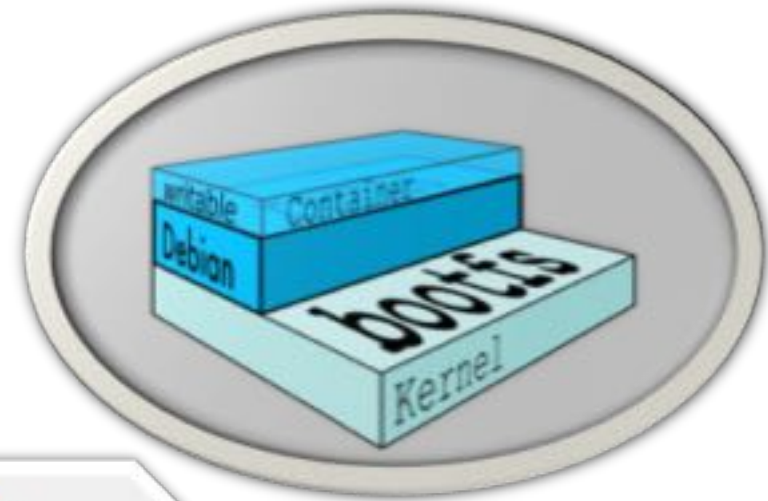
- Docker Engine
 - CLI
 - Docker Daemon
 - Docker Registry
- Docker Hub
 - Cloud service
 - Compartir Aplicacions
 - Automatitzar workflows
 - Composar apps a partir de components
- Docker images
- Docker containers



Docker images



- No es un VHD
- No es un FILESYSTEM
- Fa ús de [Union File System](#)
- És read-only [Layer](#)
- No té estat
- Basicament un fitxer tar
- Te jerarquia
 - de profunditat arbitrària
- Pot encabir-se al Docker Registry

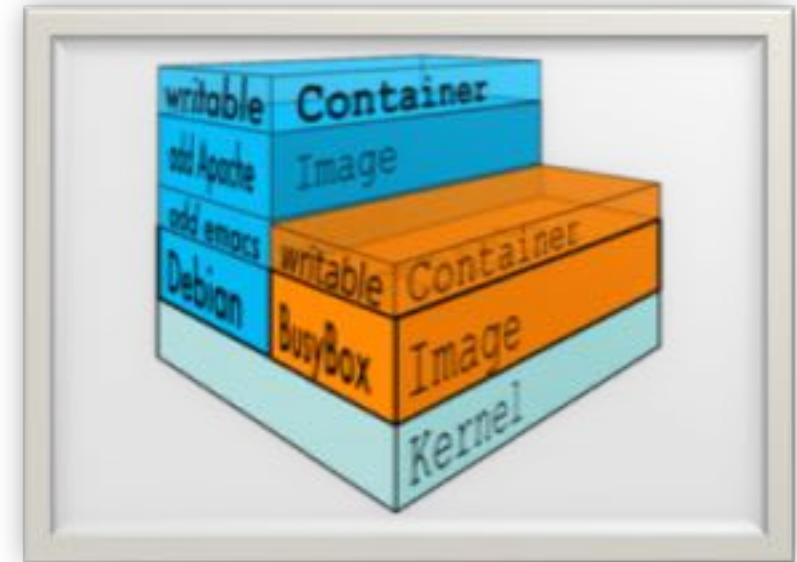


Docker container



És la unitat bàsica d'entrega de software (ship it!)

- S'executa a tot arreu
 - independentment de la versió de kernel
 - o de la distro del host
 - * però han de coincidir amb arquitectura (x32, x64, etc)
 - Ho executa tot
 - si pot correr a un host, ho pot fer dins d'un container
- *A menys que es facin servir emuladors de CPU amb **qemu** i **binfmt**



Docker CLI



\$ docker images	# mostra tots els images.
\$ docker import	# crea una imatge des d'un arxiu tar.
\$ docker build	# crea una imatge de d'un Dockerfile.
\$ docker commit	# crea una image des de un container.
\$ docker rmi	# eliina una image.
\$ docker history	# llista els canvis d'una imatge.

Cicle de vida d'un container



La vida d'un container...

- Concepció
 - **BUILD** una imatge a partir d'un Dockerfile
- Neixement
 - **RUN** (create+start) un container
- Reprocció
 - **COMMIT** (persisteix) un container a una imatge
 - **RUN** un nou container des d'una imatge
- Sleep
 - **KILL** un container en execució
- Wake
 - **START** un container aturat
- Death
 - **RM** (delete) un container aturat
- Extinció
 - **RMI** una imatge de container (delete image)

Dockerfile



- Conceptualment és un Makefile
- Extend una imatge base
- Que es converteix en una nova imatge
- Imperatiu, no Declaratiu

Defineix la recepta per construir una imatge

- S'utilitza docker build per executar un dockerfile
- Es poden definir ordres per defecte per executar, definir ports exposats, etc.

26.6.2017

```
# our base image
FROM alpine:latest

# Install python and pip
RUN apk add --update py-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

# tell the port number the container should expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]

Macbook:flask-app draba$ cat Dockerfile

# our base image
FROM alpine:latest

# Install python and pip
RUN apk add --update py-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
```


Preguntas típicas

5 preguntas típicas sobre Docker



1. Docker client versus host

- Docker client is a command line interface (CLI) Docker
- Docker host is a Linux/Windows VM running Docker daemon

2. Docker Linux and Windows hosts

- You can only create the same container as the underlying host VM - Linux host = Linux containers

3. Docker Image vs Docker Container

- Image = The definition – literally a single file [My Website]
- Container – An instance of an image [3 instances of My Website]

4. Cloud Registry Service and Public Image Repos

- Unlimited public repos, one free private or buy private repos
- 50,000+ images - Wordpress, Nginx, Redis, MySQL, Logstash, and your images!
- Docker Trusted Registry – Dedicated registry application deployable on-premise or direct from Azure Marketplace

5 common Docker questions



5. Deployments **replace** instead of **update**

“Immutable infrastructure”

Website

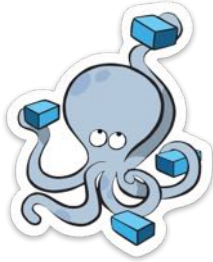
Update your app using
Web Deploy or CI/CD

Docker

Replace running
containers using CI,
Don't update the old
container

Eines complementaries

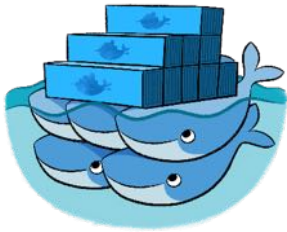
Altres conceptes Docker



Compose

Definir i fer deploy d'aplicacions multicontainer

Sessió 2 i 3



Swarm

Ús de múltiples màquines com a una de sola, per controlar múltiples entorns de containers.



Docker Machine

Crear i gestionar instàncies Docker localment i en cloud

Gràcies!

<http://www.insylo.com>

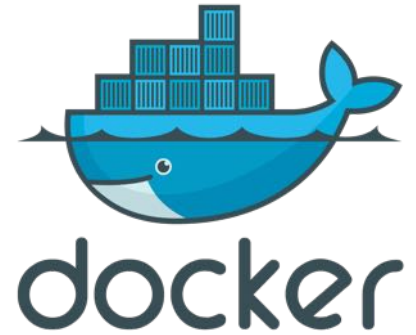
DAVID RABA

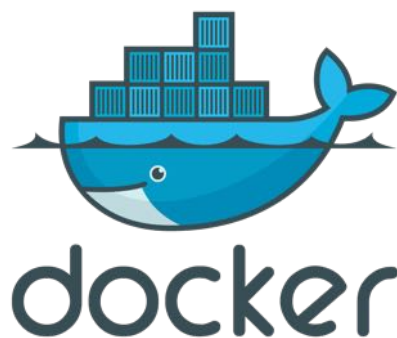


@draba



<https://es.linkedin.com/in/davidraba>





Creació d'imatges amb Docker

Institut de Ciències de l'Educació Josep Pallach (ICE)

Codi 0007190019

<http://www2.udg.edu/tabid/6126/Default.aspx?ID=1968>

INS Montilivi, Avinguda de Montilivi, 125, 17003 Girona

27, 28, 29 de Juny del 2017

1

2

3