




Ansible i Docker





Qui sóc jo?

Sergi Barroso

Cloud systems developer @  CLOUDREACH





Ansible





Què és Ansible?

- ◇ Gestor de configuracions
- ◇ Aprovisionador/Orquestrador





Què ens aporta?

- ◇ Implementació multiplataforma
- ◇ Pensar en “estats”
- ◇ Seguretat en les accions
- ◇ Capacitat de idempotència





Conceptes

- ◇ Inventory
 - Dinàmic o estàtic
 - Grups
- ◇ Tasks
- ◇ Handler
- ◇ Variables i Facts
- ◇ Playbooks
- ◇ Roles





Inventari

Definim grups de hosts

```
[webservers-europe]  
foo1.example.com  
bar1.example.com
```

```
[webservers-asia]  
foo2.example.com  
bar2.example.com
```

```
[dbservers-europe]  
db-e1.example.com  
db-e2.example.com
```

```
[dbservers-asia]  
db-a1.example.com  
db-a2.example.com
```

Definim súper grups

```
[asia:children]  
webservers-asia  
dbservers-asia
```

```
[europe:children]  
webservers-europe  
dbservers-europe
```





Tasques

Comandes que executarem en els servidors:

- ◇ Comandes de shell
- ◇ Mòduls de Ansible
- ◇ Mòduls de 3ers

Llistat de mòduls de Ansible:

http://docs.ansible.com/ansible/modules_by_category.html





Handler

- ◇ Son tasques que es criden recursivament

Variables i facts

- ◇ Emmagatzemament clau:valor





Rols

- ◇ Agrupació de tasques, variables, handlers, plantilles i fitxers destinades a un mateix propòsit (p.e. "Servidor web")
- ◇ Els rols segueixen una estructura de directoris determinada





Playbooks

- ◇ Sistema de gestió de configuracions
 - Ordenar l'execució de tasques
- ◇ Gestió de grup de servidors
 - Infraestructura com a codi
- ◇ Utiliza llenguatge YAML





Demo time!





Debugging

Tècniques per millorar la qualitat dels nostres playbooks:

- ◇ --step
- ◇ --start-at-task "nom"
- ◇ -vv
- ◇ -check (dry run)
- ◇ Tasca "debug"





Pros:

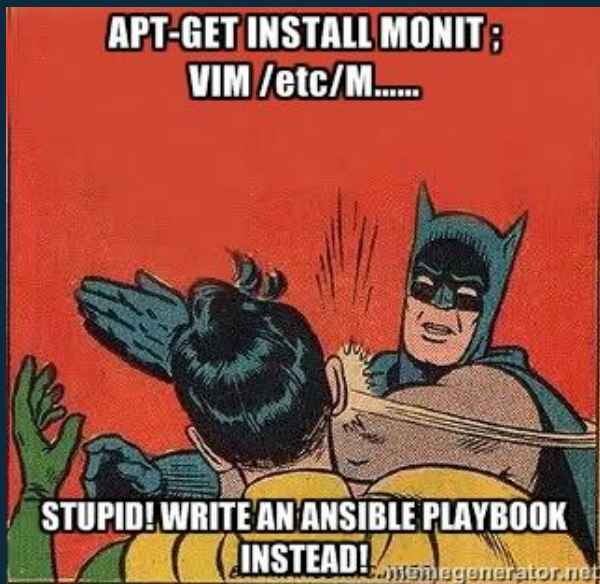
- ◇ Corba d'aprenentatge
- ◇ Agentless
- ◇ Yaml
- ◇ Python
- ◇ Plantilles Jinja
- ◇ Suport de la comunitat

Cons:

- ◇ Dependència de Python en el servidor destí

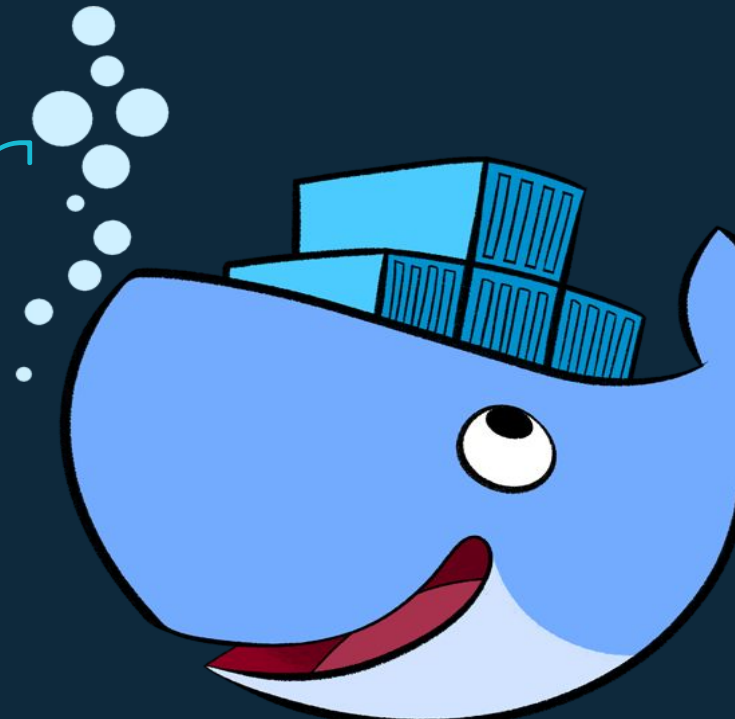


APM?





Docker





Què és Docker?

- ◇ Programari lliure que ens permet empaquetar una aplicació i totes les seves dependències en un contenidor virtual capaç de funcionar en un sistema operatiu Linux.
- ◇ Una plataforma dissenyada per cobrir necessitats de desenvolupadors/devops.
- ◇ Sistema que ens permet crear una arquitectura immutable.

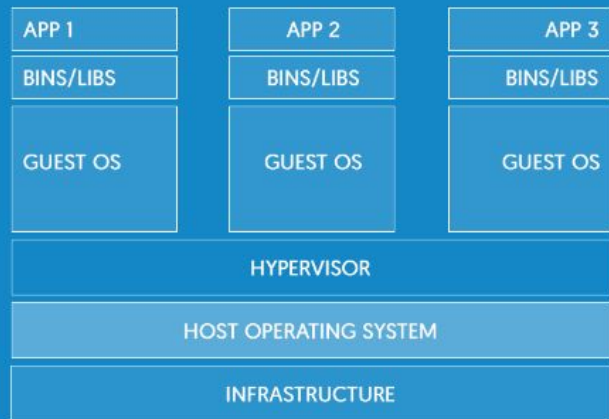
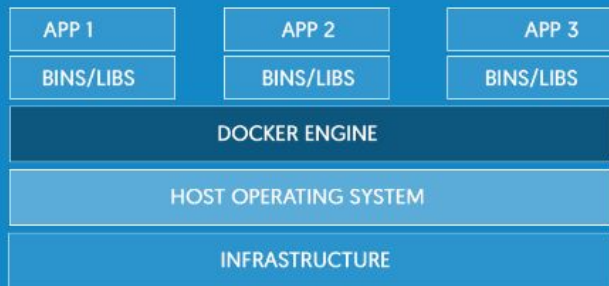
Què no és:

- ◇ Una plataforma de virtualització completa





Containers vs VM



Pros:

- ◆ No hi ha guest OS ni hipervisor

Contres:

- ◆ Menys aïllament que en un hipervisor



Com funciona?

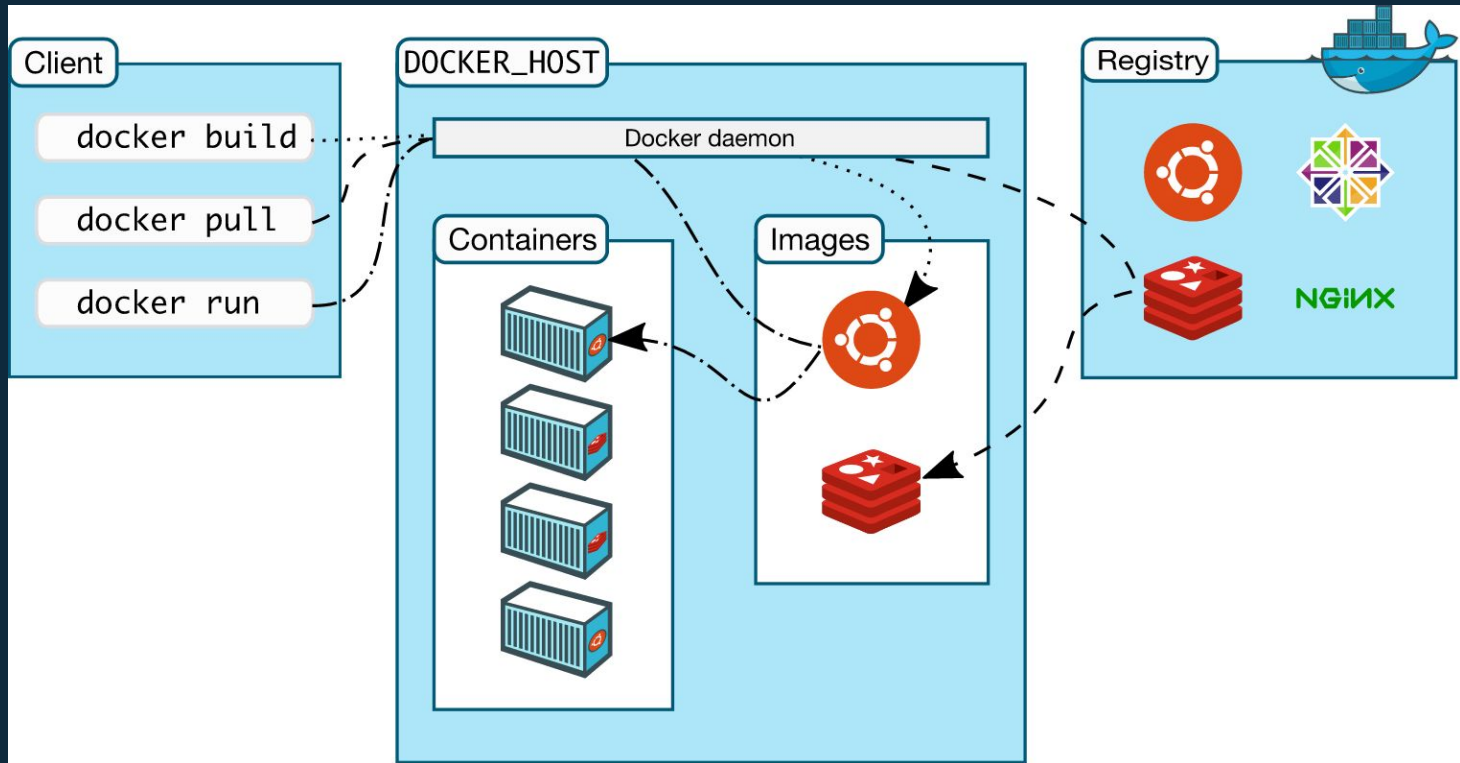
Principal components components:

- ◇ Docker File (conjunt d'instruccions per generar una imatge)
- ◇ Docker Image (plantilla de màquina virtual)
- ◇ Docker container (màquina virtual)
- ◇ Docker Registry (genera màquines virtuals a través de plantilles)

Tecnología del Linux utilitzades:

- ◇ namespaces: pid, net, ipc, mnt i uts isolation
- ◇ cgroups: permet compartir i limitar els recursos de hardware
- ◇ UnionFS: sistema de fitxers per capes per generar imatges de Docker

Com funciona?





Creació de imatges

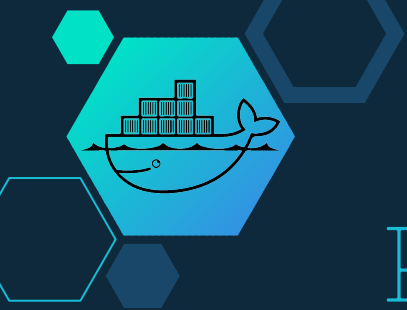




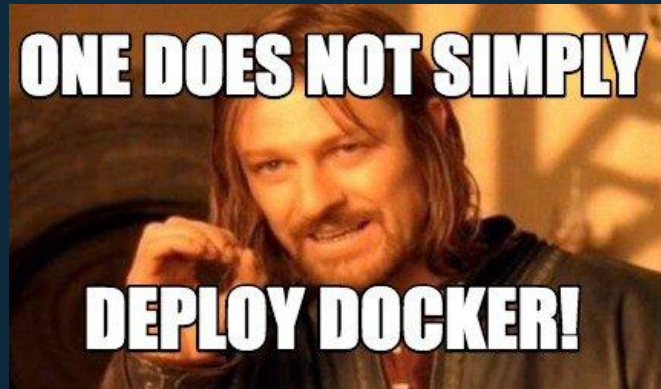
Demo time!

DOCKER





Problemes!



- ◇ Imatges de Docker gegants
 - Intentar que les imatges no sobrepassin els 150MB
 - Utilitzar busybox quan sigui possible
 - En cas contrari utilitzar p.e. Alpine en comptes de Debian
- ◇ Seguretat de les imatges
 - No utilitzar imatges no oficials
- ◇ Imatges monolítiques
 - Microserveis per la mor de Deu!!!
- ◇ SSH per entrar al contenidor
 - Si has de fer servir ssh en una imatge es que alguna cosa va amalament



Moltes gràcies!

Em podeu trobar a:



@hiroru



hiroru@lionclan.org



hiroru

