

# Linux containers

chroot con gli steroidi

*a cura di Enrico Testori*



# OS-level virtualization

L'idea di base è sfruttare le funzionalità offerte dal kernel per creare istanze di “user spaces” multiple anziché una sola.



# OS-level virtualization

Su Linux questo meccanismo può essere visto come un chroot molto avanzato a cui sono aggiunte funzionalità come l'isolamento e il controllo delle risorse.



# Uno per tutti e tutti per uno!

Linux Containers non è il nome di una tecnologia, ma un insieme di più tecnologie

- Namespaces
  - Cgroups
  - Chroots



# Uno per tutti e tutti per uno!

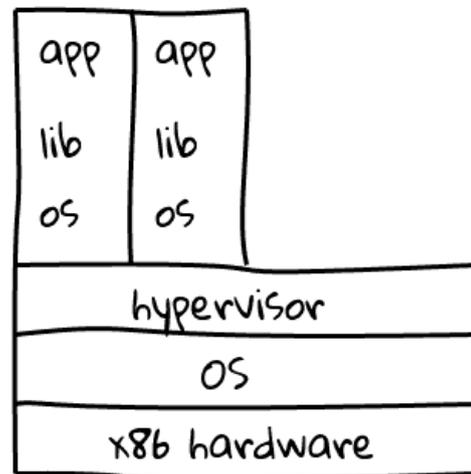
traditional  
virtualization

x86 hardware

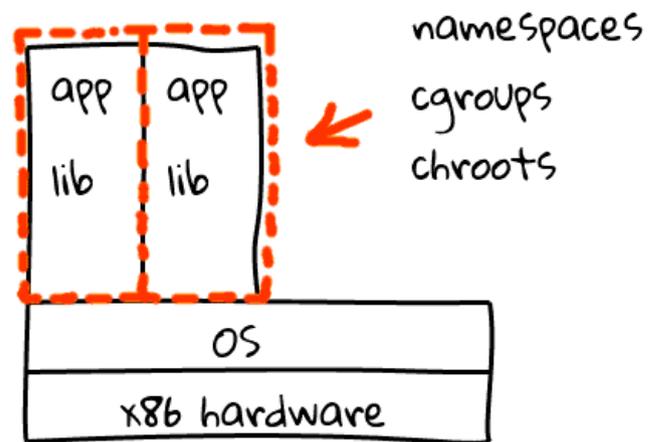


# Uno per tutti e tutti per uno!

traditional  
virtualization



containers



# Chroot

una operazione che consiste nel cambiare la directory di riferimento dei processi che sono in esecuzione (e dei loro figli)



# Chroot

- Testing
- Compatibilità
- Isolamento dipendenze
  - Recovery
  - Sicurezza?



# Namespaces

Simile a chroot, ma viene applicato a:

- Punti di mount
  - Utenti
  - Processi



# Cgroups

Definisce e controlla il modo in cui i “container” consumano risorse:

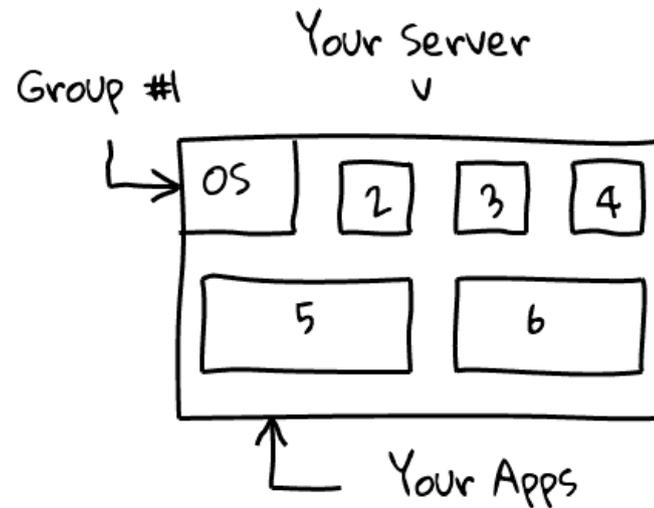
- CPU
- Disk IO
- Network
- Memoria
- Accounting



# Cgroups - Overview

Control Groups (cgroups)

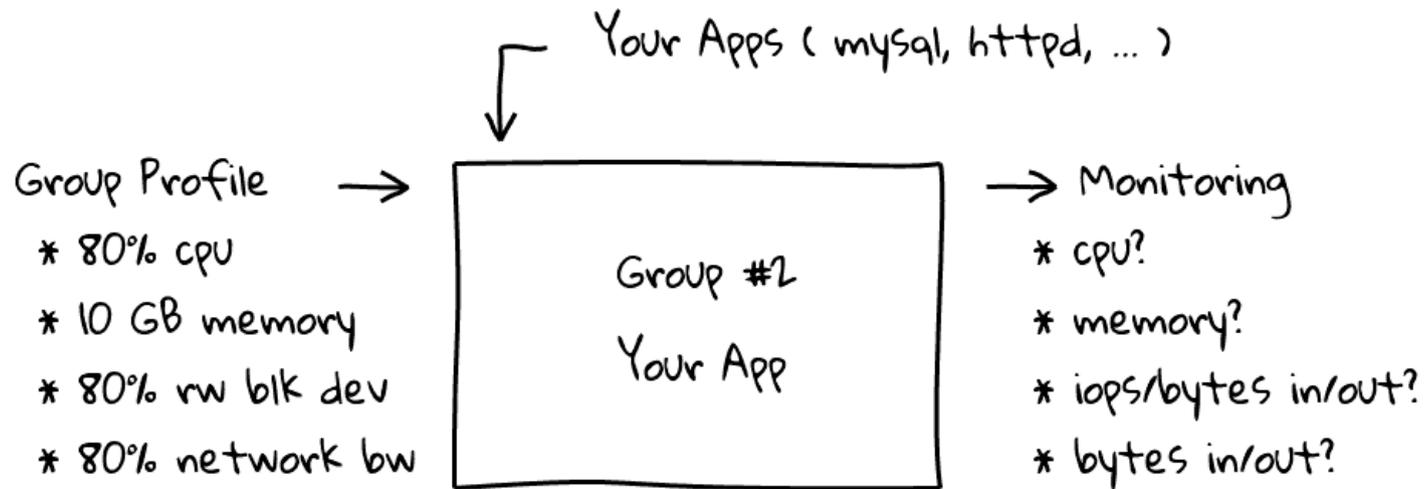
- \* CPU time
- \* System memory
- \* Disk bandwidth
- \* Network bandwidth
- \* Monitoring



# Cgroups - Apps



# Cgroups - Apps





LugAnegA Linux User Group  
Linux Day 2014

# LXC – Realtà o fantascienza?

L'idea che sta alla base di tutto esiste da più di trent'anni ma non è molto famosa anche a causa delle ambiguità che si porta dietro.



# LXC – Realtà o fantascienza?

La usano parecchi colossi dell'informatica.  
Uno a caso... Google!



Bene, ora che sappiamo “tutto” su questa tecnologia possiamo costruirci un framework che si occupi di fare il lavoro sporco per noi...



... ma è meglio di no!



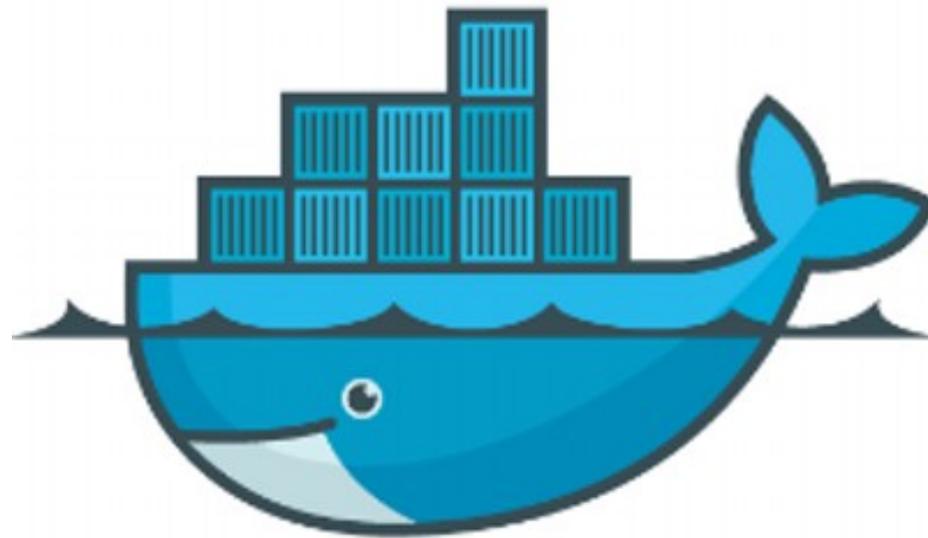
# LXC – Why not?

- Nessun tool per gestire il workflow dei container
  - Assenza di standard
  - Confusione



# Docker

Build, Ship and Run Any App, Anywhere

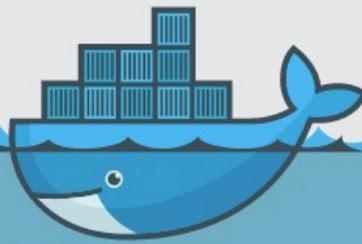


# Docker

È uno dei progetti open-source più seguiti dell'ultimo anno per tre semplici motivi:

- Definisce uno standard
- Ha una community molto numerosa
  - API semplici e robuste





**400+**

 Contributors

docker

**1.5M**

Docker Container  
Downloads

**5700+**

 Pull Requests

**11k+**

“Dockerized” Apps

**5500**

Projects on GitHub  
with the word “Docker”  
in their title



**5600+**

 Issues

**11k+**

 Stars

**1900+**

 Forks

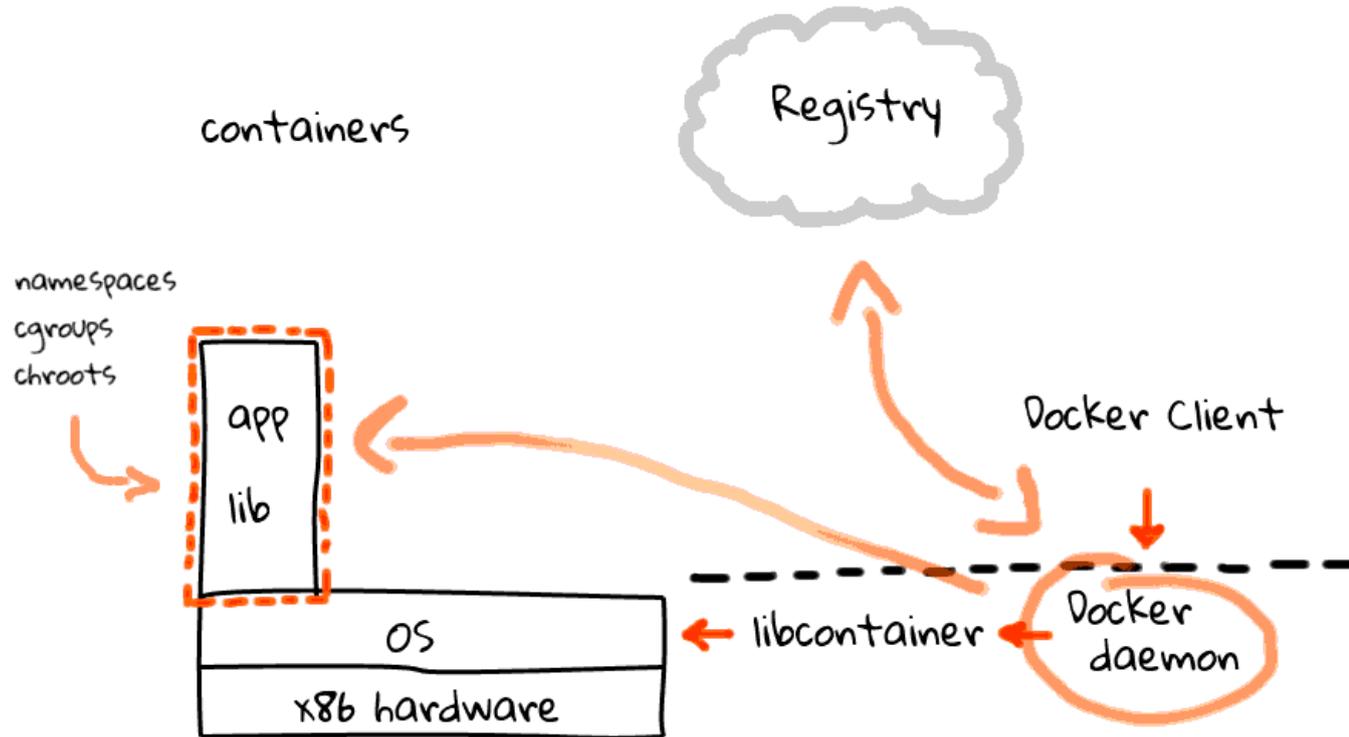
**8k+**

 Commits

 leanstack



# Docker – Com'è fatto?



# Docker – Fondamenti

- Pull
- Commit
- Push



