

Containers in a nutshell

Simone Lombardi

HCSSLUG — hcsslug.org

27 Ottobre 2017

Cosa sono i container?

I container sono un environment di esecuzione completo ed isolato: hanno a disposizione le loro risorse e condividono con il sistema host il kernel.

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità
- Sharing

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità
- Sharing
- Velocità di deploy

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità
- Sharing
- Velocità di deploy
- Footprint

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità
- Sharing
- Velocità di deploy
- Footprint
- Manutenzione

Qualche nome...

I progetti più famosi sono:

- LXC

Qualche nome...

I progetti più famosi sono:

- LXC
- LXD

Qualche nome...

I progetti più famosi sono:

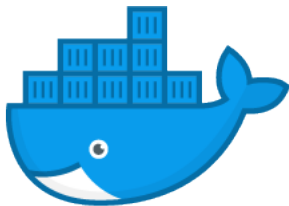
- LXC
- LXD
- systemd-nspawn

Qualche nome...

I progetti più famosi sono:

- LXC
- LXD
- systemd-nspawn
- Docker

Docker in a nutshell



docker

Docker in a nutshell

Docker è un progetto FOSS che estende i Linux container permettendo di buildare e deployare applicativi, usando solo le dipendenze runtime. Mette a disposizione molti tool per gestire i container.

Come funziona Docker?

- **cgroups**

Come funziona Docker?

- **cgroups**
- **namespaces**

Come funziona Docker?

- **cgroups**
- **namespaces**
- **OverlayFS**

Come funziona Docker?

- **cgroups**
- **namespaces**
- **OverlayFS**
- **libcontainer**

Perchè Docker?

- Isolamento degli applicativi

Perchè Docker?

- Isolamento degli applicativi
- Creazione e distruzione dei container molto rapida.

Perchè Docker?

- Isolamento degli applicativi
- Creazione e distruzione dei container molto rapida.
- Velocità e leggerezza

Perchè Docker?

- Isolamento degli applicativi
- Creazione e distruzione dei container molto rapida.
- Velocità e leggerezza
- Risorse condivise col sistema Host

Docker può buildare immagini leggendo un set di istruzioni da un file, chiamato Dockerfile. In questo file possiamo definire il comportamento del container, usando varie features implementate:

- Volumi

Docker può buildare immagini leggendo un set di istruzioni da un file, chiamato Dockerfile. In questo file possiamo definire il comportamento del container, usando varie features implementate:

- Volumi
- NAT

Docker può buildare immagini leggendo un set di istruzioni da un file, chiamato Dockerfile. In questo file possiamo definire il comportamento del container, usando varie features implementate:

- Volumi
- NAT
- Port forwarding

Dockerfile

```
1 FROM smebberson/alpine-base:3.0.0
2 MAINTAINER smlb <smlbr01@gmail.com>
3
4 # Install nginx
5 RUN echo "http://dl-4.alpinelinux.org/alpine/v3.3/main" >> /etc/apk/
    repositories && \
6     apk add --update nginx=1.8.1-r1 && \
7     rm -rf /var/cache/apk/* && \
8     chown -R nginx:www-data /var/lib/nginx
9
10 # Add the files
11 ADD root /
12
13 # Expose the ports for nginx
14 EXPOSE 80 443
```

In figura: Esempio di Dockerfile

Vi sono a disposizione innumerevoli tool per gestire applicativi multi-container contemporaneamente, è qui che entra in gioco **Docker Compose**.

È un tool per definire e avviare più container, definiti in un file .yaml, con molte features interessanti:

- Environments isolati su un singolo host

È un tool per definire e avviare più container, definiti in un file .yaml, con molte features interessanti:

- Environments isolati su un singolo host
- I dati dei volumi sono preservati

È un tool per definire e avviare più container, definiti in un file .yaml, con molte features interessanti:

- Environments isolati su un singolo host
- I dati dei volumi sono preservati
- È possibile ricreare solo i container modificati

È un tool per definire e avviare più container, definiti in un file .yaml, con molte features interessanti:

- Environments isolati su un singolo host
- I dati dei volumi sono preservati
- È possibile ricreare solo i container modificati
- Sostituzioni Variabili

È un piattaforma che permette di creare repository, buildare immagini e testarle, linkandole a Docker Cloud in modo da permettere il deploy rapido di container.

<https://hub.docker.com>

Confronto con le VM

- Le VM possono essere 'spostate' mentre sono in esecuzione.

Confronto con le VM

- Le VM possono essere 'spostate' mentre sono in esecuzione.
- I Container **NON** rimpiazzano le VM in ogni caso.

Confronto con le VM

- Le VM possono essere 'spostate' mentre sono in esecuzione.
- I Container **NON** rimpiazzano le VM in ogni caso.
- Valutare sempre gli USE Case!

Docker è uno strumento molto versatile, permette di gestire workflow e infrastrutture in maniera pulita e davvero intuitiva. Le casistiche di utilizzo sono molteplici, come per ogni tipo di strumento, bisogna sempre valutarne i pro e i contro.

Fine

Grazie a tutti per l'attenzione :D