Containers in a nutshell

Simone Lombardi

HCSSLUG - *hcsslug.org*
https://smlb.github.io

27 Ottobre 2017

hcsslug
GNU/LINUX USERS GROUP

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

hcsslug
GNU/LINUX USERS GROUP

Vantaggi dei container

I container portano numerosi vantaggi:

- Portabilità
- Sharing
- Velocità di deploy
- Footprint

hcsslug
GNU/LINUX USERS GROUP

Vantaggi dei container

I container portano numerosi vantaggi:

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

hcsslug
GNU/LINUX USERS GROUP

Qualche nome...

I progetti più famosi sono:

- LXC

Qualche nome...

I progetti più famosi sono:

- LXC
- LXD

Qualche nome...

I progetti più famosi sono:

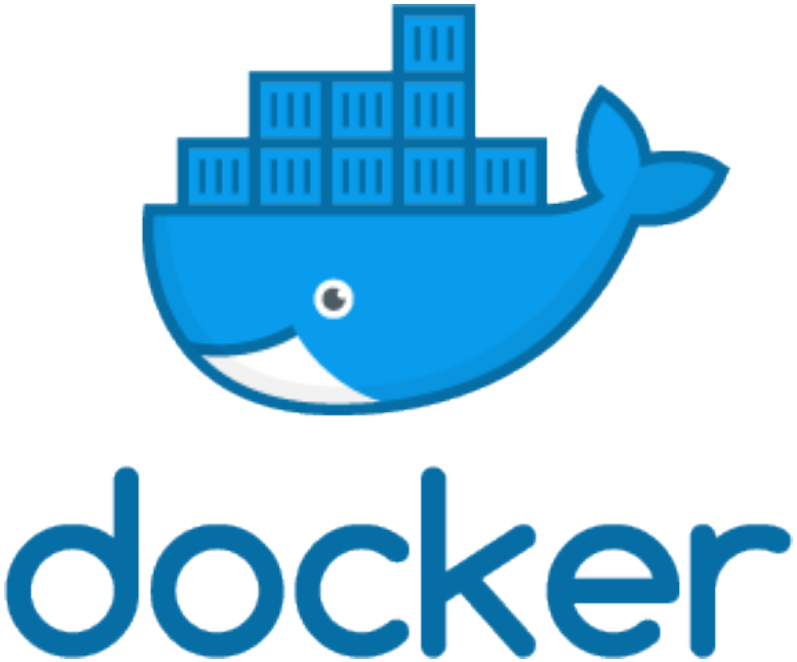
- LXC
- LXD
- systemd-nspawn

Qualche nome...

I progetti più famosi sono:

- LXC
- LXD
- systemd-nspawn
- Docker

Docker in a nutshell



hcsslug
GNU/LINUX USERS GROUP

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.

hcsslug
GNU/LINUX USERS GROUP

Come funziona Docker?

- **cgroups**

Come funziona Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Come funziona Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Come funziona Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Come funziona Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Perchè Docker?

- Isolamento degli applicativi

hcsslug
GNU/LINUX USERS GROUP

Perchè Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Perchè Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Perchè Docker?

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

hcsslug
GNU/LINUX USERS GROUP

Un piccolo assaggio

Docker può buildare immagini leggendo un set di istruzioni da un file, chiamato **Dockerfile**. In questo file possiamo definire il comportamento del container e quello che verrà eseguito.

hcsslug
GNU/LINUX USERS GROUP

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

Dockerfile: spiegazione

- **FROM**: definisce l'img di base per avviare il processo di build.

Dockerfile: spiegazione

- **FROM**: definisce l'img di base per avviare il processo di build.
- **RUN**: le direttive di esecuzione per il Dockerfile.

hcsslug
GNU/LINUX USERS GROUP

Dockerfile: spiegazione

- **FROM**: definisce l'img di base per avviare il processo di build.
- **RUN**: le direttive di esecuzione per il Dockerfile.
- **ADD**: copia i file dalla source dell'host al path specificato nel container.

hcsslug
GNU/LINUX USERS GROUP

Dockerfile: spiegazione

- **FROM**: definisce l'img di base per avviare il processo di build.
- **RUN**: le direttive di esecuzione per il Dockerfile.
- **ADD**: copia i file dalla source dell'host al path specificato nel container.
- **EXPOSE**: specifica la porta su cui verranno esposti i servizi dal container.

hcsslug
GNU/LINUX USERS GROUP

Altre istruzioni

- **CMD**: a differenza di **RUN**, non è eseguito durante la build ma quando il container è inizializzato con l'immagine buildata.

Altre istruzioni

- **CMD**: a differenza di **RUN**, non è eseguito durante la build ma quando il container è inizializzato con l'immagine buildata.
- **ENV**: usato per settare le variabili di ambiente.

hcsslug
GNU/LINUX USERS GROUP

Altre istruzioni

- **CMD**: a differenza di **RUN**, non è eseguito durante la build ma quando il container è inizializzato con l'immagine buildata.
- **ENV**: usato per settare le variabili di ambiente.
- **WORKDIR**: definisce dove **CMD** sarà eseguito.

hcsslug
GNU/LINUX USERS GROUP

Altre istruzioni

- **CMD**: a differenza di **RUN**, non è eseguito durante la build ma quando il container è inizializzato con l'immagine buildata.
- **ENV**: usato per settare le variabili di ambiente.
- **WORKDIR**: definisce dove **CMD** sarà eseguito.
- **ENTRYPOINT**

hcsslug
GNU/LINUX USERS GROUP

Container multipli

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

hcsslug
GNU/LINUX USERS GROUP

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

hcsslug
GNU/LINUX USERS GROUP

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
- I dati dei volumi sono preservati

hcsslug
GNU/LINUX USERS GROUP

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
- I dati dei volumi sono preservati
- È possibile ricreare solo i container modificati

hcsslug
GNU/LINUX USERS GROUP

Docker Hub

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

hcsslug
GNU/LINUX USERS GROUP

Svantaggi di Docker

Confronto con le VM

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

hcsslug
GNU/LINUX USERS GROUP

Svantaggi di Docker

Confronto con le VM

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

hcsslug
GNU/LINUX USERS GROUP

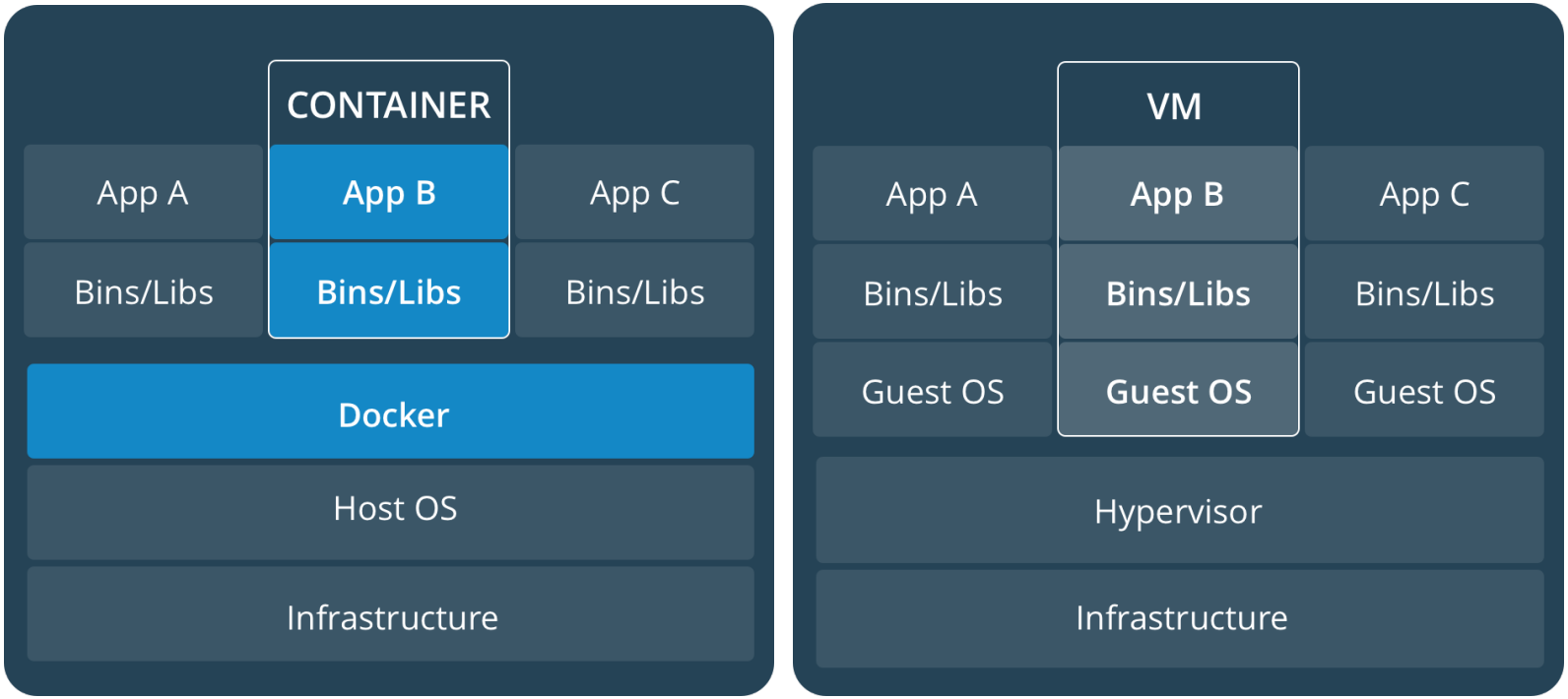
Svantaggi di Docker

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!

hcsslug
GNU/LINUX USERS GROUP

Docker vs VM



In figura: Layer effettivi

Conclusione

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.

hcsslug
GNU/LINUX USERS GROUP

Fine

~ # docker stop linuxday

Grazie a tutti per l'attenzione :D

hcsslug
GNU/LINUX USERS GROUP