

Appunti incompleti su docker

Salvo D'Asta

March 21, 2021

Contents

0.1	Installazione	2
0.1.1	Con apt-get install	2
0.2	scaricare un immagine da Docker HUB	2
0.3	visualizzare l'elenco delle immagini visualizzate	2
0.4	creare ed eseguire un container	2
0.5	visualizzare l'elenco dei container	2
0.6	eseguire un comando all'interno di un container	2
0.7	Eliminazione di un container o di una immagine	2
0.8	rimuovere i docker file e le immagini	3
0.9	creazione ed avvio di un container	3
0.10	Elenco dei container attivi e di quelli non attivi	3
0.11	Creazione di una immagine a partire da un container	3
0.12	Creazione di un'immagine con DockerFile	3
0.13	creazione di una network	3
0.14	Esempio creazione container mysql	3
0.14.1	creazione del container	3
0.14.2	creazione dell'utente remoto	4
0.15	Installazione su Raspberry Pi con Moodlebox	4

Cosa è Docker?

Si tratta di una tecnologia per la distribuzione (*deployment*) di servizi software. differenze tra virtualizzazione e "containerizzazione"

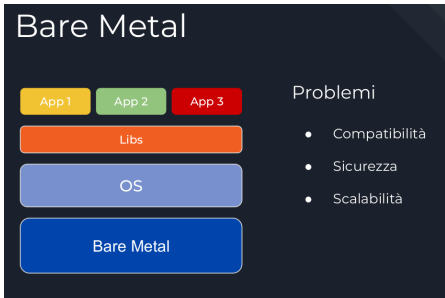


Figure 1: Il sistema "Bare metal" cioè basato sull'hardware *fisico* della macchina

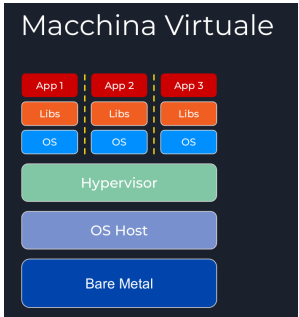


Figure 2: Il sistema basato sulla virtualizzazione della macchina (virtualbox, wmware etc.)

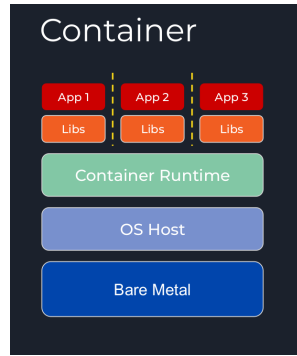


Figure 3: Il sistema basato su docker

Istruzioni di base

0.1 Installazione

0.1.1 Con apt-get install

```
sudo apt-get install docker.io
```

0.2 scaricare un immagine da Docker HUB

Collegarsi al sito <https://dockerhub.com> nella sezione *Explore* cercare l'immagine desiderata e seguire le istruzioni. da terminale digitare:

```
docker pull <nome immagine>
```

0.3 visualizzare l'elenco delle immagini visualizzate

```
docker images
```

0.4 creare ed eseguire un container

docker run -dit --name <nome container> esempio di esecuzione di un server apache:

```
docker run -dit --name miocontainer -p 8087:80 -v "/home/pi/classe5D/":"/usr/local/apache2/htdocs/" httpd:2.4
```

0.5 visualizzare l'elenco dei container

per visualizzare solo i container in esecuzione:

```
docker container ls
```

per visualizzare anche quelli non attivi:

```
docker container ls -a
```

0.6 eseguire un comando all'interno di un container

```
docker exec -it <nome container> <comando>
```

esempio di esecuzione di una shell bash all'interno di un container:

```
docker exec -it miohttpd bash
```

0.7 Eliminazione di un container o di una immagine

Dopo aver controllato l'elenco dei container:

```
docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d3be21917f8a	portainer/portainer	"/portainer"	3 days ago	Created		lucid_liskov
6583875c0997	ubuntu	"/bin/bash"	7 days ago	Exited (0)		mioubuntu

per eliminare il container *mioubuntu*:

```
docker rm 6583
```

 (non è necessario inserire tutto l'hash)

per eliminare le immagini sostituire il comando `rm` con `rmi` ed inserire i primi numeri dell'hash dell'immagine.

0.8 rimuovere i docker file e le immagini

```
docker system prune
```

0.9 creazione ed avvio di un container

```
docker container run nginx
```

Se l'immagine di `nginx` non si trova in locale, la scarica automaticamente dal *registry*

0.10 Elenco dei container attivi e di quelli non attivi

```
docker container ls
```

 (solo quelli attivi)

```
docker container ls -a
```

 (anche quelli inattivi)

0.11 Creazione di una immagine a partire da un container

```
docker container commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

esempio:

```
docker container commit <nome del container esistente> <nome della nuova immagine:<eventuale TAG>>
```

 // Supponendo che dopo aver creato il container `mioubuntu` ed aver apportato delle modifiche io voglia creare l'immagine `MIO` allora il comando sarà:

```
docker container commit mioubuntu MIO
```

0.12 Creazione di un'immagine con DockerFile

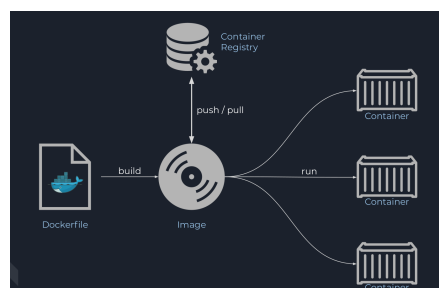


Figure 4:

0.13 creazione di una network

```
da:docker network --help
```

```
docker network create --subnet=<indirizzo_ip/subnet> <nome della subnet>
```

0.14 Esempio creazione container mysql

0.14.1 creazione del container

```
docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:tag
```

0.14.2 creazione dell'utente remoto

```
docker exec some_mysql bash
mysql -uroot -p
CREATE USER 'salvo'@'%' IDENTIFIED BY 'proprietario';
GRANT ALL PRIVILEGES ON mysql.* TO 'salvo'@'%';
```

0.15 Installazione su Raspberry Pi con Moodlebox

```
ssh moodlebox@moodlebox.home
sudo apt-get install docker.io
sudo apt-get install -y --no-install-recommends \
apt-transport-https \
ca-certificates \
curl \
software-properties-common

curl -fsSL https://apt.dockerproject.org/gpg sudo apt-key add --

root@raspberrypi:/home/pi# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest d8c0ac39cbcb 2 weeks ago 107MB
armhf/hello-world latest d40384c3f861 4 years ago 1.64kB
```

```
docker run -itd --name mio-httpd:latest -p 8080:80 httpd:latest
```

```
root@raspberrypi:/home/pi# docker exec -it mio-httpd bash
root@f2ca72141b9e:/usr/local/apache2#
```

```
cat index.html
^[[1] Container httpd in esecuzione su raspberry pi4 4GB ^[[1]
^[[1]+ Stopped cat ^[[1] index.html
```