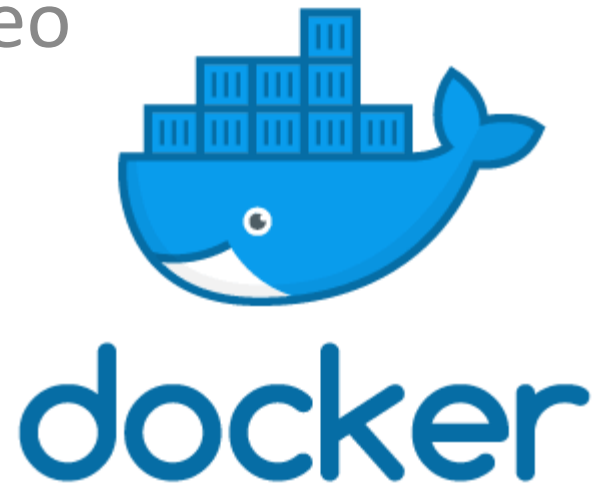


Docker

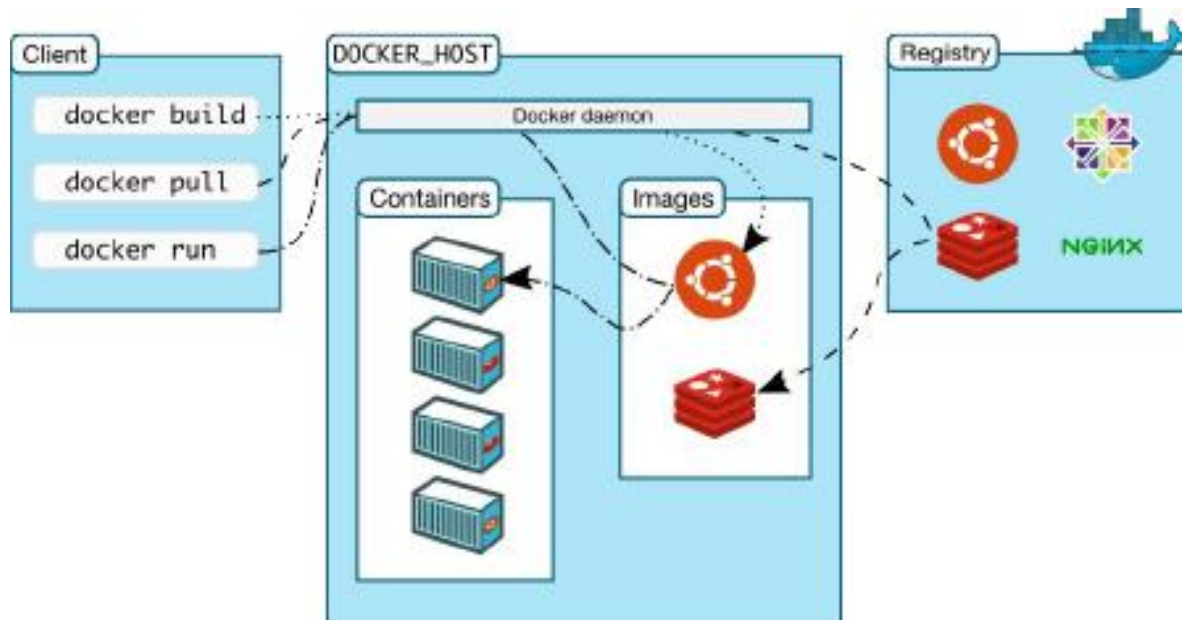
Riccardo Cattaneo



Le Immagini

Le Immagini sono dei modelli da cui eseguire uno o più container. Le immagini possono essere scaricate dal sito ufficiale «Docker Hub»

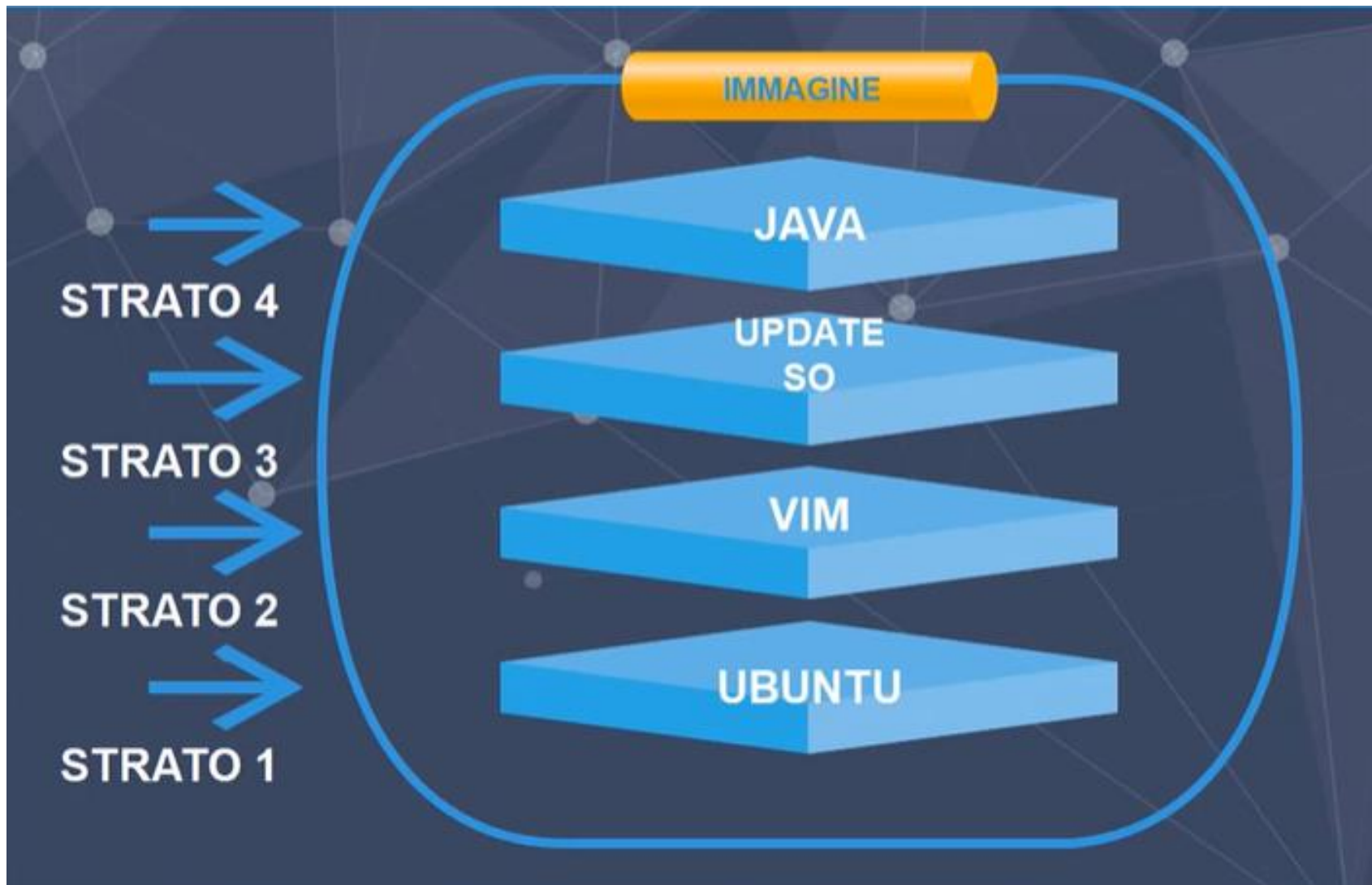
<https://hub.docker.com/>



docker pull

L'operazione di **pull** di un'immagine effettua il download dell'immagine stessa da un registro sul nostro docker locale e ci permette di eseguire il container derivato da tale immagine.

Un'immagine è composta da più strati sovrapposti. Sarà compito del Docker Engine unificare e considerare tutti questi strati come una singola entità.



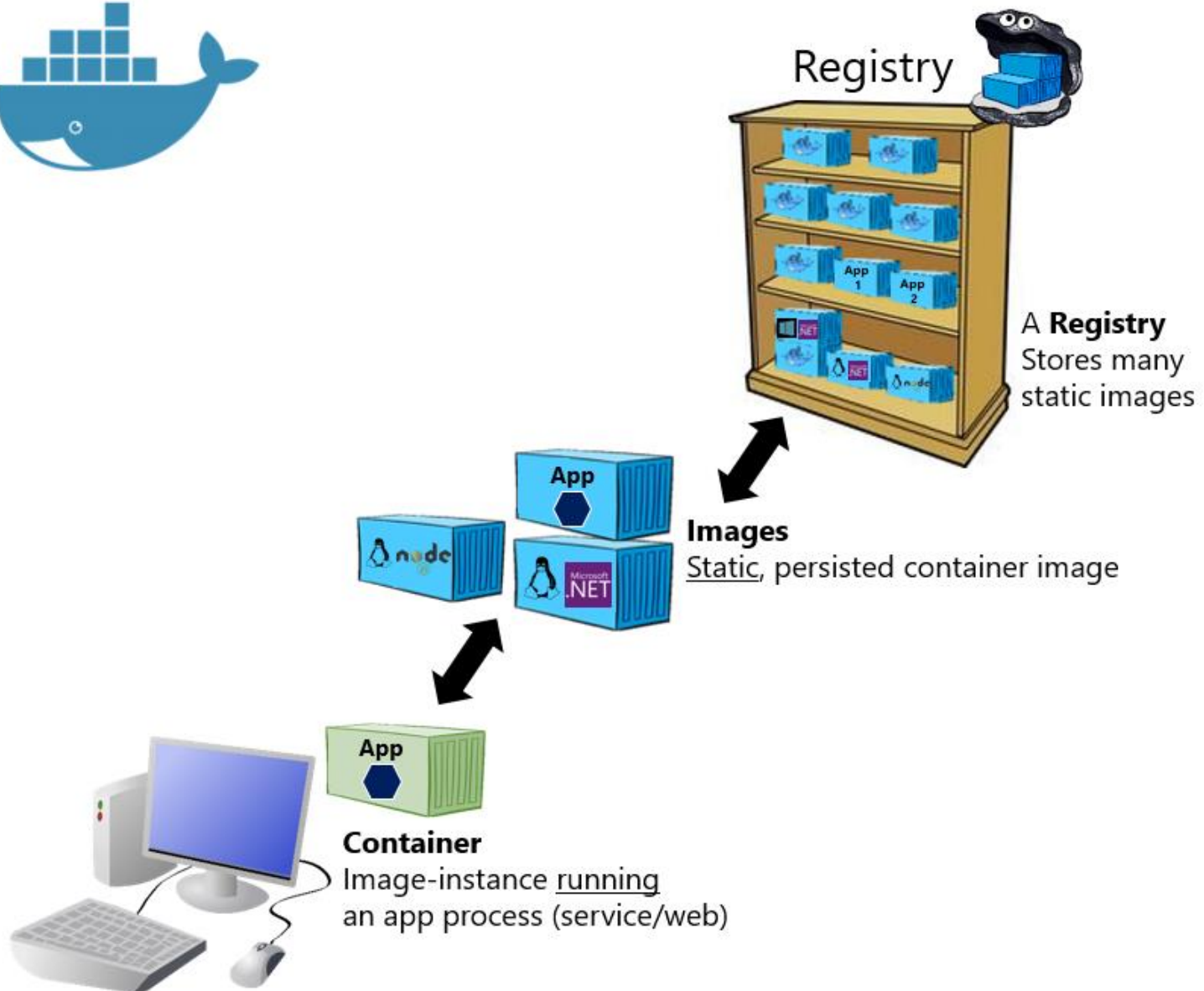
Per fare un primo esempio e per vedere da quanti layers è composta la mia immagine procediamo con l'esecuzione dei seguenti comandi (che approfondiremo più avanti) :

```
docker pull redis  
docker inspect redis
```

```
riccardo@riccardo-VirtualBox: ~  
    "MergedDir": "/var/lib/docker/overlay2/c79dac50845899d890a1288106ac8409c85  
b38169ca4e67afeb277b32a16daf1/merged",  
    "UpperDir": "/var/lib/docker/overlay2/c79dac50845899d890a1288106ac8409c85b  
38169ca4e67afeb277b32a16daf1/diff",  
    "WorkDir": "/var/lib/docker/overlay2/c79dac50845899d890a1288106ac8409c85b3  
38169ca4e67afeb277b32a16daf1/work"  
  },  
  "Name": "overlay2"  
},  
"RootFS": {  
  "Type": "layers",  
  "Layers": [  
    "sha256:e8b689711f21f9301c40bf2131ce1a1905c3aa09def1de5ec43cf0adf652576e",  
    "sha256:b43651130521eb89ffc3234909373dc42557557b3a6609b9fed183abaa0c4085",  
    "sha256:8b9770153666c1eef1bc685abfc407242d31e34f180ad0e36aff1a7feaeb3d9c",  
    "sha256:6b01cc47a390133785a4dd0d161de0cb333fe72e541d1618829353410c4facef",  
    "sha256:0bd13b42de4de0a0d0cc3f1f162cd0d4b8cb4ee20cbea7302164fdc6894955fd",  
    "sha256:146262eb38412d6eb44be1710bfe0f05d3493831f82b1c2be8dc8d9558c9f033"  
  ]  
},  
"Metadata": {  
  "LastTagTime": "0001-01-01T00:00:00Z"  
}  
}  
]  
riccardo@riccardo-VirtualBox:~$
```

Fare Attenzione a non confondere il concetto di **immagine** con quello di **container**. Se eseguiamo un container, quest'ultimo dipende dall'immagine da cui è derivato.

Infatti non è possibile eliminare un'immagine finché questa è in uso dal container.



Docker Hub

Il Docker Hub è il «raccolgitore» delle immagini e delle loro versioni. Il sito ufficiale è :

<https://hub.docker.com/>

Si consiglia di registrarsi per prima cosa attraverso il form che vediamo nella home page, dopodiché possiamo loggarci.



Search for great content (e.g., mysql)

[Explore](#)

[Pricing](#)

[Sign In](#)

[Sign Up](#)

Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.

Sign Up Today

Already have an account? [Sign In](#)



[Explore](#)

[Repositories](#)

[Organizations](#)

[Get Help](#) ▼

cattaneoweb ▼



Get to Docker Hub and Take a Tutorial

and learn how you can build, tag and share a sample image on Hub.

Started with Docker Desktop

Docker

Repositories

DockerCon Live is coming this May 27th! [Sign up to learn more.](#)



mysql)

Explore

Repositories

Organizations

Get Help ▾

cattaneoweb ▾



name...

Create Repository

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

Come possiamo vedere il sito ci dà la possibilità di creare un nostro repository. Proviamo a crearne uno cliccando sul bottone «Create Repository», scegliamo un nome, diamo una descrizione al nostro repository e scegliamo la visibilità pubblica o privata e confermare il tutto cliccando su create.

[Repositories](#)[Create](#)

Create Repository



Pro tip

You can p

docker
docker

Make sure
tag.

Visibility


Using 0 of 1 private repositories. [Get more](#)



Public 

Public repositories appear in Docker Hub search results



Private 

Only you can view private repositories

l)

Explore

Repositories

Organizations

Get Help ▾

cattaneoweb ▾



Using 0 of 1 private repositories. [Get more](#)

Webhooks


Settings

Docker commands

Public View

To push a new tag to this repository,

```
docker push cattaneoweb/corsojava-2021:tagname
```

 VULNERABILITY SCANNING - DISABLED
[Enable](#)

most recent tags here.

Recent builds

Link a source provider and run a build to see build results here.

Docker

Explore

Repositories

Organizations


Get Help ▾

cattaneoweb ▾



ages.

Most Popular ▾

VERIFIED PUBLISHER 

se Enterprise Edition

ated 3 years ago

e 12c Enterprise Edition

nux

x86-64

Databases

0

Stars

Docker

Filters

Images

Categories

☐ Verified Publisher

☐ Official Images
Official Images Published By Docker

☐ Analytics

☐ Application Frameworks

☐ Application Infrastructure

☐ Application Services

☐ Base Images

☐ Databases

☐ DevOps Tools

1 - 25 of 4.285.103 available images.

Most Popular

Oracle Database Enterprise Edition

By Oracle • Updated 3 years ago

Oracle Database 12c Enterprise Edition

ContainerLinuxx86-64Databases

VERIFIED PUBLISHER

0 Stars

MySQL Server Enterprise Edition

By Oracle • Updated 2 years ago

The world's most popular open source database system

ContainerLinuxx86-64Databases

VERIFIED PUBLISHER

0 Stars



ubuntu ☆

[Docker Official Images](#)

Ubuntu is a Debian-based Linux operating system based on free software.

↓ 1B+

[Container](#)
[Linux](#)
[386](#)
[ARM 64](#)
[x86-64](#)
[IBM Z](#)
[PowerPC 64 LE](#)
[ARM](#)
[Base Images](#)
[Operating Systems](#)
[Official Image](#)

Linux - IBM Z (latest)

Copy and paste to pull this image

```
docker pull ubuntu
```



[View Available Tags](#)

Description

Reviews

Tags

Il «pulling» di un'immagine

Una nuova installazione di docker non ha nessuna immagine disponibile in locale. E' possibile verificarlo digitando il seguente comando :

docker image ls

Proviamo a digitare il seguente comando per scaricare un'immagine come ad esempio alpine che è una distribuzione di linux :

docker pull alpine

Dopo aver confermato con invio viene scaricato in locale l'immagine di alpine.

Effettuare il «pull» di un'immagine significa scaricare l'immagine stessa sul docker installato localmente. Ed è possibile specificare anche la versione come vedremo più avanti. Docker verifica sempre se nella cache locale è già presente l'immagine o meno. Se non è presente procede con l'operazione di «pulling».

Se provo infatti a riscaricare l'immagine alpine, posso verificare che questa volta non procederà più a scaricarla in quanto già presente.

Il Tag di una immagine

Ogni immagine ha la sua versione (tag) e la sintassi corretta per effettuare il pull di un'immagine è :

```
docker image pull nome_img : tag_img
```

Per sapere le versioni disponibili per ogni immagine andare nel sito docker-hub cercare l'immagine da scaricare e vedere che versioni ci sono. Per verificare se è stata scaricata correttamente ricordiamoci di utilizzare il comando **docker image ls**

Esempi

docker image pull redis : 5.01

Verrà scaricata la versione di redis 5.01

docker image pull redis : latest

Verrà scaricata l'ultima versione di redis

docker image pull redis

Se non specifichiamo nessun tag verrà scaricata l'ultima versione di redis

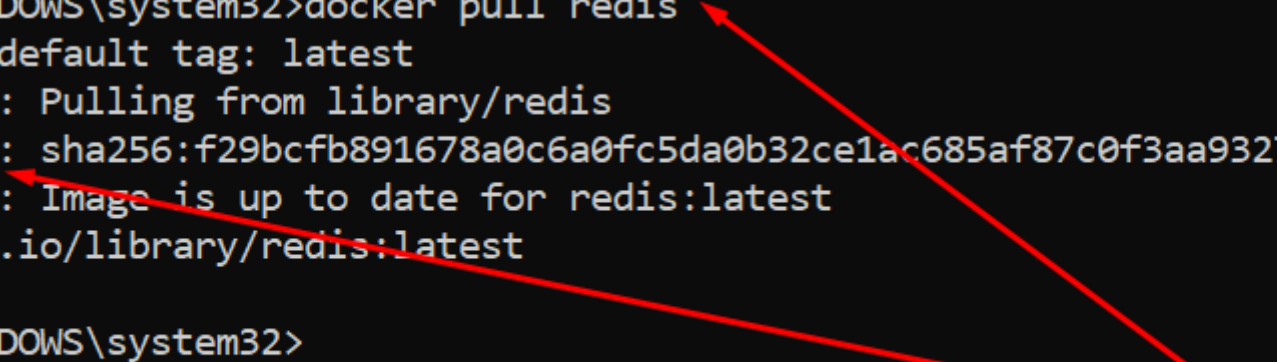
Digest

Per assicurarci che l'immagine scaricata sia effettivamente quella voluta, si può utilizzare un «hash» chiamato «digest».

Ogni volta che effettuiamo il pull di un'immagine ci viene fornito in output anche il digest di tale immagine. E' sufficiente copiarlo e conservarlo, nel caso di un nuovo pull potremmo confrontare i valori e se saranno diversi significherà che è stato apportato un cambiamento all'immagine.

```
C:\WINDOWS\system32>docker pull redis
Using default tag: latest
latest: Pulling from library/redis
Digest: sha256:f29bcfb891678a0c6a0fc5da0b32ce1ac685af87c0f3aa9327e562da8d3f3b88
Status: Image is up to date for redis:latest
docker.io/library/redis:latest

C:\WINDOWS\system32>
```



Volendo possiamo utilizzare il digest per scaricare esattamente quella versione in qualsiasi momento utilizzando la seguente sintassi :

`docker pull redis@sha256:f29bcfb8916.....`

docker rm

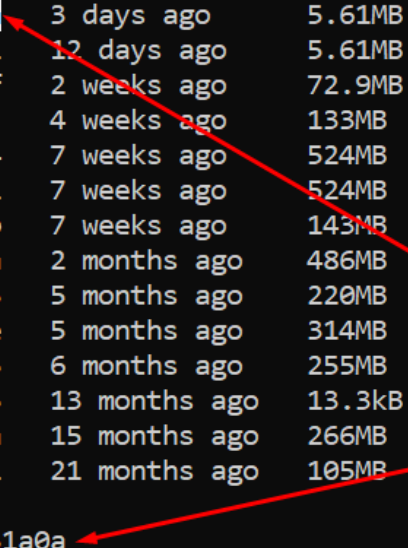
Per rimuovere un'immagine è sufficiente utilizzare il comando **docker rm** (remove) seguito dall'id dell'immagine.

NB Non è possibile cancellare un'immagine se c'è un container associato a quell'immagine.

```
C:\WINDOWS\system32>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redis	latest	eb0ab2d55fdf	17 hours ago	104MB
<none>	<none>	13a168631a0a	3 days ago	5.61MB
alpine	latest	e50c909a8df2	12 days ago	5.61MB
ubuntu	latest	f63181f19b2f	2 weeks ago	72.9MB
nginx	latest	f6d0b4767a6c	4 weeks ago	133MB
microservizio	latest	5a1fa2e88f74	7 weeks ago	524MB
<none>	<none>	a39307190751	7 weeks ago	524MB
<none>	<none>	231b0402ef3b	7 weeks ago	143MB
openjdk	latest	8f057dde245a	2 months ago	486MB
dpage/pgadmin4	latest	ae36b8785e03	5 months ago	220MB
postgres	latest	62473370e7ee	5 months ago	314MB
dpage/pgadmin4	<none>	c520f7001785	6 months ago	255MB
hello-world	latest	bf756fb1ae65	13 months ago	13.3kB
dpage/pgadmin4	4.15	048a766c7caa	15 months ago	266MB
openjdk	8-jdk-alpine	a3562aa0b991	21 months ago	105MB

```
C:\WINDOWS\system32>docker image rm 13a168631a0a
```



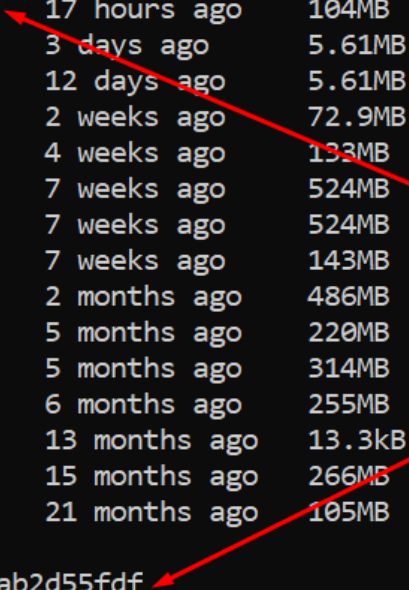
docker inspect

Con il comando **docker inspect** seguito dall'ID dell'immagine... stiamo chiedendo a docker di fornirci le informazioni dettagliate su quella immagine. Ma può essere usato anche in riferimento a container e ad altro... lo approfondiremo più avanti...

```
C:\WINDOWS\system32>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redis	latest	eb0ab2d55fdf	17 hours ago	104MB
<none>	<none>	13a168631a0a	3 days ago	5.61MB
alpine	latest	e50c909a8df2	12 days ago	5.61MB
ubuntu	latest	f63181f19b2f	2 weeks ago	72.9MB
nginx	latest	f6d0b4767a6c	4 weeks ago	133MB
microservizio	latest	5a1fa2e88f74	7 weeks ago	524MB
<none>	<none>	a39307190751	7 weeks ago	524MB
<none>	<none>	231b0402ef3b	7 weeks ago	143MB
openjdk	latest	8f057dde245a	2 months ago	486MB
dpage/pgadmin4	latest	ae36b8785e03	5 months ago	220MB
postgres	latest	62473370e7ee	5 months ago	314MB
dpage/pgadmin4	<none>	c520f7001785	6 months ago	255MB
hello-world	latest	bf756fb1ae65	13 months ago	13.3kB
dpage/pgadmin4	4.15	048a766c7caa	15 months ago	266MB
openjdk	8-jdk-alpine	a3562aa0b991	21 months ago	105MB

```
C:\WINDOWS\system32>docker image inspect eb0ab2d55fdf
```

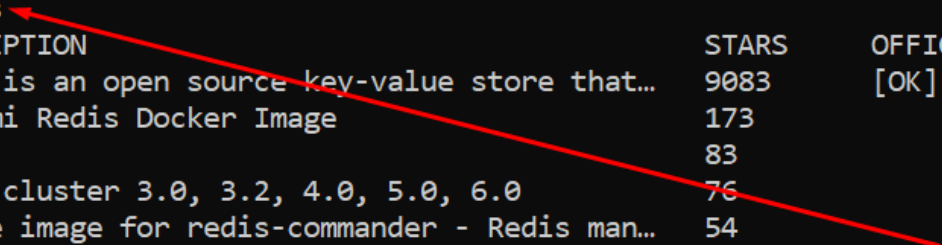


docker search

Un altro comando molto utile da utilizzare da riga di comando è **docker search** seguito dal nome dell'immagine che stiamo cercando. Se ad esempio scriviamo `docker search redis`, si collega al repository di docker e cerca tutte le immagini che hanno quel nome e le elenca in modo da poter scegliere quale scaricare senza doversi collegare al sito web.

```
C:\WINDOWS\system32>docker search redis
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
redis	Redis is an open source key-value store that...	9083	[OK]	
bitnami/redis	Bitnami Redis Docker Image	173		[OK]
sameersbn/redis		83		[OK]
grokzen/redis-cluster	Redis cluster 3.0, 3.2, 4.0, 5.0, 6.0	76		
rediscommander/redis-commander	Alpine image for redis-commander - Redis man...	54		[OK]
kubeguide/redis-master	redis-master with "Hello World!"	33		



Il comando docker run

Il comando **docker run** si utilizza per l'avvio di un container. Tuttavia possiamo utilizzarlo anche per effettuare il pull di un'immagine : prima verificherà se l'immagine è già presente localmente, poi provvederà allo **scaricamento** e **all'esecuzione** del container.

docker run hello-world

Avvio di un container

La definizione più semplice di container è : **un'istanza in esecuzione derivata da un'immagine.** Inoltre è possibile avviare più container a partire da una singola immagine.



docker container run

E' Il comando che ci permette di eseguire un container. Questo è il comando nella sua forma base, come vedremo abbiamo la possibilità di aggiungere delle opzioni e parametri.

Normalmente si utilizza un container allo scopo di utilizzare un'applicazione o un servizio. Il comando assume quindi la seguente forma :

docker container run immagine applicazione

Ad esempio : **docker container run -it ubuntu /bin/bash**
ci permette di avviare una shell di tipo bash all'interno del container ubuntu.

L'opzione **-it** collega il nostro terminale a quello del container così da poter effettuare le attività all'interno di quest'ultimo. Inoltre avviando un container ci rendiamo presto conto che alcuni comandi che normalmente utilizziamo non sono disponibili, in quanto l'immagine contiene solo ciò che è strettamente necessario. Ovviamente abbiamo la possibilità di scaricare in seguito ciò che ci occorre.

Facciamo qualche esempio per prendere un pò di pratica, se eseguo il comando

docker run alpine

Come possiamo osservare abbiamo eseguito il container che subito è terminato in quanto non abbiamo specificato altre opzioni o altre operazioni da effettuare. Se infatti proviamo a digitare il seguente comando

docker ps

Questo comando visualizza tutti i container attivi, e possiamo vedere che il nostro alpine non è presente.

Per cui posso aiutarmi ad esempio con il comando **sleep**, questo comando permette di mantenere attivo il container per un numero di secondi da noi specificato.

docker run alpine sleep 20

In questo caso sto avviando il container e lo tengo attivo per 20 secondi e poi terminerà. Possiamo verificarlo aprendo un altro terminale ed eseguire il comando `docker ps`

Adesso proviamo a far partire da una singola immagine più container (è consigliato dare dei nomi ad ogni container tramite **l'opzione --name** come nel seguente esempio) :

```
docker run --name=docker1 alpine sleep 20
```

```
docker run --name=docker2 alpine sleep 20
```

Possiamo ora verificare con il comando `docker ps` l'effettiva esecuzione dei due container.

Interazione con un Container

Con il comando **-it** posso dire a docker di voler interagire con il container, ma devo specificare anche la tipologia di console interattiva che voglio, ad esempio /bin/bash :

```
docker run -it ubuntu /bin/bash
```

Se invece vogliamo uscire dalla console interattiva mantenendo comunque attivo il container basta fare **CTRL + P e CTRL + Q**. Una volta uscito se poi voglio rientrare, il comando da eseguire è :

```
docker exec -it ID_CONTAINER /bin/bash
```

docker start ID_CONTAINER (avvio di un container)

docker stop ID_CONTAINER (arresto di un container)

docker rm ID_CONTAINER (rimuove un container)

docker container prune (rimuove tutti i container stoppati)

Se aggiungo **l'opzione -d** avvio un container in background e posso poi accedere in seguito con il comando **exec**.

Esempio Riepilogo

docker image pull ubuntu

docker container run -it ubuntu /bin/bash

CTRL+P CTRL+Q

docker ps

docker stop ID_CONTAINER

docker ps -a

docker start ID_CONTAINER

docker exec -it ID_CONTAINER /bin/bash