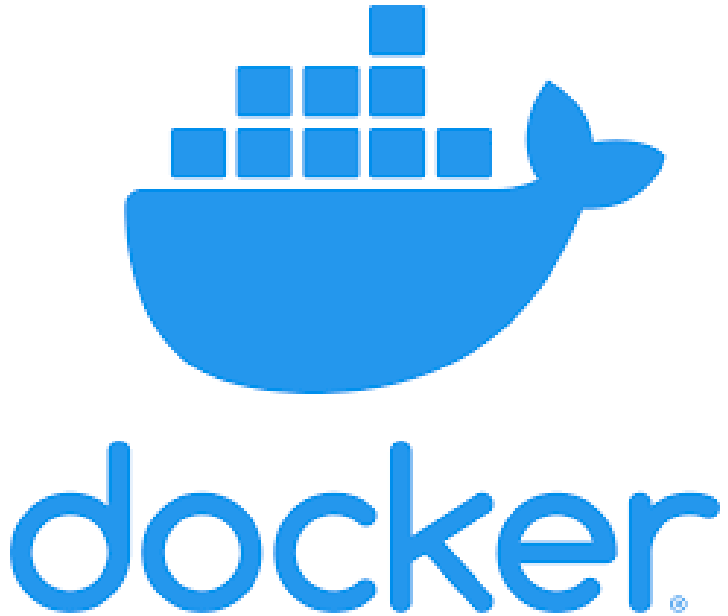


Introduzione a Docker

Vincenzo Racca

Chi sono

```yaml
apiVersion: v2022
kind: CV
metadata:
  name: "Vincenzo Racca"
spec:
  jobs:
  - name: "Java Developer"
    company:
    - name: "System Management S.p.A."
      date: 2019-04
  - name: Blogger
    site: "www.vincenzoracca.com"
  certifications:
  - "Spring Professional"
  - "CKAD - Certified Kubernetes Application Developer"
```

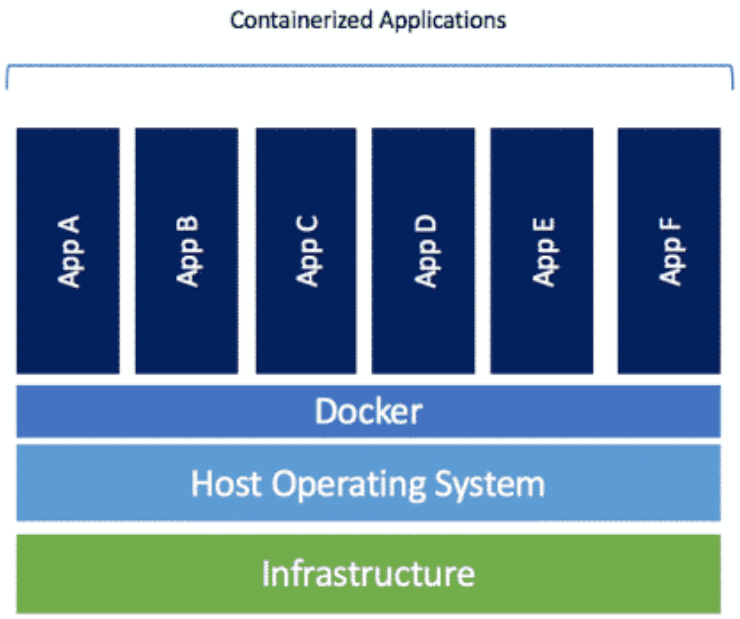
Sommario

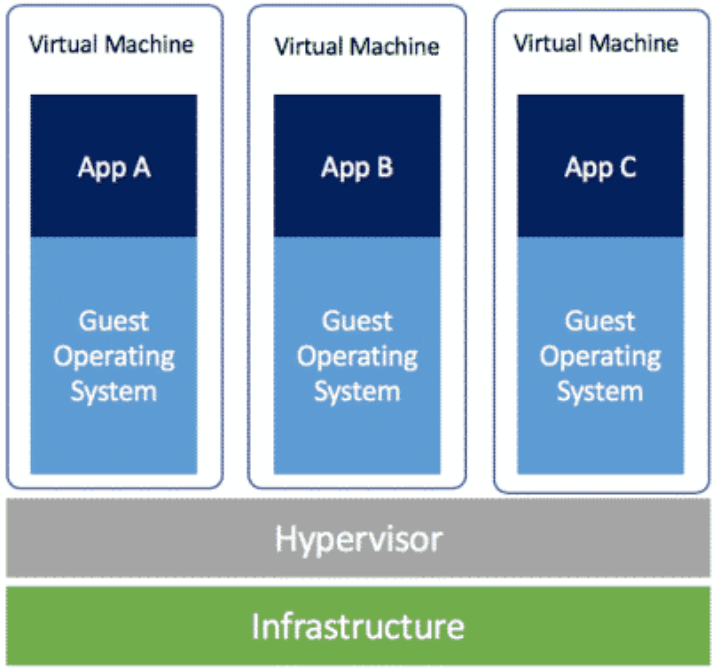
1. Cos’è un container?
2. Virtual Machine vs Container
3. Cos’è Docker?
4. Installazione di Docker
5. Immagini Docker per creare container
6. Comandi utili
7. Demo di un’applicazione Java containerizzata
8. Applicazioni containerizzate: change your mind!

Cos’è un Container

- Tecnologia che permette di pacchettizzare e isolare applicazioni dal suo ambiente di runtime (esempio: diverse versioni JDK).
- Sono quindi processi isolati, grazie alle features Linux di cgroup e namespace.
- Sono processi leggeri, poiché condividono il Kernel dell’host.
- I container avviano un processo principale (ad esempio un «java –jar»). Finché è in esecuzione il processo principale, il container è in esecuzione.

Container vs Virtual Machine





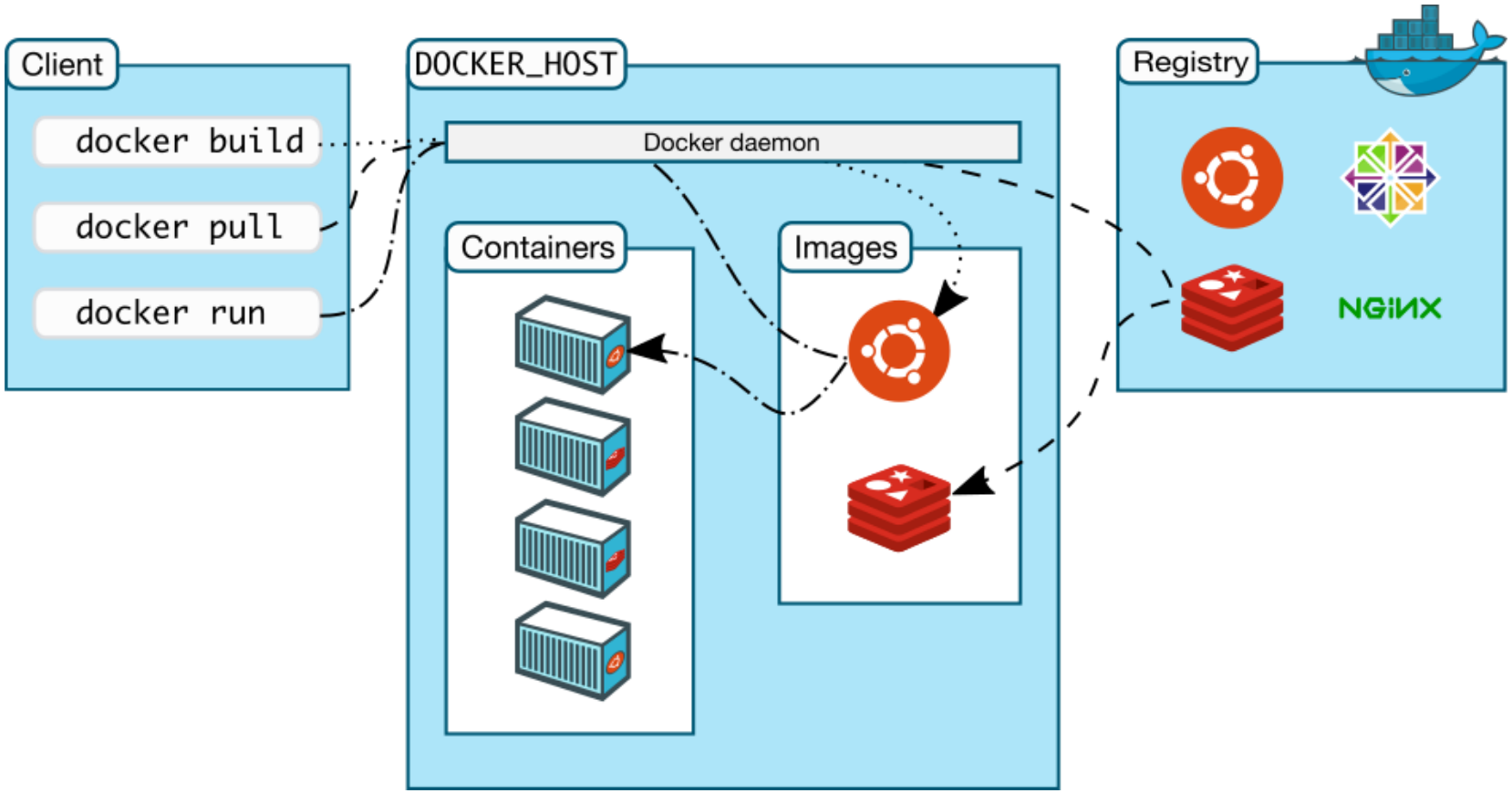
Perché la tecnologia a container?

- Isolamento tra diverse applicazioni (non più problemi di incompatibilità dati dall'ambiente, esempio versioni JDK diverse).
- Architettura a microservizi, esigenza di isolare più (micro)applicazioni.
- Possono essere creati facilmente (processo veloce e ripetibile).
- Maggiore scalabilità.
- Possono essere eseguiti su qualsiasi OS.
- Eseguire una applicazione containerizzata in locale è uguale ad eseguirla in qualsiasi altro ambiente (stesso ambiente di runtime).
- Utili anche nell'utilizzo di test di integrazione (esempio: TestContainer).

Cos’è Docker?

«Tecnologia che permette la creazione e la gestione di Container»

«È un container runtime»



Installazione di Docker

https://docs.docker.com/get-docker/

Immagini Docker per creare container

Cosa sono le immagini

- Template di sola lettura che contiene istruzioni per creare container.
- Facendo un paragone con la OOP, potremmo associare le immagini alle classi, mentre i container agli oggetti.
- Vengono messe a disposizione diverse immagini dal DockerHub, un registro pubblico di immagini.
- Possono essere create immagini custom a partire da un'immagine esistente, mediante Dockerfile.
- Può essere formata da più layers (ogni istruzione del Dockerfile che modifica il filesystem, crea un nuovo layer). Quando si esegue un container, viene creato, sopra gli altri layers,l'unico layer di scrittura, chiamato "container layer".

```
docker run --name=nginx -d -p 8080:80 nginx
```

Immagini Docker per creare container

Il Dockerfile (https://docs.docker.com/engine/reference/builder/)

```
FROM openjdk:11-jdk
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} application/app.jar
RUN groupadd -g 1000 java_group
RUN useradd -u 1000 -ms /bin/bash -g java_group java_user
USER java_user
EXPOSE 8080
WORKDIR application
ENTRYPOINT ["java", "-jar", "app.jar"]
CMD ["arg1", "arg2"]
```

```
docker build -t my-image:0.0.1 .
```

Comandi utili

Lista di immagini scaricate in locale: **docker images**

Creare un container da un'immagine: **docker run --name <container_name> -d <img_name>**

Lista container attivi**: docker ps**

Lista di tutti i container**: docker ps -a**

Creare un'immagine da un Dockerfile (comando eseguito nel path del Dockerfile): **docker build -t <img_name>:<img_tag> .**

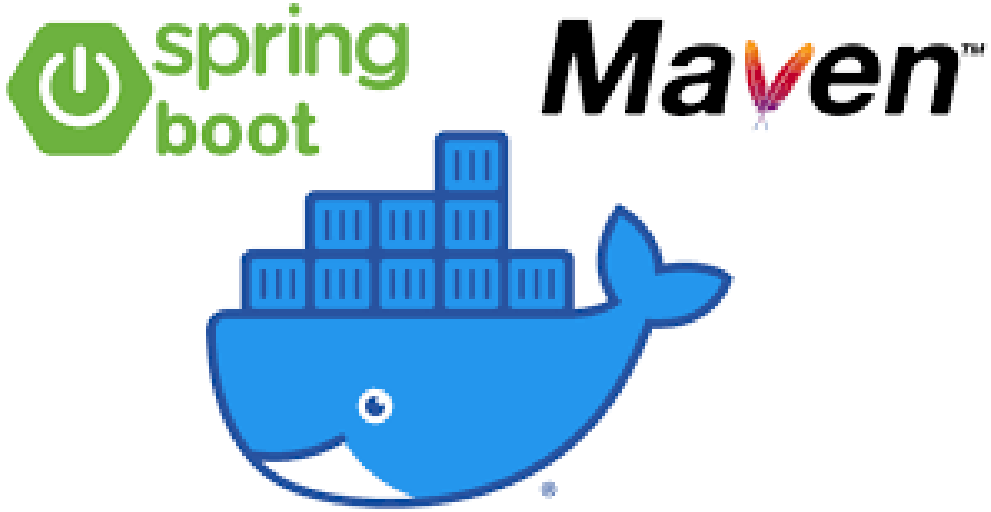
Log di un container: **docker logs -f <container_name>**

Eseguire un comando su un container: **docker exec <container_name> <command>**

Entrare in un container tramite bash: **docker exec -it <container_name> /bin/bash**

Demo di un’applicazione Java containerizzata

Progetto su GitHub: https://github.com/vincenzo-racca/spring-docker



Applicazioni containerizzate: change your mind!

- Tieni sempre a mente che i container sono effimeri.
- Applicazioni stateless.
- Scalabili orizzontalmente e non più verticalmente.
- Log su standard output e non più su file.
- Properties di configurazione come variabili d'ambiente.

Per altre indicazioni, leggere i "12-factor app": https://12factor.net/it/