



Docker da zero

Fabio Ros Moku s.r.l.

1.1

L'idea alla base di Docker

Docker è una tecnologia che consente creazione ed utilizzo dei container linux.

Ok. Ma cos'è un container?

Immaginatevi una configurazione che determina come deve essere fatto il sistema che andrà eseguito. Questa sarà un'immagine.

Quell'immagine sarà la base per avviare un processo a partire da quelle caratteristiche. Quel processo è un container.

Questa soluzione si sposa alla perfezione
con l'approccio a microservizi.

1.2

Come siamo arrivati
ai container?

1.2 Come siamo arrivati ai container?

Prima c'erano le **macchine fisiche**, spesso sovradimensionate o sottodimensionate.

- Costi elevati, difficoltà a scalare o a replicare il codice in altre macchine
- Difficoltà di monitoraggio delle risorse in modo centralizzato
- Tempistiche d'intervento in caso di failure molto lunghe
- ...

Poi sono arrivate le **macchine virtuali**.

Spesso usate per il delivery di applicazioni complete.

Per rendere l'idea, immaginate di avere una macchina virtuale per ogni servizio gestibile individualmente.

Sarebbe davvero efficace.

Ma quanto costerebbe in termini di risorse?

Sarebbe davvero la miglior soluzione possibile?

- La soluzione a macchine virtuali non sarebbe la più efficace.
- Docker permette di fattorizzare tutte le risorse comuni ed evitare inutili sprechi.
- L'utilizzo dei container linux, senza uno strumento come Docker sarebbe davvero complicato.
- Docker su linux utilizza i container nativi di linux

Ok, per Linux ci siamo.

Ma Windows? e Mac?

- Recentemente Windows ha introdotto il concetto di “**Windows container**”.
- Supportati solo da Windows 10 e Windows server 2016.
- Docker for Windows richiede la versione 10 Professional o Enterprise 64 bit
- Per le altre versioni di Windows, docker-toolbox predispone un ambiente in una macchina virtuale su VirtualBox e permette di utilizzare Docker.

Docker for Mac

- MacOS non supporta nativamente i container.
- Docker for Mac fornisce un pacchetto completo che esegue i container in una macchina virtuale Linux.

Probabilmente avete sentito
parlare di Kubernetes



kubernetes

- Kubernetes permette di orchestrare i container agendo ad alto livello.
- Progetto open-source lanciato da Google
- Docker è impostato come container runtime di default.
- Docker ha riconosciuto Kubernetes, anche se ha un sistema nativo per gestire queste funzionalità: Swarm.
- In questo corso vedremo nel dettaglio Swarm in quanto sistema fornito direttamente da Docker Inc.

1.3

Perché dovrei usare
Docker?

1.3 Perché dovrei usare Docker?

- Puoi replicare l'ambiente senza sorprese ed istantaneamente
 - tra sviluppatori
 - negli ambienti (sviluppo / staging / produzione)
- lo stesso deploy, può essere eseguito su N server
- Cambiare hosting provider è immediato
- Pensato per scalare

1.3 Perché dovrei usare Docker?

- Tutto ciò che c'è in un container, può essere spedito assieme
- Mettiamo tutto in un container, chiudiamo le porte, ci mettiamo un tag e lo spediamo.
- Lo shipping non è più un nostro problema
- Quando verrà eseguito, il corretto funzionamento tornerà ad essere di nostra responsabilità
- Lo stesso vale per chi esegue un container, non si preoccupa di cosa contenga. Lo esegue e basta, in conformità con l'architettura.