

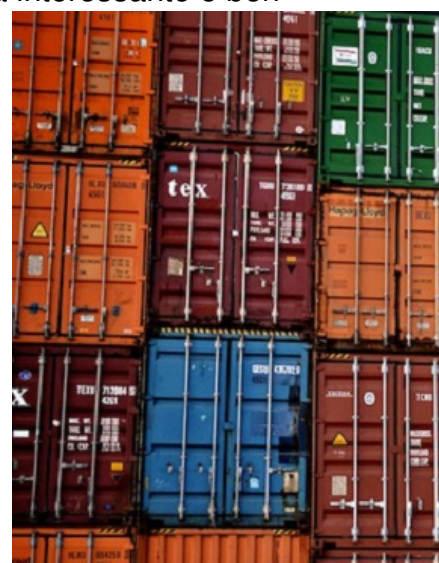
La tecnologia di containerizzazione di **Docker** sta vivendo un [momento](#) di grande gloria, forse più che altro perché Docker è la prima vera piattaforma che

mette ordine nel

[caos](#)

dei container Linux-based e rende finalmente accessibile questa tecnologia agli sviluppatori.

Il successo che [arride](#) a Docker, spinge diverse testate IT a parlare della startup e della piattaforma, perché, in ambito cloud e virtualizzazione, l'argomento risulta interessante e ben gradito da una folta schiera di utenti e lettori target.



Come tutte le volte in cui il successo giunge quasi dal nulla, repentinamente e in modo inaspettato, intorno all'argomento si creano a volte dei **falsi miti**, alimentati dalla cattiva informazione che viene poi replicata da diversi media.

La stesso avviene anche per Docker, che raggiunge la [versione 1.5](#) e su cui si è dato il via alla diffusione di alcune credenze che bisogna sfatare subito, sia per una questione di corretta informazione, sia per non ritrovarsi a usare la piattaforma scoprendo alcuni limiti di cui non si era a conoscenza.

Primo mito da sfatare: Docker è una macchina virtuale

Anche se il paragone potrebbe calzare dal punto di vista teorico e solo per spiegare la base di

funzionamento della containerizzazione, in realtà assimilare Docker a una semplice macchina virtuale sarebbe troppo riduttivo e anche errato.

Docker è molto di più ed è un' [evoluzione](#) della tecnologia di virtualizzazione, o, meglio ancora, è **l'evoluzione del formato container LXC** già integrato in Linux. Infatti, fra Docker e le macchine virtuali vi sono differenze sostanziali, come **l'isolamento risorse** che in Docker non è totale come nelle macchine virtuali, come **il minore overhead** di Docker e **la condivisione del medesimo kernel** fra tutti i container (che si contraddistingue dall'uso di un solo kernel per ogni macchina virtuale).

Secondo mito da sfatare: Docker scala le applicazioni

Falso. Questa credenza si è diffusa partendo dal fatto che Docker consente di fare un passo in avanti nella scalabilità, migliorando l'implementazione del codice su diversi server in un lasso di tempo più breve. Tutto ciò non equivale a dire che Docker scala le applicazioni. **Docker non riscrive il codice di un'applicazione e l'implementazione delle proprietà di scalabilità spetta ancora agli sviluppatori**

Terzo mito da sfatare: Docker è già usato negli ambienti di produzione

Come risultato del successo mediatico, alcuni utenti sono portati a pensare che Docker sia già utilizzato in produzione. **Falso. La piattaforma è molto nuova e ancora viene sfruttata principalmente negli ambienti di testing**

Quarto mito da sfatare: Docker è indipendente dal sistema operativo

Si dice che Docker è *OS Agnostic*, ma non è vero! Non funziona universalmente in tutti gli ambienti e su tutti i sistemi operativi, ma si esegue **in modo nativo su Linux**, mentre su OS X e [Windows](#) può essere messo in opera con opportune accortezze.

Quinto mito da sfatare: Docker rende l'applicazione sicura

No! No! No! **I container in Linux hanno diverse vulnerabilità di sicurezza** e Docker non aggiunge a essi patch o sistemi di sicurezza tali da renderli invulnerabili. Inoltre, Docker ha problemi di sicurezza anche a livello di

[distribuzione](#)

dei container, a cui molte aziende e Docker stessa stanno cercando di trovare una soluzione.

Sesto mito da sfatare: Docker per Java non rappresenta la panacea a tutti i mali

Docker per Java non è solo vantaggi. Se da un lato, a differenza dei JAR, con Docker è possibile *imballare* tutte le dipendenze (compresa la JVM) in un'immagine pronta alla distribuzione, dall'altra parte l'ottimizzazione delle prestazioni è davvero più difficile. Usando i container non si sa a priori quanta memoria venga allocata a ciascuno di essi e questo **rende la messa a punto delle prestazioni davvero complicata**