

Containers

utilizzo di Docker: Docker è un popolare software libero che utilizza la virtualizzazione a livello di sistema operativo per eseguire applicazioni in ambienti isolati chiamati container. I container Docker sono un'estensione dei container del sistema operativo.

Sono molto utili perchè non hanno all'interno il sistema operativo quindi sono molto veloci ad esempio l'avvio e lo spegnimento delle v.m. sono nell'ordine dei minuti mentre nei containers nell'ordine dei millisecondi

I **container** sono una tecnologia di virtualizzazione leggera che permette di isolare e impacchettare applicazioni e i loro ambienti di esecuzione in unità autonome.

Per definizione i container sono **volatili**: allo spegnimento perdi tutto.

Sono molto usati nello sviluppo e nel deployment di software perchè garantiscono **portabilità**, **scalabilità** e **coerenza** tra ambienti diversi.

Paragonandoli alla programmazione le immagini (spiegate dopo) sono le classi e i containers sono gli oggetti: della classe uso un'istanza, cioè l'oggetto, dell'immagine uso un'istanza cioè un containers.

Per utilizzare Docker abbiamo 3 sistemi:

- 1) GUI: uso di Docker desktop
- 2) CLI: uso di un terminale con comandi su riga di comando
- 3) SDK: mi aggancio via software a Docker

Le **Virtual Machine (VM)** sono un tipo di virtualizzazione che consente di eseguire più sistemi operativi indipendenti su un unico hardware fisico. Ogni VM emula un computer completo, con il proprio sistema operativo, memoria, processore e storage.

Una VM è un'istanza software che gira sopra un **hypervisor**, che può essere di due tipi:

- **Hypervisor di tipo 1 (bare-metal)** → Gira direttamente sull'hardware (es. VMware ESXi, Microsoft Hyper-V, Xen).
- **Hypervisor di tipo 2 (hosted)** → Gira sopra un sistema operativo (es. VirtualBox, VMware Workstation).

Vantaggi delle Virtual Machine

- ✓ **Isolamento forte** → Ogni VM è indipendente dalle altre, ideale per test e sicurezza.
- ✓ **Compatibilità** → Può eseguire qualsiasi sistema operativo (Windows, Linux, macOS).
- ✓ **Snapshot & Backup** → Possibilità di salvare lo stato della VM per un ripristino rapido.
- ✓ **Multi-tenant** → Più utenti possono usare VM diverse sullo stesso hardware.

Caratteristica	Virtual Machine (VM)	Container
Isolamento	Completo (ha un proprio SO)	Condivide il kernel del SO host
Avvio	Più lento (minuti)	Molto rapido (secondi)
Consumo risorse	Più elevato (ogni VM ha un SO)	Più leggero
Portabilità	Minore	Alta (compatibile ovunque ci sia un runtime container)

Immagini nei container

Un'**immagine di container** è un pacchetto immutabile (è in sola lettura, spesso basata su un'altra immagine "madre" dalla quale si ereditano tutte le cose che già ha più c'è la possibilità di customizzare alcune cose) che contiene tutto il necessario per eseguire un'applicazione in un ambiente containerizzato. Le immagini sono la base da cui vengono creati i **container**, proprio come un file ISO è la base per creare una macchina virtuale.

Un'immagine di container include:

- ✓ **Il codice dell'applicazione**
- ✓ **Le dipendenze necessarie** (librerie, framework, ecc.)
- ✓ **Un sistema di file di base** (ad es. una distribuzione Linux minimale)
- ✓ **Le variabili di configurazione**
- ✓ **Le istruzioni per l'esecuzione** (punto di ingresso, comandi, ecc.)

Le immagini sono **immutabili**, il che significa che non cambiano dopo essere state create. Se vuoi modificarle, devi crearne una nuova.

SU DOCKER HUB TROVI MOLTISSIMI ESEMPI DI IMMAGINI!! <https://hub.docker.com/>

Elemento	Immagine	Container
Cos'è?	Un modello statico e immutabile	Un'istanza in esecuzione di un'immagine
Persistenza	Immutabile	Può salvare dati, ma non permanentemente (a meno di usare volumi)
Modificabile?	No, serve creare una nuova immagine	Sì, ma le modifiche si perdono alla chiusura

Architettura di Docker

