

Capitolo 1 – Cos'è Docker

Docker è una piattaforma per creare, distribuire e eseguire **container**.

I container permettono di eseguire applicazioni in ambienti isolati, leggeri e portabili, senza la necessità di macchine virtuali complete.

1.1 Differenze tra Container e VM

Caratteristica	VM	Container
Isolamento	Completo (OS separato)	Processo separato, stesso OS
Peso	Pesante (GB per VM)	Leggero (MB per container)
Avvio	Lento (minuti)	Veloce (secondi)
Portabilità	Limitata al hypervisor	Elevata, qualsiasi host Docker
Risorse	Dedicate	Condivise con host

1.2 Componenti Principali di Docker

- **Docker Engine**
Servizio principale che esegue container e gestisce immagini.
- **Docker CLI**
Strumento a linea di comando per interagire con Docker.
- **Docker Daemon**
Processo in background che gestisce container e immagini.
- **Docker Hub / Registry**
Repository pubblico o privato per salvare e distribuire immagini.

1.3 Vantaggi dell'uso di Docker

- **Portabilità**: stesso container su qualsiasi host con Docker.
- **Isolamento**: applicazioni indipendenti tra loro.
- **Efficienza**: occupa meno risorse di una VM completa.
- **Ripetibilità**: stessi container identici in sviluppo, test e produzione.
- **Integrazione CI/CD**: ideale per pipeline DevOps.

1.4 Caso d'uso tipico

1. Uno sviluppatore crea un'applicazione web.
 2. Scrive un **Dockerfile** per definire ambiente e dipendenze.
 3. Costruisce un'immagine e la carica su Docker Hub.
 4. In produzione, il container viene eseguito senza preoccuparsi del sistema operativo host.
-

1.5 Conclusione

Docker permette di astrarre l'applicazione dall'infrastruttura sottostante, rendendo:

- Sviluppo più veloce
- Deploy più semplice
- Gestione di ambienti coerente e sicura

Nel prossimo capitolo entreremo nel dettaglio dell'**architettura di Docker**, spiegando come Engine, Daemon e registri lavorano insieme.

Revision #1

Created 2026-02-26 14:18:03 UTC by Pe

Updated 2026-02-26 14:18:35 UTC by Pe