05 Febbraio 2025

# Containers

Docker → ambiente dove mettere le applicazioni di riferimento (di virtualizzazione) → applicazioni che puoi dare ai clienti persone che servono i file

**Definizione di Container in Informatica**

Un **container** è una struttura o un'unità che incapsula dati, risorse o applicazioni per facilitarne la gestione, il trasporto e l'esecuzione in ambienti controllati. Il concetto di container può essere applicato in diversi ambiti dell'informatica, tra cui la virtualizzazione, il web development e la gestione dei dati.

---

**1. Container Software (Docker, Kubernetes)**

I container software sono ambienti isolati che includono codice, dipendenze e configurazioni necessarie per eseguire un'applicazione.

✅ **Vantaggi:**

- **Portabilità**: Funzionano su qualsiasi piattaforma che supporta i container.
- **Isolamento**: Evitano conflitti tra applicazioni diverse.
- **Leggerezza**: Consumo di risorse inferiore rispetto alle macchine virtuali.
- **Scalabilità**: Facili da replicare e distribuire su larga scala.

❌ **Svantaggi:**

- **Sicurezza**: Condividono il kernel del sistema operativo, aumentando il rischio di attacchi.
- **Gestione complessa**: Richiedono strumenti come Kubernetes per orchestrare più container.
- **Persistenza dati**: I dati possono andare persi se non si usano volumi esterni.

---

**2. Container nel Web Development (HTML & CSS)**

Un container in web development è un elemento HTML (come  o <section>) che organizza il layout della pagina.

✅ **Vantaggi:**

- **Organizzazione del contenuto**: Aiuta a strutturare il layout.
- **Facile personalizzazione**: Consente l'applicazione di stili CSS mirati.
- **Compatibilità con Flexbox e Grid**: Facilita la creazione di layout responsivi.

❌ **Svantaggi:**

- **Eccessivo annidamento**: Troppi container possono rendere il codice difficile da leggere.
- **Performance**: Un uso errato può rallentare il rendering della pagina.

---

**3. Container nelle Strutture Dati (Array, Liste, HashMap)**

Un container in programmazione è una struttura dati che memorizza e organizza oggetti (es. array, liste, stack).

✅ **Vantaggi:**

- **Efficienza**: Permette la gestione e il recupero rapido dei dati.
- **Flessibilità**: Disponibile in diverse forme (dinamico, statico, indicizzato, non indicizzato).
- **Struttura modulare**: Facilita la programmazione e il riutilizzo del codice.

❌ **Svantaggi:**

- **Uso errato della memoria**: Alcuni container consumano più memoria di quanto necessario.
- **Tempo di accesso variabile**: Strutture diverse hanno tempi di accesso differenti (es. liste vs array).

---

💡 **Conclusione**

I container sono strumenti fondamentali in vari ambiti dell'informatica. La loro efficacia dipende dal contesto d'uso e dalla corretta implementazione. 🚀

**Docker** → crea e gestisce i container → container tecnologies

**Docker** → puoi usare a riga di comando oppure dall'applicazione web

**Docker** → come funzionamento sotto c'è Linux

**Docker** → tecnologia che serve per sviluppare un applicazione → solo quello che serve → con tutte le dipendenze necessarie

**Docker** → unita organizzativa per scambiare molte funzionalità in poco spazio si sposta i dati rimangono uguali → sposto dati rimane tutto uguale

---

✅ **Vantaggi:**

- Più facile spostare i dati da un dispositivo ad un altro e anche molto più veloce
- diversi elementi di produzione si adatta molto facilmente

- Condividere ambiente di sviluppo con tutte le persone con le caratteristiche che ti servono → più container collegati tra di loro

**Framework** → insieme di dati che servono per collegarsi alle sue opzioni corrette con le varie funzioni

**Definizione di Framework**

Un **framework** è un insieme di strumenti, librerie e regole che fornisce una struttura predefinita per lo sviluppo di software. Aiuta i programmatori a scrivere codice in modo più efficiente, evitando di dover costruire tutto da zero.

---

✅ **Vantaggi di un Framework**

- **Maggiore produttività**: Fornisce componenti pronti all'uso.
- **Struttura organizzata**: Impone regole e schemi per scrivere codice più ordinato.
- **Sicurezza**: Include meccanismi di protezione contro vulnerabilità comuni.
- **Manutenibilità**: Favorisce il riutilizzo del codice e facilita gli aggiornamenti.
- **Comunità e supporto**: Spesso ha una vasta documentazione e una community attiva.

---

❌ **Svantaggi di un Framework**

- **Curva di apprendimento**: Alcuni framework sono complessi e richiedono tempo per essere imparati.
- **Rigidità**: Impone una struttura che può limitare la libertà dello sviluppatore.
- **Overhead**: Può introdurre codice e funzionalità non necessarie, rallentando le prestazioni.

---

**Esempi di Framework**

- **Web Development**: React, Angular, Vue.js
- **Backend**: Django (Python), Spring (Java), Express.js (Node.js)
- **Mobile**: Flutter, React Native
- **Machine Learning**: TensorFlow, PyTorch

💡 **Conclusione**: Un framework semplifica lo sviluppo, ma richiede studio per sfruttarne al meglio le potenzialità! 🚀

**Virtual Machine (VM) vs Docker (Containerization)**

**Virtual Machine (VM)**

Una **macchina virtuale** (VM) è un ambiente emulato che simula un sistema operativo completo. Utilizza un hypervisor (es. VMware, VirtualBox, Hyper-V, KVM) per eseguire più sistemi operativi su una singola macchina fisica.

**Vantaggi delle VM**

✅ **Isolamento completo** – Ogni VM esegue un intero sistema operativo, garantendo un forte isolamento tra le applicazioni.
✅ **Compatibilità** – Supporta qualsiasi sistema operativo, indipendentemente dall'host.
✅ **Sicurezza** – L'isolamento tra VM rende difficile la compromissione di una VM da parte di un'altra.
✅ **Affidabilità** – Adatto per eseguire applicazioni legacy o software che richiedono una configurazione complessa.

**Svantaggi delle VM**

❌ **Consumo di risorse** – Ogni VM esegue un intero sistema operativo, occupando molta RAM, CPU e spazio su disco.
❌ **Lentezza nell'avvio** – Avviare una VM richiede tempo perché bisogna caricare un intero OS.
❌ **Meno portabilità** – Spostare e ridistribuire VM è più pesante rispetto ai container.

---

**Docker (Containerization)**

Docker è una piattaforma di **containerizzazione** che permette di eseguire applicazioni in ambienti isolati (container) senza avviare un intero sistema operativo.

**Vantaggi di Docker**

✅ **Leggero** – I container condividono il kernel dell'OS host, riducendo il consumo di risorse.
✅ **Avvio rapido** – I container si avviano in pochi secondi.
✅ **Portabilità** – I container sono indipendenti dalla piattaforma e funzionano su qualsiasi macchina con Docker.
✅ **Scalabilità** – Permette di eseguire e gestire molteplici istanze di un'applicazione con facilità.
✅ **Facile gestione** – Si integra bene con sistemi di orchestrazione come Kubernetes.

**Svantaggi di Docker**

❌ **Isolamento limitato** – I container condividono il kernel dell'host, quindi possono avere problemi di sicurezza rispetto alle VM.
❌ **Compatibilità limitata** – Funziona meglio su Linux; su Windows richiede workaround o WSL2.
❌ **Meno adatto per applicazioni complesse** – Non è ideale per applicazioni che necessitano di un intero sistema operativo o dipendenze molto specifiche.

**Quando scegliere VM o Docker?**

| Scenario | VM | Docker |
|---|---|---|
| Eseguire software con dipendenze OS-specifiche | ✅ | ❌ |
| Massima sicurezza e isolamento | ✅ | ❌ |
| Avvio rapido e leggerezza | ❌ | ✅ |
| Deploy di microservizi | ❌ | ✅ |
| Applicazioni legacy | ✅ | ❌ |
| Scalabilità orizzontale (Kubernetes, cloud) | ❌ | ✅ |

**Docker** → aiuta a gestire e creare strutture di container

**Linux** → pronto a gestire container

## Configurazione Docker Desktop

**Esegui i Seguenti comandi nel cmd come amministratore**

Riattivazione hypervisior → bcdedit /set hypervisorlaunchtype auto

**Esegui i Seguenti comandi nel poweshell come amministratore**

enable-windowsoptionalfeatures -online -featurename microsoft-hyper-v -all → aggiunge caratteristiche Hyper Visor che servono per il container rispondi y

enable-windowsoptionalfeature -online -featurename container -all → aggiunge i requisiti per docker rispondi y

**Esegui i Seguenti comandi nel cmd come amministratore**

wsl —update → aggiorni wsl

Crei account di docker da docker desktop installando l'applicazione docker desktop dal file .exe

Riavi PC → e avvi docker desktop

Username mio → nicolamarano

per vedere se funziona vai su cmd → scrivi docker e vedi le varie configurazioni possibili

```
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes

Global Options:
      --config string      Location of client config files (default
                           "C:\\Users\\Marano Nicola\\.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket to connect to
  -l, --log-level string   Set the logging level ("debug", "info",
                           "warn", "error", "fatal") (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\Marano Nicola\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\Marano Nicola\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\Marano Nicola\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

C:\Users\Marano Nicola>ù
```

metti installazione nelle impostazioni → start docker when you sign in your conputer

## Creazione Primo Container di una pagina web in rete

scarichi estensioni in visual studio code → scaricando estensioni docker exstension pack e prettier e docker

Container → basati su immagine

Apri terminale in visual studio code → con ctrl + maiusc + ò

docker build .→ crea immagine sul container nel docker → dove appoggia container interpretando le istruzioni del docker file

docker images → vedi immagini presenti nel disco

docker run -p porta:porta nome immagine → avvi docker l'immagine sta girando

fai nel browser cerci 127.0.0.1 o localhost:porta → e vedi il sito funzionare

docker ps → vedo container che stanno funzionando in questo momento → quando tempo è creato e da quanto gira → su che porta funziona → con un nome casuale

docker stop nome container → bloccata il container

docker gira cancella immagine → non gira più il container

1 immagine → anche + container

su docker hub vedi immagine già presenti che tu puoi usare

docker run servizio → crea un container con il servizio che mi interessa

docker run node → crei container creando il container con quel servizio→ fa girare container in base a quello che scarichi

docker ps -a → vedi docker attivi e non attivi (tutti)

docker run -it node → esegue node

11 Febbraio 2025

Puoi anche eliminare immagini già create e anche i container

docker -rmi  prune→ elimini tutte le immagini create

docker run nome servizio → creo servizio basato sul servizio che ti serve e dopo la scarica lui in automatico e crea il container

docker images → vedi immagine

docker stop nome container → blocca il container

docker rm nome o id container → elimina container

docker build . → esegui file dockerfile e crea immagine

docker run id → esegui il container

docker rm nome container o id → elimini container

docker run -p porta destinazione iniziale:porta destinazione finale nome container o id

Immagini in sola lettura → i container fanno funzionare il sito

Docker –help → aiuto ai comandi docker

Docker ps –help → tutti opzioni dopo ps

docker start nome id o nome container → fa ripartire container

Attack mode → blocco terminale → con il container

deattack mode → terminale non bloccato solo se web

Docker run -it nome id o nome dell'immagine di riferimento → esegui in modo iteratico il software di riferimento

docker container prune → elimino tutti i container

Docker image inspect nome immagine → tutti i dati dell'immagine