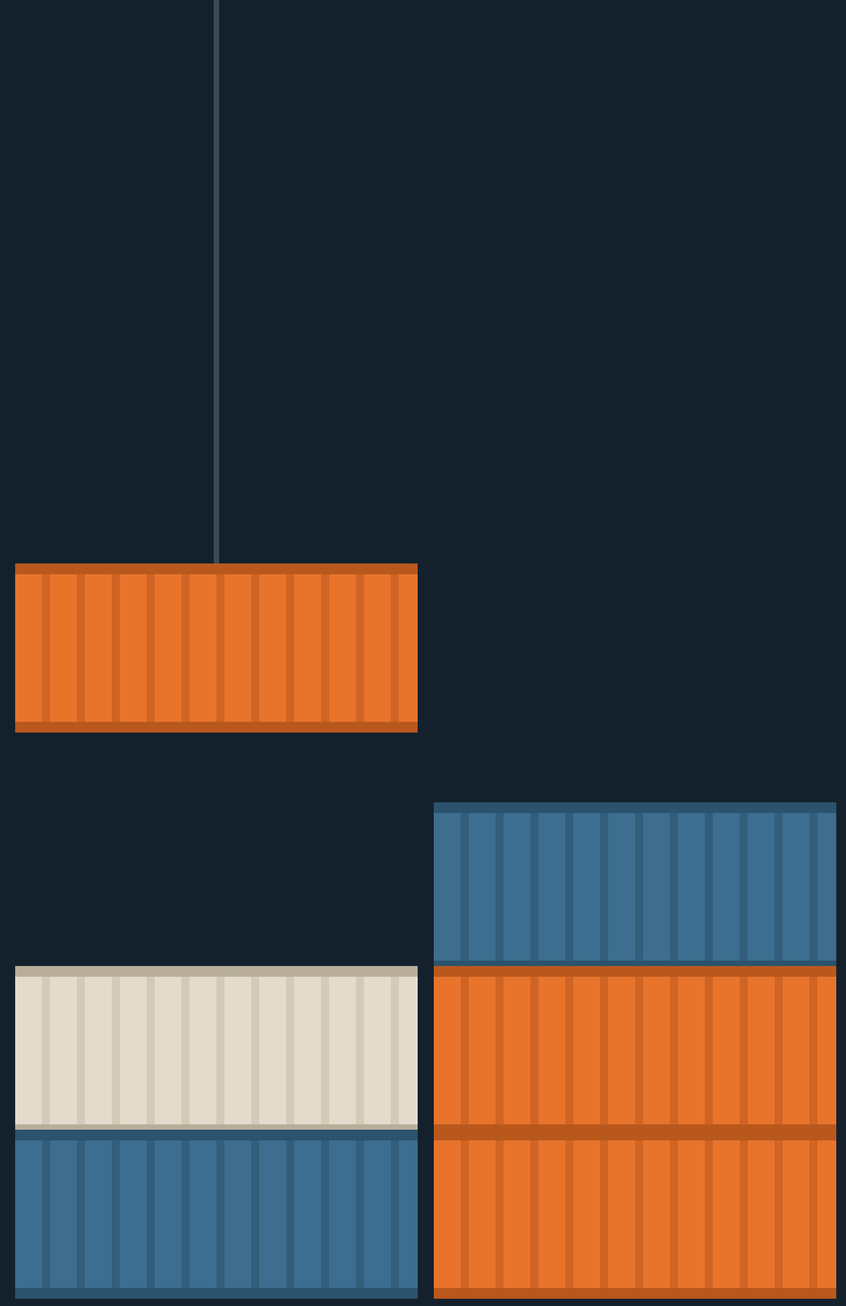
Linux Day 2026

Casorate Sempione (Va) · 2026-10-24

INTRODUZIONE AI CONTAINER

Introduzione a Podman

Cosa sono, a cosa servono e come si usano: un’introduzione per chi parte da zero.



Relatore: Lorenzo Lobba

ILS (Italian Linux Society) Casorate Sempione (Va)

1 · I CONTAINER

«Sul mio computer funziona!»

Il mio PC

`libreria v2.1 · Ubuntu 24.04`

✓ **Funziona**

Il server

`libreria v1.4 · Debian 11`

⚠ **Errore all'avvio**

Il cliente

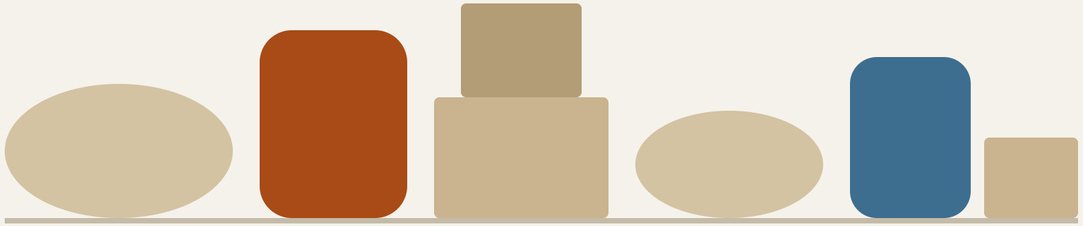
`libreria assente · Windows 11`

⚠ **Non parte**

Un programma dipende da tutto ciò che trova intorno: **sistema operativo, librerie, versioni, configurazioni**.

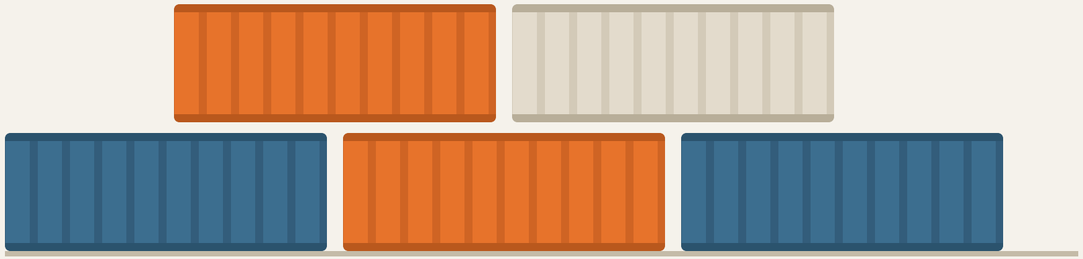
1 · I CONTAINER

L’idea viene dal trasporto merci



Prima

Sacchi, casse e barili caricati a mano, uno per uno.



Dal 1956

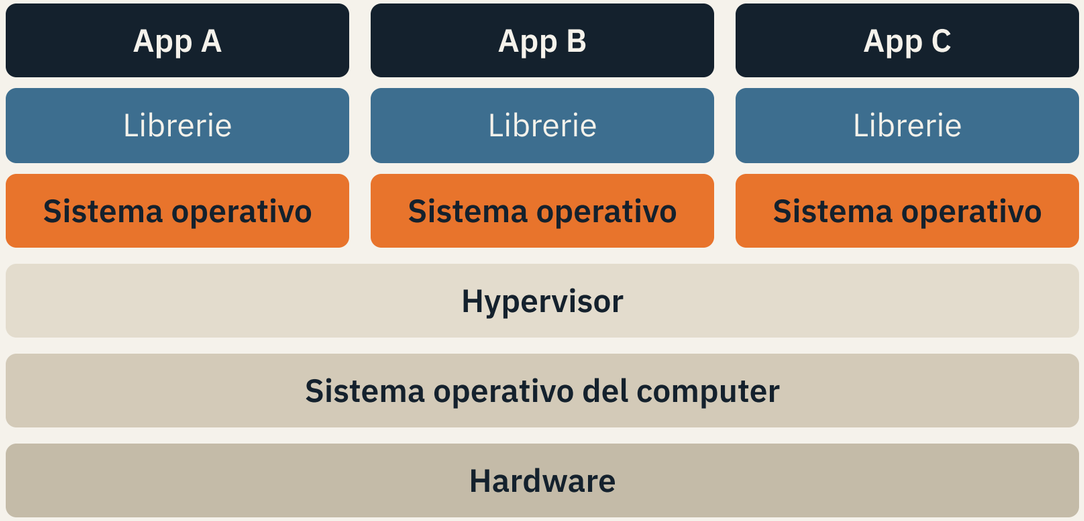
Scatole di misura standard: da nave a treno a camion, senza mai aprirle.

Nel software è lo stesso: il programma viaggia insieme a tutto ciò che gli serve, in una scatola standard che funziona su qualsiasi computer.

1 · I CONTAINER

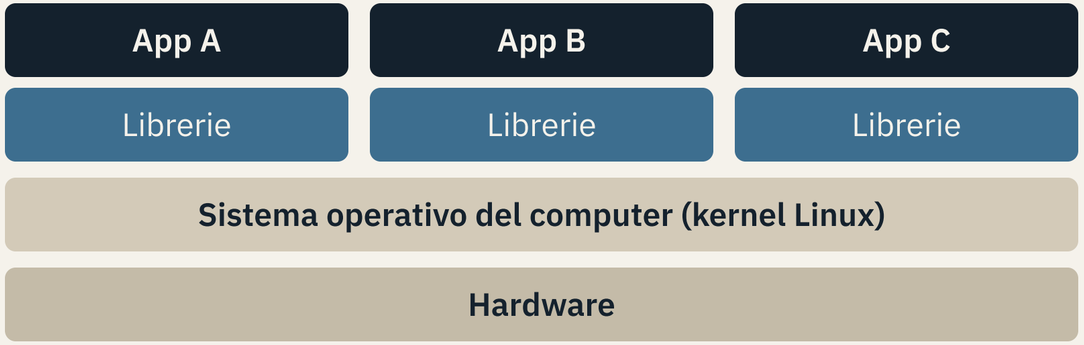
Container e macchine virtuali

Macchine virtuali



Ogni VM ha un sistema operativo intero: **gigabyte** di spazio, avvio in **minuti**.

Container



I container condividono il kernel: **megabyte** di spazio, avvio in **secondi**.

1 · I CONTAINER

Tre parole da conoscere

Registro

Il magazzino online delle immagini: Docker Hub, Quay.io o uno aziendale.

Come un app store

`pull` →

Immagine

Il pacchetto pronto: programma, librerie e configurazione. Non cambia.

Come uno stampo per biscotti

`run` →

Container

Un’immagine in esecuzione. Da una sola immagine, tanti container.

Come i biscotti: tutti uguali

`podman pull` scarica un’immagine

`podman run` avvia un container

2 · PODMAN

Da Docker a Podman

2013

Docker rende i container facili da usare.

2015

Nasce lo standard **OCI** per immagini e container.

2019

Red Hat rilascia **Podman 1.0**, senza demone.

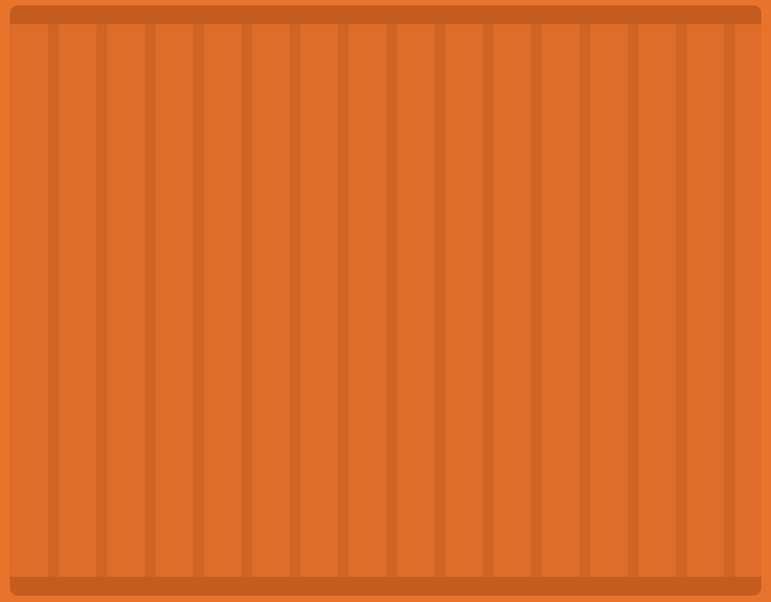
Oggi

Due motori diversi, **le stesse immagini**.

OCI (Open Container Initiative): le regole comuni che rendono le immagini compatibili tra Docker, Podman e gli altri motori.

PODMAN IN SINTESI

Senza demone. Senza root. Compatibile.

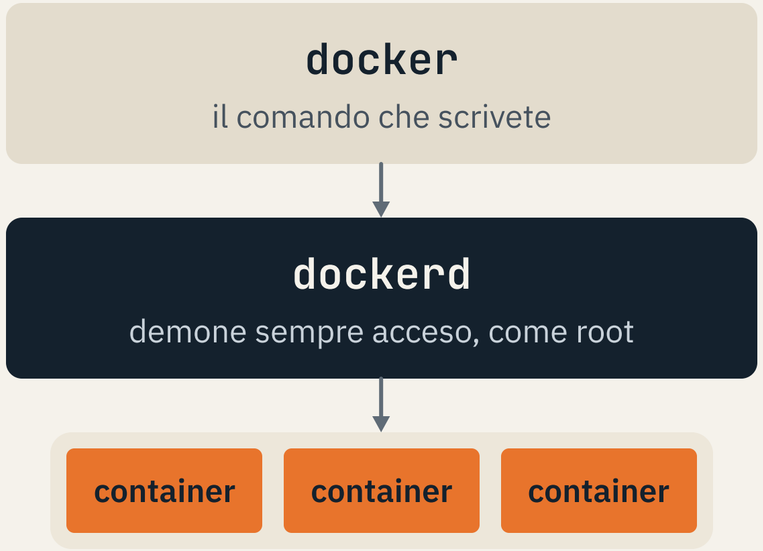


Usa le stesse immagini e quasi tutti i comandi di Docker, ma funziona in un altro modo.

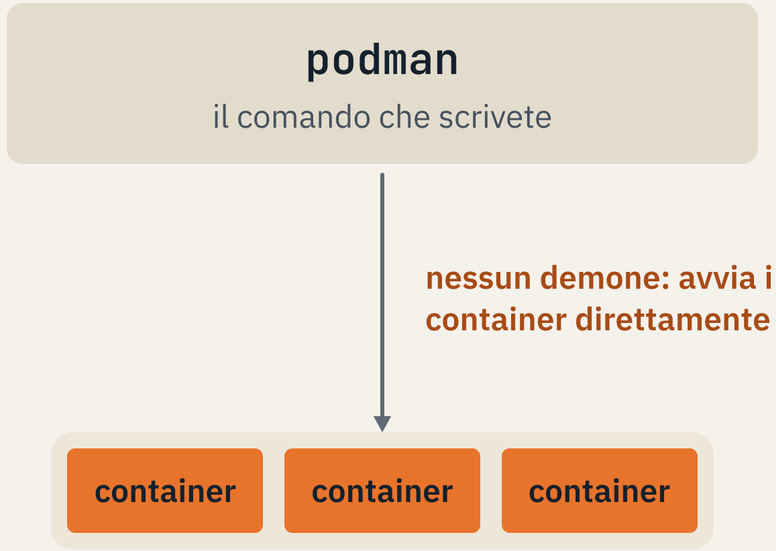
2 · PODMAN

Senza demone

Un **demone** è un programma di servizio che resta sempre acceso in background.



Tutto passa da un servizio centrale.

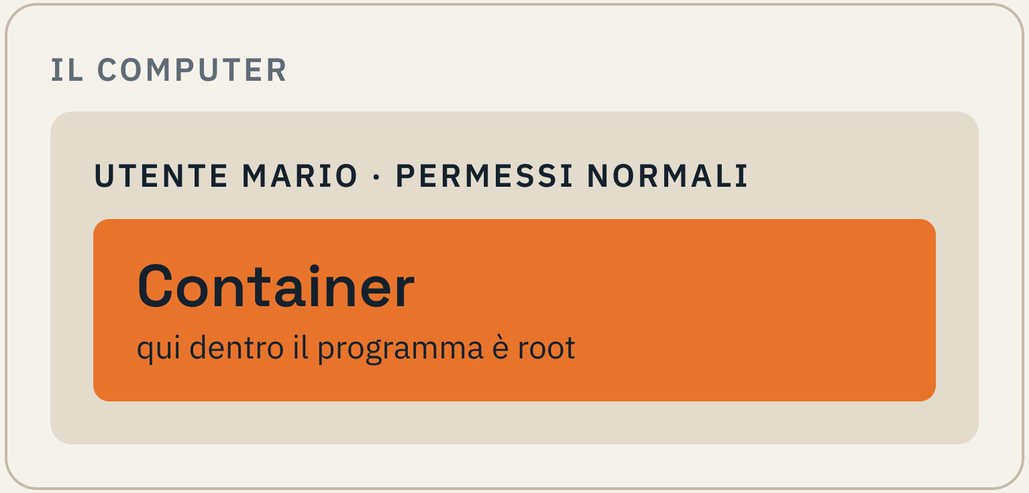


Ogni container è un normale processo.

2 · PODMAN

Senza root

Root è l’amministratore di Linux: l’utente che può fare tutto sul sistema.



`root nel container = mario fuori`

- Se un container viene attaccato, il danno resta dentro i permessi di **mario**.
- Ogni utente avvia i propri container, senza chiedere all’amministratore.
- Docker può farlo, ma va configurato. In Podman è **il comportamento predefinito**.

2 · PODMAN

Compatibile con Docker

```
docker run -d -p 8080:80 docker.io/library/nginx
podman run -d -p 8080:80 docker.io/library/nginx
alias docker=podman # per chi è abituato a Docker
```

Stesse immagini

Docker Hub, Quay.io e gli altri registri: è tutto standard OCI.

Stessi comandi

run, ps, logs, build, pull, push: stessi nomi e stesse opzioni.

Stessi file

Dockerfile e file di Docker Compose funzionano anche con Podman.

2 · PODMAN

Pod e servizi di sistema

Pod da qui il nome: POD MANager

Container che lavorano insieme e condividono la rete, come in Kubernetes.

pod: blog

App web | Database

si avviano e si fermano insieme

`podman kube generate` → un file per Kubernetes

Servizi di sistema

Con Quadlet diventa un servizio: parte all’accensione e si riavvia da solo.

```
# /etc/containers/systemd/web.container
[Container]
Image=docker.io/library/nginx
PublishPort=8080:80
[Service]
Restart=always
[Install]
WantedBy=multi-user.target
```

2 · PODMAN

Docker o Podman?

	Docker	Podman
Architettura	Un demone centrale (dockerd)	Nessun demone
Permessi	Root di default, rootless opzionale	Rootless di default
Immagini e comandi	Standard OCI	Gli stessi di Docker
Più container insieme	Docker Compose	Pod, podman compose, Kubernetes
Avvio automatico	Gestito dal demone	Servizi systemd (Quadlet)
App per Windows e Mac	Docker Desktop: a pagamento per le aziende grandi*	Podman Desktop: gratuita
Diffusione	La più usata, tanti esempi online	Predefinito su Red Hat e Fedora

* Gratuito per uso personale e per aziende sotto i 250 dipendenti e i 10 milioni di dollari di fatturato annuo.

3 · IN PRATICA

Demo: un server web in un minuto

```
# 1 · scarica l'immagine e avvia il container
podman run -d --name web -p 8080:80 docker.io/library/nginx
# 2 · apri il browser su http://localhost:8080
# 3 · elenca i container attivi
podman ps
# 4 · leggi i messaggi del server
podman logs web
# 5 · ferma e cancella il container
podman rm -f web
```

LE OPZIONI

`-d`

in background: il terminale resta libero

`--name web`

un nome per ritrovarlo

`-p 8080:80`

porta 8080 del PC → porta 80 del container

3 · IN PRATICA

Creare la propria immagine

Containerfile · lo stesso formato del Dockerfile

```
FROM docker.io/library/python:3.12-slim   si parte da un’immagine con Python
RUN pip install flask                     si installano le librerie
COPY app.py /app/                         si aggiungono i propri file
CMD ["python", "/app/app.py"]             il comando da eseguire all’avvio
```

```
podman build -t mia-app .
```
costruisce l’immagine

```
podman run -d -p 5000:5000 mia-app
```
la avvia, uguale su ogni computer

3 · IN PRATICA

Come iniziare

Windows e Mac

Installate Podman Desktop: prepara da solo la macchina Linux per i container.

`podman-desktop.io`

Linux

Il pacchetto si chiama podman:

```
sudo apt install podman
sudo dnf install podman
```

apt: Debian e Ubuntu · dnf: Fedora e Red Hat

Il primo comando

```
podman run quay.io/podman/hello
```

Se compare il messaggio di benvenuto di Podman, è tutto pronto.

Per approfondire: podman.io · docs.podman.io · podman-desktop.io

CHI FA COSA · CONTAINER LINUX

Dall’avvio alla supervisione

Compito	Podman	Docker
Comando	podman	docker (CLI)
Coordinamento	Podman, senza demone centrale	dockerd, il demone centrale
Gestione dei container	Podman	containerd
Avvio e isolamento	crun oppure runc (runtime OCI)	runc (runtime OCI, predefinito)
Supervisione e output	conmon, per ogni container	containerd-shim, per ogni container

Podman: coordina · crun/runc avvia · conmon sorveglia.
Docker: dockerd coordina · containerd gestisce · runc avvia · lo shim sorveglia.

Il runtime normalmente termina dopo l’avvio. Il monitor resta attivo.
Con podman run -d, il comando Podman termina e il container continua a funzionare.

RIEPILOGO

1. Un container è un programma con tutto ciò che gli serve: funziona uguale ovunque.
2. Podman usa le stesse immagini e gli stessi comandi di Docker.
3. In più: niente demone, niente root, pod e servizi di sistema.

Domande?

