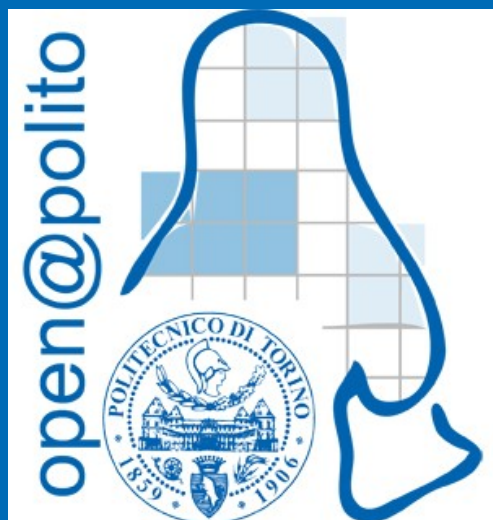




CONTAINER: Cosa, Come e Perche'

Con il supporto di:



Di Andrea Gabbani

Immagina...

- Vuoi installare Nextcloud...
- Avrai bisogno di:

Immagina...

- Vuoi installare Nextcloud...
- Avrai bisogno di:



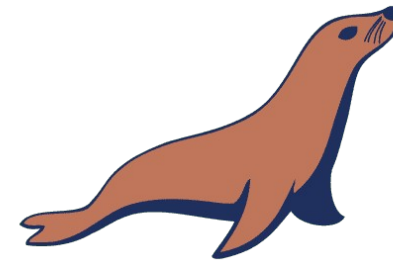
Immagina...

- Vuoi installare Nextcloud...
- Avrai bisogno di:



Immagina...

- Vuoi installare Nextcloud...
- Avrai bisogno di:

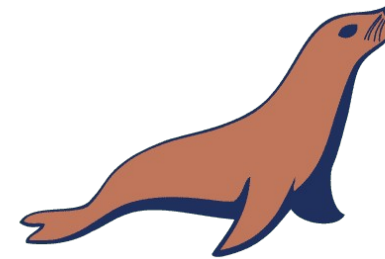


MariaDB

(Con relativo plugin php)

Immagina...

- Vuoi installare Nextcloud...
- Avrai bisogno di:



MariaDB

(Con relativo plugin php)



Immagina...

- Vuoi creare un video?
- Avrai bisogno di:

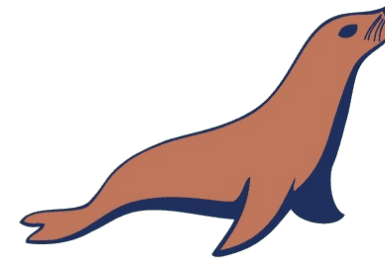
samba



FFmpeg



gdLibrary



MariaDB

(Con relativo plugin php)

Immagina...

- Vuoi un sito...
- Avrai bisogno di:

samba



Ffmpeg

redis

MariaDB



(Con relativo plugin php)

Immagina..

- Vuoi un sito?
- Avrai bisogno di



Ampeg

redis

MariaDB

(Con relativo plugin php)

Cosa e' un Container?

- Software e dipendenze impacchettate assieme
 - Si evita il "dependency hell" visto prima...
- Riproducibili
 - "It works on my machine."
- Immutabili
- Isolati

Perche' un Container?

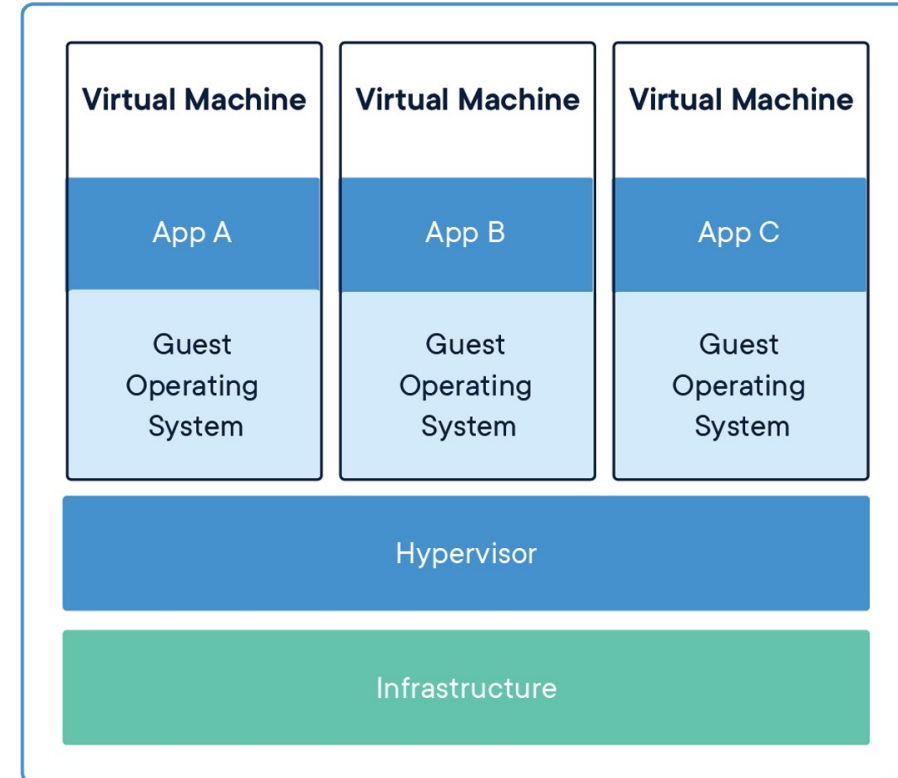
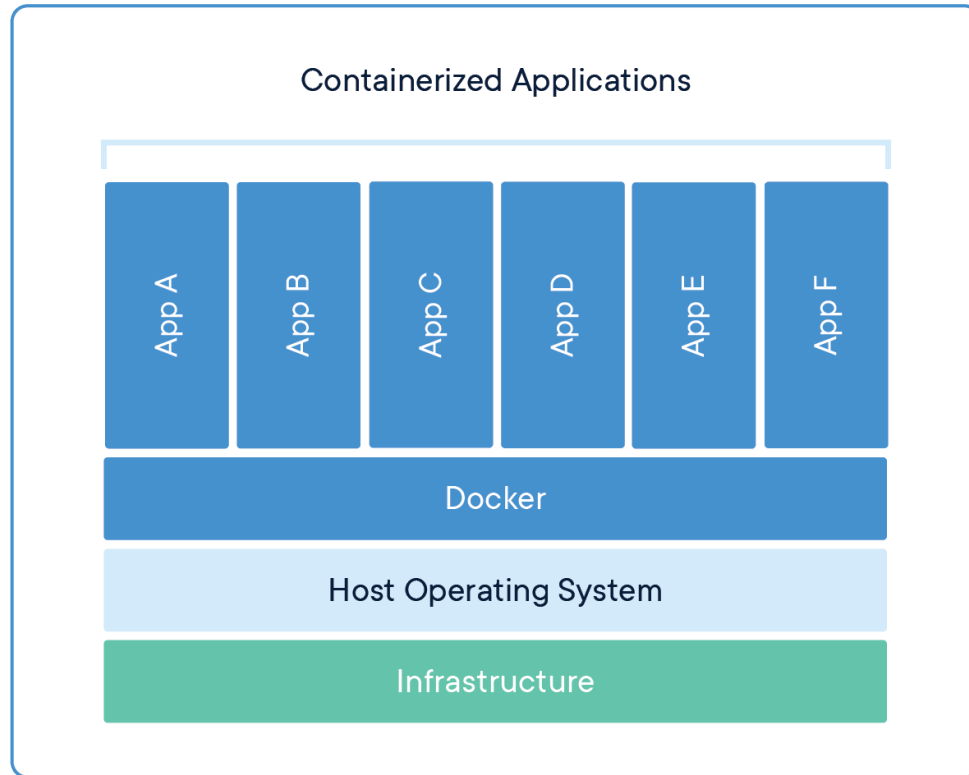
- **Utente**
 - Deployment veloce e semplice di applicazioni di ogni genere
- **Developer**
 - Ambienti di test facilmente controllabili e riproducibili
- **Distributore**
 - Packaging "universale" di software

Come funziona un Container?

- Immagini
 - I pacchetti che contengono il software, dipendenze, script per il setup...
- Container Engine
 - Il software che crea, gestisce e manipola i Container, a partire dalle immagini
- Namespace e cgroups
 - Strumenti pre-esistenti di linux per isolare e regolare i processi
- Volumi e networking
 - Per far interagire i Container con il mondo esterno

Forse starai pensando...

- “Ma quindi e’ tipo una Virtual Machine?”
- Non proprio...



Immagini

- Contengono tutto il necessario per eseguire un programma
- Contenute in un filesystem isolato
- Definite in un file chiamato "Dockerfile" o "Containerfile"
- Rese disponibili sulle Container Registry
- Costruite "a strati"

Docker

- Il Container Engine piu' famoso, nasce nel 2013
- Semplice da usare
- Ampiamente supportato
- Enorme libreria di immagini – Docker Hub
- Estendibile – Docker API
- Docker Daemon
- Docker Compose

Podman

- Container Engine nato nel 2018
- Creato da RedHat
- Sostituto “drop-in” a Docker
 - alias docker=podman
- Pod – Gruppi di Container
- Supporto nativo per la modalita’ “rootless” – Niente daemon!
- Integrazione con systemd
- Update automatici