From 0 to Docker

Introduzione a utilizzo di Docker da parte di uno sviluppatore

Chi sono

g.mikele@gmail.com
michele@qricambi.com

mikele_g

kidrock

Perchè

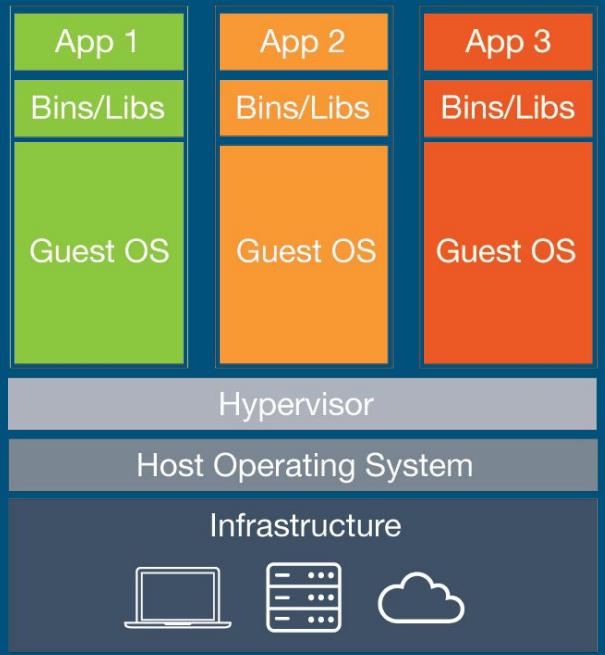
- Possibilità di testare completamente lo stack dell’applicazione
- Rimuove la classica frase *“A me in locale funziona”*
- Ottimizzare l’utilizzo delle risorse

Cos’è Docker

Piattaforma open-source per la creazione di Containers *Portabile, leggeri e autosufficienti* per eseguire applicazioni

DockerConf 2017 parlano di 14 milioni di host, 900mila apps, 3300 partecipanti al progetto e 170mila membri della community e 12 miliardi di immagini scaricate.

Differenza tra virtual machine e container



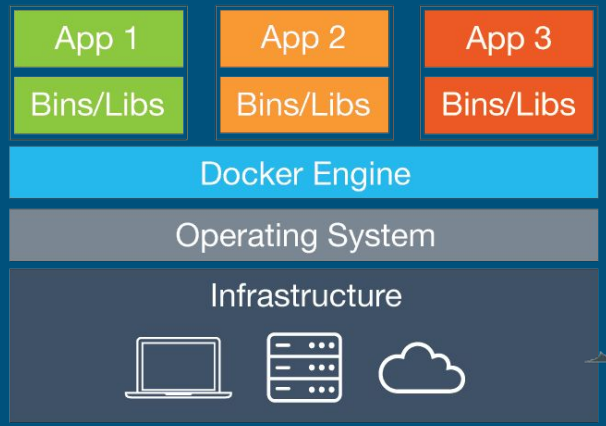
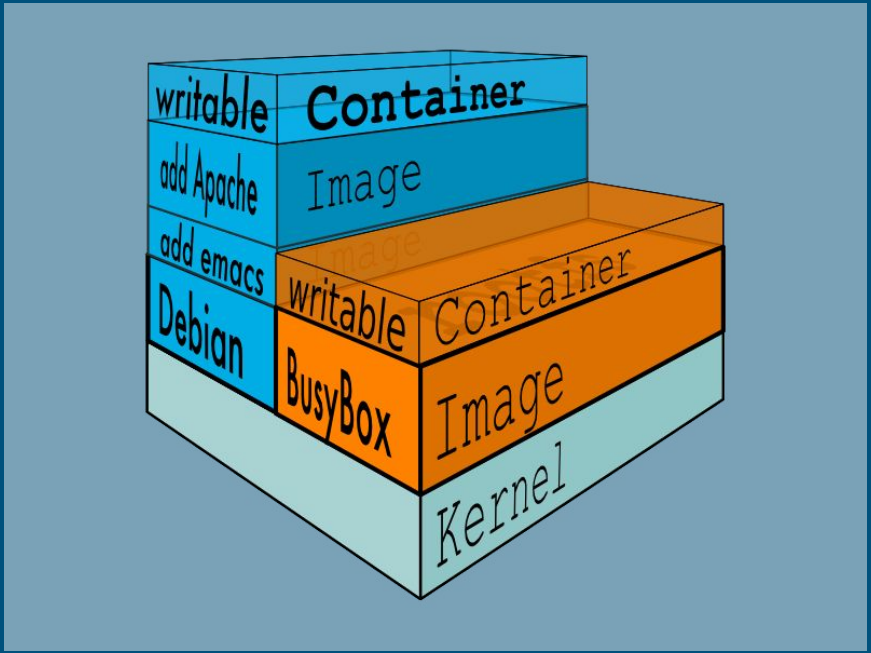
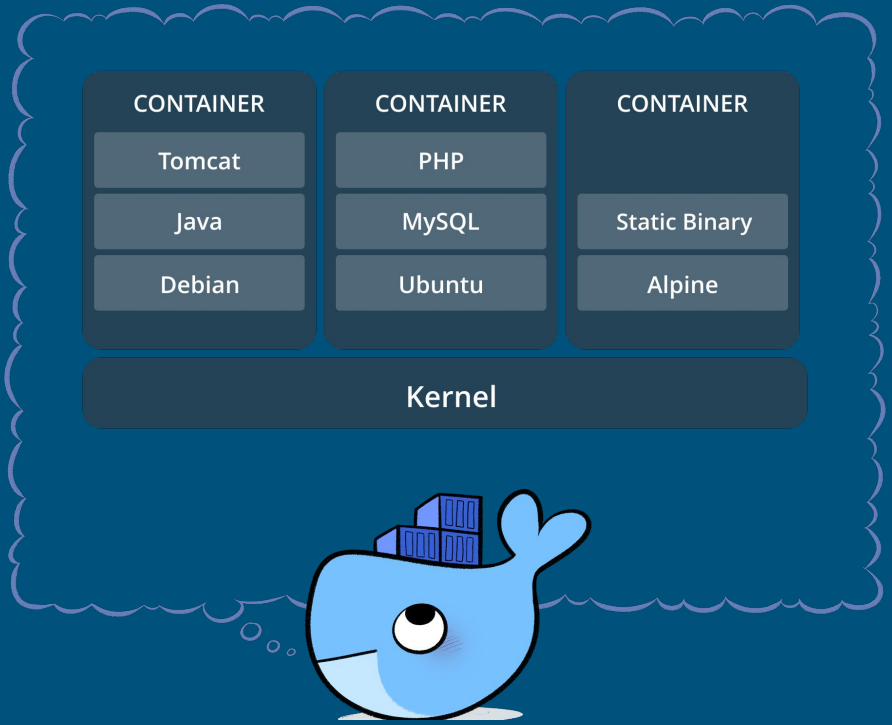


Immagine docker



Container/Image



Docker su Windows

- Disponibile nativamente su Windows Server 2016
- I principi di base sono gli stessi di Linux: accesso controllato e isolato delle risorse del kernel
- Un linux container NON può essere ospitato su host windows e viceversa, perché i kernel sono diversi, le API utilizzate sono specifiche per i due sistemi.
- Il client di Docker supporta entrambe le API: un solo prodotto per entrambi i sistemi operativi

Deploy di un’applicazione

Sviluppare con docker

- Dockerfile
- Docker-compose
- Docker registry

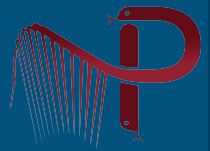
Demo

#1

```
FROM ubuntu:latest
LABEL maintainer = "g.mikele@gmail.com"
RUN apt update -y
RUN apt install -y python-pip python-dev build-essential
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Demo

#2
Qualche miglioramento

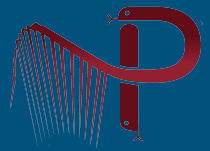


```
FROM ubuntu:latest
LABEL maintainer = "g.mikele@gmail.com"
RUN apt update -y
   &&  apt install -y python-pip python-dev build-essential
   && rm -rf /var/lib/apt/lists/*
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Demo

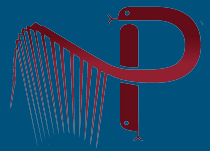
#3
Sostanziale miglioramento

```
FROM python:3-alpine
LABEL maintainer = "g.mikele@gmail.com"
COPY . /app
WORKDIR /app
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 5000
ENTRYPOINT ["python"]
CMD ["app.py"]
```



Demo

#4
Multi-Stage



```
FROM microsoft/aspnetcore-build AS build
LABEL maintainer = "g.mikele@gmail.com"
WORKDIR /code
COPY . .
RUN dotnet restore && dotnet publish --output /output

FROM microsoft/aspnetcore
LABEL maintainer = "g.mikele@gmail.com"
COPY --from=build /output /app
WORKDIR /app
EXPOSE 5000
ENTRYPOINT [ "dotnet" ]
CMD ["/app/WebApiSwaggerAndMvc.WebApi.dll"]
```

Demo

#5
Docker-Compose

```
version: '3'
services:
  proxy:
    image: pyre_proxy
    restart: always
    build: nginx
    container_name: pyre_proxy
    links:
      - pyreWWW
    ports:
      - "80:80"
    volumes:
      - ./logs/nginx:/var/log/nginx/
  pyreWWW:
    image: pyre_dot_it
    restart: always
    build: https://github.com/kidrock/PyRE.git
    container_name: pyre_www
    ports:
      - "8080"
```

Q&A

Grazie