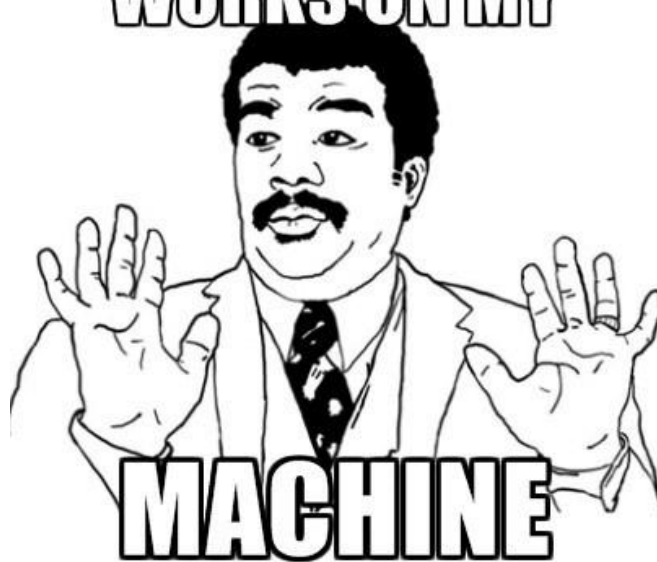


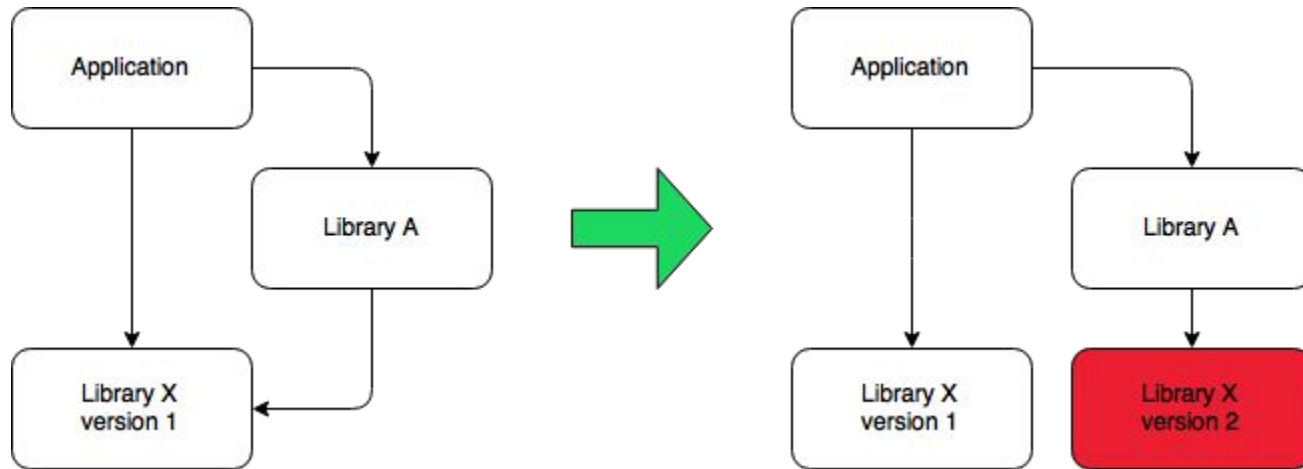
docker

WORKS ON MY



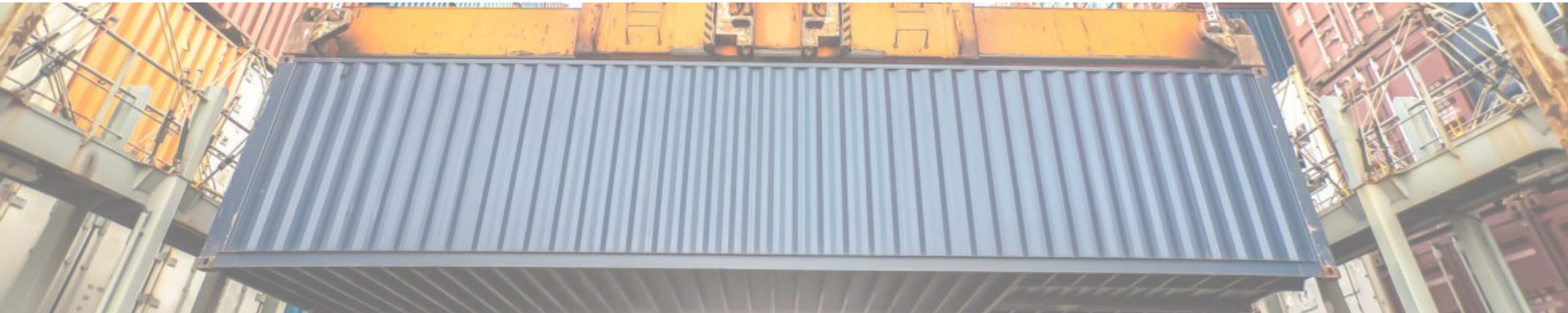
MACHINE

Dependency Hell



O que é container?

Um contêiner é uma unidade padrão de software que empacota o código e todas as suas dependências para que o aplicativo seja executado de maneira rápida e confiável de um ambiente de computação para outro.

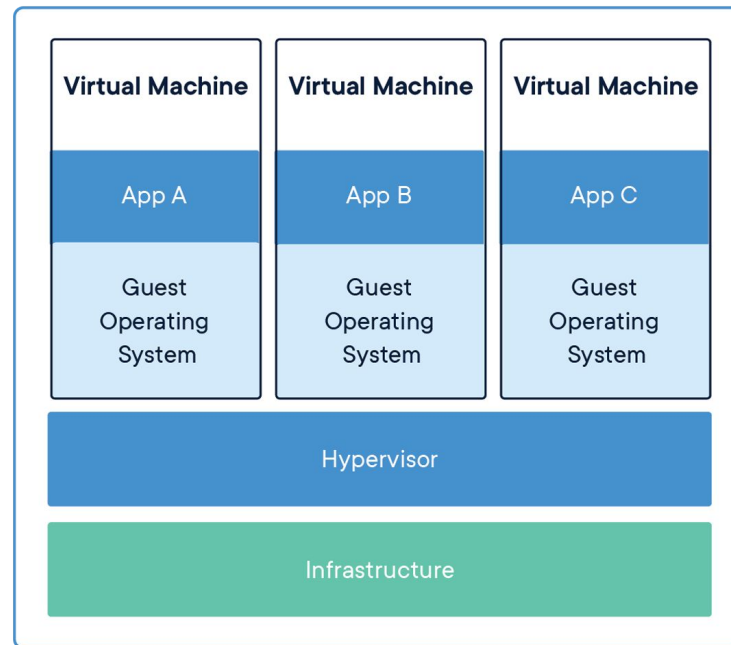
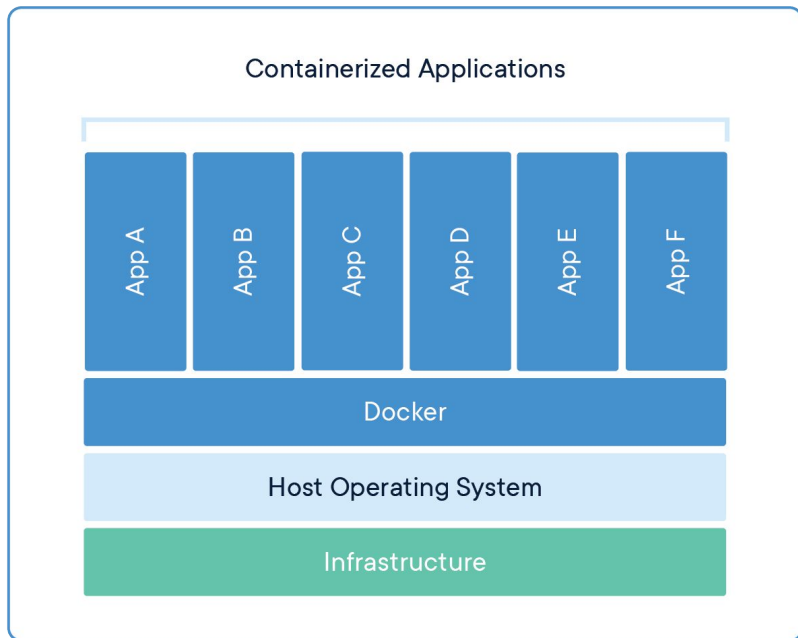




Container vs VM

Os contêineres e as máquinas virtuais têm benefícios semelhantes de isolamento e alocação de recursos, mas funcionam de maneira diferente porque os contêineres virtualizam o sistema operacional em vez do hardware. Os contêineres são mais portáteis e eficientes.

Container vs VM





O que é Docker?

- Ferramenta Open Source
- Gerenciador de containers
- Escrito em Go (Google)
- Criado pelo dotCloud, foi disponibilizado para a comunidade em 2013 com o nome de "Docker"



Por que utilizar?

- Ambientes semelhantes
- Aplicação toda contida em uma imagem
- Padronização
- Comunidade



Instalando o Docker

```
~ ➤ curl -sSL https://get.docker.com | sh
```

```
~ ➤ sudo usermod -aG docker <user>
```



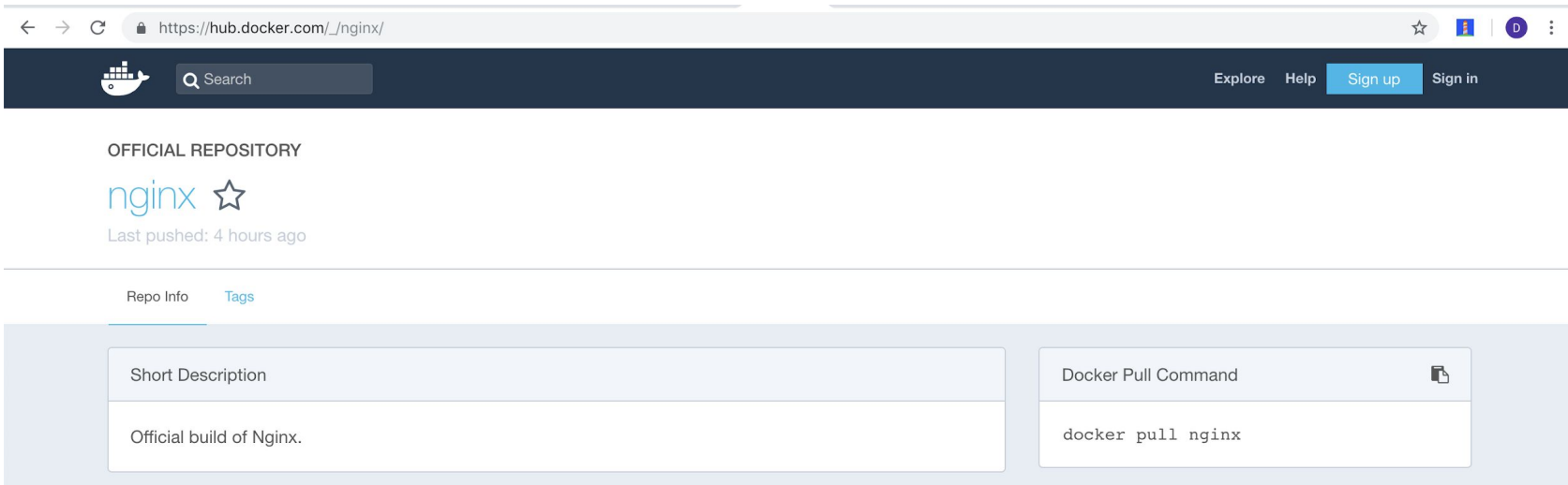
Testando o Docker

```
~ ➤ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world
```




Imagem

Uma imagem é um modelo/template utilizado para rodar um container.



The screenshot shows the Docker Hub interface for the 'nginx' repository. At the top, there's a browser address bar with the URL 'https://hub.docker.com/_/nginx/'. Below it is a dark blue navigation bar with the Docker logo, a search bar, and links for 'Explore', 'Help', 'Sign up', and 'Sign in'. The main content area is titled 'OFFICIAL REPOSITORY' and features the 'nginx' logo with a star icon and the text 'Last pushed: 4 hours ago'. There are two tabs: 'Repo Info' (selected) and 'Tags'. The 'Repo Info' tab displays a 'Short Description' box containing 'Official build of Nginx.' and a 'Docker Pull Command' box containing the command 'docker pull nginx'.

← → ↻ 🔒 https://hub.docker.com/_/nginx/ ☆ 🇧🇷 | D ⋮

 Search Explore Help Sign up Sign in

OFFICIAL REPOSITORY


nginx ☆

Last pushed: 4 hours ago

Repo Info Tags

Short Description

Official build of Nginx.

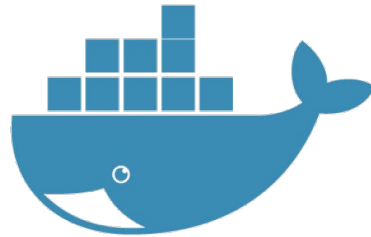
Docker Pull Command 

```
docker pull nginx
```



Docker HUB

Dockerhub é um Serviço de Web Hosting compartilhado para imagens Docker





Subindo um container

```
~ ➤ docker run --name hacktoberfest-ubuntu ubuntu:18.04  
Unable to find image 'ubuntu:18.04' locally
```

Subindo um container

```
~ ➤ docker run --name hacktoberfest-hello-world \  
> ubuntu:18.04 \  
> /bin/echo 'Hello World'  
  
Hello World
```

```
~ ➤ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
~ ➤						

Subindo um container

```
~ ➤ docker run -ti --name hacktoberfest-bash \  
> ubuntu:18.04 /bin/bash  
root@3d1675eecd48:/# pwd  
/  
root@3d1675eecd48:/# ls  
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var  
root@3d1675eecd48:/#
```

```
~ ➤ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3d1675eecd48	ubuntu:18.04	"/bin/bash"	45 seconds ago	Up 44 seconds		hacktoberfest-bash

```
~ ➤
```



Dockerfile

```
FROM nginx
# RUN echo "Hello Nginx"
# COPY nginx.conf /etc/nginx/nginx.conf
# WORKDIR /app
# COPY . /app
# VOLUME /app
EXPOSE 8080
CMD ["echo", "Ready!"]
```

```
~/Desktop/docker ➔ docker build -t hacktoberfest-nginx:1.0 .
```




Dockerfile

```
~/Desktop/docker ➤ docker run --name hacktoberfest-nginx-container -d -p 8080:80 hacktoberfest-nginx:1.0
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Deploy

- `docker build ...`
- `docker push ...`
- `docker pull ...`
- `docker run ...`

```
version: '3.6'
```

```
services:
```

```
# -----
```

```
# Postgres
```

```
# -----
```

```
postgres:
```

```
  container_name: prisma-social-postgres
```

```
  image: postgres:latest
```

```
  ports:
```

```
    - 54320:5432
```

```
  volumes:
```

```
    - ./docker/postgres/docker-entrypoint-initdb.d:/docker-entrypoint-initdb.d
```

```
    - ./docker/postgres/data:/var/lib/postgresql/data
```

```
  env_file:
```

```
    - ./docker/postgres/.env
```

```
# -----  
# Services  
# -----  
  
prismasocial:  
  container_name: prismasocial  
  restart: always  
  build:  
    context: .  
    dockerfile: Dockerfile  
  ports:  
    - 8025:8025  
  links:  
    - postgres:postgres  
  env_file:  
    - ./docker/postgres/.env  
  volumes:  
    - ./srv/app  
    - ./docker/scripts:/usr/src/docker_scripts  
  command: bash -c "/usr/src/docker_scripts/wait-for-postgres.sh postgres python /srv/app/src/manage.py runserver 0.0.0.0:8025"  
  depends_on:  
    - postgres
```