# Introdução ao Docker

Maio/2018

**Profº Thales Faggiano**
E-mail: thales.faggiano@sp.senai.br

senai·sp


Prof. **Thales Faggiano**

# Quem sou ?

- Educador e instrutor de formação profissional;
- Especialista com experiência nas áreas de gestão, análise de sistemas, redes e segurança em TI;
- Formação superior em Gestão de TI;
- Formação técnica em processamento de dados e redes de computadores;
- Certificados: ITIL Foundation, LPI - LPIC 1, CompTIA A+, CompTIA Linux+, CompTIA Systems Support Specialist, Cisco - CCNA Routing & Switching (2014).

senai·sp

# Onde trabalhei...








senai·sp

# Clientes...















senai·sp

# Parceiros...

NETSCOUT

Symantec

THOMSON REUTERS

Bloomberg

Check Point
SOFTWARE TECHNOLOGIES LTD.

Goldman Sachs

CISCO

BNP PARIBAS
The bank for a changing world

websense

CME Group

senai·sp

# Fornecedores...







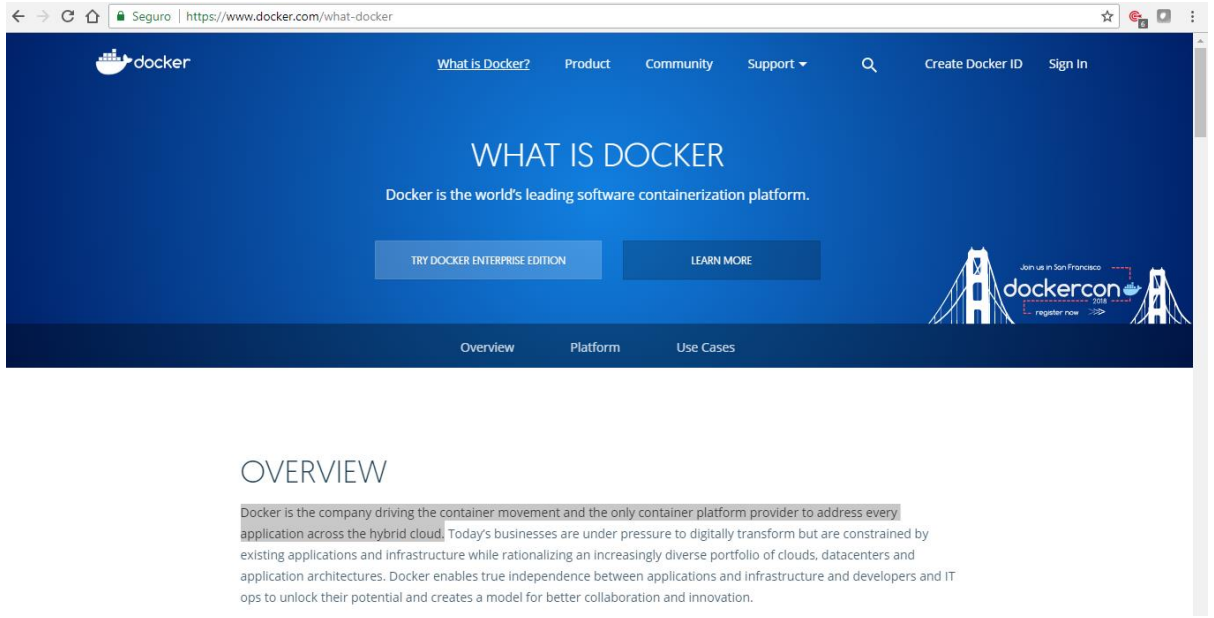




senai·sp

# O que é Docker ?

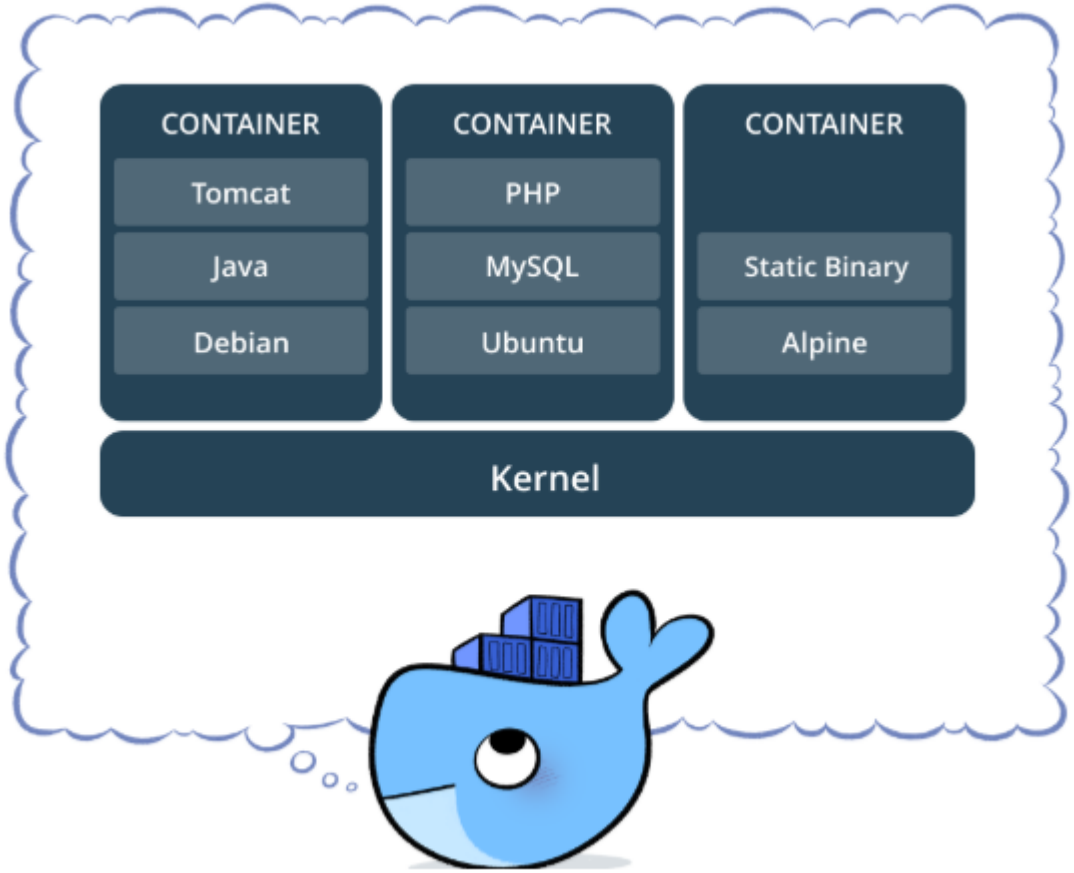
- Docker é a empresa que impulsiona o movimento de ***"conteinerização"*** no mundo.
- É a única plataforma que permite que as aplicações possam beneficiar-se de *nuvens híbridas* por meio de ***"conteiners"*.**




*"Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud."*
https://www.docker.com/what-docker

senai·sp

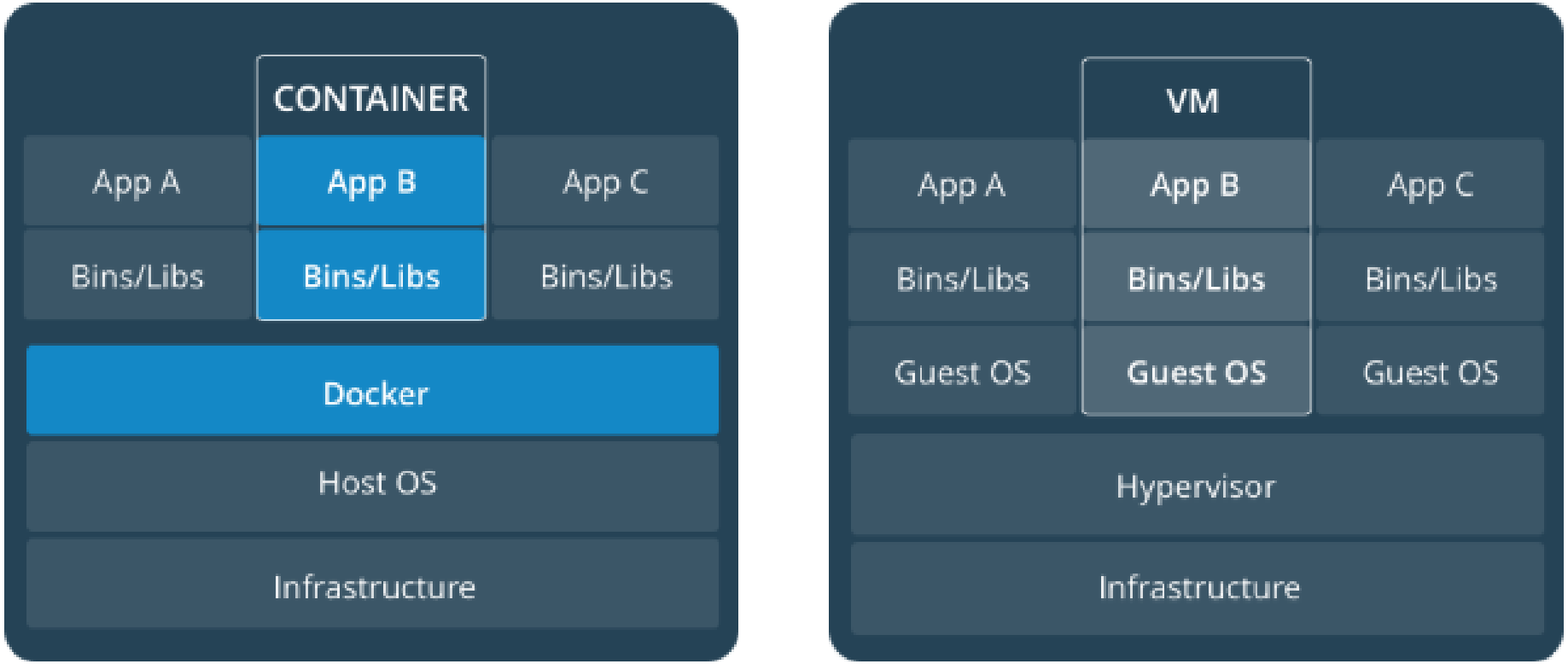
# O que são containers?




- São “pacotes de softwares” organizados em unidades padronizadas.
- São leves, autônomos e incluem tudo o que é preciso para executá-los: código, ferramentas e bibliotecas do sistema, configurações etc.
- Ajudam a transportar as aplicações desenvolvidas em um ambiente para outro, e.g.:
  - Desenvolvimento -> Teste;
  - Teste -> Homologação;
  - Homologação -> Produção.

https://www.docker.com/what-container

# Containers vs Máquinas Virtuais




https://www.docker.com/what-container

senai·sp

# Instalação - Procedimentos

**Passo 1 - Remover antigas instalações:**

- # apt-get remove docker --purge
- # apt-get remove docker-engine --purge
- # apt-get remove docker.io --purge
- # apt-get remove docker-ce –purge

**Passo 2 – Instalar ferramentas acessórias:**

- # apt-get update && apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg2 \
  software-properties-common

**Passo 3 – Baixar e adicionar a chave GPG no repositório:**

- # curl -fsSL https://download.docker.com/linux/debian/gpg \ | apt-key add -

**Passo 4 – Configurar o repositório:**

- # add-apt-repository "deb [arch=amd64] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable“

**Passo 5 – Instalar o Docker Community Edition:**

- apt-get update && apt-get install docker-ce –y

**Passo 6 – Confirmar a instalação:**

- # docker run hello-world

senai·sp

# Instalação - Hello from Docker!

```
root@jessie:/# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9bb5a5d4561a: Pull complete
Digest: sha256:f5233545e43561214ca4891fd1157e1c3c563316ed8e237750d59bde73361e77
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/

root@jessie:/#
```

senai·sp

# Primeiros Passos – Testes!

**Passo 1 - Listar comandos do docker:**

- docker
- docker container --help

**Passo 2 - Mostrar versão e informações:**

- docker --version
- docker version
- docker info

**Passo 3 - Executar uma imagem do docker:**

- docker run hello-world

**Passo 4 - Listar imagens do docker:**

- docker image ls

**Passo 5 - Listar containers (running, all, all em modo quiet):**

- docker container ls
- docker container ls --all
- docker container ls -aq

# Primeiros Passos – Dockerfile

Dockerfile:

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

requiriments.txt:

```
Flask
Redis
```

app.py:

```
from flask import Flask
from redis import Redis, RedisError
import os
import socket

# Connect to Redis
redis = Redis(host="redis", db=0, socket_connect_timeout=2, socket_timeout=2)

app = Flask(__name__)

@app.route("/")
def hello():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>cannot connect to Redis, counter disabled</i>"

    html = "<h3>Hello {name}!</h3>" \
           "<b>Hostname:</b> {hostname}<br/>" \
           "<b>Visits:</b> {visits}"
    return html.format(name=os.getenv("NAME", "world"), hostname=socket.gethostname(), visits=

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```

senai·sp

# Primeiros Passos – Executando o Container

**Passo 1 - Criando uma imagem Docker:**

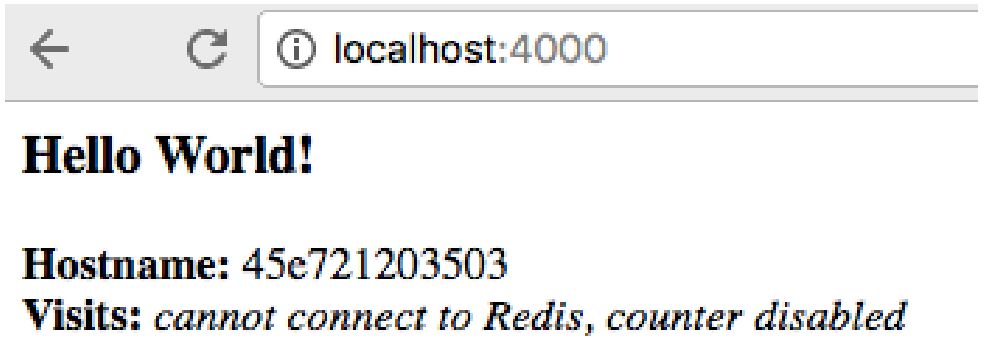
- docker build -t friendlyhello .

**Passo 2 - Listando nossa imagem:**

- docker image ls

**Passo 3 - Executando a aplicação:**

- docker run -p 4000:80 friendlyhello




```
root@jessie:/PrimeiroContainer# docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
friendlyhello       latest              78dc6e80a7b0        9 seconds ago       151MB
python              2.7-slim            46ba956c5967        2 weeks ago         140MB
hello-world         latest              e38bc07ac18e        5 weeks ago         1.85kB
root@jessie:/PrimeiroContainer# docker run -p 4000:80 friendlyhello
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

senai·sp

# Primeiros Passos – Continuando ...

https://docs.docker.com/get-started/part2/

# Próximos Passos - ☺

https://asciinema.org/a/blkah0l4ds33tbe06y4vkme6g

senai·sp