



## DOCKER – CONTAINERS NÃO SÃO VM's

**Walker Douglas Garcia Pinto**<sup>1</sup>

**Fagner Coin Pereira**<sup>2</sup>

### RESUMO

Containers são máquinas virtuais? Essa é uma pergunta muito frequente quando estamos falando de containers. Neste artigo serão abordados conceitos básicos sobre as principais diferenças entre containers e máquinas virtuais (VM's), especificamente containers utilizando Docker. A Docker é uma empresa de software que desenvolve, mantém e distribui “O Docker”, sistema para gerenciamento de containers o qual é Open Source, podendo ser implementado na versão “Community” ou “Enterprise”, a empresa foi fundada em 2013 e antes de se chamar Docker era chamada de DotCloud. Este projeto foi criado dentro da Google a qual liberou os códigos fonte que são escritos em uma linguagem de programação chamada Golang ou Go. A tecnologia Docker de containers é utilizada atualmente por diversas empresas tais como, Spotify, Netflix, Ebay, IBM, a própria Google e muitas outras. Possui uma comunidade ativa e comprometida com o desenvolvimento de recursos, ajustes, testes e correção de falhas. Docker é leve, portátil e híbrido, características que serão abordadas no decorrer deste artigo e lhe darão uma visão ampla sobre containers e suas vantagens em relação a infraestrutura de virtualização comumente utilizada para prover serviços.

Palavras-chave: Docker, containers, open source, gerenciamento, tecnologia.

---

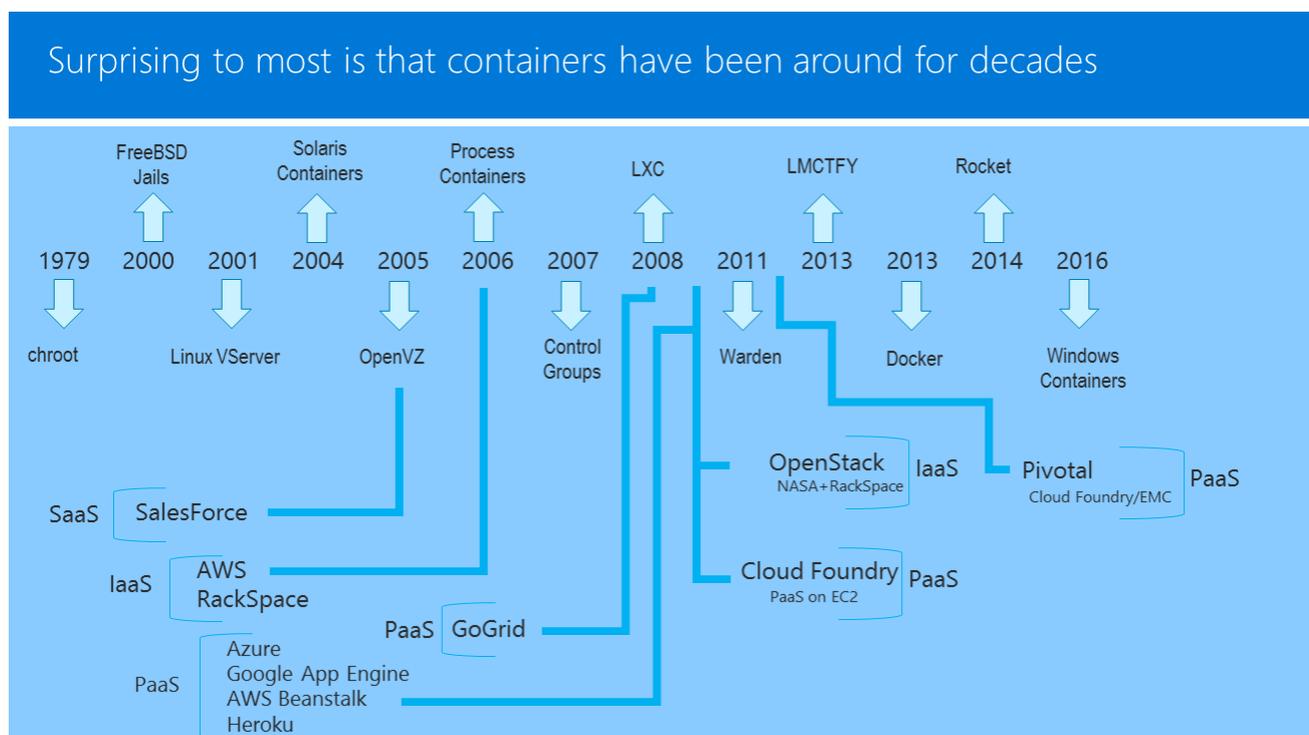
<sup>1</sup> Acadêmico do Curso Superior em Tecnologia em Redes de Computadores – Faculdade Alcides Maya. [walkerdgp@gmail.com](mailto:walkerdgp@gmail.com)

<sup>2</sup> Professor do Curso Superior em Tecnologia em Redes de Computadores – Faculdade Alcides Maya. [coin.pereira@gmail.com](mailto:coin.pereira@gmail.com)

## CONTAINERS

Containers são ambientes isolados dentro de um sistema operacional, podendo este sistema operacional ser Windows, Linux ou Mac OS e obrigatoriamente ser 64 bits. Esta técnica de isolamento de ambientes já era utilizada em 1979 através do isolamento de processos no sistema operacional UNIX com o [chroot](#), na imagem abaixo é possível observar a linha do tempo da evolução dos containers até chegada do Docker em 2013 com a mudança do nome da DotCloud para Docker Inc.

Figura 1 - Linha do tempo da tecnologia de containers até a chegada do Docker:

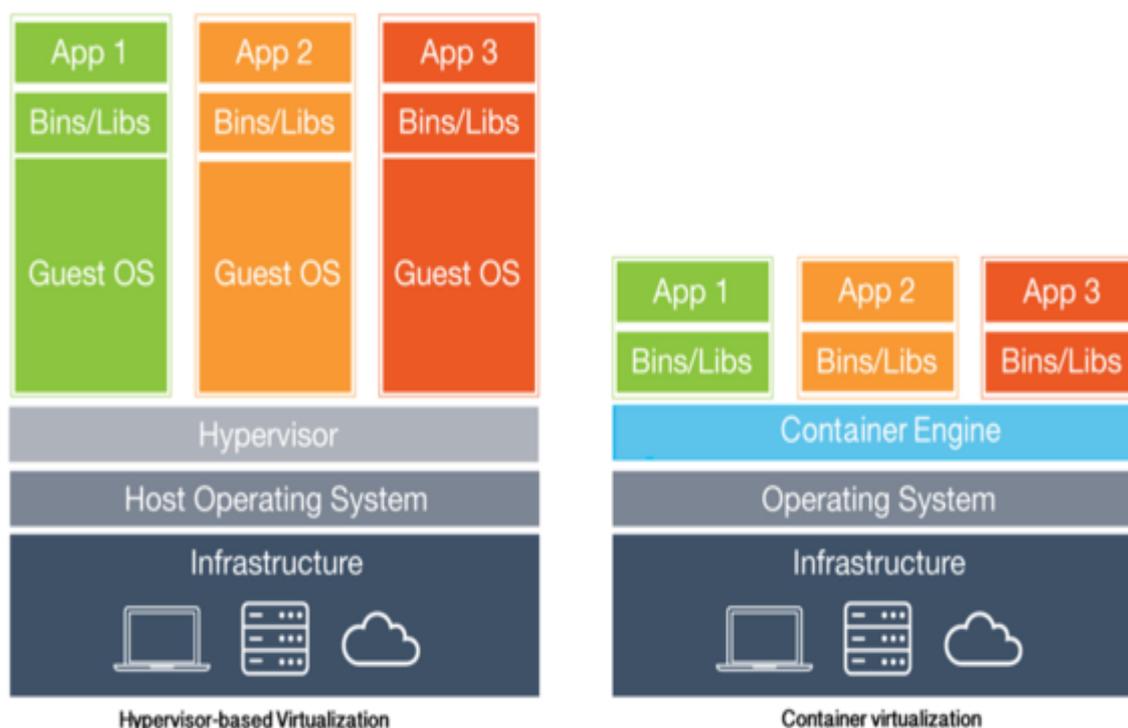


Fonte: Terkaly (2016)

O principal objetivo de um container é prover micro serviços, sem a necessidade que todo um sistema operacional seja instalado para que este um serviço funcione corretamente, isso trás maior agilidade quando é necessário subir um novo servidor web por exemplo com o Apache ou Tomcat.

Uma máquina virtual possui seu próprio kernel, binários e bibliotecas já o Docker compartilha o Kernel, binário e bibliotecas do S.O sob o qual ele está sendo executado, na imagem abaixo é possível observar estas diferenças.

Figura 2 - Comparação entre sistemas virtualizados e containers:



Fonte: Izidório (2017)

O fato de containers em Docker serem portáveis, é devido ao fato de que existe uma camada chamada “Docker Engine” ou “Container Engine” como é representada na imagem acima. Esta camada é responsável pela abstração de comandos e execução dos serviços, não é possível acessar diretamente esta camada, para isso utilizamos um terminal de linha de comandos, como o “CMD” presente em sistemas Windows ou o “BASH” presente em sistemas Unix e Linux.

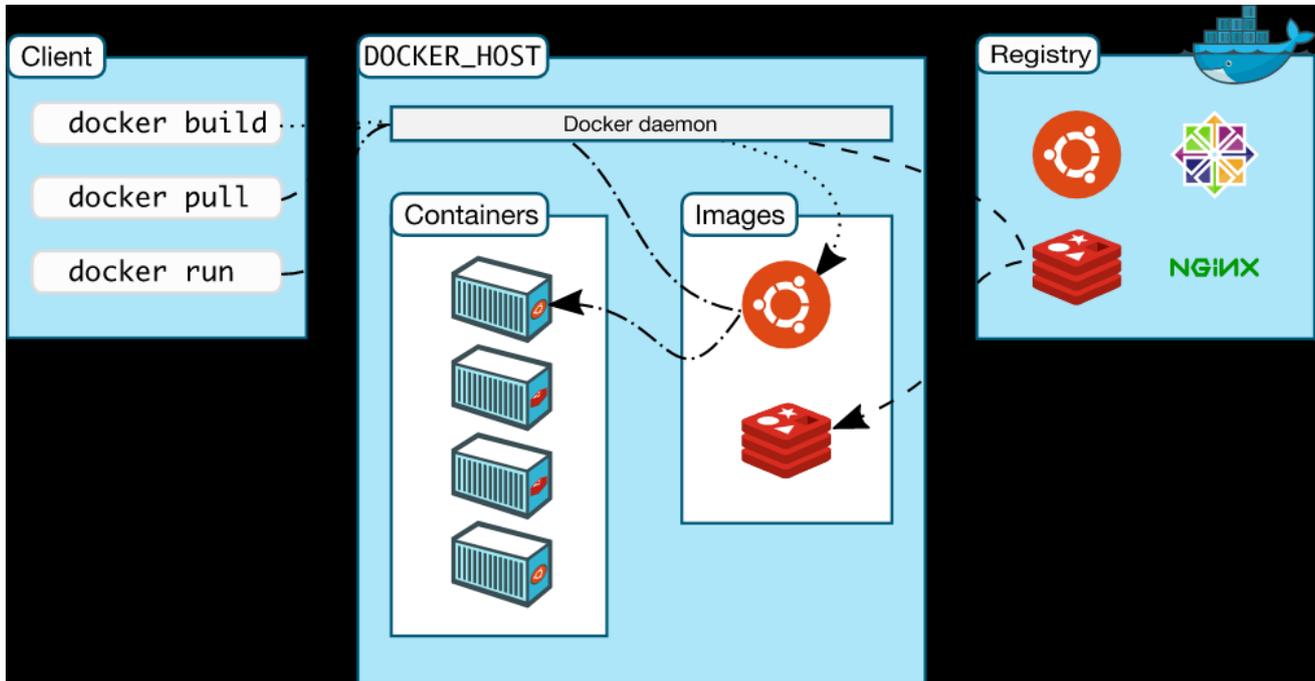
Os comandos digitados nestes terminais (CLI), são interpretados pela engine do Docker os quais possuem uma nomenclatura que iniciasse pelo comando chamado “docker”, seguido de parâmetros adicionais.

Abaixo é possível observar a nomenclatura básica de execução de três comandos, o build, pull e o run.

- Build – cria uma imagem baseada em um container

- Pull – Faz o download de uma imagem no Docker HUB
- Run – Executa uma imagem e a torna em um container

Figura 3 - Execução de comandos básicos docker:



Fonte: Docker... (2019)

Outra facilidade que o docker dispõem é de criar imagens customizadas, caso seja necessário criar um novo container, mas com as mesmas configurações de outro container já em execução, basta utilizar uma imagem customizada, também é possível utilizar imagens prontas disponíveis no “Registry” que nada mais que o Docker Hub, um repositório público de imagens prontas.

A partir de uma imagem é possível criar vários containers iguais ao mesmo tempo utilizando um “recurso” adicional chamado orquestrador.

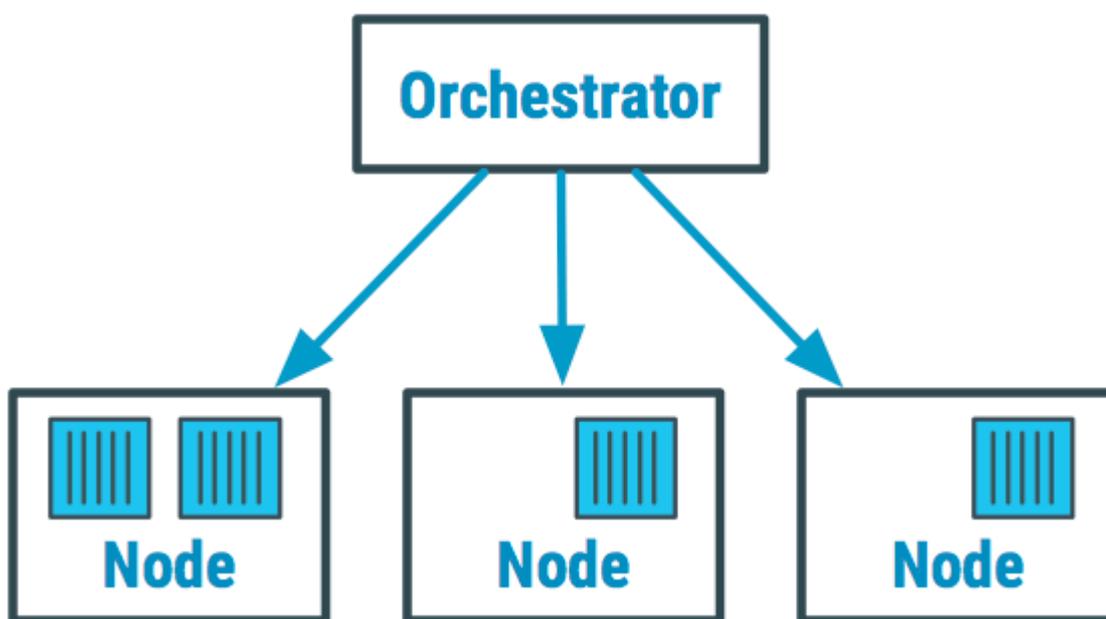
Os orquestradores nada mais são que gerenciadores de containers, através deles é possível criar centenas ou milhares de containers ao mesmo tempo, distribuir a carga de utilização entre os containers, garantir alta disponibilidade e escala-los quando necessário.

É muito comum escalar containers em sistemas onde em alguns momentos há uma necessidade maior de recursos de memória e processamento e outros não é necessário operar com estes recursos na sua capacidade total, para que isso seja gerenciado de forma inteligente e automatizada, entram os orquestradores, os quais irão entregar mais recursos

ao container ou balancear sua carga, isso traz maior segurança para o administrador de micro serviços, pois não é necessário desligar ou pausar os containers para realizar os ajustes, provendo assim alta disponibilidade e flexibilidade.

O docker conta com um sistema nativo de orquestração chamado “Swarm”, porém existe outras ferramentas mais completas para orquestração, como por exemplo o “Kubernetes”.

Figura 4 - Visão básica de um orquestrador para gerência de vários hosts docker ou nodes:



Fonte: Bashir (2018)

Dados os exemplos citados neste artigo, é possível observar algumas das principais diferenças entre máquinas virtuais e containers, o quão eficaz é utilizar containers para prover serviços e forma híbrida e flexível.

A capacidade de orquestrar tais containers e gerencia-los de uma forma centralizada e escalável, os tornam superiores a máquinas virtuais, as quais necessitam que o sistema operacional seja instalado por completo, enquanto containers utilizam recursos compartilhados do host sob o qual estão sendo executados.

Containers Docker podem ser executados dentro de máquinas virtuais ou dentro de um sistema que está sendo executado diretamente de uma máquina físico o qual é chamado de bare metal.

O docker possui uma documentação aberta completa e gratuita no seu site, a qual pode ser consultada a qualquer momento para que dúvida sejam sanadas, lá estão descritos todos os comandos docker e seus parâmetros.

Sendo assim containers não são de fato VM's e também é errado dizer que containers são máquinas virtuais "enxutas" as diferenças e arquitetura e funcionamento limitam tal comparação.

## REFERENCIAS:

BASHIR, Faizan. **Container Orchestration**. 2018. Disponível em:

<<https://devopedia.org/container-orchestration>>. Acesso em: 05 nov. 2019.

DOCKER overview. 2019. Disponível em: <<https://docs.docker.com/engine/docker-overview/>>. Acesso em: 05 nov. 2019.

IZIDÓRIO, Bruno. **O que é Docker?** 2017. Disponível em:

<<https://brunoizidorio.com.br/docker-o-que-e-docker/>>. Acesso em: 05 nov. 2019.

TERKALY, Bruno. **Docker Container Overview for Business Leaders**. 2016. Disponível em: <<https://blogs.msdn.microsoft.com/allthingscontainer/2016/11/07/docker-container-overview-for-business-leaders/>>. Acesso em: 05 nov. 2019.