

# Conceitos

## docker

O Docker é uma plataforma de código aberto que facilita a criação de ambientes isolados durante o desenvolvimento de software. Isso é importante para a gestão da infraestrutura da aplicação, desde a criação até a modificação dos serviços.

Na prática, o Docker age como uma `máquina virtual` que roda software dentro de outro sistema operacional. O conceito de `container Linux` é anterior à tecnologia do docker ([Leia mais - O que é um container Linux](#)).

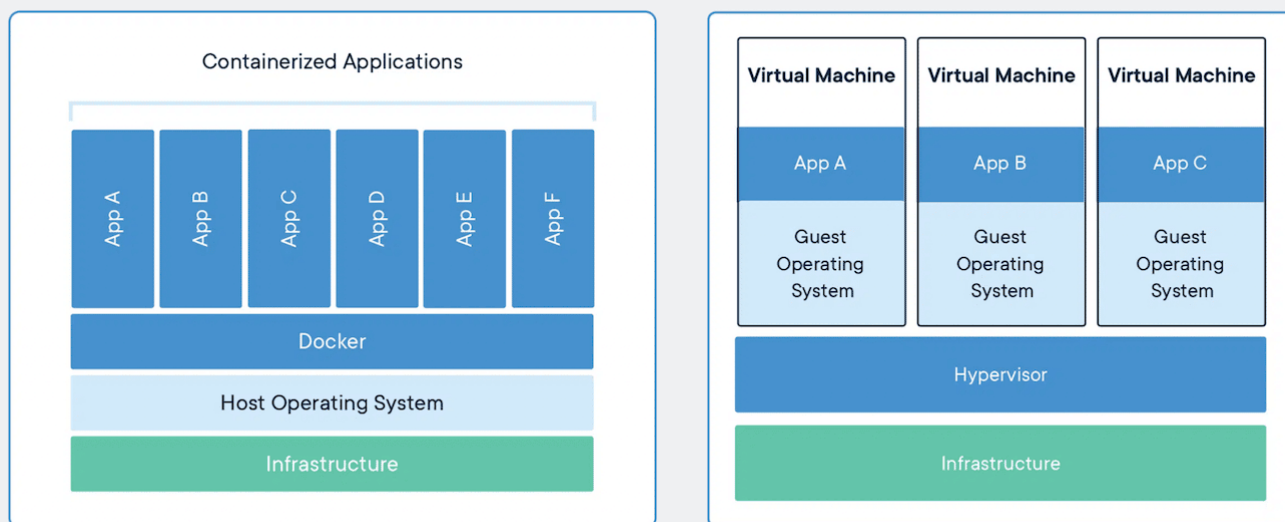
Vantagens:

- rodar seus contêineres com menor gasto de recursos;
- funcionalidades de segurança que tornaram a tecnologia bem popular nos dias atuais;
- evitar o problema do [inferno das dependências](#);
- garantir que as mesmas versões e dependências utilizadas no ambiente de desenvolvimento e testes seja utilizada no ambiente de produção.

## docker x virtualização

Os contêineres do Docker têm seu próprio sistema de arquivos, estrutura de dependências, processos e recursos de rede. A aplicação tem tudo o que precisa dentro do contêiner e pode ser executada em qualquer lugar. A tecnologia de contêiner do Docker usa **diretamente** os recursos subjacentes do kernel do sistema operacional host.

As máquinas virtuais (VMs) fornecem a virtualização de uma **máquina inteira** (servidor). Uma máquina virtual emula os componentes de hardware de uma máquina física, como CPU, memória, placa de interface de rede, controladores USB e placas de som. Você pode executar um sistema operacional convidado e várias aplicações no ambiente virtual.



## Comparação de docker e máquinas virtuais

### container

O que é construído pelo Docker. É uma aplicação encapsulada, construída com as bibliotecas e dependências necessárias, e que pode ser executada em qualquer lugar que o Docker esteja instalado.

### host

É a máquina que possui o Docker instalado e que executará os contêineres (no nosso caso, a máquina local).

### porta

O mapeamento de portas é utilizado para acessar o serviço que roda dentro do contêiner. Nós abrimos uma porta no host para nos dar acesso à porta correspondente dentro do contêiner docker. Dessa forma, todas as requisições que são feitas no host podem ser direcionadas para dentro do contêiner docker.

### volume

O sistema de arquivos do contêiner é volátil, quando o contêiner é destruído, todos os arquivos se perdem. O volume são diretórios externos ao contêiner, que são montados diretamente nele e, dessa forma, não seguem esse padrão de volatilidade padrão dos demais arquivos, portanto os volumes são persistidos.

### image

Uma imagem é como uma versão *congelada* de um contêiner. As imagens são criadas através de um arquivo chamado `Dockerfile` ou de um contêiner em execução.

## tag

A tag é uma versão específica de uma imagem docker. Geralmente ela é associada da uma determinada versão do serviço principal que é oferecido em um determinado contêiner. Por convenção, a versão mais recente de uma imagem é disponibilizada na tag `latest`.

## workdir

É a pasta padrão de trabalho do contêiner. Quando você entra dentro do contêiner, ou executa um comando nele, caso não seja fornecido o caminho completo, será executado por padrão a partir desta pasta.

## docker compose

Uma ferramenta para definição e execução de múltiplos containers Docker. Com ela é possível configurar todos os parâmetros necessários para executar cada container a partir de um arquivo de definição.

## hub

É um local (repositório docker) onde são armazenadas as imagens docker. Pode ter acesso público ou restrito.

## Referências

[Docker \(software\)](#)

[O que é Docker, principais vantagens e quando utilizá-lo no desenvolvimento de software?](#)

[Docker: tudo o que você precisa saber](#)

[O que é Docker: Um Guia Completo](#)

[Qual é a diferença entre o Docker e uma VM?](#)

---

Revision #12

Created 2 July 2024 19:37:04 by Éverton de Oliveira Paiva

Updated 5 July 2024 11:44:58 by Éverton de Oliveira Paiva