

Centro Paula Souza - Fatec Votorantim

Docker (Containers)

Alinne Martins Cardoso

Bruno Henrique Pascoal

Maycon Basilio de Moura

Pietro Adrian R. Da Silva

Rafael Martins Cardoso

Docker (Containers)

O Docker possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um container, e a partir desse momento o ambiente inteiro torna-se portátil para qualquer outro host que contenha o Docker instalado.

Isso reduz de maneira drástica o tempo de deploy de alguma infraestrutura ou até mesmo aplicação, pois não há necessidade de ajustes de ambiente para o correto funcionamento de serviço, o ambiente é sempre o mesmo, configure-o uma vez e replique-o quantas vezes quiser.

O tempo de deploy se refere ao período necessário para mover uma aplicação ou atualização de software do desenvolvimento para a produção, tornando-a acessível aos usuários finais.

O processo de deploy, geralmente envolve várias etapas, incluindo:

1. Desenvolvimento: Esta é a fase inicial, onde o software é desenvolvido e testado na máquina do programador.
2. Staging/Teste: Nesta fase, o software é movido para um ambiente teste, onde é submetido a mais testes para garantir que está funcionando corretamente.
3. Produção: Após todos os testes serem concluídos e quaisquer problemas serem corrigidos, o software é finalmente movido para o ambiente de produção, onde está disponível para os usuários finais.

O tempo de deploy pode variar dependendo da complexibilidade do software, da estrutura de TI, da organização e das práticas de deploy utilizadas. Por exemplo, algumas organizações utilizam práticas de Continuous Deployment (Implantação Contínua), que visam minimizar o tempo entre o desenvolvimento de uma nova linha de código e seu uso pelos usuários.

Outra facilidade do Docker é poder criar suas imagens (contêiner prontos para deploy) a partir de arquivos de definição chamados Dockerfiles.

O Docker utiliza como backend default o LXC, com isso é possível definir limitações de recursos por container, como memória, CPU, I/O, etc.

Como o Docker faz isso?

Ele trabalha utilizando cliente e servidor (toda a comunicação entre o Docker Daemon e Docker cliente é realizada através de API), basta apenas que você tenha instalado o serviço do Docker em um lugar, e aponte em seu Docker Client para esse servidor. A plataforma do Docker em si, utiliza alguns conjuntos de recursos, seja para a criação ou administração dos contêiners, entre esses conjuntos podemos destacar a biblioteca libcontainer, que é responsável pela comunicação entre Docker Daemon e o backend utilizado, é ela responsável pela criação do contêiner, e é através dela que podemos setar os limites de recursos por container.

O Docker utiliza recursos padrão kernel do Linux para sua execução. Isso faz com que seus processos sejam independentes, com a possibilidade de serem operados individual e simultaneamente.

Quaisquer arquivos do Docker pode ser “Empacotados”, ou reunidos, em contêiners, e podem ser transportados de um ambiente a outro de modo simples e rápido. Assim, a portabilidade é uma das características mais marcantes do Docker.

O Docker é uma plataforma open source, que viabiliza a virtualização e criação de ambientes isolados, ou contêiners no Linux.

Com todas as ferramentas que compõe, é possível executar esses ambientes de forma eficiente e otimizada.

Essa tecnologia também permite que qualquer equipe opere e tenha acesso às funcionalidades da infraestrutura do sistema e do servidor, sem precisar de qualquer permissão especial.

Afinal, podemos comparar sua aplicação com a das máquinas virtuais (do inglês, Virtual Machine ou VM). A diferença é que ele também é compatível com o armazenamento em nuvem.

Tais características facilitam muito o trabalho em conjunto dos profissionais e desenvolvedores, contribuindo para a flexibilidade e agilidade das entregas.

Tais características facilitam muito o trabalho em conjunto dos profissionais desenvolvedores, contribuindo para a flexibilidade e agilidade das entregas.

Vantagens de usar Docker:

Comunidade com repositório:

Existe uma grande comunidade com um imenso repositório de imagens do Docker, com modelos prontos para adaptação.

Desde pilares de infraestrutura até sistemas de integração, é possível encontrar no repositório, que tem imagens originais, já configuradas e devidamente regularizadas.

Basta, então, dar o seu “toque final” nas imagens e adicionar configurações e adaptações que seu sistema precisar.

Infraestrutura e Desenvolvimento com a mesma linguagem:

Ter uma linguagem só para a infraestrutura e o desenvolvimento em Docker era, até então, impossível.

Muitos problemas e impasses surgiram devido à divergência de linguagens, mas foram amenizados com um idioma em comum.

Ambientes semelhantes:

Basta transformar um arquivo em imagem Docker e ele pode ser armazenado e transportado para qualquer contêiner. São vários ambientes possíveis para manter o seu arquivo.

É possível integrar banco de dados com o contêiner e criar uma estrutura ainda mais completa e funcional. Essa opção é altamente útil para aqueles que fazem testes em diferentes ambientes ou servidores.

Toda aplicação e dependências em um só pacote:

Com a metodologia Docker, você pode ter todas as aplicações e dependências em um só pacote.

Precisa certificar de que as imagens são facilmente acessíveis, e disponibilizar a configuração de execução para os usuários que baixarem o arquivo.

Fases como o armazenamento e atualização também são viabilizadas com Docker. Afinal, é possível manter diversas versões de uma mesma imagem no container.

Quando um arquivo é modificado, basta executar alguns poucos comandos Docker para que a atualização aconteça também nos arquivos baixados.

Ganho tempo com replicação do ambiente Docker:

Com tantas funcionalidades, é nítido que o usuário ou profissional ganhará tempo. Ainda outra utilidade é a padronização e possibilidade de replicação que o sistema oferece.

Para reproduzi-las fielmente, as imagens precisam seguir todas as práticas de construção durante seu processo de estruturação.

O que são containers?

Containers representam a virtualização de um ambiente isolado no kernel do sistema operacional do Linux.

Isso, por sua vez, permite que cada nível seja executado separadamente, otimizando tempo e configurações do sistema.

Quando é necessário executar camadas de desenvolvimento simultaneamente. A tecnologia Docker em contêiners é a mais indicada.

O que são imagens?

As imagens Docker são os arquivos que constroem a aplicação, que geralmente é feita em camadas.

São responsáveis por armazenar todas as informações que um processo precisa para ser executado, como:

- Bibliotecas;
- Sistema de arquivos;
- Pacotes e Recursos;

- Módulos do Kernel.

Assim, é como se o processo sempre dependesse de uma imagem para ser executado corretamente, trazendo resultados expressivos para o usuário.

A diferença entre container e imagem:

Ao passo que a imagem é como um arquivo responsável por armazenar todas as informações necessárias, o container corresponde ao processo que é executado por ela.

Cada imagem pode criar mais de um container e eles só podem ser virtualizados, como é a proposta, se tiverem acesso a imagens completas, com pacotes de arquivos exatos.

Os containers, quando estão em execução, são capazes de manter uma tarefa por vez, com uma responsabilidade por container.

DIEDRICH, C. **O que é Docker? | Mundo Docker**. Disponível em:
<<https://www.mundodocker.com.br/o-que-e-docker/>>. Acesso em: 28 set. 2023.

O que é deploy, para que serve, vantagens e como fazer deploy. Disponível em:
<<https://kenzie.com.br/blog/o-que-e-deploy/>>. Acesso em: 2 dez. 2023.

RIBEIRO, U. **O que é docker, como funciona e quais as vantagens**. Disponível em:
<<https://www.certificacaolinux.com.br/o-que-e-docker/>>. Acesso em: 28 set. 2023.

Saiba o que é deploy e as formas de realizá-lo. Disponível em:
<<https://www.hostgator.com.br/blog/o-que-e-deploy-e-como-realiza-lo/>>. Acesso em: 2 dez. 2023.