

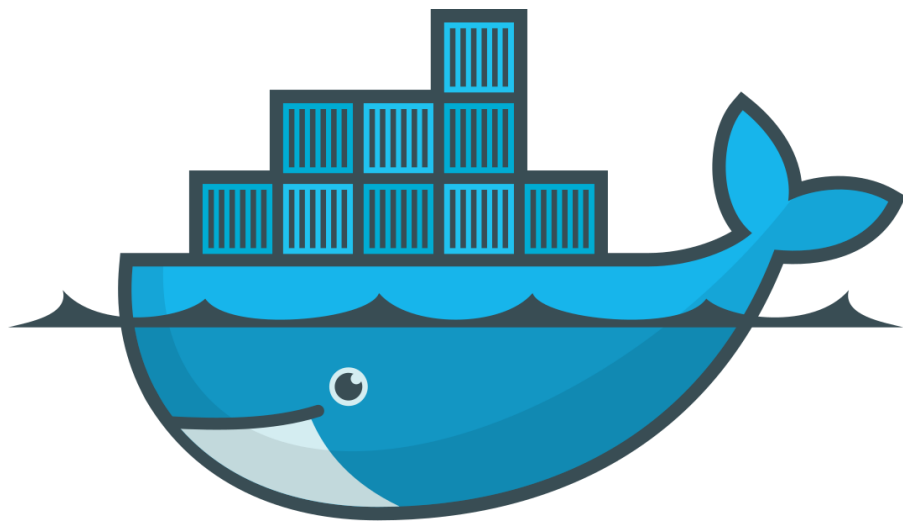


Java WebDeveloper – Formação FullStack

Professor Sergio Mendes

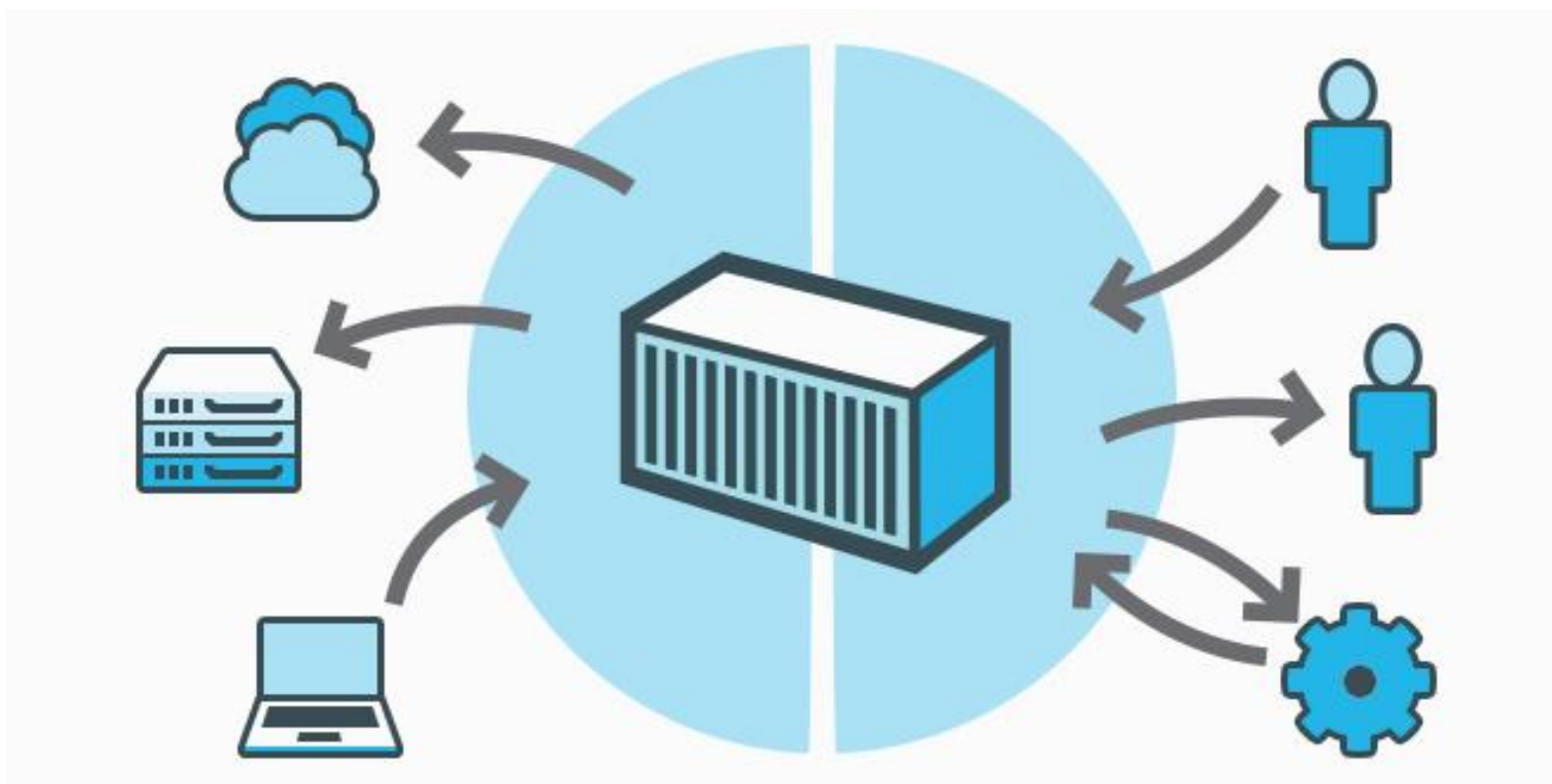
Aula 24 (20/03/23)





docker

Docker é uma plataforma aberta, criada com o objetivo de facilitar o desenvolvimento, a implantação e a execução de aplicações em ambientes isolados. Foi desenhada especialmente para disponibilizar uma aplicação da forma mais rápida possível.



Usando o Docker, você pode facilmente gerenciar a infraestrutura da aplicação, isso agilizará o processo de criação, manutenção e modificação do seu serviço. Todo processo é realizado sem necessidade de qualquer acesso privilegiado à infraestrutura corporativa.

Assim, a equipe responsável pela aplicação pode participar da especificação do ambiente junto com a equipe responsável pelos servidores.

O Docker viabilizou uma "linguagem" comum entre desenvolvedores e administradores de servidores. Essa nova "linguagem" é utilizada para construir arquivos com as definições da infraestrutura necessária e como a aplicação será disposta nesse ambiente, em qual porta fornecerá seu serviço, quais dados de volumes externos serão requisitados e outras possíveis necessidades.

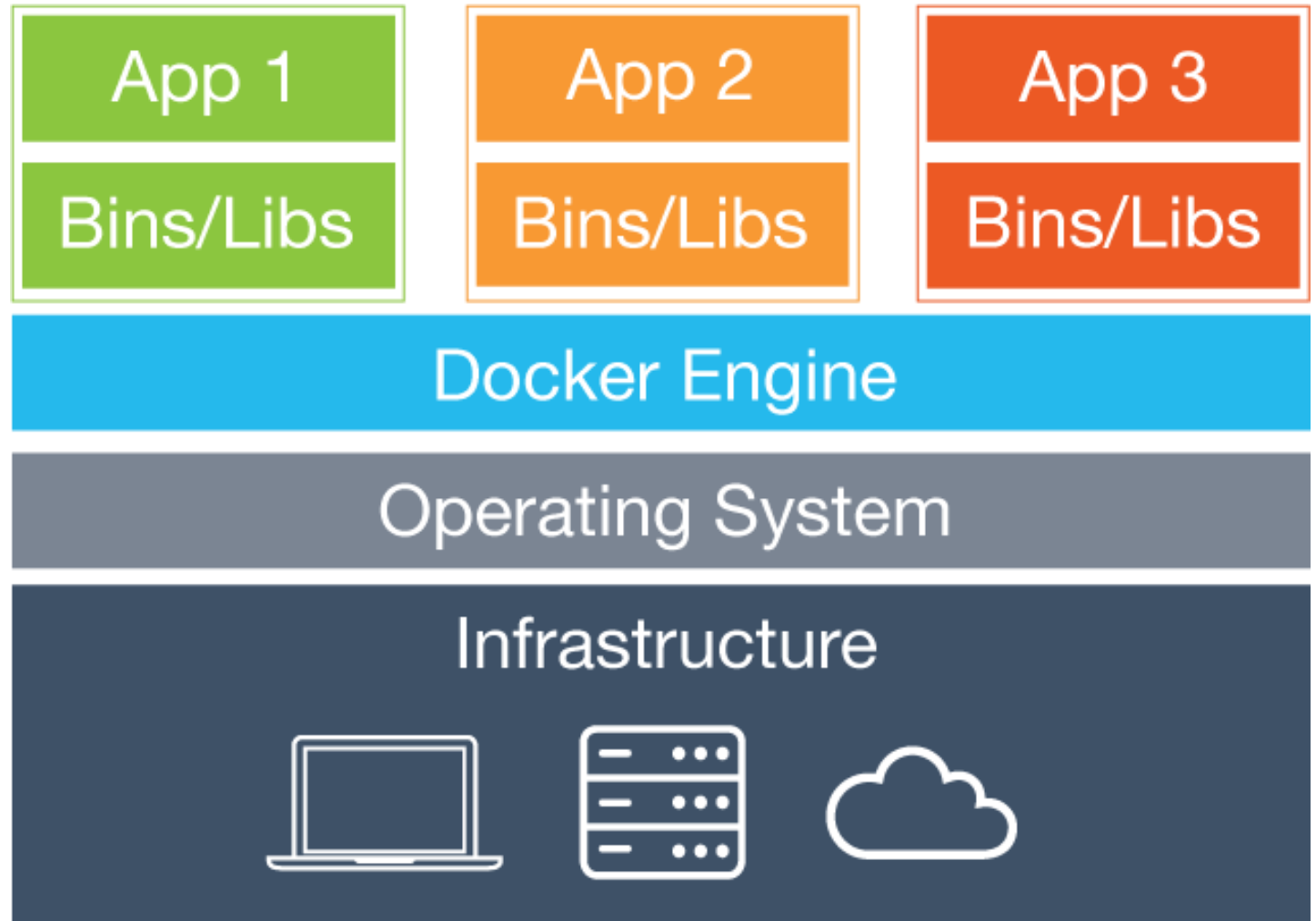
O Docker utiliza o modelo de container para “empacotar” a aplicação que, após ser transformada em imagem Docker, pode ser reproduzida em plataforma de qualquer porte; ou seja, caso a aplicação funcione sem falhas em seu notebook, funcionará também no servidor ou no mainframe. Construa uma vez, execute onde quiser.

Os containers são isolados a nível de disco, memória, processamento e rede. Essa separação permite grande flexibilidade, onde ambientes distintos podem coexistir no mesmo host, sem causar qualquer problema. Vale salientar que o overhead nesse processo é o mínimo necessário, pois cada container normalmente carrega apenas um processo, que é aquele responsável pela entrega do serviço desejado.

Em todo caso, esse container também carrega todos os arquivos necessários (configuração, biblioteca e afins) para execução completamente isolada.

O modelo de isolamento utilizado no Docker é a virtualização a nível do sistema operacional, um método de virtualização onde o kernel do sistema operacional permite que múltiplos processos sejam executados isoladamente no mesmo host.

Esses processos isolados em execução são denominados no Docker de container.



Para criar o isolamento necessário do processo, o Docker usa a funcionalidade do kernel, denominada de namespaces, que cria ambientes isolados entre containers: os processos de uma aplicação em execução não terão acesso aos recursos de outra.

A menos que seja expressamente liberado na configuração de cada ambiente.

Para evitar a exaustão dos recursos da máquina por apenas um ambiente isolado, o Docker usa a funcionalidade cgroups do kernel, responsável por criar limites de uso do hardware a disposição.

Com isso é possível coexistir no mesmo host diferentes containers sem que um afete diretamente o outro por uso exagerado dos recursos compartilhados.

Qual a diferença entre Imagem e Container?

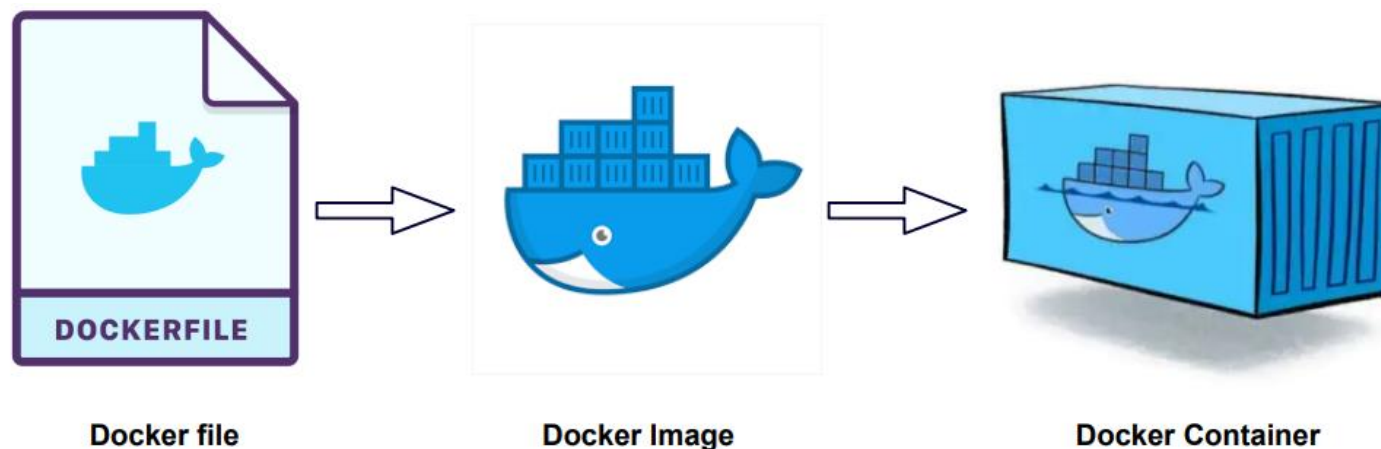
Traçando um paralelo com o conceito de orientação a objeto, a **imagem** é a classe e o **container** o objeto. A imagem é a abstração da infraestrutura em estado somente leitura, de onde será instanciado o container.

Todo container é iniciado a partir de uma imagem, dessa forma podemos concluir que nunca teremos uma imagem em execução.

Um container só pode ser iniciado a partir de uma única imagem. Caso deseje um comportamento diferente, será necessário customizar a imagem.

Ao usar o Docker, um desenvolvedor cria um aplicativo ou serviço e empacota a ele e suas dependências em uma imagem de contêiner. Uma imagem é uma representação estática do aplicativo ou do serviço e de sua configuração e dependências.

Para executar o aplicativo ou o serviço, uma instância da imagem do aplicativo é criada para criar um contêiner, que estará em execução no host do Docker. Inicialmente, os contêineres são testados em um ambiente de desenvolvimento ou em um computador.



O container nada mais é do que um ambiente isolado, disposto em um servidor, que divide um único host de controle. Vamos voltar ao exemplo dos containers tradicionais para explicar melhor esse conceito.

Um navio cargueiro pode carregar diversos containers. Caso um dos recipientes seja danificado, os demais não são afetados. Afinal, são isolados, protegidos e estão carregando seus próprios produtos.

Trazendo para o mundo do desenvolvimento, cada container possui uma função e sua responsabilidade. Caso um deles sofra um dano, o funcionamento do sistema não para e a função afetada é redirecionada para um novo container.

As vantagens do uso do Docker

- O Docker permite um uso mais eficiente dos recursos do sistema;
- As instâncias de aplicativos em container usam muito menos memória do que as máquinas virtuais, elas são inicializadas e interrompidas mais rapidamente e podem ser armazenadas muito mais densamente em um hardware host. Tudo isso equivale a menos gastos com TI;
- A redução de custos irá variar dependendo de quais aplicativos estão em jogo e de quão intensivos os recursos podem ser, mas os containers funcionam invariavelmente como mais eficientes que as VMs.

As vantagens do uso do Docker

- O Docker permite ciclos de entrega de software mais rápidos;
- O software corporativo deve responder rapidamente a mudanças de condições. Isso significa que o escalonamento fácil atende à demanda e facilita a atualização para adicionar novos recursos conforme a necessidade do negócio;
- Os containers Docker facilitam a colocação rápida de novas versões de software, com novos recursos de negócios, e a rápida reversão para uma versão anterior, se necessário. Eles também facilitam a implementação de estratégias como implantações em azul / verde;

As vantagens do uso do Docker

- O Docker permite a portabilidade de aplicativos;
- Docker ajuda a arquitetura de microsserviços;
- Os containers são leves, portáteis e facilitam a construção de software em linhas de pensamento avançadas, de modo que o desenvolvedor não está tentando resolver os problemas de amanhã com os métodos de desenvolvimento de ontem.

{ COTI INFORMÁTICA }
{ ESCOLA DE NERDS }