



## Curso Bônus

### Planejando Sua Carreira Para as Profissões do Futuro

Docker - O Que é, Quando Usar e Por Que  
Usar?



O Docker é uma plataforma de código aberto que permite criar, implantar, executar, atualizar e gerenciar containers — componentes executáveis padronizados que combinam o código-fonte de uma aplicação com as bibliotecas e dependências do sistema operacional (SO) necessárias para executar esse código em qualquer ambiente.

Pense no container como uma máquina virtual “superleve”, apenas com aquilo que precisamos para executar uma aplicação, por exemplo. Lembre-se que o sistema operacional sempre deverá estar presente.

Os containers simplificam o desenvolvimento e a entrega de aplicativos distribuídos. Eles se tornaram cada vez mais populares à medida que as organizações mudam para ambientes em nuvem e ambientes multicloud híbridos.

É possível criar containers sem o Docker, trabalhando diretamente com recursos integrados ao Linux e outros sistemas operacionais. Mas o Docker torna a containerização mais rápida, fácil e segura. O Docker vem ganhando popularidade no universo da Ciência de Dados e aqui mesmo na DSA estamos trazendo cada vez mais o Docker em diversos cursos, para que os alunos fiquem atualizados com a evolução da tecnologia.

O Docker também se refere à Docker, Inc. (a empresa que vende uma versão comercial do Docker) e ao projeto de código aberto Docker para o qual a Docker, Inc. e muitas outras organizações e indivíduos contribuem.

O Docker pode ser usado para os mais variados fins de computação. Mas usamos o Docker principalmente quando o uso de containers for essencial para a performance ou arquitetura de uma aplicação. A tecnologia do Docker é totalmente baseada no Linux.

Os containers permitem o isolamento de processos e recursos de virtualização integrados ao kernel do Linux. Esses recursos permitem que vários componentes de aplicativos compartilhem os recursos de uma única instância do host (o computador físico) da mesma forma que um hipervisor permite que várias máquinas virtuais (VMs) compartilhem a CPU, a memória e outros recursos de um único servidor, por exemplo. A virtualização de servidores já é uma realidade há muitos anos.

Como resultado, a tecnologia de container oferece todas as funcionalidades e benefícios das VMs (máquinas virtuais), incluindo isolamento de aplicativos, escalabilidade econômica e descartabilidade, além de vantagens adicionais importantes:

- **Mais Leve:** ao contrário das VMs, os containers não sobrecarregam a instância de sistema operacional do host físico. Eles incluem apenas os processos e dependências do SO necessários para executar o código. Os tamanhos dos containers são medidos em megabytes (vs. gigabytes para algumas VMs), fazem melhor uso da capacidade do hardware e têm tempos de inicialização mais rápidos.
- **Produtividade Aprimorada:** aplicativos em container podem ser escritos uma vez e executados em qualquer lugar. E em comparação com as VMs, os containers são mais rápidos e fáceis de implantar, provisionar e reiniciar. Isso os torna ideais para uso em pipelines de integração contínua e entrega contínua (CI/CD) e são mais adequados para equipes de desenvolvimento que adotam práticas Agile e DevOps, bem como em DataOps e MLOps.
- **Maior Eficiência de Recursos:** com containers podemos executar várias vezes mais cópias de um aplicativo no mesmo hardware do que usando VMs. Isso pode reduzir os gastos com ambiente em nuvem.

O Docker é tão popular atualmente que “Docker” e “containers” são usados de forma intercambiável. Mas as primeiras tecnologias relacionadas a containers estavam disponíveis anos antes que o Docker fosse lançado ao público em 2013.

Mais notavelmente, em 2008, o LinuxContainers (LXC) foi implementado no kernel do Linux, permitindo totalmente a virtualização para uma única instância do Linux. Embora o LXC ainda seja usado hoje, tecnologias mais recentes usando o kernel Linux estão disponíveis. O Ubuntu, um sistema operacional Linux moderno e de código aberto, também oferece esse recurso.

O Docker permite que façamos acesso a recursos nativos de containerização usando comandos simples e que façamos a automação por meio de uma interface de programação de aplicativos (API) que economiza trabalho. Comparado ao LXC, o Docker oferece:

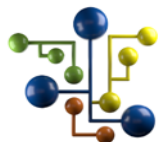
- Portabilidade de container aprimorada: enquanto os containers LXC geralmente fazem referência a configurações específicas da máquina, os containers Docker são executados sem modificação em qualquer desktop, data center e ambiente de nuvem.
- Atualizações ainda mais leves e mais granulares: com o LXC, vários processos podem ser combinados em um único container. Isso possibilita a construção de um aplicativo que pode continuar em execução enquanto uma de suas partes é desativada para uma atualização ou reparo.
- Criação automatizada de containers: o Docker pode criar automaticamente um container com base no código-fonte do aplicativo.
- Controle de versão de container: o Docker pode rastrear versões de uma imagem de container, reverter para versões anteriores e rastrear quem criou uma versão e como. Ele pode até carregar apenas os deltas entre uma versão existente e uma nova.
- Reutilização de containers: os containers existentes podem ser usados como imagens base, ou seja, como modelos para criar novos containers.
- Bibliotecas de containers: podemos acessar um repositório contendo milhares de containers criados por voluntários. Veremos isso no curso.

## ***Cursos de Aperfeiçoamento Profissional - Bônus da Formação***

---

Hoje, a containerização do Docker também funciona com Microsoft Windows e Apple MacOS. Podemos executar containers Docker em qualquer sistema operacional, e a maioria dos principais provedores de nuvem, incluindo Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform e IBM Cloud, oferecem serviços específicos para ajudar a criar, implantar e executar aplicativos em containers com o Docker.

Logo, aprender Docker é cada vez mais importante para trabalhar com Ciência de Dados.



**Equipe DSA**

**Muito Obrigado!**  
**Continue Trilhando Uma Excelente Jornada de Aprendizagem.**