



CONTÊINERES

Material Complementar



Marianne Salomão

Cloud Engineer na IBM Brasil, desenvolvedora de software full - stack e instrutora de TI.



linkedin.com/in/mariannesalomao



github.com/mariannesalomao

Índice

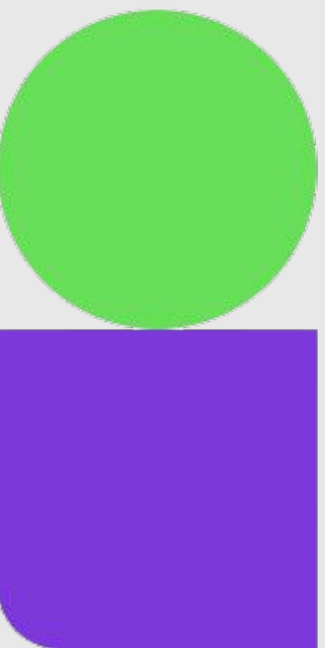
- + [Objetivos](#)
- + [Introdução](#)
- + [Contêineres](#)
- + [Contêineres VS virtualização](#)
- + [Benefícios](#)
- + [Casos de Uso](#)
- + [Docker](#)
- + [Economia de recursos](#)
- + [Conclusão](#)
- + [Referências](#)



Objetivos

Ao final da leitura, esperamos que você seja capaz de:

- Entender o que é são contêineres;
- Usar os contêineres.



Introdução

Nesta apostila iremos aprender mais sobre os **Contêineres**, as suas propostas, funcionalidades e quais problemas esta tecnologia resolve.

Para nossa jornada apresentarei os conceitos de containerização, bem como suas vantagens.

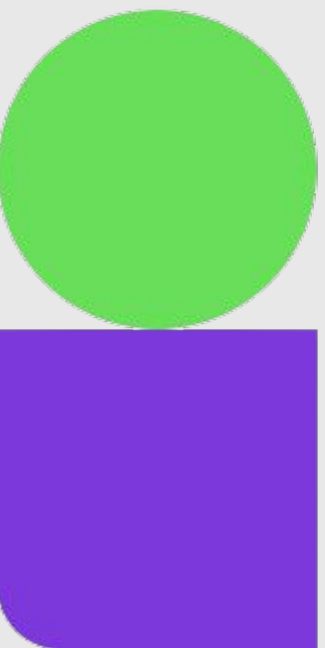
Iremos a fundo nas funcionalidades e assuntos importantes desta tecnologia. Vamos abordar conhecimentos importantes sobre as funcionalidades nos projetos de aplicações.

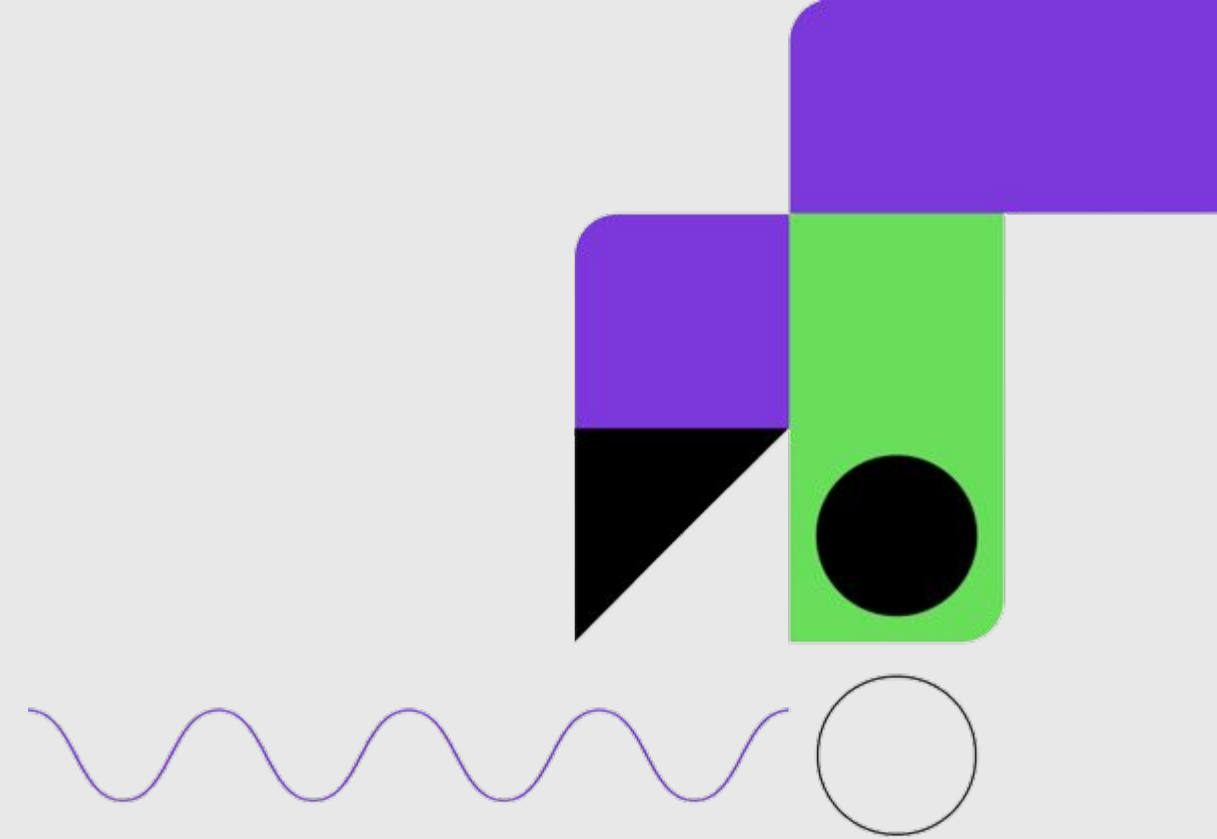


O que são os Containers?

Os containers são unidades executáveis de software nas quais o código do aplicativo é empacotado, junto às suas bibliotecas e dependências, de maneira semelhante para que possa ser executado em qualquer lugar, seja no desktop, na estrutura de TI tradicional ou na cloud.

Para isso, os containers utilizam um tipo de virtualização de sistema operacional (S.O.) na qual os recursos do S.O. (no caso do kernel do Linux, os namespaces e as primitivas de cgroups) são usados para ambos os processos isolados a fim de controlar a quantia de CPU, de memória e de disco à qual esses processos têm acesso.



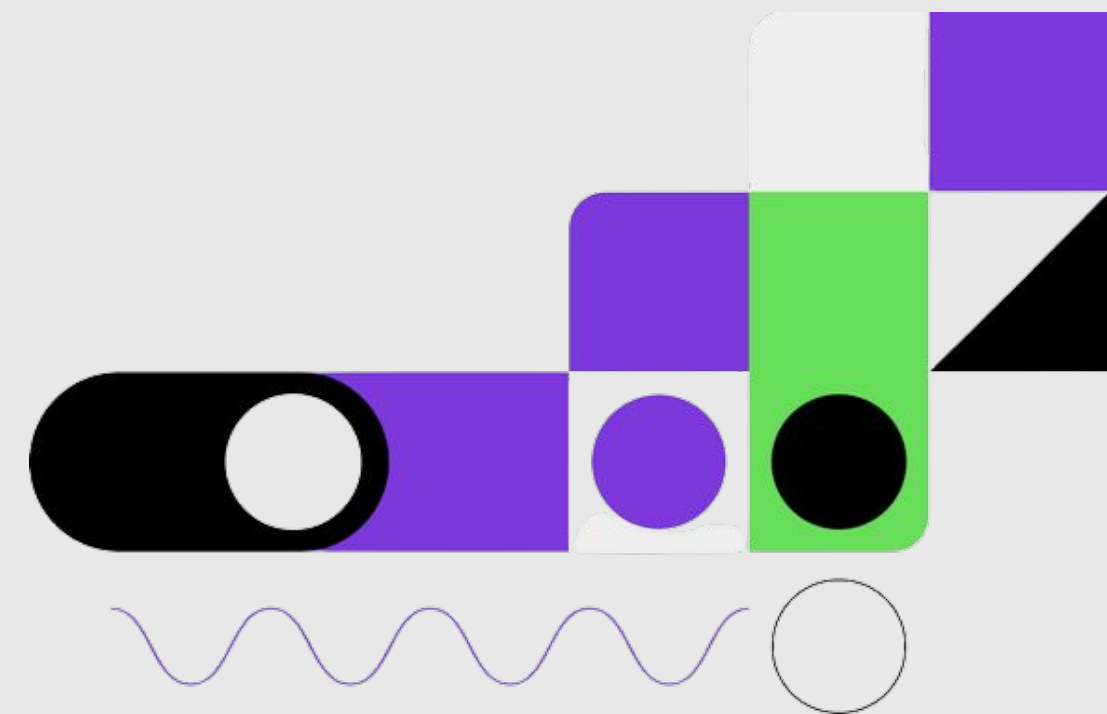


Contêineres

Contêineres

Os contêineres são pequenos, rápidos e portáteis, pois, ao contrário de uma máquina virtual, os contêineres não precisam incluir um sistema operacional convidado em todas as instâncias e podem apenas usar as funções e os recursos do sistema operacional de host.

Os contêineres surgiram pela primeira vez décadas atrás com versões como FreeBSD Jails e AIX Workload Partitions, mas a maioria dos desenvolvedores modernos lembra de 2013 como o início da era moderna dos contêineres com o surgimento do Docker.

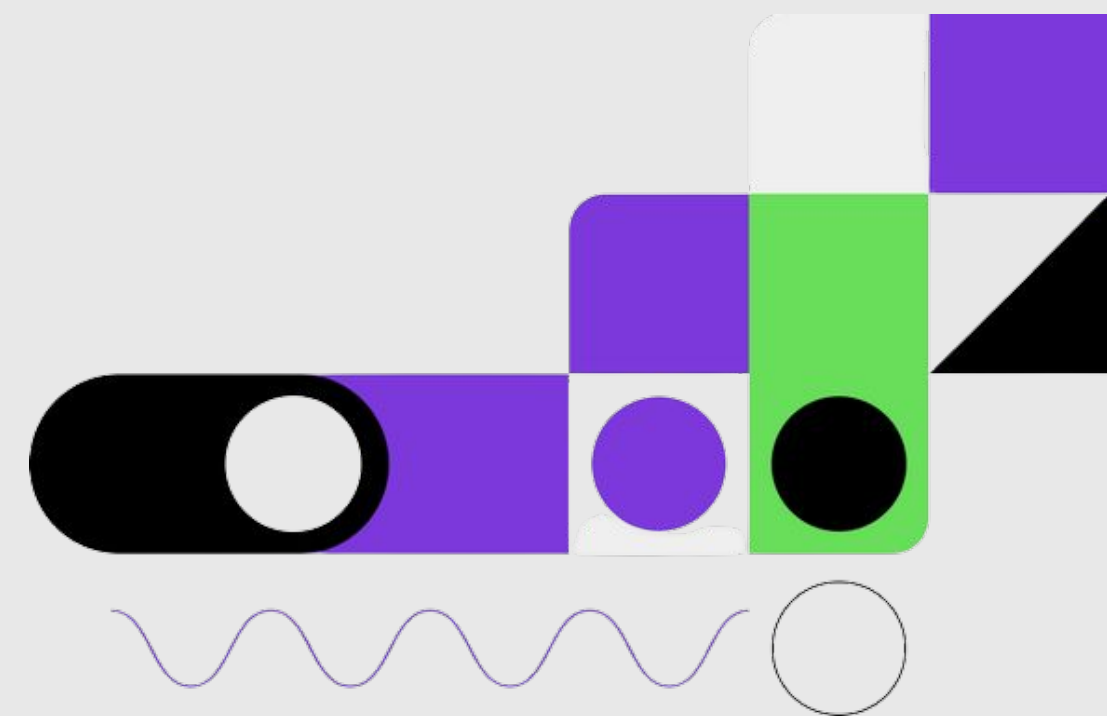


Contêineres VS Virtualização

Uma maneira de entender melhor um contêiner é entender como ele se difere de uma máquina virtual(VM) tradicional.

Na virtualização tradicional, seja on-premises ou na cloud, um hypervisor é potencializado para virtualizar o hardware físico.

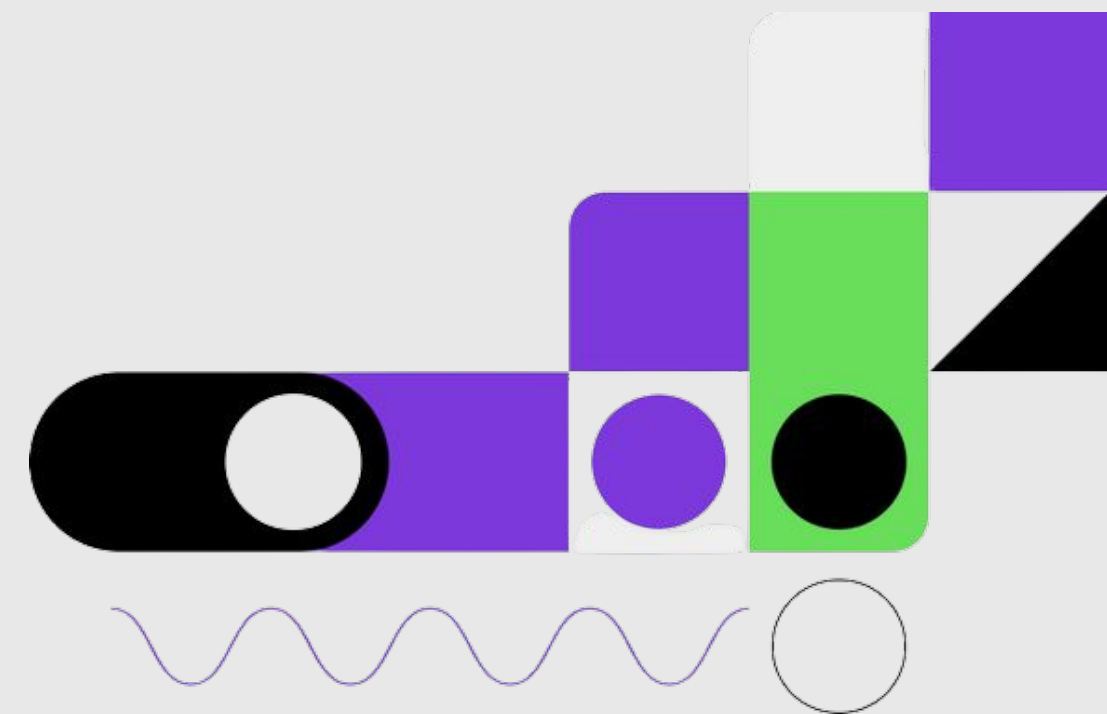
Cada VM contém um sistema operacional guest, uma cópia virtual do hardware que ele precisa para ser executado e um aplicativo com suas bibliotecas e dependências associadas.



Contêineres VS Virtualização

Em vez de virtualizar o hardware subjacente, os contêineres virtualizam o sistema operacional (geralmente Linux) de modo que cada contêiner individual contenha apenas o aplicativo e suas bibliotecas e dependências.

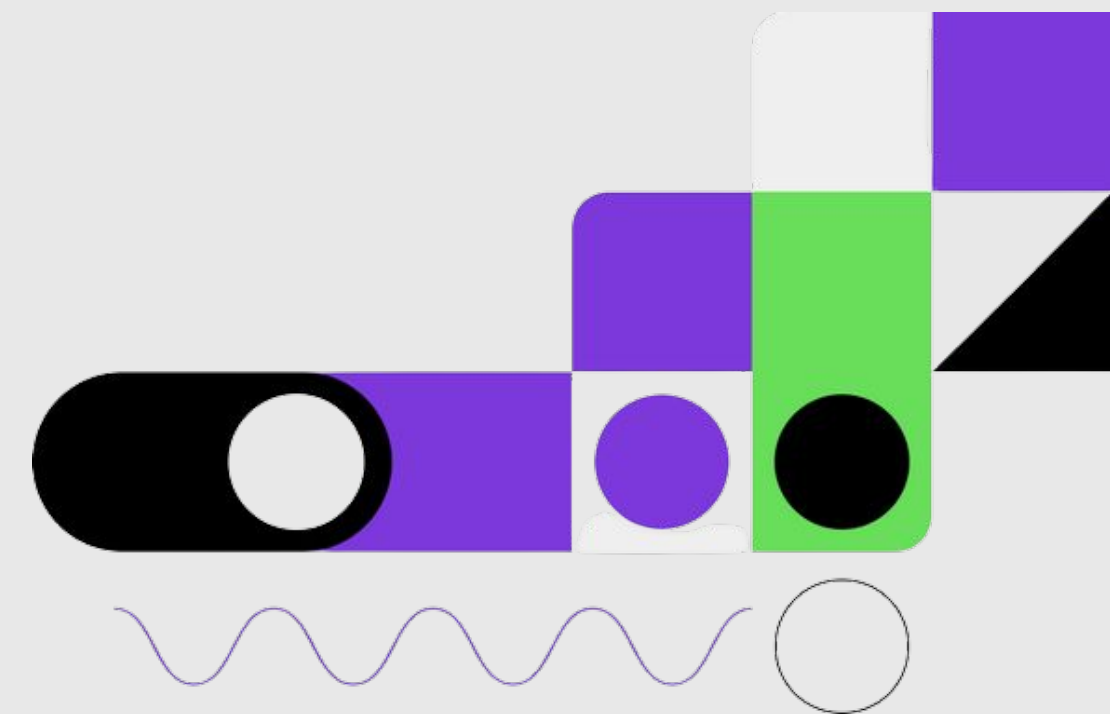
A ausência do sistema operacional guest é o motivo pelo qual os contêineres são tão leves e, portanto, rápidos e portáteis.



Benefícios

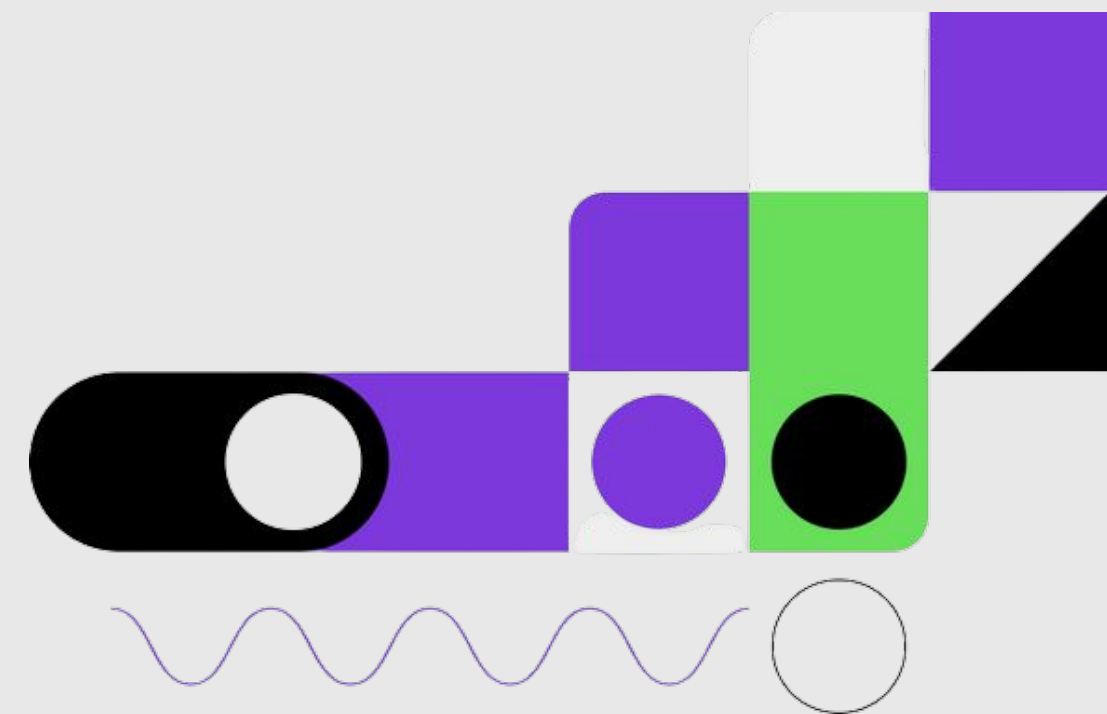
A principal vantagem dos contêineres, especialmente em comparação com uma VM, está em fornecer um nível de abstração que os torna leves e portáteis.

- **Leve:** os contêineres compartilham o kernel do sistema operacional da máquina, eliminando a necessidade de uma instância completa do sistema operacional por aplicativo e tornando os arquivos contêineres pequenos e fáceis de usar. Seu tamanho menor, especialmente comparado a máquinas virtuais, significa que eles podem se adaptar rapidamente e oferecer melhor suporte aos aplicativos nativos em cloud que ajusta a escala horizontalmente.
- **Portátil e independente de plataforma:** os contêineres carregam todas as suas dependências com eles, significando que um software pode ser codificado uma vez e ser executado sem precisar ser reconfigurado nos ambientes de computação como notebooks, clouds e on-premises.



Benefícios

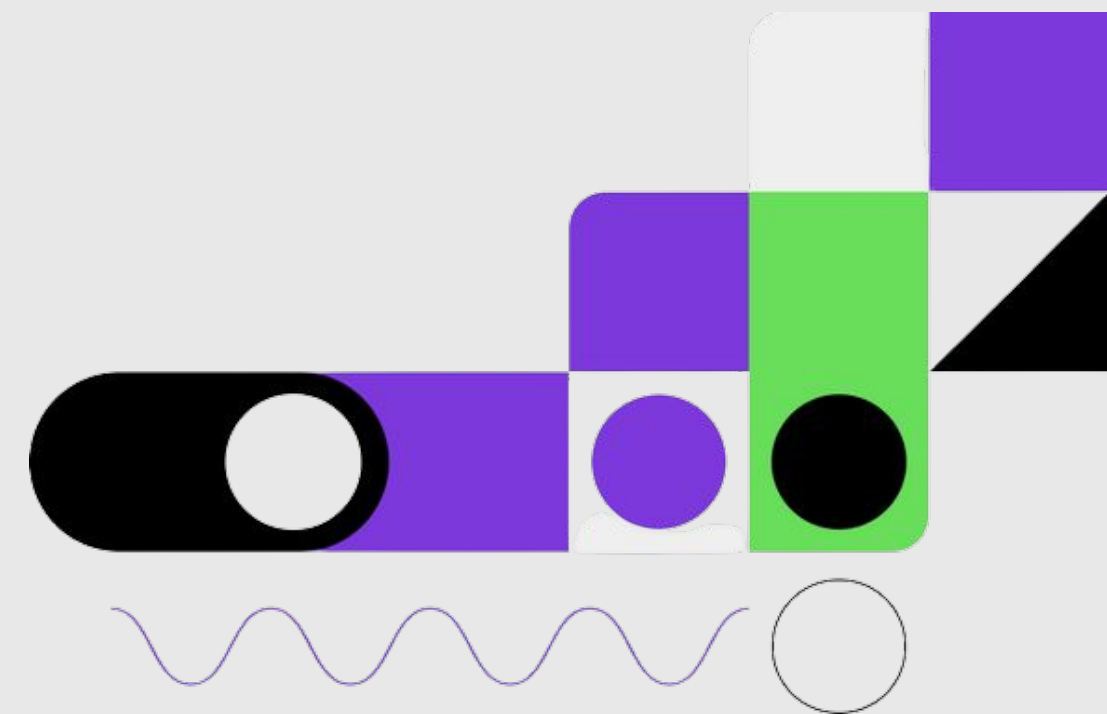
- **Oferece suporte ao desenvolvimento e arquitetura modernos:** devido a uma combinação de sua portabilidade/consistência de implementação entre plataformas e seu tamanho pequeno, os contêineres são ideais para o desenvolvimento moderno e padrões de aplicativos, como DevOps, sem servidor e microsserviços, que são desenvolvidos e são implementações regulares de código realizados em pequenos incrementos.
- **Melhora a utilização:** Assim como as VMs, os contêineres permitem que desenvolvedores e operadores melhorem a utilização da CPU e da memória de máquinas físicas. Um dos principais benefícios dos contêineres é que, como também permitem arquiteturas de microsserviços, os componentes de aplicativos podem ser implementados e serem ajustados em escala de modo mais granular, uma alternativa interessante a ter que aumentar a escala de um aplicativo monolítico inteiro, porque um único componente está em dificuldade para processar a carga.



Casos de uso

Contêineres estão sendo cada vez mais utilizados, especialmente em ambientes de cloud. Muitas organizações estão mesmo considerando substituir as VMs por contêineres como uma plataforma de computação de propósito geral para seus aplicativos e cargas de trabalho. Mas dentro desse escopo muito amplo há casos de uso que se destacam, nos quais os contêineres são especialmente relevantes.

Microserviços: Os contêineres são pequenos e leves, o que os torna ideais para arquiteturas de microserviço em que os aplicativos são desenvolvidos com muitos serviços menores, fracamente acoplados e que podem ser implementados de forma independente.

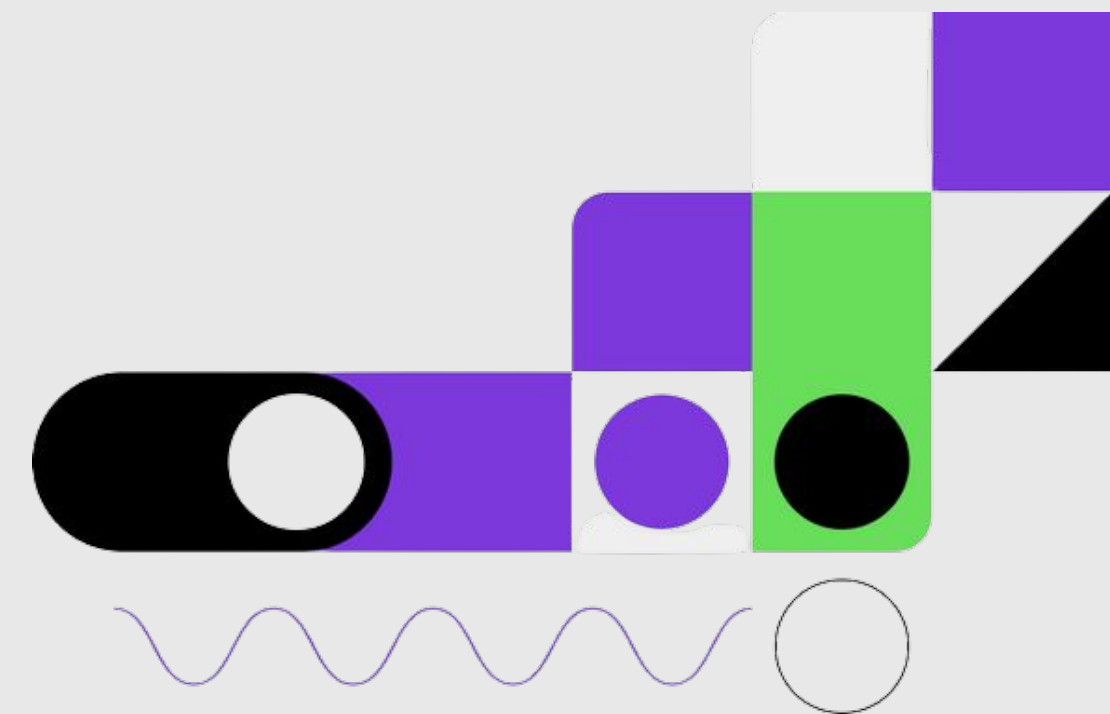


Casos de uso

DevOps: A combinação de microsserviços como arquitetura e contêineres como plataforma é uma base comum para muitas equipes que adotam o DevOps como forma de criar, entregar e executar o software.

Híbrido e multicloud: Como os contêineres podem ser executados de maneira consistente e em qualquer lugar, em ambientes de notebook, on-premises e na cloud, eles são uma arquitetura subjacente ideal para cenários de cloud híbrida e de multicloud, nos quais as organizações operam em diversas clouds públicas combinadas com seu próprio datacenter.

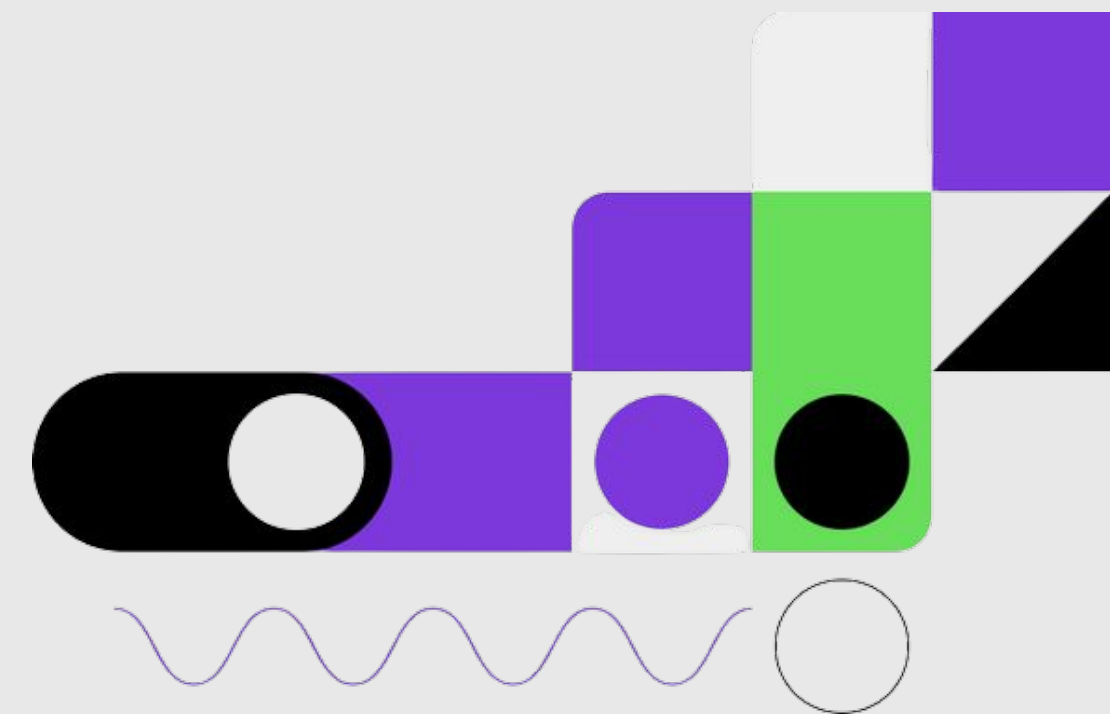
Modernização e migração de aplicativos: Uma das abordagens mais comuns de modernização de aplicativos consiste em colocá-los em contêineres para que sejam migrados para a cloud.



Docker

Quando o assunto é container, é impossível não lembrar de docker. Afinal, trata-se de uma plataforma open source desenvolvida justamente para gerenciá-los.

Dentre as suas funcionalidades, destaca-se o fato dele agrupar as partes de softwares de um sistema de arquivos completo e reunir os recursos mais importantes para sua execução. Logo, tudo que for instalado é armazenado nos contêineres, a fim de facilitar a execução em qualquer ambiente de desenvolvimento.

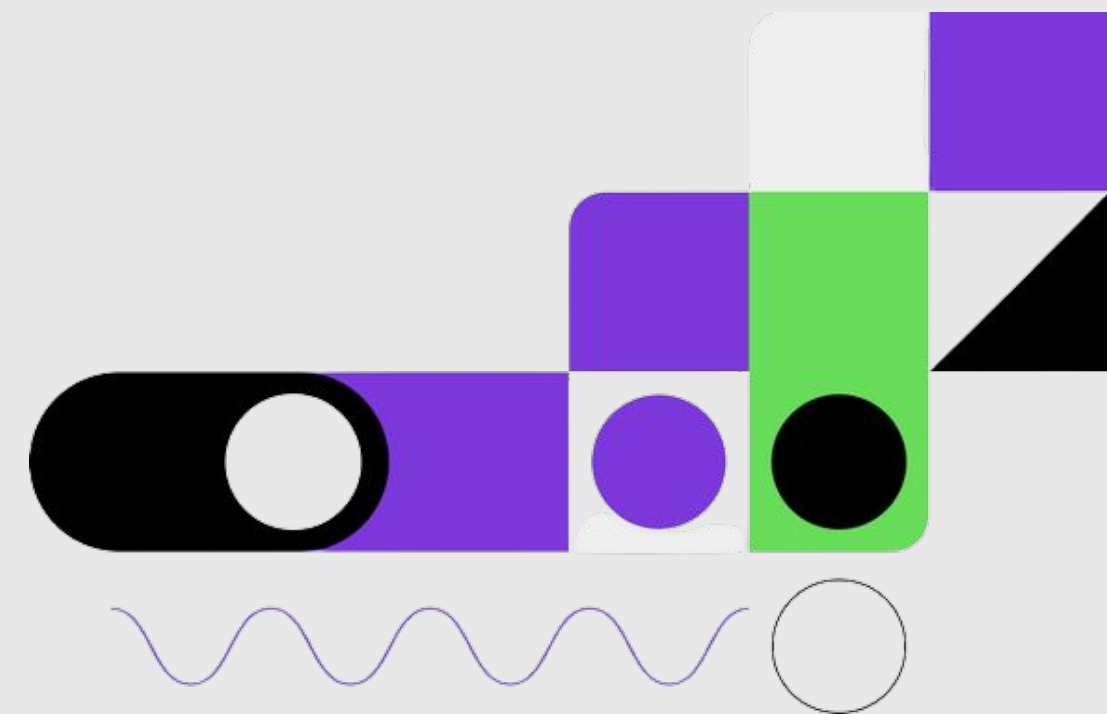


Docker

Tudo que você precisa para conteineirizar sua aplicação com Docker é efetuar o download da ferramenta e instala-la na sua máquina local (acesse esse [link](#) de download), e depois configurar o arquivo **Dockerfile** no seu projeto.

Exemplo de arquivo Dockerfile:

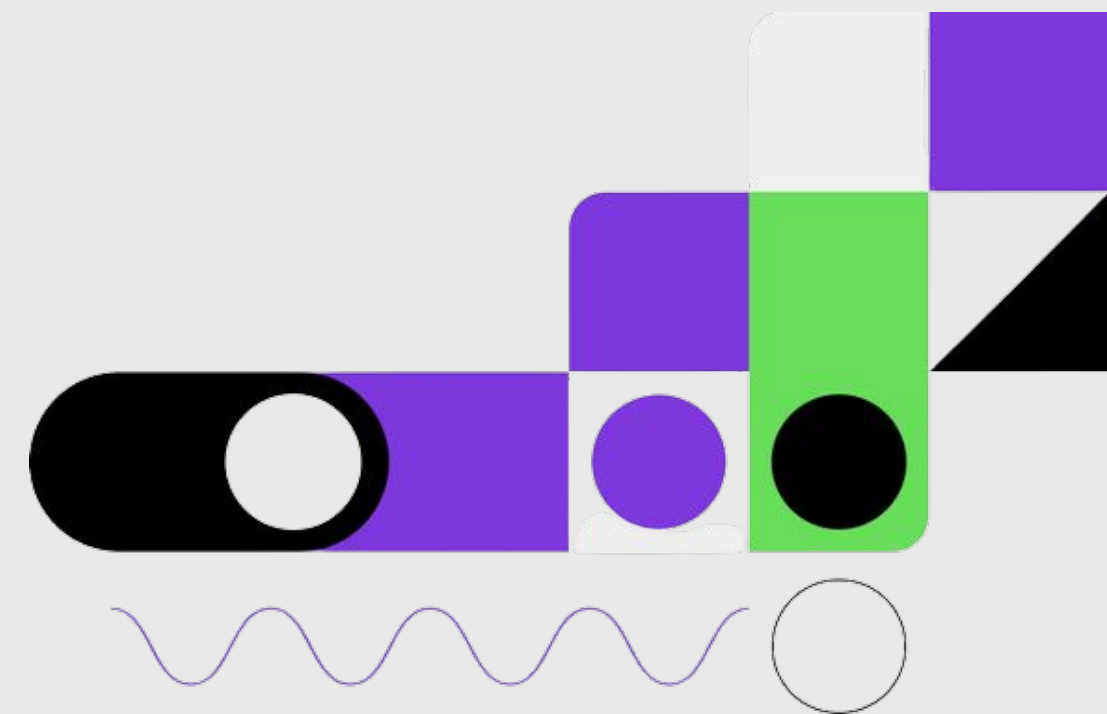
```
FROM node:12-alpine
RUN apk add --no-cache python2 g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```



Docker

O Dockerfile depende dos requisitos funcionais e da regra de negócio do seu aplicativo.

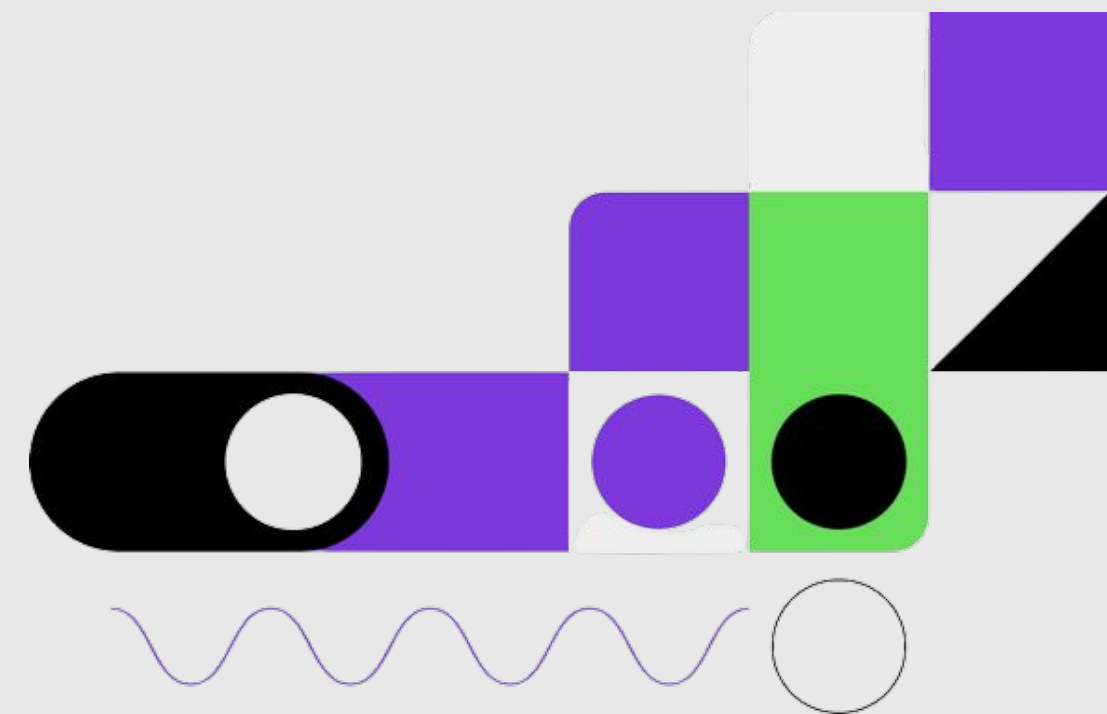
Para saber mais acesse esse [link](#) com mais exemplos de arquivos Dockerfile.



Porque Contêineres precisam do DOCKER?

Os Contêineres são práticos em ambientes de desenvolvimento, porém, o seu poder é ainda maior quando falamos em ambientes de produção. Uma vez que eles começam a tomar conta dos DataCenters, é preciso organizar, gerenciar e também monitorá-los.

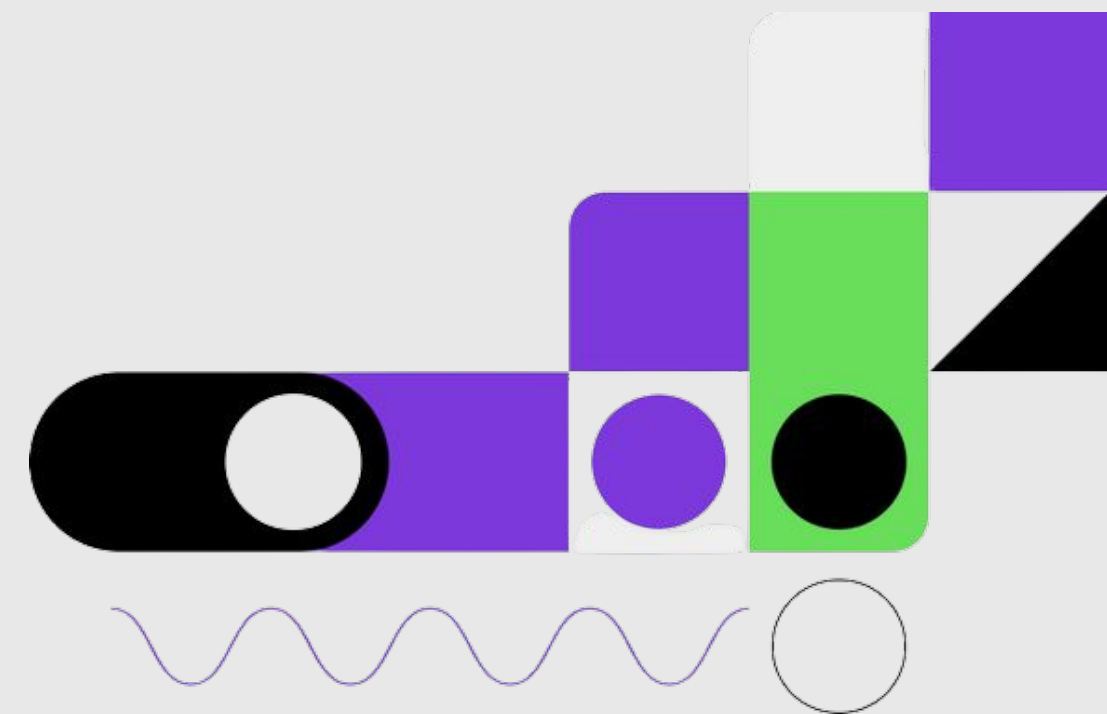
E é aqui que entra o papel dos orquestradores como o docker. Esses orquestradores são ferramentas essenciais porque conseguem provisionar e remover Contêineres de acordo com a demanda de requisições.



Porque Contêineres precisam do DOCKER?

Esse agrupamento de Contêineres, também conhecido como **Cluster**, consegue compartilhar armazenamento, tornando possível rodar dezenas e até centenas simultaneamente em um único local.

Imagine que você tem um site de vendas e toda vez que você anuncia nas redes sociais, o número de visitantes aumenta vertiginosamente. A solução de Contêineres pode ser utilizada nesse caso, pois com o orquestrador bem configurado é possível provisionar quantos Contêineres forem necessários.

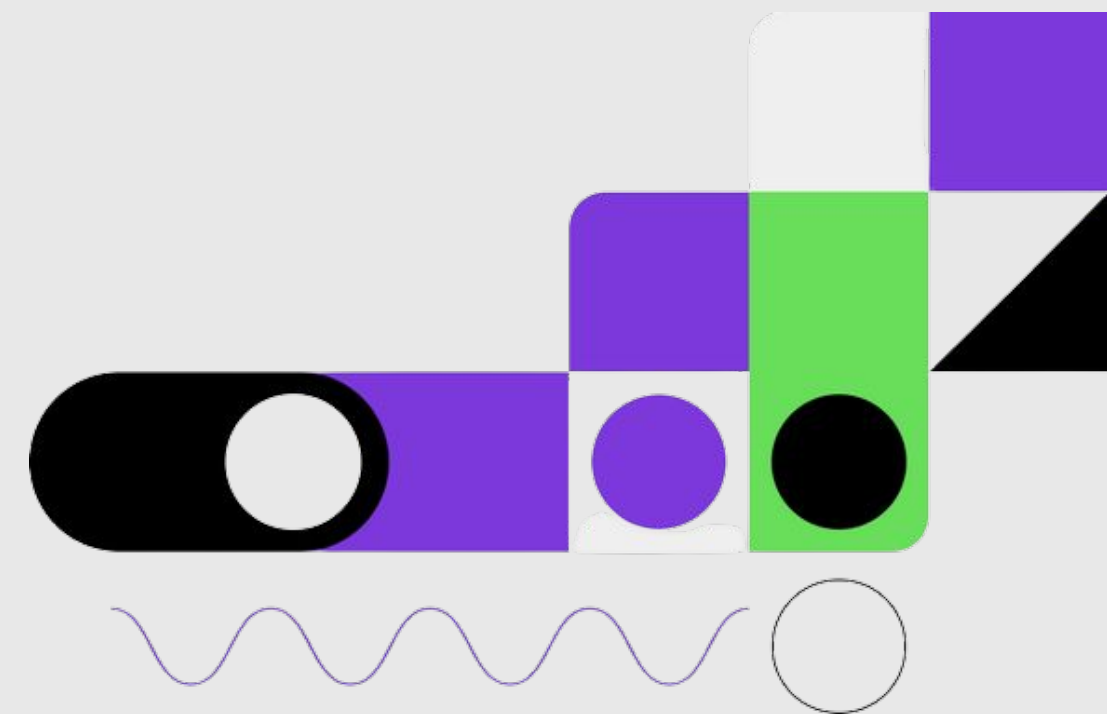


Porque Contêineres precisam do DOCKER?

Se cada Contêineres dá conta de 500 usuários fazendo requisições na loja virtual, se aumentar para 10 mil visitantes, o orquestrador vai criar novos. No momento em que as visitas forem diminuindo, os Contêineres vão sendo removidos aos poucos.

Além do docker, existem outras ferramentas de orquestramento, como:

- **OpenShift;**
- **Kubernetes.**



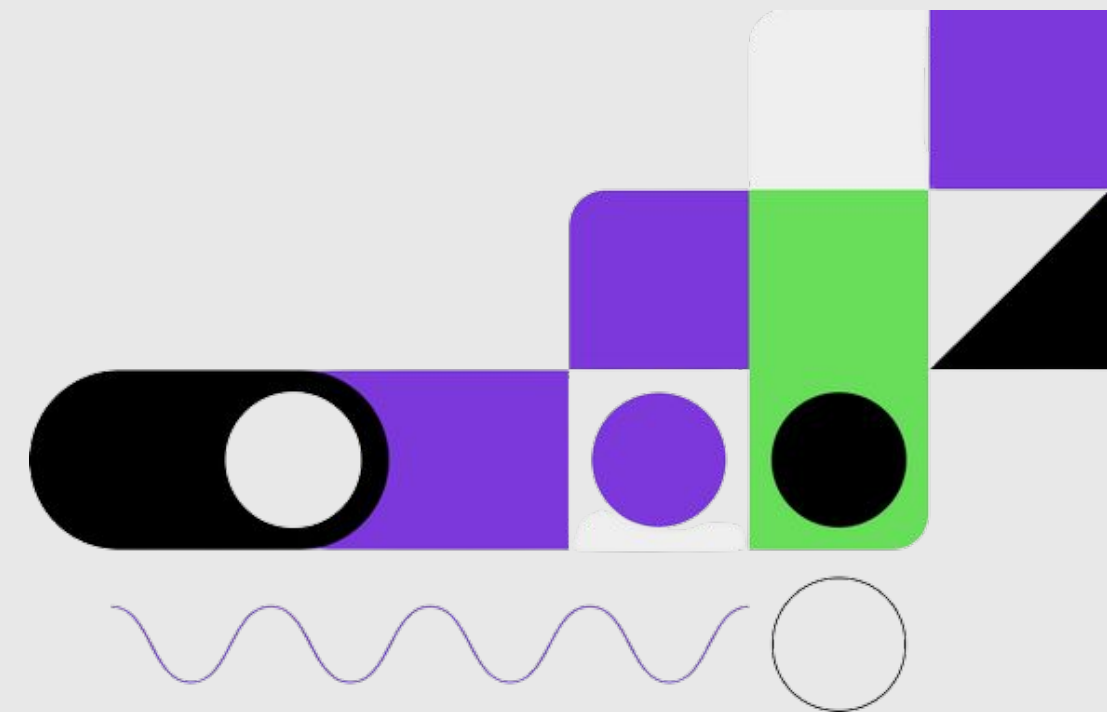
Compartilhamento

Quando pensamos em Contêineres, não estamos necessariamente replicando todas as aplicações nele. Até é possível fazer isso, mas na verdade, a replicação se dá apenas na infraestrutura que vai executar a aplicação.

Porém, o docker permite que sejam compartilhados:

- Arquivos;
- Banco de dados;
- Imagens.

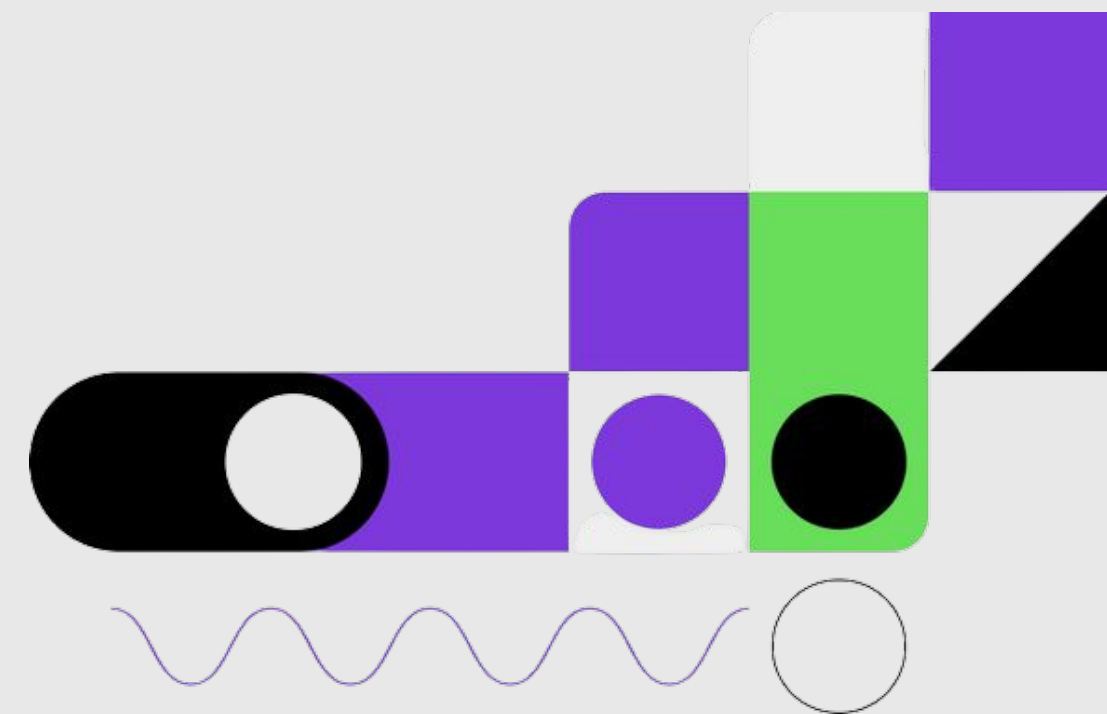
Neste caso, eles se tornam acessíveis para todos os Contêineres, como se eles fossem volumes próprios. Esse compartilhamento pode ser realizado via cloud, ou seja, através da nuvem.



Economia de Recursos

Com a possibilidade de ampliação ou redução na quantidade de Contêineres, não há mais a necessidade de investir em equipamentos robustos e ampla infraestrutura.

Quando há necessidade, como no exemplo da loja virtual, o docker se encarrega de garantir que as requisições serão atendidas.





Conclusão

Sobre o nosso fechamento:

Por ora aprendemos os conceitos e algumas funcionalidades sobre os Contêineres.

Tenha em mente que para solidificar esse conhecimento é interessante criar do zero uma aplicação e depois containerize ela. Configure o Dockerfile da sua aplicação para isso.

Teste.

Siga nos no LinkedIn clique [aqui](#).

Há várias outras ferramentas e tecnologias de containerização de projetos. Mas a mais famosa é o Docker Experimente e teste essa tecnologia.

"Sucesso é o acúmulo de pequenos esforços, repetidos dia e noite." - Robert Collier



Referência Bibliográfica

Referência: IBM (BR) (org.). Containers. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/containers>. Acesso em: 25 abr. 2022.

Referência: HostGator (BR) (org.). Containers o que são no mundo da programação. Disponível em: https://www.hostgator.com.br/blog/containers-o-que-sao-no-mundo-da-programacao/?gclid=CjwKCAjwu_mSBhAYEiwA5BBmf3j9BIN--6H53XSi_JtijnMp3BMjJuQ-pAz27_Va5YuQ0WMqReVZGxoC2xoQAvD_BwE. Acesso em: 25 abr. 2022.

