



# Como criar aplicações modernas com containers Linux

Containers: a tecnologia leve e nativa em nuvem que está revolucionando o desenvolvimento de aplicações.

# Conteúdo

## 3 INTRODUÇÃO

## 4 HISTÓRICO

## 5 CAPÍTULO 1: Princípios básicos

- 5 O que é um container Linux?
- 5 Melhor, mais rápido e mais econômico
- 6 Virtualização e containers
- 7 Containers + microsserviços = a combinação perfeita
- 7 Reduza, reutilize e recicle com imagens de container

## 8 CAPÍTULO 2: Melhore sua produtividade

- 8 O que você ganha com os containers?
- 8 Flexibilidade
- 8 Escalabilidade
- 8 Eficácia
- 9 Padronização real
- 9 Portabilidade
- 9 Qualidade de entregas
- 10 Ferramentas e linguagens familiares
- 10 Reputação no mercado

## 11 CAPÍTULO 3: Containers na prática

- 11 Lift-and-shift
- 11 Refatoração
- 12 Desenvolvimento de novas aplicações
- 12 Microsserviços
- 12 Aplicações híbridas
- 12 Tarefas repetitivas
- 12 Inteligência artificial e aprendizado de máquina

## 13 CAPÍTULO 4: Considerações e desafios

- 13 Antes de começar
- 13 Determine sua estratégia de dados
- 13 Estabeleça a comunicação entre seus containers
- 13 Faça a sincronização e padronização
- 14 Colete todos os logs
- 14 Reforce a segurança
- 14 Desafios
- 14 Acompanhar a evolução da tecnologia
- 14 Adotar a cultura DevOps
- 14 Controlar a segurança
- 14 Gerenciar e o monitorar

## 15 CONSIDERAÇÕES FINAIS

- 15 Saiba mais
- 15 Conheça mais recursos

# Introdução

## O que está incluído?

Arquiteturas monolíticas são coisa do passado. Ainda que "transformação digital" possa parecer só mais um termo da moda, ela é muito mais do que isso. A transformação digital proporciona velocidade, consistência e eficiência inéditas que estão mudando radicalmente a forma como os desenvolvedores trabalham.

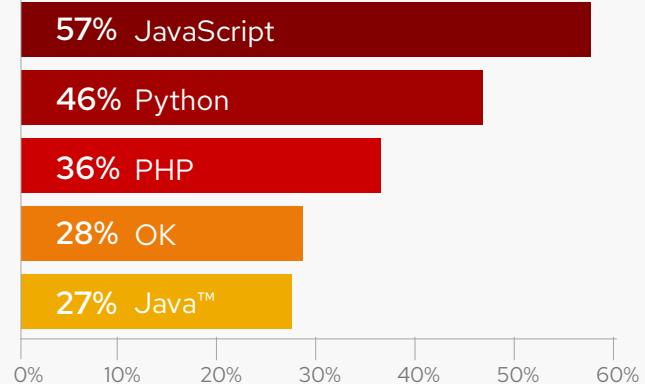
## Incluindo você.

Boa parte da responsabilidade e da pressão em cumprir com as promessas da empresa recai sobre os desenvolvedores de TI. No mundo atual, os usuários exigem novas aplicações, funcionalidades e atualizações para seus dispositivos em tempo real. Para atender a essas demandas, a tecnologia de containers é essencial. Eles permitem que você trabalhe de um jeito mais inteligente, criando ambientes consistentes capazes de acelerar o desenvolvimento e a entrega de aplicações nativas em nuvem executáveis em qualquer ambiente. Além disso, você pode adotar microsserviços para eliminar ciclos demorados de testes de regressão, fazer implantações sem interrupção e fornecer um mecanismo de patches e reversão de código de acordo com cada funcionalidade.

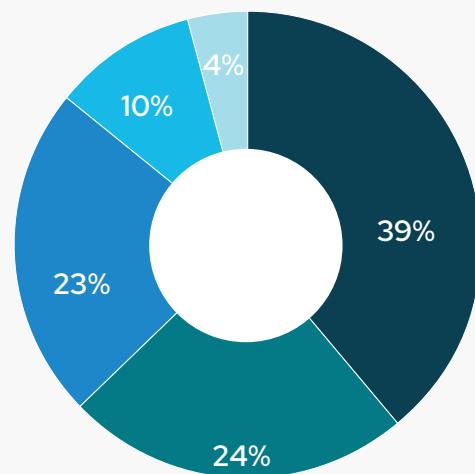
**Veja como implantar aplicações em containers.**  
Inscreva-se no curso de treinamento ►

Apesar de serem uma tecnologia relativamente nova, os containers já são adotados por desenvolvedores de todo o mundo e considerados uma ferramenta essencial para seu trabalho. O rápido aperfeiçoamento dos containers e as inovações nas ferramentas de gerenciamento e orquestração fizeram essa tecnologia alcançar uma adoção recorde, sendo usada atualmente por 49% dos desenvolvedores. É o que mostra o relatório da Digital Ocean de 2018.<sup>1</sup>

## LINGUAGENS USADAS NO DESENVOLVIMENTO EM CONTAINERS<sup>1</sup>



## VANTAGENS DOS CONTAINERS<sup>1</sup>



- Escalabilidade fácil
- Sem dependência de fornecedor
- Testes simplificados
- Testes mais rápidos
- Outros

# Histórico

## A jornada é mais importante que o destino

*"A invenção da prensa por Johannes Gutenberg aumentou muito a demanda por óculos. Com a nova prática de leitura, europeus em todo o continente descobriram que tinham problemas de visão. A demanda por óculos no mercado fez com que cada vez mais pessoas produzissem e testassem novas lentes, o que levou à invenção do microscópio. Logo depois, fomos capazes de descobrir que os nossos corpos são formados por células."*

### Steven Johnson

Como chegamos até aqui: seis inovações que moldaram a vida moderna

Como muitas outras grandes inovações, os containers são o ápice de vários conceitos e tecnologias que evoluíram ao longo do tempo. Nas décadas de 1970 e 1980, começamos a separar os códigos em objetos, experimentando as ideias de abstração e isolamento. Logo descobrimos que proteger algumas partes do código e expor outras proporciona maior controle sobre o processamento e o gerenciamento de dados, além de garantir flexibilidade para integrar sistemas adjacentes. Com essas inovações, desenvolvemos ainda mais componentes e processos de abstração e aplicação de camadas. Além disso, evoluímos para ambientes multicamada e arquiteturas orientadas a serviços (SOAs) que aumentaram o isolamento das camadas de dados, separando-as das interfaces de usuário e do código corporativo. Durante esse tempo, nossas metodologias também evoluíram. Passamos pelo desenvolvimento monolítico em cascata e pelo ciclo de vida de desenvolvimento de software (SDLC). Então, avançamos para o desenvolvimento ágil e scrum e, enfim, chegamos ao DevOps e à entrega contínua.

Do ponto de vista do negócio, essas inovações serviram para simplificar o processo de desenvolvimento, proporcionando uma produção melhor, mais rápida e mais econômica. Do ponto de vista do desenvolvedor, cada nova iteração de inovação encurtou o ciclo de desenvolvimento e gerou padrões e metodologias mais rigorosas a serem seguidos. Embora essas inovações tenham sido progressos, a invenção dos containers foi a consolidação de todos eles, levando a uma solução que nos permite criar códigos realmente flexíveis, interoperáveis e portáteis.

Basta observar a evolução dos containers Linux® para entender o tamanho do impacto que eles causaram nos últimos 20 anos.

### 2000

A tecnologia dos containers surge como jails do FreeBSD. Isso possibilitou o particionamento dos servidores em subsistemas que podem ser disponibilizados para os desenvolvedores sem comprometer o sistema como um todo.

### 2001

O conceito dos containers chega ao Linux por meio do projeto Linux-VServer. com o objetivo era executar vários servidores Linux de uso geral em uma única caixa.

### 2007

Mais tecnologias são incorporadas para possibilitar a containerização – mais especificamente, a inclusão de grupos de controle (cgroups), systemd e namespaces do usuário/kernel. Essas tecnologias trouxeram funcionalidades gerais de controle e virtualização que serviram de framework para a separação dos ambientes.

### 2008

O Docker surge com sua tecnologia trazendo ainda mais conceitos e ferramentas destinadas a acelerar o desenvolvimento e permitir o compartilhamento de novos containers em camadas.

### 2012

A arquitetura de microserviços evolui como um aprimoramento da SOA usado para criar aplicações flexíveis, independentes e implantáveis.<sup>4</sup>

### 2015

O Kubernetes é lançado como um sistema open source de orquestração de containers para automatizar o gerenciamento, o escalonamento e a implantação de aplicações.

# Capítulo 1: Princípios básicos

## O que é container Linux?

O container é simplesmente uma maneira inovadora de abstrair um ou mais processos do restante do sistema. Os containers deixam a carga "mais leve", possibilitando o trabalho em pequenos subconjuntos de código sem impactar o ambiente de execução geral. Eles também fornecem uma forma padrão de empacotar e isolar dependências, configurações e código da aplicação em um objeto.



### Melhor, mais rápido e econômico

O valor real dos containers é a portabilidade. Os containers Linux incluem todos os arquivos necessários para executar um componente, funcionalidade ou aplicação em uma mesma imagem. Eles fornecem consistência e previsibilidade em todas as fases, do desenvolvimento à produção. Isso torna sua implantação mais rápida, confiável e econômica em comparação com os pipelines monolíticos que dependem de ambientes de desenvolvimento, teste e produção replicados. Além disso, é possível gravar e desenvolver códigos em containers apenas uma vez e implantá-los em vários ambientes operacionais. Isso elimina a necessidade de prazos adicionais, ciclos de testes demorados e processos de implantação específicos.

Os containers têm um mesmo sistema operacional instalado no servidor e são executados como processos isolados do recurso, o que permite implantações rápidas, confiáveis e consistentes em qualquer ambiente.

O valor real dos containers é a **portabilidade**.

## Virtualização x containers

Containers e virtualização parecem a mesma coisa, mas eles são bem diferentes.

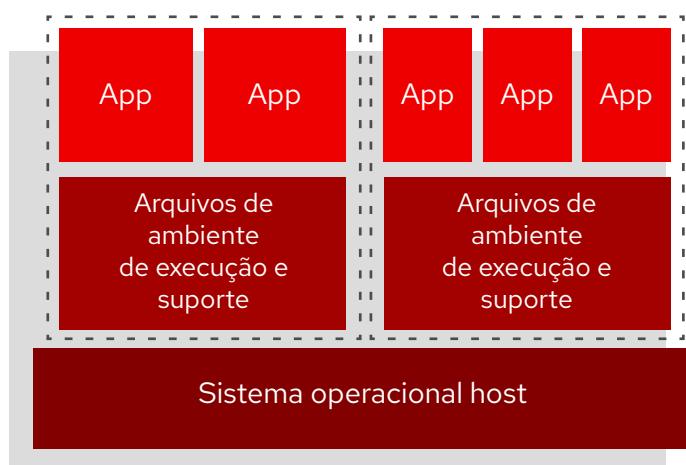
### VIRTUALIZAÇÃO



Com a virtualização, você executa vários computadores independentes em um mesmo hardware. Os sistemas operacionais e suas aplicações compartilham os recursos de hardware de um servidor host. Cada máquina virtual requer o próprio sistema operacional subjacente. Um hipervisor cria e executa as máquinas virtuais.

VS.

### CONTAINERS



O container isola os processos da aplicação do restante do sistema, incluindo apenas o necessário para executar uma aplicação específica, como alguns arquivos do sistema operacional, bibliotecas e programas de suporte e recursos do sistema. Os containers são leves, inicializam muito mais rápido do que as máquinas virtuais e exigem bem menos memória do que elas.

## Containers + microserviços = a combinação perfeita

Agora que você sabe um pouco mais sobre containers, vamos falar sobre seu principal uso para os profissionais: o desenvolvimento de microserviços.

Os microserviços são aplicações pequenas, independentes e de função única que se comunicam por meio de interfaces de programação de aplicações (APIs). Um dos princípios básicos dessa arquitetura é que cada microserviço gerencia apenas uma função e conta com uma API avançada que possibilita a comunicação de entrada e saída do código. Microserviços são o mecanismo de encapsulamento. E como são totalmente independentes, implementar alterações neles gera menos riscos para a aplicação do que num código com estrutura monolítica. Além disso, os microserviços são mais rápidos e ágeis do que as aplicações tradicionais. Isso ocorre porque eles têm natureza independente e usam os recursos do sistema de forma autônoma.

Os containers e os microserviços não dependem um do outro e, muitas vezes, são usados separadamente. São tecnologias com finalidades diferentes. Quando implementados em conjunto, eles são uma ferramenta poderosa para a criação de aplicações portáteis e nativas em nuvem.

Pense nos containers como uma tecnologia que capacita os microsserviços. Os containers são abstraídos do sistema operacional host e incluem todos os arquivos de ambiente de execução e de suporte necessários para executar o código contido neles. Quando você implanta um container, ele é executado seja qual for o sistema operacional subjacente. Os containers são portáteis e podem ser implantados em várias nuvens e dispositivos sem a necessidade de novas compilações ou testes.

A criação de microserviços com implantação em containers está se tornando o padrão do desenvolvimento corporativo. Essa arquitetura proporciona agilidade, velocidade e eficiência de recursos sem precedentes para diversas tarefas que fazem parte do seu trabalho diário como desenvolvedor. Ao usar os containers e os microserviços em um ambiente de DevOps, os desenvolvedores podem implantar cada serviço de maneira independente. Essa prática elimina a necessidade de consolidar as alterações no código, aumenta a qualidade dos testes, e ajuda a prevenir falhas tanto na fase de testes quanto na produção. Além disso, diferentes equipes de desenvolvedores podem trabalhar nas aplicações levemente acopladas e escolher o melhor stack de tecnologia para suas necessidades sem impor esses requisitos às outras equipes.

## Reduza, reutilize e recicle códigos

Os desenvolvedores adoram reaproveitar códigos. Com os containers, é possível criar imagens base, adicioná-las a um repositório e extraí-las sempre que você estiver pronto para iniciar um novo projeto. Imagens base são arquivos estáticos que não podem ser modificados e não incluem executáveis. Elas oferecem consistência e portabilidade, com capacidade para executar processos isolados em qualquer infraestrutura. As imagens são formadas por ferramentas e bibliotecas do sistema e outras configurações de plataforma que suas aplicações precisam executar.

Você pode criar suas próprias imagens de container ou escolher opções dos repositórios públicos. Diversos fornecedores de software como a Red Hat e a Microsoft disponibilizam imagens públicas das soluções deles.

0001100101000110111000100100101  
0100100011111001001001010101000  
I0101100  
  
**Microsserviços**  
**e containers são**  
**uma combinação**  
**poderosa,**  
**principalmente**  
**quando integrados**  
**em um ambiente**  
**de DevOps.**

## PRINCIPAIS VANTAGENS DAS IMAGENS DE CONTAINER

1

## Automatize os processos de compilação e implantação

2

Crie tags para facilitar a localização e o download

3

## Simplifique as verificações de vulnerabilidades

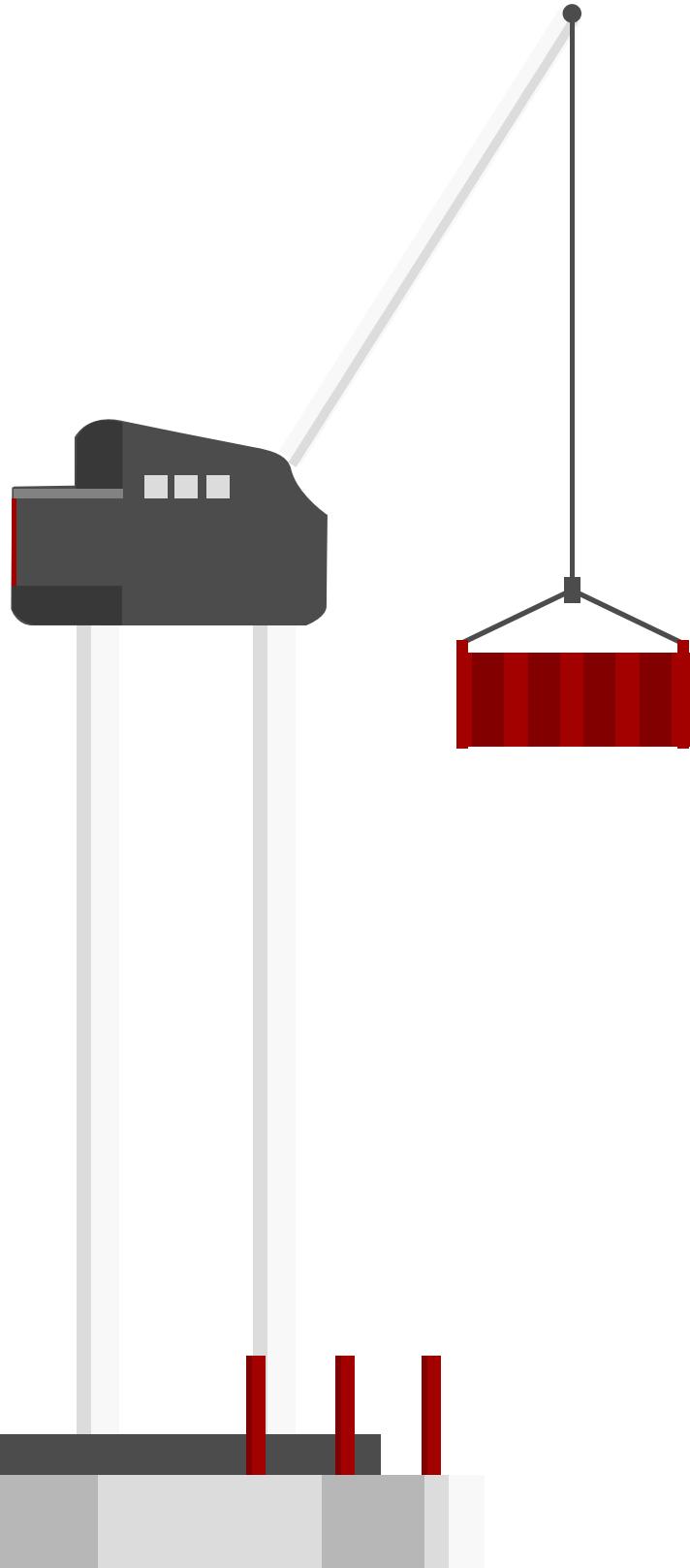
4

## Aplique políticas e padronização de código

5

## Economize tempo e estimule o reuso

# Capítulo 2: Aumente sua produtividade



## Vantagens do container para você

Com os containers, você adota uma abordagem otimizada para criar, testar, implantar e reimplantar aplicações. Eles são uma ótima opção para diversos casos, dos projetos mais simples às aplicações de missão crítica. Os containers oferecem diversas vantagens para a TI e para os negócios. Vejamos agora como eles podem beneficiar os desenvolvedores.

### Mais Flexibilidade

Os ciclos de desenvolvimento demandam muito tempo. A empresa está focada em seus objetivos estratégicos, a equipe de testes tem um backlog de falhas complicadas para resolver, o time de operações está lutando manter os sistemas atualizados e seguros e os clientes têm suas próprias prioridades. Infelizmente, a quantidade de horas de trabalho em um dia e de desenvolvedores em um projeto é limitada. A execução das aplicações em containers possibilita que você, a empresa e as equipes de operações respondam em tempo real a logs de alteração, erros, problemas de segurança, níveis de patch e novas solicitações dos clientes. E como a implantação de um container não afeta o restante da arquitetura da aplicação, é possível fazer alterações rapidamente, sem esperar que outras equipes terminem seus ciclos de desenvolvimento.

### Pense grande

Os containers são leves e, muitas vezes, podem ser inicializados em milissegundos. Eles não exigem uma inicialização do sistema operacional e carregam apenas as dependências de que precisam. Criar, replicar ou excluir containers também são tarefas muito rápidas. Vamos supor que os clientes criem cargas temporárias, ou a empresa decida incorporar novas equipes de usuários. Para responder a esses requisitos, as equipes de operações adicionam recursos que os containers usarão conforme necessário. Além disso, é possível fazer implantações em novas nuvens para escalar horizontalmente as aplicações em containers para novos usuários ao redor do mundo.

### Trabalho mais inteligente

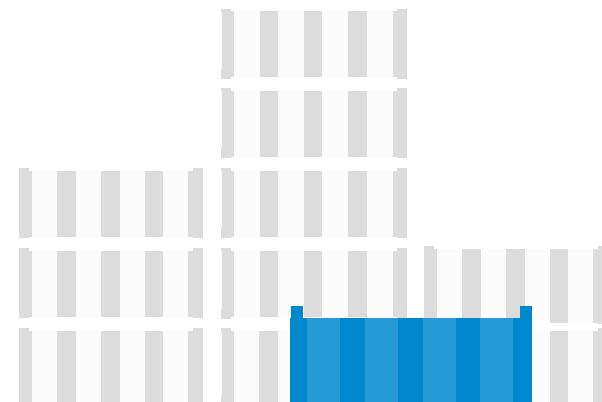
Com os containers, você pode se concentrar na lógica da aplicação sem se preocupar com configurações ou versões específicas do sistema operacional. Afinal, é por isso que você tem uma equipe de operações de TI, certo? Além disso, é possível empacotar o código, as dependências e as configurações em uma unidade encapsulada para controlar versões, testar e implementar com toda facilidade. Combinar os containers a uma arquitetura baseada em serviços também simplifica o suporte, os testes e o aprimoramento das suas aplicações.

## Conquiste a verdadeira padronização

Todos sabemos que manter a padronização dos ambientes do desenvolvimento à produção sempre foi complicado. Até agora. Uma das maiores vantagens dos containers é que eles padronizam ambientes locais, de desenvolvimento, testes, controle de qualidade e produção. Com essa previsibilidade, é possível iniciar ambientes isolados e economizar tempo na depuração e diagnóstico de problemas causados por diferenças nos níveis de patches, sistemas operacionais e aplicações. Com os containers, o seu tempo é investido no desenvolvimento e na entrega de novas funcionalidades. E mais: desenvolvedores novos na equipe podem começar a trabalhar imediatamente, sem precisar instalar e configurar ambientes locais. Basta extrair uma imagem de container do repositório e começar a programar.

## Grave uma vez, execute em qualquer lugar

O conceito "grave uma vez, execute em qualquer lugar" surgiu com algumas limitações, sendo a maior delas é relacionada ao sistema operacional no qual a aplicação foi desenvolvida. Você pode executar sua aplicação em qualquer lugar, desde que o sistema-alvo rode no mesmo sistema operacional. Com os containers, o código é realmente executado em todos os ambientes, seja qual for o lugar onde ele foi desenvolvido. Quando você cria em um container, é possível implantar em bare-metal, máquinas virtuais, nuvens públicas e privadas e ambientes híbridos baseados no Linux, Windows e MacOS. Além disso, a ampla adoção do projeto open source do Docker, permite que você automatize a implantação das aplicações em containers de forma estável. Com isso, você tem mais tempo para iniciar seu próximo ciclo de desenvolvimento.



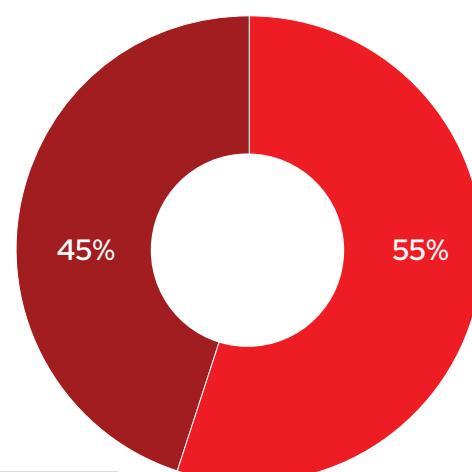
## Crie aplicações de alta qualidade

Os containers aceleram os testes e a solução de problemas para todos os envolvidos. Ao conter o desenvolvimento em uma única funcionalidade da aplicação, você diminuir o risco de incluir erros no código adjacente por descuido. Depois de colocar o container no controle de versão, as equipes de controle de qualidade podem testá-lo diretamente em relação à imagem. Assim, elas não precisam extrair e criar a aplicação inteira a partir de um servidor de integração contínua (CI). Isso simplifica os testes e economiza tempo.

Os containers também ajudam as equipes de suporte e controle de qualidade a identificar a causa de um problema na aplicação com mais rapidez. Como o código é levemente acoplado e orientado a APIs, há menos interdependências para analisar quando algo dâ errado. Os engenheiros descobrem o motivo do erro rapidamente e localizam o container que o causou. Após a identificação do problema, é fácil reverter as mudanças. Containers individuais podem ser revertidos sem impactar o restante da aplicação. Depois que o container voltar para você, será necessário menos tempo para solucionar o problema. Assim, você pode se concentrar mais no desenvolvimento das funcionalidades que os líderes da empresa e os clientes desejam. Com os containers, corrigir erros no mesmo dia é uma realidade. Isso que mantém os clientes – e seu chefe – satisfeitos.

De acordo com um estudo recente da IDC,

55% dos líderes de TI implantaram containers localmente, enquanto 45% o fizeram em nuvens públicas.<sup>5</sup>



### Use suas ferramentas e linguagens favoritas

Como desenvolvedor, você sabe que é libertador poder escolher as ferramentas e as linguagens que serão usadas em um projeto. Com os containers, você tem flexibilidade para fazer isso. Alguns ambientes de execução de aplicações são mais adequados para determinados tipos de cargas de trabalho ou arquiteturas. Por exemplo, o Vert.x é compatível com arquiteturas distribuídas e reativas, que são ideais para aplicações responsivas em tempo real, como as exigidas em dispositivos de Internet das Coisas (IoT). Claro que é possível criar esse tipo de aplicação em outra linguagem. No entanto, você precisaria modificar o que o Vert.x fornece como algo nativo. Os containers são verdadeiramente independentes e fornecem flexibilidade e controle. Por isso, é possível escolher a ferramenta adequada para uma aplicação sem obrigar equipes inteiras a adotarem as mesmas ferramentas e linguagens nos projetos delas.

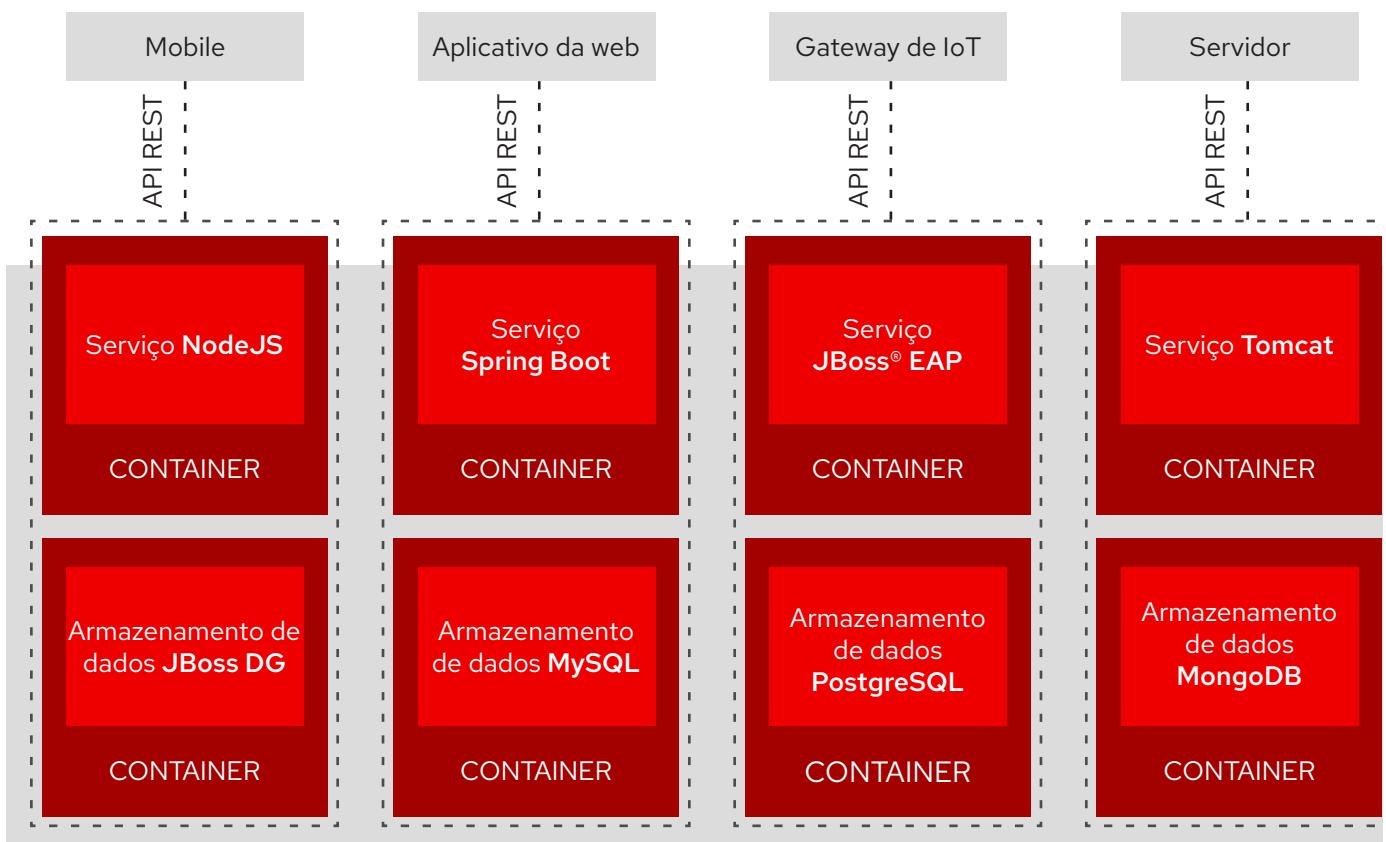


Figura 1. Aumente a flexibilidade do desenvolvimento e simplifique a implantação com os containers

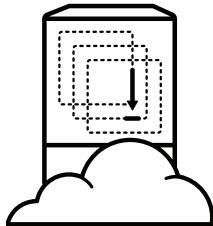
### Aumente seu reputação

Essa última vantagem é muito importante. Com o avanço das tecnologias, você pode acabar ficando para trás se não desenvolver novas habilidades. Ou pior, ficar trabalhando em aplicações legadas até se aposentar. Oportunidades de crescimento e desenvolvimento são essenciais para o seu sucesso. Além disso, desenvolvedores costumam adorar aprender novas tecnologias. É um privilégio desse trabalho.

As tecnologias de microsserviços, containers e soluções de orquestração, além de muito procuradas, são também tecnicamente desafiadoras. De acordo com uma pesquisa da Red Hat® de 2018, a estimativa é que o uso dos containers aumente em 89% nos próximos dois anos. E essa tendência de adoção rápida e massiva continuará no futuro.<sup>6</sup> Ao se especializar no desenvolvimento de microsserviços baseados em containers, você abre caminho para novas oportunidades de crescimento profissional e agrega valor à sua empresa.

# Capítulo 3: Containers na prática

É provável que muito brevemente peçam a você para entregar o código em containers (se isso já não tiver acontecido). À medida que as empresas passam pelos processos de transformação digital, as equipes de liderança aprendem algo importante: com os containers, as empresas de TI usam melhor seus recursos humanos e tecnológicos para reduzir os custos, aumentar a eficiência e agregar valor. Ao planejar seus projetos, considere estes casos de uso. E não deixe de mencionar a ideia dos containers na próxima reunião da sua equipe.



## "Lift and shift"

As aplicações nativas em nuvem vêm sendo bastante discutidas atualmente, no entanto, a maioria das aplicações corporativas ainda é monolítica. O lift-and-shift é o processo de migração das aplicações para uma arquitetura nativa em nuvem mais moderna. Elas são migradas para a nuvem com o mínimo possível de alterações no código. Esse processo geralmente rápido, é o preferido dos líderes empresariais. Isso acontece porque é possível usar aplicações de missão crítica existentes para agregar mais valor e aumentar o desempenho. Tudo isso, sem precisar de grandes investimentos em restruturação.

Mesmo sendo simples e rápido, o lift-and-shift é considerado uma solução temporária, já que ele não aproveita totalmente as ferramentas específicas da nuvem. Ao realizar o lift-and-shift de uma das suas aplicações, é preciso reavaliá-la em algum momento para garantir que ela ainda esteja agregando valor e determinar a necessidade de refatoração ou substituição.



## Refatoração

A refatoração sempre foi uma parte importante da modernização de aplicações. Nesse caso, estamos falando especificamente sobre a refatoração de aplicações para containers. Mesmo sendo mais intensiva do que uma migração lift-and-shift, a refatoração possibilita que as aplicações utilizem todas as vantagens dos ambientes nativos em nuvem e em containers.

Antes, os trabalhos de refatoração costumavam ser negligenciados por serem complexos e demorados. Com os containers, a refatoração é um processo gradual. Você modifica primeiro apenas as partes mais importantes de uma aplicação e realiza o lift-and-shift do que sobrou. Depois, você refatora o restante ao longo do tempo. Assim, é possível manter os negócios em operação enquanto implementa novas melhorias de acordo com o cronograma. Muitas vezes, esse processo é chamado de "substituição do monólito" (strangling) em livros e blogs relacionados a microserviços.

Claro que ainda é possível fazer a refatoração do jeito antigo, modificando toda a aplicação antes de liberá-la para uso. Ambas as estratégias têm seus prós e contras. A principal vantagem de uma refatoração gradual é o tempo. A desvantagem é a chance de não concluir totalmente a migração para uma plataforma nativa em nuvem, o que gera riscos e complexidade no gerenciamento.

## Desenvolvimento de novas aplicações

Como você já viu, os containers são uma ferramenta poderosa e flexível para desenvolver novas aplicações nativas em nuvem. Quando você desenvolve do zero, é possível aproveitar todas as vantagens dos containers. Eles fornecem uma plataforma ideal para microsserviços, aplicações híbridas, automação de tarefas repetitivas e soluções modernas como inteligência artificial (AI) e aprendizado de máquina (ML).

### Arquitetura de microsserviços

Os containers e as arquiteturas de microsserviços combinam perfeitamente. É possível isolar, implantar e escalar microsserviços e aplicações distribuídas com facilidade usando containers separados como elementos essenciais.

### Aplicações híbridas

Com os containers, é possível padronizar a forma como o código é implantado. Assim, você facilita a criação de fluxos de trabalho de aplicações executados entre ambientes na nuvem e on-premise.

### Tarefas repetitivas

Para acelerar a inicialização das tarefas, basta desenvolver as que são repetitivas com facilidade nos containers. Por exemplo, o processamento em lote e a extração, transformação e carregamento (ETL). Depois, é possível automatizá-las para facilitar as operações e escalá-las dinamicamente para atender às demandas.

### Inteligência artificial e aprendizado de máquina

Os containers oferecem uma maneira inovadora de criar e implantar aplicações portáteis na nuvem que incorporem a AI e o ML. Eles escalam os modelos de AI e ML com rapidez para atender às necessidades de processamento dos algoritmos avançados de treinamento. Além disso, é possível implantar os containers de AI e ML perto das fontes de dados para aprimorar o desempenho e encurtar os ciclos de treinamento.

# Capítulo 4: Considerações e desafios

## Dicas para facilitar a sua vida

Já falamos bastante sobre o impacto e o poder dos containers. No entanto, assim como qualquer outra tecnologia, eles trazem desafios que você precisa enfrentar. Quando isso não é feito corretamente, os desafios podem se tornar um problema muito grande à medida que o número de containers corporativos implantados aumenta. Você precisa ficar de olho na evolução da tecnologia e acompanhar o desempenho da sua organização ao adotar, gerenciar e manter uma arquitetura baseada em containers.

### Considerações antes de começar...

Agora provavelmente você já deve ter algumas ideias para começar um novo projeto e refatorar uma aplicação legada. O melhor jeito de ganhar experiência e ampliar seus conhecimentos em containers é começar logo a codificação.

No entanto, você precisa considerar alguns fatores antes de começar. Criar uma aplicação em containers traz alguns desafios que não ocorrem em um ambiente tradicional. Veja nas informações a seguir um resumo das considerações que impactam o seu trabalho de desenvolvimento. Várias ideias desta seção foram inspiradas no [Project Atomic](#). Não deixe de conferi-lo quando terminar de ler este e-book.

#### Determine sua estratégia de dados

A maioria das aplicações nativas em nuvem é stateless, mas muitas exigem um armazenamento persistente. Os containers têm armazenamento imutável: os dados são perdidos quando os containers são encerrados. Você precisa considerar esse fator e projetar suas aplicações de modo que os dados persistam em qualquer estado do container.

Caso seja necessário preservar os dados da aplicação após o encerramento de um container, basta atribuir um volume de armazenamento a ele. Esses volumes persistem em qualquer estado de container. Os desenvolvedores precisam garantir que as aplicações sejam projetadas para fazer gravações em um armazenamento compartilhado. No caso de aplicações corporativas, ferramentas como o Red Hat OpenShift®

Container Storage fornecem armazenamento definido por software criado especificamente para ambientes de containers. Essa solução garante um local de armazenamento persistente de dados, até mesmo durante a inicialização e o encerramento dos containers. Além disso, é possível escalá-la com facilidade em implantações virtuais, de bare-metal, nuvem e containers para aumentar a portabilidade sem limitar os containers à sua arquitetura de armazenamento.

#### Garanta a comunicação entre os containers

É necessário que os componentes distribuídos de uma aplicação se comuniquem uns com os outros para a realização dos fluxos de trabalho. As tecnologias de container possibilitam que os desenvolvedores criem pontos explícitos de interconexão. Além disso, elas fornecem um mecanismo de comunicação entre containers usando APIs. Isso é excelente para a comunicação entre containers, mas e os bancos de dados?

Geralmente, os bancos de dados tradicionais se comunicam usando um soquete por meio de uma rede. Esse tipo de mecanismo legado não funciona com os containers, já que o namespace deles é alterado de acordo a mudança de estado. Em uma aplicação em containers, é necessário ter uma plataforma de orquestração como o Kubernetes ou o Red Hat OpenShift Container Platform. Eles facilitam a comunicação de rede entre containers, bancos de dados ou outros recursos.

#### Sincronize e padronize

Algumas aplicações em containers exigem que determinados atributos do host e do container sejam sincronizados para garantir um funcionamento consistente. Por exemplo, imagine um servidor de registros centralizado que recebe dados de vários containers implantados em diversas regiões geográficas. As informações e os detalhes de data e hora do registro não terão muita utilidade para a equipe de operações se cada container informar um horário diferente do que o host, além de não reportar o local do servidor. Ao sincronizar e padronizar um conjunto de atributos do ambiente, você garante a precisão, a relevância e a utilidade dos dados comunicados de volta ao armazenamento central.

### Colete todos os logs

Todas as aplicações precisam registrar informações apropriadas para facilitar a solução de problemas. Quando a aplicação registra ações, erros e avisos em algum tipo de mecanismo, você precisa considerar a forma como os usuários poderão coletá-los, revisá-los e possivelmente guardá-los. Os containers são separados por namespace e não podem acessar diretamente os componentes de bare-metal, incluindo os discos rígidos locais. Por isso, você precisa repensar sua estratégia de logging.

O jeito mais fácil de coletar logs é usar uma ferramenta criada para esse trabalho. Com plataformas de orquestração de containers como o OpenShift Container Storage, você coleta dados de log relacionados automaticamente. É possível salvá-los em um armazenamento persistente ou seguir as recomendações de envio de logs à saída ou erro padrão para visualizá-los no console central de gerenciamento.

### Reforçar a segurança

Em uma aplicação em containers, é essencial armazenar dados confidenciais, como credenciais de acesso. Assim, os containers podem se comunicar sem pedir essas informações repetidamente aos usuários. No entanto, o armazenamento de credenciais pode ser algo complicado e expor sua aplicação a possíveis riscos de segurança. O jeito mais comum de transmitir dados confidenciais em um container é por meio de variáveis de ambiente que não são expostas ao público. As plataformas de orquestração de containers como o Kubernetes e o OpenShift Container Platform fornecem mecanismos nativos para proteger variáveis de ambiente e transmitir dados confidenciais em uma aplicação em containers.

## Desafios

### Acompanhe a evolução da tecnologia

A evolução e a expansão do ecossistema dos containers é extremamente rápida. Esse ecossistema inclui ferramentas para criar, implantar, configurar, automatizar e gerenciar containers. Os projetos open source compatíveis com esse ecossistema são muito ativos. Claro que esse é um ótimo sinal sobre o futuro da tecnologia, mas pode ser complicado manter as habilidades necessárias para entregar aplicações por meio dessas soluções. Como desenvolvedor, você precisa se atualizar sobre os projetos open source em andamento, inscrever-se em cursos de treinamento e ampliar suas habilidades. Assim, você continuará acompanhando o ritmo da evolução da tecnologia dos containers.

### Adote a cultura de DevOps

Como falamos no começo deste e-book, os containers estão mudando a forma como as aplicações são desenvolvidas. Por conta disso, as equipes de aplicações precisam modificar seus

processos e cultura para se adaptar. Para usar os containers com sucesso, as organizações precisam adotar uma cultura de DevOps íntegra, eliminando a separação tradicional causada pelo modelo de código monolítico. Elas precisam seguir e automatizar os princípios de DevOps para criar um fluxo de trabalho rápido, confiável, consistente e voltado à segurança, do desenvolvimento à implantação. Como desenvolvedor, você fará parte do processo geral, incluindo desenvolvimento, implantação e satisfação do cliente. Na sua nova função, você será responsável pelos containers e por certificar-se de que as lógicas de aplicação criadas possam ser executadas no ambiente de produção. Você será capaz de realizar correções em tempo real para depurar, corrigir e implantar códigos com rapidez, de modo que a parte da aplicação sob sua responsabilidade atinja todo seu potencial.

### Controle a segurança

A segurança é sempre um desafio para os desenvolvedores de aplicações que trabalham ou não com containers. O isolamento da aplicação é uma vantagem de segurança dos containers. No entanto, a proliferação deles em uma organização gera um novo tipo de risco de segurança.

Geralmente, os containers exigem que as aplicações sejam separadas em microserviços menores. Isso aumenta o tráfego de dados e as regras complexas de controle de acesso. Quanto mais containers você tem, maiores são as chances de criar controles de acesso fracos entre os eles. É necessário atender aos protocolos de segurança e acesso adequadamente para evitar que novas vulnerabilidades sejam geradas no seu ambiente de produção. Além disso, muitas organizações usam repositórios de imagens de containers. Elas precisam confirmar que essas imagens atendem aos requisitos de conformidade e segurança da organização. Você precisa entender os riscos relacionados aos containers e sempre atender às políticas de controle de acesso.

### Gerencie e monitore

À medida que a quantidade de containers implantados aumenta, fica cada vez mais difícil monitorá-los. Ao implantar soluções de nuvem híbrida, em que os containers são executados em nuvens públicas e privadas, a complexidade do gerenciamento aumenta consideravelmente. A implementação de um armazenamento de dados consistente para guardar, analisar e reagir aos eventos gerados a partir de todos os containers e aplicações pode ser um desafio. Isso também se aplica à localização de erros depois de coletar um evento. No entanto, há novas tecnologias, como o Istio, o Prometheus e o Jaeger, que ajudam você a enfrentar esses obstáculos.

# Considerações finais

A conteinerização é a metodologia escolhida por quase todas as equipes corporativas e de desenvolvimento modernas. Hoje, 91% dos gerentes de desenvolvimento e de desenvolvedores em nuvem implantam containers on-premise de alguma forma.<sup>7</sup> Essa é uma tecnologia que possibilita um trabalho mais inteligente e eficiente. Os containers fornecem um ambiente consistente para o desenvolvimento e a entrega rápida de aplicações nativas em nuvem executáveis em qualquer lugar.

A finalidade deste e-book é apresentar os princípios básicos dos containers e ajudar você a entender o potencial dessa tecnologia e o impacto que ela pode ter na sua empresa. Há muitos recursos disponíveis para aplicar os conhecimentos que você aprendeu neste documento aos seus projetos atuais. Em breve, você conseguirá separar suas aplicações da arquitetura, inspirar sua organização e impulsionar sua carreira com uma prática de desenvolvimento mais moderna e ágil: a conteinerização.

A melhor hora de começar é agora. Boa sorte na sua jornada para se tornar um desenvolvedor de aplicações nativas em nuvem.

## Saiba mais

Quer sair na frente? Aprenda como implantar aplicações em containers. Participe de um curso online gratuito da Red Hat e veja como os conceitos da conteinerização funcionam na prática. Descubra como colocar aplicações e serviços em containers, testá-los usando o Docker e implantá-los em um cluster do Kubernetes usando o Red Hat OpenShift Container Platform. Você também aprenderá a criar e implantar uma aplicação a partir do código-fonte usando o recurso source-to-image do OpenShift Container Platform.

[Inscreve-se ▶](#)

## Conheça mais recursos

[Udemy](#)  
[Stack Overflow](#)  
[Docker](#)  
[AWS](#)  
[Google](#)  
[Red Hat](#)

## Sobre a Red Hat

A Red Hat é a líder mundial no fornecimento de soluções de software open source corporativas, utilizando uma abordagem de parceria com as comunidades para oferecer tecnologias confiáveis e de alto desempenho de Linux, nuvem híbrida, container e Kubernetes. A Red Hat ajuda os clientes a integrar aplicações de TI novas e existentes, desenvolver aplicações nativas em nuvem e definir padrões com nosso sistema operacional líder do setor, além de automatizar, proteger e gerenciar ambientes complexos. Com serviços de consultoria, treinamento e suporte premiados, a Red Hat é a parceira de confiança das empresas listadas na Fortune 500. Como parceira estratégica de provedores de serviços em nuvem, integradores de sistemas, fornecedores de aplicações, clientes e comunidades open source, a Red Hat ajuda empresas a se prepararem para o futuro digital.

Copyright © 2019 Red Hat, Inc., Red Hat, Red Hat Enterprise Linux, Ansible, Ceph, CloudForms, Gluster, JBoss e OpenShift são marcas registradas da Red Hat, Inc. nos EUA e em outros países. Linux® é uma marca registrada da Linus Torvalds nos Estados Unidos e em outros países.

A marca nominativa e o logotipo OpenStack, em conjunto ou separados, são marcas comerciais ou registradas da OpenStack Foundation nos Estados Unidos e em outros países, usadas com a permissão da OpenStack Foundation. A Red Hat, Inc. não é afiliada, endossada ou patrocinada pela OpenStack Foundation ou pela comunidade OpenStack.

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou registradas da Oracle America, Inc. nos Estados Unidos e em outros países.

<sup>7</sup> Evans Data Corporation. "Cloud Development Survey 2017, Volume 2".