

# GUIA EXECUTIVO PARA SELECIONAR UMA PLATAFORMA DE DESENVOLVIMENTO NATIVA DA CLOUD

## INTRODUÇÃO

Quando qualquer organização começa a se planejar para adotar aplicações nativas da cloud, é importante considerar todo o tempo necessário desde a seleção de uma plataforma de desenvolvimento até que uma aplicação esteja realmente em nível de produção e pronta para entrega na cloud. A jornada pode ser longa, com muitas decisões ao longo do caminho que podem favorecer ou dificultar o progresso.

Por exemplo, no início de uma migração para o desenvolvimento nativo da cloud, é comum ocorrer ineficiências caso os desenvolvedores selecionem ferramentas e estruturas antes de saberem onde a aplicação será implantada. Enquanto os desenvolvedores corporativos desejam ter opções de ambientes de execução, estruturas e linguagens, as empresas precisam de padrões que atendam a todo o ciclo de vida da aplicação para reduzir custos operacionais, diminuir riscos e cumprir com os requisitos de conformidade. As empresas também não querem ficar dependentes de um único fornecedor de infraestrutura de cloud nem do estilo de arquitetura mais recente.

Além disso, a curva de aprendizado no desenvolvimento da cloud é acentuada. Por isso, não é recomendável pensar sobre a criação de aplicações robustas, escaláveis e resilientes somente no final do ciclo de desenvolvimento. Um dos principais motivos para migrar para a cloud é a capacidade elástica da escalabilidade dinâmica. Portanto, a resiliência não pode ser algo secundário.

Para uma estratégia eficaz de aplicação nativa da cloud, é necessário considerar todos os fatores: de ferramentas de desenvolvimento e plataforma de cloud a automação de implantação e operações. O benefício de uma abordagem holística para o desenvolvimento nativo da cloud é que ela dá uma direção a fim de evitar etapas desnecessárias.

## INTRODUÇÃO AO DESENVOLVIMENTO NATIVO DA CLOUD

O desenvolvimento nativo da cloud se resume a responder às mudanças com rapidez, resiliência e agilidade. O que possibilita essa resposta são implantações mais frequentes que reduzem substancialmente o tempo de provisionamento para reagir a mudanças.

Grande parte da discussão sobre aplicações nativas da cloud remete ao manifesto chamado “[A aplicação de doze fatores](#)”. Trata-se de um conjunto de princípios provenientes da experiência de criação e operação de aplicações de Software como Serviço (SaaS). As metas são:

- Reduzir a curva de aprendizado do desenvolvedor.
- Criar aplicações adequadas para implantação em plataformas de cloud.
- Maximizar a agilidade com implantação contínua.
- Capacitar aplicações para escalar verticalmente sem a necessidade de mudanças significativas.



facebook.com/redhatinc  
@redhat  
linkedin.com/company/red-hat

Ao trabalhar com os clientes, a Red Hat encontrou os principais elementos a seguir para desenvolver aplicações nativas da cloud:

- **Uma arquitetura baseada em serviços.** O estilo pode ser de microsserviços ou qualquer modelo de arquitetura modular, levemente acoplado. Os serviços representam uma atividade corporativa que é razoavelmente autônoma e prontamente compreensível. A meta é ter serviços que possam ser atualizados ou substituídos com facilidade. É mais fácil testar um serviço bem definido do que uma aplicação completa.
- **Containers.** Os containers Linux®, que usam imagens Docker, são um modelo de empacotamento comum e um ambiente de execução autônomo que fornece portabilidade, bem como isolamento. Os containers possibilitam a automação avançada que torna as plataformas de cloud atrativas.
- **Automação de DevOps.** Um conjunto de práticas e processos colaborativos, destinados a unificar o desenvolvimento e as operações. Os objetivos são: melhorar a frequência de implantação e fornecer lançamentos de qualidade mais alta. Isso resulta em um time to market mais rápido, menos risco e maior satisfação do usuário. A integração/entrega contínuas (CI/CD) estão intimamente ligadas às melhorias que o DevOps proporciona. Outra meta fundamental é a instrumentação e o monitoramento para entender o desempenho, assim como garantir uma experiência de usuário final de qualidade.
- **Comunicação baseada em API.** A comunicação entre processos somente ocorre por meio de interfaces de programação de aplicações (APIs) usando interfaces simples, baseadas em contrato, pela rede. Isso elimina o acoplamento não intencional que restringe alterações e é uma fonte comum de interrupções. A importância disso se torna mais clara em ambientes híbridos, nos quais as aplicações não estão mais localizadas no mesmo datacenter.

## INTRODUÇÃO AOS MICROSERVIÇOS

A arquitetura de microsserviços é interessante para a solução de muitos desafios de desenvolvimento de software. As aplicações são divididas em um conjunto de serviços levemente conectados que implementam funções corporativas específicas. O principal objetivo é acelerar a inovação, com base na ideia de que é muito mais fácil entender, aprimorar, testar e implantar um microsserviço individual do que uma aplicação monolítica. Como o propósito e a função de cada microsserviço devem ser bem definidos, o teste automatizado e a entrega contínua se tornam muito mais práticos de implementar.

## REQUISITOS PARA APLICAÇÕES NATIVAS DA CLOUD

Uma arquitetura nativa da cloud impõe vários requisitos técnicos às aplicações. Entre eles estão:

- **A implantação precisa ser automatizada.** Para agilizar a frequência de lançamentos, existe a necessidade de incorporar a automação de implantação. Dessa forma, o processo pode ocorrer com um simples clique em um botão para aprovação. É necessário ter um pipeline automatizado de criação e implantação, além da habilidade de reverter para uma versão anterior rapidamente, se for o caso.
- **As informações de configuração devem ser separadas da aplicação.** Os detalhes de configuração necessários para que uma aplicação seja executada dentro de um ambiente específico devem ser armazenados no ambiente de implantação. Isso permite que a mesma imagem de aplicação seja usada em todos os ambientes, incluindo desenvolvimento, teste e produção. Além disso, é necessário ter armazenamento seguro com credencial, pois todas as aplicações precisarão se comunicar, com segurança, com fontes de dados e outros componentes pela rede.

- **Os serviços de aplicação devem ser localizados dinamicamente.** Isso é necessário para ter portabilidade entre ambientes, alta disponibilidade e escalabilidade dinâmica com o uso de balanceamento de carga.
- **Armazenamentos de dados separados são necessários para dados persistentes.** Os processos e os containers devem ser sem monitoração de estado. Os processos podem falhar ou ser reiniciados, possivelmente em diferentes máquinas, em poucos segundos. Dados, informações de sessão e registros devem ser mantidos em armazenamentos de dados externos.

Veja algumas considerações ao iniciar o desenvolvimento nativo da cloud:

- Qual é o nível de experiência da sua empresa com a criação e a implantação de aplicações em containers a serem executadas em plataformas de cloud? Os desenvolvedores sabem o que é necessário para gerenciar dados persistentes e configuração separadamente das aplicações?
- Qual será o tempo necessário para criar um ambiente de desenvolvimento a fim de iniciar um desenvolvimento nativo da cloud? O ambiente de desenvolvimento se equipara ao ambiente da cloud de produção?
- Qual será o tempo necessário para criar e integrar, com eficácia, CI/CD aos processos de desenvolvimento e implantação?

Todos esses itens devem ser considerados ao selecionar uma plataforma de desenvolvimento nativa da cloud.

## CONSIDERAÇÕES SOBRE MICROSERVIÇOS

Os microsserviços e as aplicações nativas da cloud são intimamente relacionados. Na verdade, é difícil implantar microsserviços com êxito sem uma plataforma de cloud, containers e automação de DevOps. Um dos fatores principais para o uso de microsserviços é o ganho de agilidade ao usar componentes de aplicação menores, que podem ser lançados com mais frequência e menos riscos. No entanto, o que antes era uma única aplicação poderá se tornar uma dezena de microsserviços com ciclos de lançamento independentes. Veja algumas considerações sobre a migração para microsserviços:

- Qual será o tempo necessário para provisionar novos ambientes de desenvolvimento, teste e produção para cada novo microsserviço?
- Há automação de criação e implantação suficiente para oferecer suporte ao número de lançamentos que serão necessários?
- Como as aplicações descobrirão onde os microsserviços estão sendo executados? Há flexibilidade para realocar microsserviços conforme necessário para disponibilidade ou escalabilidade?
- Os microsserviços podem ser lançados sem incorrer em tempo de inatividade que impactará os usuários finais?
- Qual será o nível de dificuldade para diagnosticar problemas? Qual será o nível de dificuldade para rastrear eventos entre serviços diferentes e ambientes possivelmente híbridos?
- Os problemas com um serviço serão bem controlados ou eles disseminarão várias falhas?
- Considerando que os microsserviços são distribuídos, componentes de infraestrutura de software adicionais são necessários para executar funções, como descoberta de serviço, balanceamento de carga do lado do cliente, gerenciamento de estado persistente, rastreamento distribuído e resiliência.
- Os desenvolvedores precisarão implementar esses serviços ou localizar componentes fora da Internet?

- Qual será o peso da manutenção agora que esses componentes são parte de cada microsserviço implantado? Como os componentes serão atualizados para resolver bugs e vulnerabilidades de segurança?
- Esses são serviços comuns que deverão ser integrados aos ambientes de execução ou às plataformas de cloud?

Quando os desenvolvedores começam a criar microsserviços, uma das primeiras questões a ser resolvida é quais tipos de ambientes de execução, estruturas e linguagens são necessários. Veja algumas considerações para solucioná-la:

- Os desenvolvedores precisam estudar novas estruturas para começar?
- É possível usar o código Java™ EE existente e a experiência da empresa ao migrar para microsserviços?
- É possível usar novos estilos de desenvolvimento, como programação reativa? A plataforma oferece opções necessárias para atender a novas demandas, como cargas de trabalho orientadas por eventos criadas por ambientes mobile e a Internet das coisas (IoT)?
- Para usar a melhor ferramenta para a tarefa, é bom ter diferentes microsserviços implementados usando diferentes ambiente de execução, estruturas ou linguagens? Haverá diferenças na forma como eles são configurados, implantados ou protegidos?
- Qual será o tempo necessário para que novos microsserviços possam ser testados em um ambiente de cloud pública real?

Os microsserviços são muito interessantes. No entanto, pode ser difícil dominar os microsserviços e as arquiteturas de aplicação distribuída. É importante selecionar uma plataforma que possa remover a complexidade e simplificar o desenvolvimento.

### **MODERNIZAÇÃO DE APLICAÇÕES MONOLÍTICAS ATUAIS**

Nem sempre é prático ou econômico reescrever as aplicações monolíticas atuais na forma de microsserviços, embora isso seja interessante para desenvolvedores. No entanto, essas aplicações ainda requerem manutenção e apresentam uma lista de melhorias pendentes que precisam ser implementadas. Pode ser difícil reduzir essa lista em função dos longos ciclos de lançamento. Com a re-hospedagem dessas aplicações em uma plataforma de cloud, é mais fácil implementar CI/CD junto com os lançamentos, tal como implantações “azul-verde” ou canário. Isso permite que os desenvolvedores forneçam lançamentos com mais frequência e menos riscos. A aplicação hospedada na cloud pode se tornar uma “aplicação monolítica ágil”. Assim, os desenvolvedores podem trabalhar nas melhorias pendentes.

Para implementar microsserviços de maneira eficaz, uma boa opção também é re-hospedar uma aplicação na cloud. A flexibilidade da cloud facilita a implantação de novos microsserviços em containers em conjunto com a aplicação existente. Dessa forma, os desenvolvedores podem implementar microsserviços que migram a funcionalidade para fora da aplicação monolítica.

Levando em conta os benefícios da migração “lift-and-shift” das aplicações existentes para a cloud, veja aqui algumas considerações a serem feitas ao escolher uma plataforma de desenvolvimento nativa da cloud:

- A plataforma de desenvolvimento tem opções para modernizar as aplicações existentes assim como desenvolvimentos inteiramente novos?
- A plataforma de cloud pode oferecer suporte a aplicações que ainda não são totalmente nativas da cloud?

- Quais opções existem para migrar aplicações Java EE atuais?
- A plataforma de desenvolvimento oferece suporte à agilização da entrega de aplicações de estilo monolítico?

Os servidores de aplicação podem ser um ambiente de execução viável para aplicações nativas da cloud. No entanto, a função desse servidor tradicional muda com a migração para uma plataforma nativa da cloud. Tradicionalmente, os servidores de aplicação forneciam um ambiente de execução e implantação, assim como serviços de gerenciamento para um domínio central executado em um cluster de sistemas. No entanto, muitas das ferramentas operacionais incluídas, como o console de administração, não são mais necessárias e são contraproducentes. As plataformas de cloud são interessantes, pois incluem recursos de gerenciamento automatizados que possibilitam a escalabilidade dinâmica e a entrega contínua. Se um servidor de aplicação tradicional for empacotado com seus componentes de gerenciamento em um container, os desenvolvedores talvez não possam aproveitar esses recursos. Alguns pontos a serem considerados:

- A plataforma inclui uma opção para um ambiente de execução Java EE que é integrado aos recursos de gerenciamento da plataforma de cloud?
- Há ferramentas para ajudar a migrar aplicações que estão em execução em um servidor de aplicação legado para um ambiente de Java EE moderno, baseado na cloud?
- Para aplicações que não usam todos os recursos do Java EE, como aplicações web, há alternativas que possam ser uma opção melhor para a cloud?

Com a plataforma e as ferramentas certas, as empresas podem obter mais valor das aplicações atuais e, ao mesmo tempo, migrar para um modelo nativo da cloud e de microsserviços.

### **SUPOORTE A CLOUDS ON-PREMISE E VÁRIAS CLOUDS**

A maioria das organizações de TI prefere não se limitar a um único fornecedor de infraestrutura de cloud pública. Além disso, muitas empresas acreditam que, mesmo daqui a muitos anos, pelo menos metade das suas aplicações ainda será executada no local. Como a maioria das organizações tem esses dois requisitos, considere o seguinte:

- As aplicações precisarão mudar para usar a infraestrutura de cloud de diferentes fornecedores? E quanto a implantação, operações e monitoramento? Eles serão diferentes em cada plataforma de cloud?
- Como minimizar as diferenças entre a infraestrutura de cloud pública e os sistemas no local? A plataforma de cloud usada em clouds públicas também está disponível para execução no local ou na Infraestrutura como Serviço (IaaS) escolhida pela empresa?

### **O PANORAMA GERAL**

Ao avaliar uma plataforma de desenvolvimento nativa da cloud, também vale a pena considerar como a escolha impactará outras áreas:

- Há ferramentas de desenvolvimento prontas para funcionar com o desenvolvimento da cloud e a plataforma de implantação? Se não houver, quanto tempo será usado na configuração e integração dessas ferramentas?
- A plataforma fornece aos desenvolvedores uma abordagem prescritiva ou orientada para aumentar a produtividade?
- Há um conjunto de serviços de middleware baseados na cloud para sistema de mensageria, armazenamento de dados e processo de negócios ou gerenciamento de regras disponível e pronto para ser executado na plataforma da cloud?

- Há containers de terceiros pré-criados disponíveis para integração com as aplicações?
- Há serviços de treinamento e consultoria disponíveis? Os recursos internos disponíveis têm experiência com modernização de aplicações?

### **A IMPORTÂNCIA DE SELECIONAR A PLATAFORMA DE DESENVOLVIMENTO DE CLOUD CERTA**

Ao começar a desenvolver uma cloud, é comum tentar resolver um problema de cada vez seguindo tutoriais e selecionando componentes de software na Internet. Primeiro, os desenvolvedores precisam aprender como montar um ambiente de execução suficiente para compilar o código da aplicação em um container. Depois, eles precisam decidir quais outros componentes de software são necessários para criar aplicações completas. Esse processo pode ser demorado e apresentar uma série de desvantagens:

- A curva de aprendizado e a quantidade de integrações necessárias podem diminuir a produtividade do desenvolvedor, o que é difícil de explicar para a empresa. Qualquer outro desenvolvedor recém-contratado precisará entender o stack que foi criado internamente.
- Qualquer componente de software selecionado exigirá manutenção para garantir que continue sem bugs e vulnerabilidades de segurança. Não há certeza se os componentes open source são adequados para o uso corporativo.
- Muitas organizações relatam que os desenvolvedores podem escrever códigos para funções, como configuração e implantação, que não agregam valor direto aos negócios. Ou, pior ainda, muitas dessas funções são atreladas à plataforma de implantação escolhida e são serviços comuns que já deveriam fazer parte dela.

Talvez o maior problema na montagem de uma coleção de componentes seja que isso não reduz a curva de aprendizado geral para desenvolvimento e operações. A alternativa é selecionar uma plataforma de desenvolvimento de cloud que seja projetada para lidar com todas as fases: da seleção de um ambiente de execução para iniciar o desenvolvimento até a implantação de produção.

A migração para aplicações nativas da cloud não ocorre de um dia para outro. Para muitas organizações, o processo melhora com a experiência. Considerando tudo o que o desenvolvimento nativo da cloud envolve, é evidente que uma abordagem completa, do desenvolvimento à implantação, provavelmente será mais bem-sucedida. A Red Hat oferece essa plataforma com o Red Hat® OpenShift Application Runtimes e o Red Hat OpenShift Container Platform.

### **RED HAT OPENSIFT APPLICATION RUNTIMES E RED HAT OPENSIFT CONTAINER PLATFORM**

O Red Hat OpenShift Application Runtimes foi projetado para simplificar o desenvolvimento da aplicação nativa da cloud. Um conjunto selecionado de estruturas e ambientes de execução integrados fornece uma direção com uma experiência prescritiva que pode ser usada para dar início ao desenvolvimento. Viabilizamos uma plataforma de desenvolvimento e implantação completamente otimizada, porque o OpenShift Application Runtimes é baseado no Red Hat OpenShift Container Platform e também otimizado para ele, uma plataforma de aplicações em container projetada para clouds híbridas.

O Red Hat OpenShift Container Platform fornece uma plataforma de autosserviço para que os desenvolvedores e a equipe de operações criem e executem aplicações em container. Com o OpenShift, é possível provisionar um ambiente para um novo microsserviço ou uma nova aplicação em questão de minutos. O OpenShift possibilita ciclos de implantação substancialmente reduzidos, com risco muito menor. Pipelines de criação e implantação

de CI/CD avançados e automatizados estão disponíveis com alguns cliques. Juntos, o Red Hat OpenShift Application Runtimes e o Red Hat OpenShift Container Platform, com ferramentas de desenvolvimento da Red Hat e a Red Hat Consulting, podem ajudar as empresas a migrar para aplicações nativas da cloud em menos tempo e com menos riscos.

Os ambientes de execução do OpenShift Application Runtimes são selecionados para que os desenvolvedores tenham a ferramenta certa para essa tarefa. Para o desenvolvimento de microsserviços, os desenvolvedores podem escolher o Java EE para aproveitar a experiência existente, o novo padrão de microperfil Java que evoluiu para atender às necessidades de microsserviços ou uma estrutura orientada por evento para criar microsserviços reativos, capazes de escalar para cargas de trabalho de alta simultaneidade e baixa latência. Um ambiente de execução Node.js é fornecido para serviços de back-end do JavaScript que são usados frequentemente com aplicações mobile e web. Para migrar aplicações existentes, é fornecido um ambiente de execução baseado no Red Hat JBoss® Enterprise Application Platform, um servidor de aplicação com certificação Java EE 7 baseado em uma arquitetura moderna, modular e pronta para a cloud. A Red Hat verifica e testa todos os ambientes de execução.

Para que os desenvolvedores possam começar rapidamente, disponibilizamos no site Red Hat Developers uma ferramenta SaaS gratuita, baseada na cloud, executada no Red Hat OpenShift Online, chamada Launch. Após escolher uma aplicação e um ambiente de execução de amostra, os desenvolvedores obtêm uma base de código completa que está pronta para ser compilada e executada na cloud no OpenShift Online. Como é uma ferramenta educacional, as mesmas aplicações de amostra estão disponíveis para todos os ambientes de execução. Isso permite que os desenvolvedores comparem os benefícios dos estilos de arquitetura disponíveis com facilidade.

Para saber mais, consulte: <https://www.redhat.com/pt-br/technologies/cloud-computing/openshift/application-runtimes>



## **SOBRE A RED HAT**

A Red Hat é a líder mundial no fornecimento de soluções de software open source, utilizando uma abordagem de parceria com as comunidades para oferecer tecnologias confiáveis e de alto desempenho de cloud, Linux, middleware, armazenamento e virtualização. A Red Hat conta com premiados serviços de suporte, treinamento e consultoria. Como um hub de conectividade em uma rede global de empresas, parceiros e comunidades open source, a Red Hat ajuda a criar tecnologias relevantes e inovadoras que permitem a ampliação recursos disponíveis e preparam os clientes para o futuro da TI.

Saiba mais em <http://www.redhat.com/pt-br>.



facebook.com/redhatinc  
@redhat

linkedin.com/company/red-hat

### **AMÉRICA LATINA**

+54 11 4329 7300

latammktg@redhat.com

### **BRASIL**

+55 11 3629 6000

marketing-br@redhat.com