

Arquitetura Servless versus Arquitetura de Microservices

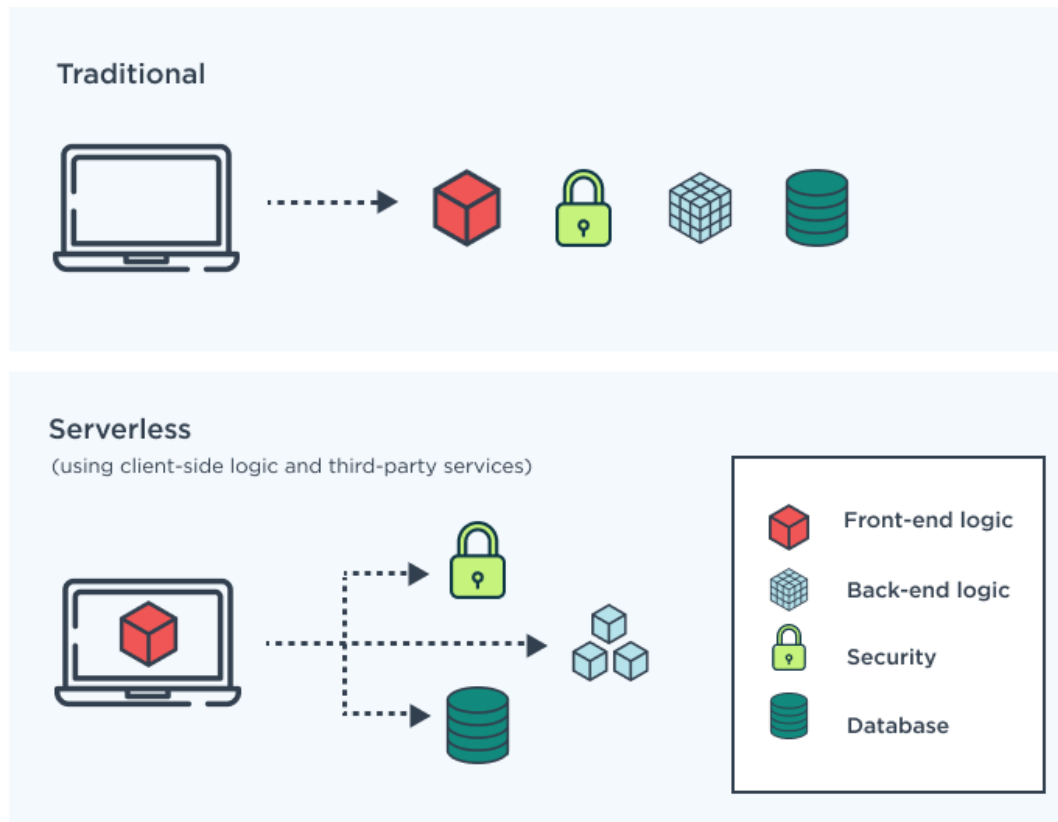
Visão geral

A computação sem servidor, comumente referida como apenas sem servidor, é um modelo de tecnologia baseado em nuvem promissor que surgiu no horizonte de desenvolvimento de aplicativos e arquitetura de software nos últimos anos. Tentando aproveitar o enorme potencial de frameworks sem servidor, muitos grandes participantes do mercado rapidamente aderiram [ao movimento dos serviços em nuvem](#). Gigantes de software como Google, Microsoft, IBM e Amazon já oferecem aos clientes a migração de todas as eficiências operacionais de negócios locais para serem hospedadas em suas plataformas sem servidor carro-chefe como [AWS Lambda](#) e [Azure Functions](#).

Simplificando, a arquitetura sem servidor é uma solução de tecnologia orientada a eventos e solicitações que permite aos desenvolvedores de aplicativos criar ambientes de trabalho acionáveis na nuvem que tenham todos os recursos computacionais necessários para um fluxo de codificação tranquilo. Essa estrutura é útil, especialmente quando o tempo é um problema e as tarefas atribuídas exigem muitos recursos.

Além disso, enquanto opta por [serviços sem servidor](#) para agilizar seus processos de desenvolvimento de aplicativos, você pode melhorar drasticamente suas outras iniciativas de otimização de negócios ao longo do caminho - como DevOps e práticas Ágile.

Traditional vs Serverless Architecture



Arquitetura tradicional vs sem servidor

Modelo Sem Servidor e de Microserviços: qual é a diferença

Por um lado, essas duas arquiteturas possuem propriedades funcionais semelhantes - ou seja, podem minimizar os custos operacionais, bem como o ciclo de implantação do aplicativo, lidar com os requisitos de desenvolvimento em constante alteração e otimizar as tarefas diárias sensíveis ao tempo e aos recursos. Por outro lado, existem algumas diferenças marcantes entre microserviços e modelos sem servidor que você definitivamente deve conhecer.

- **Microserviços.** Esta solução habilitada para tecnologia é um tipo menor de SOA (arquitetura orientada a serviços) que fornece aos programadores acesso total às bibliotecas relevantes hospedadas em servidores em nuvem e permite que eles implantem vários módulos funcionais junto com vários protocolos e API, como JSON, RESTful, AMQP, SOAP e mais.

Como regra, a arquitetura de microserviço permite a containerização de dados por meio de pacotes executáveis de software como o Docker, com todos os dados necessários sendo descompactados sob demanda conforme os desenvolvedores acessam uma determinada API. Numerosos padrões de microserviços garantem a implantação contínua de aplicativos por meio de operações executadas na plataforma de nuvem.

Além disso, você deve se lembrar que utilizar a arquitetura de microserviço pode ser bastante caro para os proprietários de negócios, uma vez que mesmo microserviços ociosos estão, de fato, ativos - portanto, você terá que pagar pelas capacidades dos servidores host diariamente / mensalmente.

- **Sem servidor.** A arquitetura sem servidor é muito mais promissora para desenvolvedores de aplicativos, pois fornece a execução de ambientes de trabalho baseados em nuvem sob demanda. Isso significa que as funções sem servidor são acionadas apenas quando um determinado evento está sendo definido. Feito isso, as funções executam uma sequência de operações dependendo dos comandos que os usuários inseriram. Em seguida, a plataforma sem servidor aplica um conjunto de algoritmos e regras pré-condicionados para processar cálculos e gerar resultados acionáveis.

Depois que todas as operações são concluídas, o ambiente no qual você está trabalhando se dobra imediatamente até que a próxima tarefa seja solicitada, o que aparentemente nos dá o direito de chamar a arquitetura sem servidor de um ambiente de desenvolvimento de aplicativo sob demanda real.

Em suma, os microsserviços são muito melhores para serem empregados em casos de uso como sites de comércio eletrônico, onde a containerização de vários conjuntos de dados em grande escala é uma prioridade. Serverless mostra seu grande potencial quando se trata de otimizar custos e tempo de desenvolvimento.

Além disso, é mais preferível optar por uma arquitetura sem servidor se seu fluxo de negócios for baseado em eventos e você não precisar executar suas tarefas operacionais o tempo todo. Isso é verdadeiro especialmente para aplicativos [alimentados](#) por [IoT](#), onde alarmes e notificações urgentes geralmente ocorrem. Para obter uma imagem mais clara, vamos examinar mais de perto alguns recursos e características benéficas sem servidor.

Como funciona a Arquitetura Servless

Sem servidor como um modelo de desenvolvimento conduzido por eventos pré-configurados, utiliza uma série de serviços baseados em nuvem distribuídos para agilizar o fluxo de trabalho de implantação de software. Normalmente, ele incorpora dois modelos de serviço complementares:

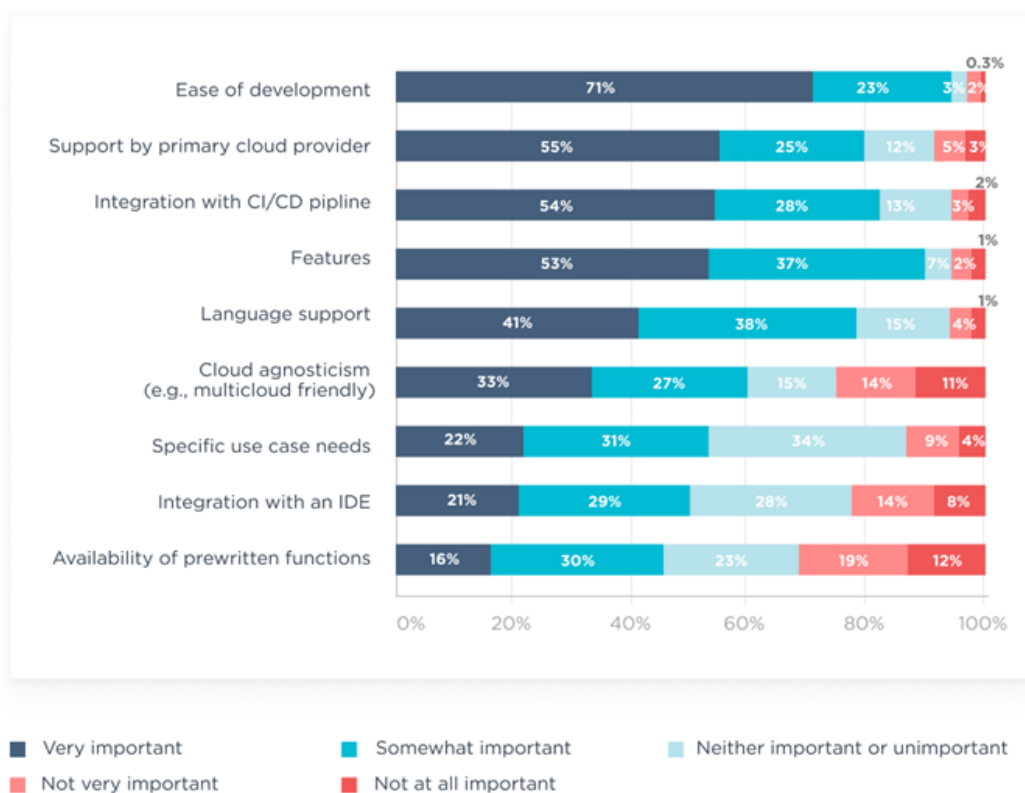
- Back-end como serviço (BaaS)
- Função como serviço (FaaS)

Vamos nos aprofundar em alguns de seus recursos peculiares. Portanto, o BaaS tem tudo a ver com o aproveitamento dos recursos de arquitetura multilocatário de terceiros para aprimorar a implementação de seu próprio aplicativo e as eficiências de entrega. Seus recursos comuns são geralmente fornecidos como SDKs ou gateways de API e todas as ações necessárias são executadas apenas na nuvem e geralmente são chamadas de partes de sua infraestrutura local legada ou dados de código. Essa solução desfruta de um ciclo de desenvolvimento realmente rápido e transfere a responsabilidade total pela manutenção de software e infraestrutura para os ombros do provedor de BaaS. Além do mais, suas capacidades são virtualmente infinitas, o que permite que os desenvolvedores de aplicativos aumentem os fluxos de implantação em trânsito.

Plataformas FaaS como AWS Lambda, por exemplo, são tipos de arquitetura de desenvolvimento orientada a eventos que garantem serviços de hospedagem robustos enquanto usam um modelo de pagamento por uso e são acionados somente quando algumas funções ou eventos predeterminados pelos usuários estão programados para ação. Os gatilhos comuns para ativar funções FaaS são eventos pré-agendados, uploads de arquivos, novos commits em um repositório, e-mails, atualizações de banco de dados, chamadas HTTP API e outros.

Os principais critérios aos quais os programadores estão prestando atenção ao escolher uma solução FaaS são a simplicidade de desenvolvimento e suporte do provedor de nuvem. Esses e outros fatores importantes são apresentados nos gráficos a seguir.

Selection Criteria When Choosing a FaaS Offering

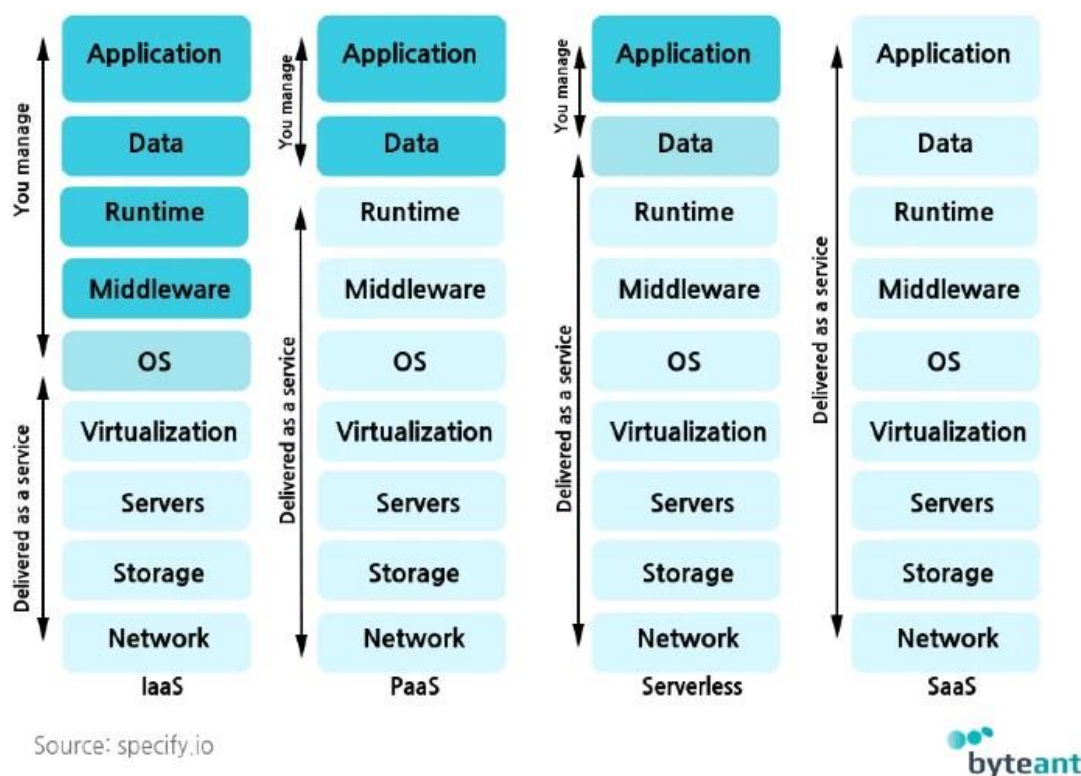


Source: The New Stack Serverless Survey 2018



Ao trabalhar em duo, os modelos BaaS e FaaS podem ajudar os desenvolvedores a lidar com tarefas complexas de uso intensivo de recursos que envolvem hospedagem de seus dados, infraestrutura e elementos lógicos de negócios. Dito isso, sem servidor compreende essas duas abordagens, oferecendo aos usuários um amplo escopo de benefícios funcionais. O esquema abaixo é para esclarecer a diferença entre sem servidor e outras soluções populares de serviço baseadas em nuvem

IaaS vs PaaS vs Serverless vs SaaS: who manages what?



Sem servidor: casos de uso principais

Depois que a arquitetura sem servidor assumiu o comando dos microsserviços, DevOps e outras tendências de tecnologia, ela está ganhando ampla adoção em empresas de pequeno e médio porte e empresas de grande porte. Hoje, as empresas optam por soluções sem servidor que se esforçam para capitalizar em tempos de lançamento de produtos mais rápidos, reduzir custos operacionais e aumentar o desempenho geral. Com todo o seu enorme potencial a ser descoberto, já existem algumas áreas de utilização sem servidor onde seu significado prático para os negócios é indiscutível.

1. Escalabilidade automática de sites. Depois de mudar para uma arquitetura sem servidor, o fornecimento de infraestrutura não será mais seu problema. Os desenvolvedores podem escrever

facilmente seu código e criar e implantar aplicativos e sites acionáveis por períodos menores. Além disso, esta solução baseada em nuvem permite que os usuários tenham escalabilidade totalmente automatizada se recursos de computação adicionais forem necessários.

2. **Análise de big data.** Sem servidor pode ser extremamente eficaz quando se trata de orquestrar conjuntos enormes e díspares de ativos de dados analíticos que costumavam ser espalhados em servidores locais e não eram unificados entre as equipes de BackOffice e FrontOffice. Em vez disso, um aplicativo separado pode ser escrito para reunir e processar conjuntos de dados isolados em todos os canais de negócios. Esse aplicativo sem servidor fornece coleta, classificação e análise de big data em um único banco de dados.
3. **Conectividade aprimorada entre dispositivos IoT.** A integração de dispositivos baseados em IoT, como vários sensores, tags RDID, smartphones e outros gadgets, tornou-se um pilar para a maioria das empresas. É aqui que as funções sem servidor entram em jogo, ajudando os usuários finais a evitar problemas frustrantes com baixo tráfego de Internet em algumas áreas problemáticas. Além disso, as capacidades de escalonamento automático tornam possível economizar nos gastos operacionais, reduzir a latência e, como resultado, melhorar muito a experiência do usuário.

Benefícios de usar sem servidor

Main Benefits Of Serverless For Business Owners



A eficiência da EDA (arquitetura orientada a eventos) que está por trás da computação sem servidor é bastante óbvia para os profissionais, uma vez que possui as seguintes vantagens competitivas:

- **Auto-escalável por design.** A solução se orgulha de sua grande flexibilidade, permitindo dimensionar seus recursos e capacidades computacionais sempre que houver uma necessidade específica. E o contrário - em relação a cargas de processamento menores, o ambiente de trabalho diminui quando o horário de pico termina.

- **Implementação aprimorada.** Com recursos sem servidor, torna-se muito fácil e mais confortável para os programadores desenvolver e implantar seus aplicativos. Eles não precisam mais pensar em atividades secundárias, como manutenção de infraestrutura ou sincronização oportuna de dados. A natureza flexível da arquitetura sem servidor permite uma automação suave da maioria dos processos de trabalho.
- **Estabilidade.** A arquitetura sem servidor passa a maior parte das responsabilidades de gerenciamento de dados e infraestrutura para o provedor de serviços em nuvem. Ele libera uma grande parte do tempo de negócios de sua equipe, permitindo que não se preocupe em facilitar a infraestrutura, orquestrar e alocar dados de código e lidar com problemas de tempo de inatividade como você costumava fazer em sua plataforma de software local legada.
- **Custo-efetividade.** A minimização de custos é o que realmente faz o servidor sem servidor se destacar da multidão de outras soluções baseadas em nuvem, pois sua abordagem de pagamento por uso ajuda a evitar despesas desnecessárias de desenvolvimento e implantação de aplicativos. Você paga apenas pelos períodos de execução do servidor quando alguns eventos programados dão origem às respectivas ações na nuvem.
- **Corte de latência.** Pelo fato de todos os ativos de dados de uma empresa estarem hospedados em servidores baseados em nuvem, permite que uma organização aloque seus recursos no servidor mais próximo disponível para usuários finais. Isso garante conectividade e interoperabilidade de dados sem problemas, sem falhas críticas de tempo e tempo de inatividade do sistema.