Serverless: Desacoplando os containers

Matheus de Oliveira Boni | Anderson Nascimento (orientador)

Bacharel em Sistemas de Informação – Universidade do Grande Rio (Unigranrio)
Duque de Caxias – Rio de Janeiro – RJ – Brasil

matheusboni@unigranrio.br | anderson.nascimento@unigranrio.edu.br

Resumo. *Atualmente na área de computação em nuvem muito se fala sobre containers Docker em arquiteturas de microsserviços para um ambiente altamente escalável, gerenciado e de prática comunicação entre os diversos componentes. Apesar de ser uma arquitetura amplamente utilizada pela facilidade que traz ao gerenciar as instâncias e propriedades das máquinas em que estão hospedadas as aplicações e suas dependências ainda é necessário um certo trabalho de implementação e suporte da infraestrutura. Portanto existe uma forma ainda mais prática de subir uma aplicação na nuvem sem configurar nenhum tipo de máquina e propriedades computacionais, neste artigo será abordada a arquitetura Serverless que possibilita rodar códigos "sem servidor", podendo focar somente no desenvolvimento da sua aplicação e passar a responsabilidade de gerenciamento dos servidores para o provedor de serviço de nuvem.*

Abstract. *Currently in cloud computing area, one of the most commented subjects is about Docker containers in microservices architectures for a highly scalable, managed and practical communication between the various components that contemplate the environment. Despite being an architecture widely used because of the ease it brings in managing instances and properties of the machines which the applications and their dependencies are hosted, some implementation works and infrastructure support is still needed. So there an even more practical way to deploy an application to the cloud without configure any kind of computational properties, this paper presents the Serverless architecture that makes possible runs code "without server", being able to focus only on the application development and the responsibility of managing servers to the cloud service provider.*

1. Introdução

Também conhecida como Função como Serviço (FaaS) a Serverless Computing possui muitos diferenciais comparada as arquiteturas mais tradicionais, dentre as que merecem mais destaque é a forma em que a aplicação é executada através de eventos que a ativam ficando inativa enquanto não ocorre nenhum tipo de chamada, ao contrário dos casos mais comuns de aplicações, que se mantém rodando mesmo sem nenhuma interação. Outra curiosidade já citada e que se percebe pelo nome é a não utilização de servidores, apesar dessa definição ela não anula a utilização de servidores apenas tiram dos desenvolvedores e analistas de infraestrutura a responsabilidade de provisionar e gerenciar os servidores na qual as aplicações são executadas, sendo auto escalável, ou seja, os recursos são disponibilizados automaticamente de acordo com o número de requisições que são recebidas. Essas informações são apresentadas e explicadas pelos próprios provedores em seus sites onde falam da visão geral dos produtos como, por exemplo, no portal da AWS.

2. Principais plataformas, vantagens e desvantagens.

Por ter a característica de se manter inativa até que um evento invoque a sua execução as funções serverless podem se tornar uma boa alternativa para redução de custos uma vez que você passará a pagar por evento e não mais por tempo de execução da máquina, ou seja, irá pagar exatamente pela quantidade que foi utilizada durante um determinado período de tempo. Segundo FERNANDES (2019) é recomendado que estas funções sejam de rápida execução afim de aproveitar essa possibilidade de menores cobranças, por isso não se deve utilizar qualquer aplicação dentro desse modelo e sim os casos em que são realmente vantajosos, o mesmo afirma que geralmente essa arquitetura é utilizada para códigos que não precisam ser executados imediatamente e que podem causar processamento desnecessário na aplicação principal, outros pontos para se levar em consideração são debugging e testes que podem ser um pouco complexos já que dependem de outros eventos.

As principais plataformas que disponibilizam esse serviço são: AWS (AWS lambda), Google Cloud Platform (Cloud functions), Azure (Azure functions) suportando linguagens como: Java, Python, JavaScript (NodeJS) e C#. Porém algumas podem ser mais simples que outras por consequência sendo mais utilizadas nesse tipo de arquitetura.

3. Utilização e Perspectivas

Segundo CANALTECH (2018) em 2016 o termo foi bastante falado e muitas vezes chegaram a cotar o ano de 2017 como o ano do Serverless Computing, mas acabou não acontecendo, o aumento na utilização da tecnologia em questão vem crescendo e atraindo cada vez mais interesses ano após ano e pode chegar ao seu ápice nos próximos 5 a 10 anos, como afirma o próprio líder de arquitetura de soluções da AWS para a América Latina, Eduardo Horai. No Brasil já existem empresas que apostam na tecnologia como a Nubank e SemParar enquanto mundialmente a Netflix por exemplo, também se desfruta da ideia.

4. Arquitetura do processo proposto

A arquitetura desenvolvida para prova de conceito está ilustrada na Figura 1. A ideia é demonstrar na prática a diferença entre as duas arquiteturas: A primeira baseada em Serverless Computing e a segunda em Docker, ambas utilizando recursos da AWS.

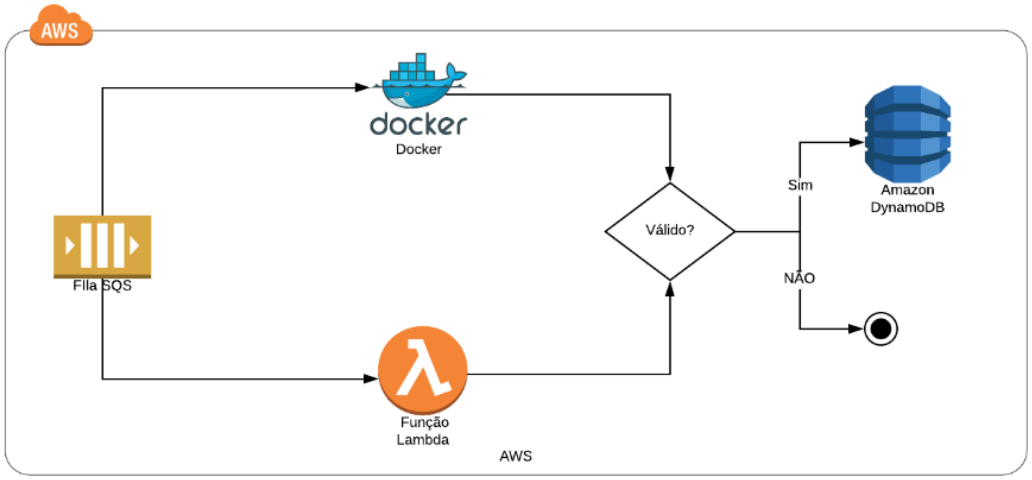

Figura 1 – Arquitetura
Fonte: Do autor

- **Fila SQS**: Fila gerenciada pela AWS utilizada para processamentos assíncronos, com mensagens contendo informações no formato JSON que serão consumidas pelas aplicações.

- **Docker**: Uma API em Java rodando em um container Docker que irá consumir as mensagens da fila e salvar no banco de dados. O código do projeto está disponível no endereço:

  - https://github.com/matheusboni/sqs-to-dynamodb-api

- **Função Lambda**: Aplicação Serverless em Python utilizando AWS Lambda que irá consumir as mensagens da fila e salvar no banco de dados. O código do projeto está disponível no endereço:

  - https://github.com/matheusboni/sqs-to-dynamodb-aws-lambda

- **Amazon DynamoDB**: Banco de dados NoSQL gerenciado pela AWS.

5. Prova de conceito

Utilizando a arquitetura apresentada será demonstrado as principais diferenças entre um microsserviço com Docker e uma função Lambda, cada aplicação executará o mesmo processo: Receber um evento da fila SQS, validar e salvar no banco de dados, porém de forma independente e em momentos distintos. Assim conseguindo identificar o comportamento de cada uma, suas características e melhores formas de utilização das mesmas.

6. Conclusão

A arquitetura de computação “sem servidores” é realmente muito poderosa, podendo receber e processar diversos eventos em tempo real, desde mensagens de uma fila, streams de dados e até mesmo uma requisição http. Mas ela não veio para substituir os microsserviços e sim para ajudá-los servindo como um “desafogo” para algumas soluções, segmentando melhor as responsabilidades, diminuindo gargalos e custos, basta saber usá-la da forma correta. Há um imenso mundo para se aprofundar em relação a computação em nuvem envolvendo Serverless, mas para futuras pesquisas a demonstração de outras funcionalidades a partir de novos eventos seria um ótimo caminho a seguir, sempre focando em desempenho e rentabilidade.

7. Referências Bibliográficas

AWS Lambda. Amazon Web Services. Disponível em: <https://aws.amazon.com/pt/lambda/>. Acesso em: 19 de abr. de 2020.

FERNANDES, Diego. Serverless: Quando utilizar e aplicações com NodeJS. Rocketseat, 2018. Disponível em: <https://blog.rocketseat.com.br/serverless-nodejs-lambda/>. Acesso em: 16 de mai. De 2020.

KOHN, Stephanie. O que é e para quem é indicado a Serverless Computing. Canaltech, 08 de ago. de 2018. Disponível em: <https://canaltech.com.br/infra/o-que-e-e-para-quem-e-indicado-a-serverless-computing-119782/>. Acesso em: 19 de abr. de 2020.