



Flutter
Talks 2019

Serverless

Function as a Service

Sobre o Palestrante

- André Luiz de Oliveira, 33 anos
- Analista de Sistemas no Grupo Tecnospeed
- Professor na Universidade Cidade Verde
- Formado em Análise de Desenvolvimento de Sistemas - UniFCV
- Pós Graduado em WebMobile - UniFCV
- Pós Graduando em Data Science - UniFCV
- Atualmente trabalho com Java, AngularJS, AWS, Devops, Delphi
- <https://github.com/FactorySolution>.

O que é Serverless

Arquitetura Serverless, ou “computação sem servidores”, é uma arquitetura de computação **orientada a eventos**.

Sua principal proposta é permitir que as empresas de software criem e mantenham seus aplicativos web sem se preocupar com a infraestrutura em que esses aplicativos estão rodando.

Antes da Arquitetura Serverless

- Para compreender o conceito da Arquitetura Serverless, é necessário compreender como surgiu a demanda por ela. Para isso, vamos analisar os paradigmas que precederam a Arquitetura Serverless.

Infraestrutura In-house

- No início das aplicações web, era comum utilizar um hardware contendo um sistema operacional e a aplicação. Alguns chamam essa abordagem de “**infraestrutura in-house**“. Em muitos casos, esse hardware era hospedado em um data center, e a software house pagava pelo espaço e infra que utilizava.

Computação Cloud Tradicional

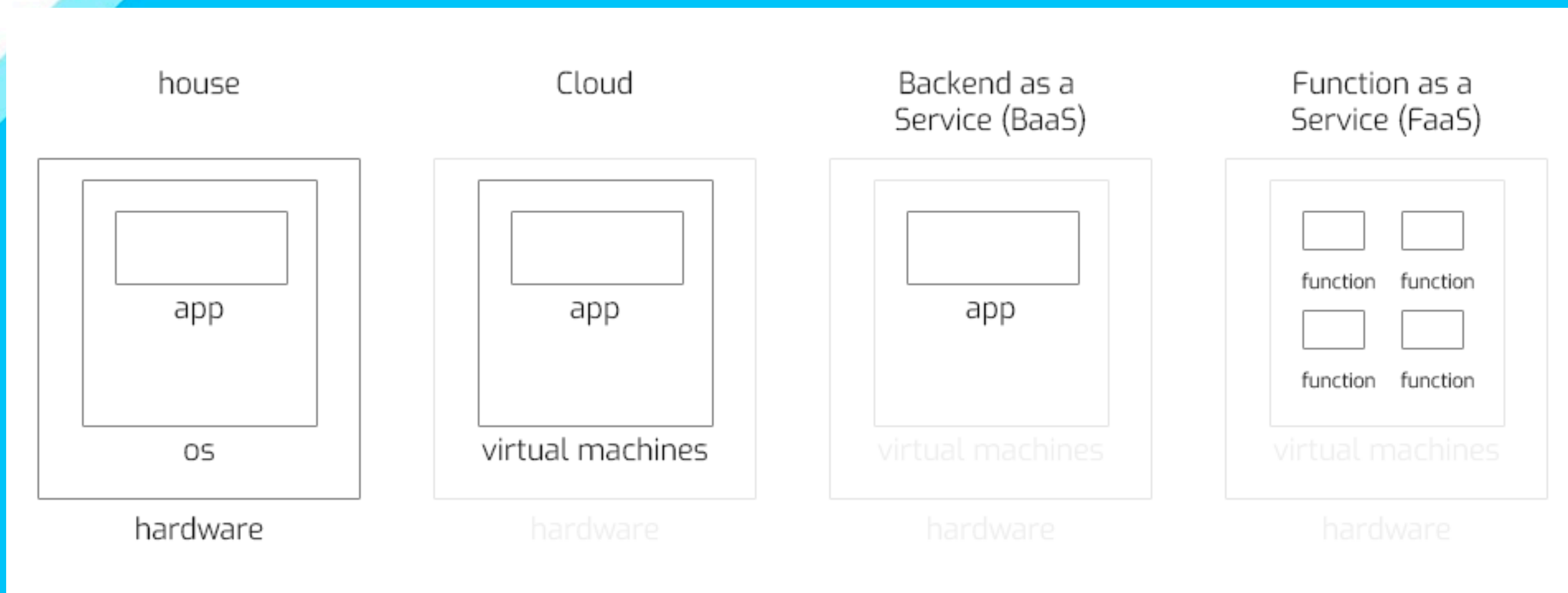
- Com o ganho de performance em virtualização de máquinas, começamos a trabalhar com a **computação cloud**. Na computação cloud tradicional, nossas aplicações ficam em máquinas virtuais onde era possível aumentar (upgrade) ou diminuir (downgrade) o poder de processamento.
- Atualmente, a maiorias das aplicações online estão em máquinas virtuais.

Backend as a Service (BaaS)

- Backend as a Service, ou apenas BaaS, caracteriza-se por um pacote de soluções hospedadas na nuvem, com um propósito final que dispõe APIs ou SDKs para estes serviços serem utilizados por outras aplicações.
- O Backend as a Service é um middleware para a sua aplicação. Ele fornece APIs para suas aplicações, web e mobile, prontas para serem integradas online

Function as a Service (FaaS)

- Function as a Service (FaaS) é um serviço de computação em nuvem que busca abstrair a questão de servidor e máquinas onde rodam os nossos programas.
- Ela deriva da filosofia Servless, que delega ao provedor de computação em nuvem (Cloud Provider) provisionar uma plataforma que permita ao cliente: desenvolver, rodar e gerenciar aplicações.
- Nestes moldes, reduz-se a complexidade de uma pessoa ter que montar e manter a infraestrutura, atual padrão de todo mercado corporativo. Montando uma aplicação com este padrão, implementamos a teoria Servless e criamos uma arquitetura nativa de micros serviços



Quais as vantagens da Arquitetura Serverless?

- Paga somente pelo que usa.
- Cada função pode ser criada em uma linguagem diferente.
- Muitos eventos pré-configurados na Cloud ajudam a criar arquiteturas orientadas a eventos.
- Não existe mais servidor.
- Auto-escalável por natureza.

Quais as desvantagens da Arquitetura Serverless?

- Tempo de execução.
- Vendor Lock-In.
- Difícil de debugar.
- Dependência de fornecedor.
- É necessário configuração extra para controlar (parcialmente) o ambiente de execução (Lambda layers).

Suporte para diversos mundos

- AWS
- Google Cloud
- Azure
- IBM Cloud
- Kubernetes
- NodeJS
- .Net Core
- Python
- Ruby
- GoLang
- Java



- Um framework web e open-source, escrito em Node.js. Quando foi desenvolvido, o Serverless Framework era destinado exclusivamente para a criação de aplicações para o AWS Lambda, a plataforma da Amazon. Posteriormente, foi compatibilizado com outros fornecedores de cloud, como a Microsoft Azure, IBM BlueMix, Google Cloud, Oracle Cloud, entre outros.
- Utilizando o Serverless Framework, **seu código deve respeitar a interface da fornecedora cloud** que você optou por utilizar. Sendo assim, é muito mais difícil adaptar um software criado para rodar em cloud tradicional utilizando o Serverless Framework.

Como utilizar?

```
#install serverless  
npm i serverless -g  
  
#create template  
serverless create --template aws-nodejs --path myService
```


Estrutura básica



```
service: myService

provider:
  name: aws
  runtime: nodejs8.10

functions:
  hello:
    handler: handler.hello
```



```
'use strict';

module.exports.hello = async (event, context) => {
  return {
    statusCode: 200,
    body: JSON.stringify({
      message: 'Go Serverless v1.0! Your function executed successfully!',
      input: event,
    }),
  }
}
```

Debug



- Serverless Offline
- `set SLS_DEBUG=* && node --inspect ./node_modules/serverless/bin/serverless offline start`





Flutter
Talks 2019

Perguntas e Respostas



Flutter
Talks 2019

Obrigado!