Lambda

## O que é?

* + Lambda se trata de um serviço da AWS que permite efetuar a execução de código sem a necessidade de prover e gerenciar um servidor. É Utilizado a arquitetura de função como serviço (Function as a service - FaaS).
  + Algumas alternativas fora da AWS são Google Cloud Functions, Azure Functions, Netlify Functions e [Serverless Framework](https://www.serverless.com/framework/docs/).

## Por que usar?

* + Permite o uso das principais linguagens do mercado como Node.js, Python, java e etc;
  + Elimina a necessidade de gerenciar infraestrutura;
  + Escalabilidade continua, Lambda executa o código em paralelo, sendo cada processo iniciado individualmente;
  + Custo, sendo cobrado apenas pelo tempo de utilização, eliminando custos de máquina ociosa;
  + Performance consistente, Lambda permite otimizar a escalabilidade vertical por código, por exemplo determinando a quantidade de RAM utilizada em certo código.

## Por que não usar?

* + Estado do processo, uma vez criado para realizar um processamento, este estado não é preservado para futuras chamadas, necessitando a utilização de uma base de dados para armazenar estados;
  + Negação de serviço (Denial of Service - DoS), Lambda possui um limite de 1000 execuções concorrentes por conta, em situações de teste, enquanto produção esteja ativa, existe o risco de causar DoS em produção;
  + Limite de duração, por padrão, Lambda pode abortar um processo caso esteja em execução por mais de 5 minutos, podendo cancelar processo como conversão de vídeo;
  + Latência de inicialização, Alguma situações podem demorar para iniciar o processo, principalmente em funções utilizando JVM (Java Virtual Machine);
  + Entrega, como cada processo deve se encontrar em seu próprio arquivo, a entrega deve ser individual, ou seja, em situação onde exista 20 funções, cada uma deve ser construída e entregue individualmente.

## Como utilizar?

* + Realizar o upload do código como ZIP ou container image, Lambda Automaticamente aloca e executa o código através de requisições ou eventos.

# Outros pontos:

## Disponibilidade (multi-zonas)

* + Possui uma vasta disponibilidade:
    - US East (N. Virginia)
    - US East (Ohio)
    - US West (Northern California)
    - US West (Oregon)
    - Africa (Cape Town)
    - Asia Pacific (Hong Kong)
    - Asia Pacific (Mumbai)
    - Asia Pacific (Osaka)
    - Asia Pacific (Seoul)
    - Asia Pacific (Singapore)
    - Asia Pacific (Sydney)
    - Asia Pacific (Tokyo)
    - Canada (Central)
    - Europe (Frankfurt)
    - Europe (Ireland)
    - Europe (London)
    - Europe (Milan)
    - Europe (Paris)
    - Europe (Stockholm)
    - Middle East (Bahrain)
    - South America (São Paulo)
    - AWS GovCloud (US-East)
    - AWS GovCloud (US-West)

## Análise de custos

* + É realizada a cobrança por requisição e duração (preços baseado em N. Virginia):
    - Requests -> $0.20 per 1M requests
    - Duration\* -> $0.0000166667 for every GB-second
  + \* O preço de duração varia de acordo com a quantidade de memória alocada por função:
    - **Memory (MB) Price per 1ms**
    - 128 $0.0000000021
    - 512 $0.0000000083
    - 1024 $0.0000000167
    - 1536 $0.0000000250
    - 2048 $0.0000000333
    - 3072 $0.0000000500
    - 4096 $0.0000000667
    - 5120 $0.0000000833
    - 6144 $0.0000001000
    - 7168 $0.0000001167
    - 8192 $0.0000001333
    - 9216 $0.0000001500
    - 10240 $0.0000001667

## Automação de infra

* + AWS SAM

## POC

## Segurança

* + O usuário é responsável pela segurança do código e identidade e acesso do serviço.

## Fontes

* + <https://hevodata.com/blog/pros-and-cons-aws-lambda/>
  + <https://aws.amazon.com/lambda/?nc1=h_ls>
  + <https://aws.amazon.com/lambda/pricing/>
  + <https://www.youtube.com/watch?v=drJwMlD9Mjo>
  + <https://docs.aws.amazon.com/lambda/latest/dg/lambda-security.html>
  + <https://youtu.be/Fx3ZGy-mbV4>