



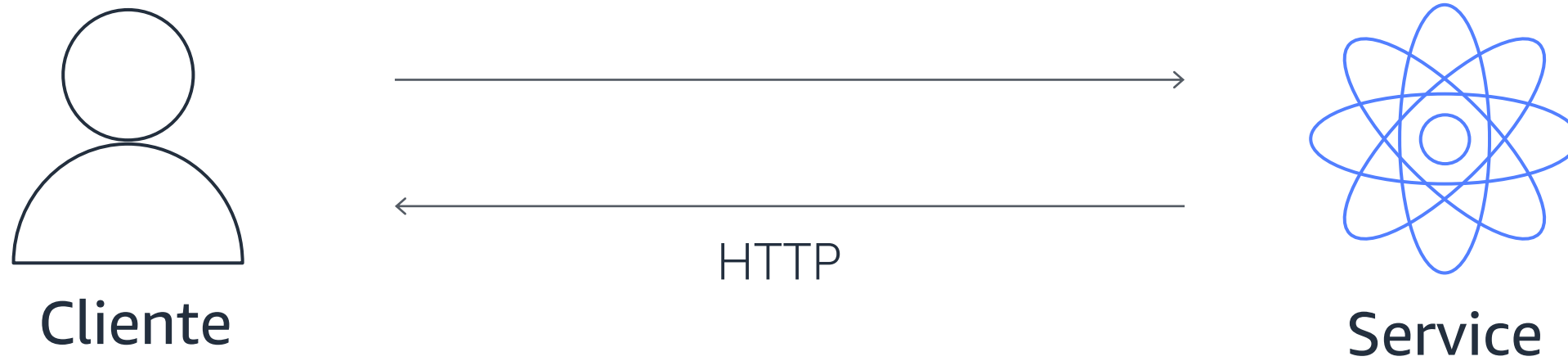
# O Canivete Suíço para o Desenvolvimento Serverless

Ana Cunha  
Developer Advocate

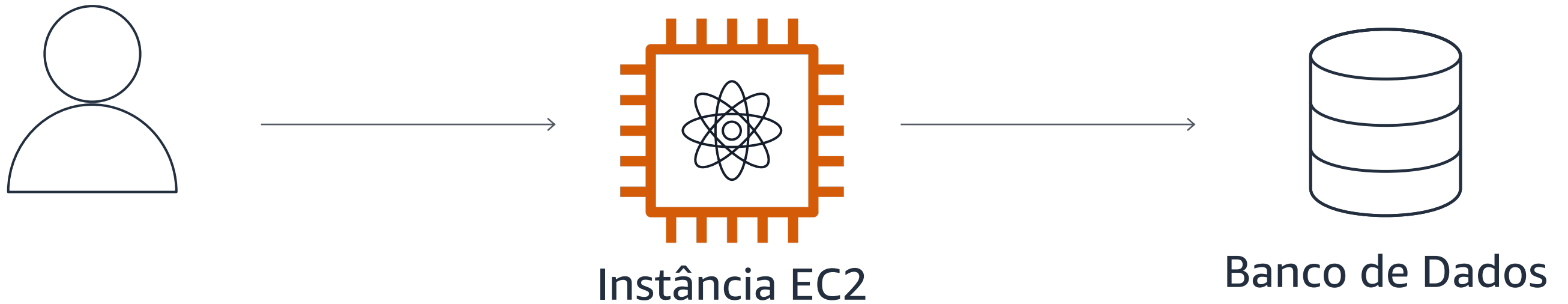


# Arquiteturas Tradicionais

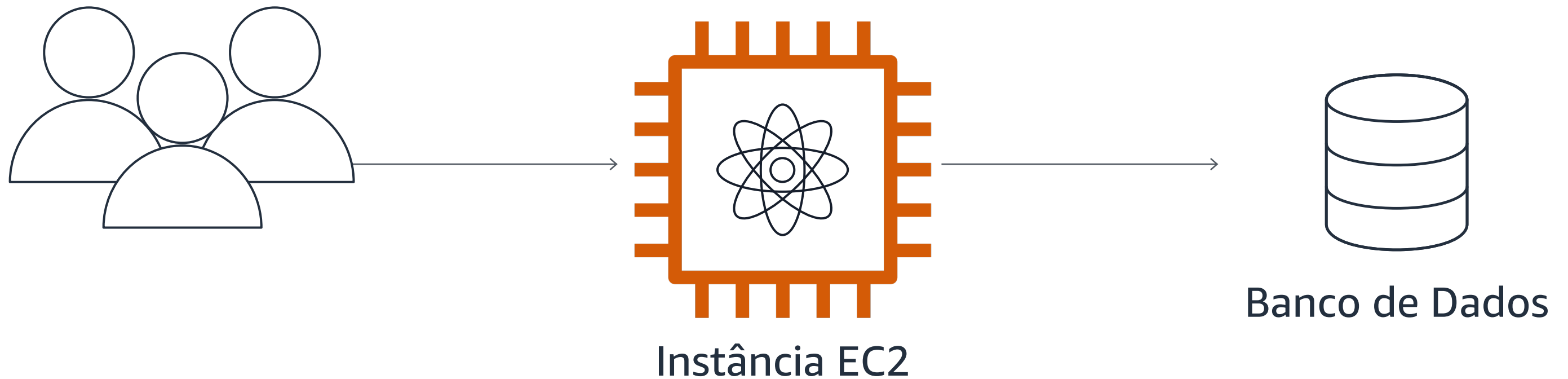
# Web Service



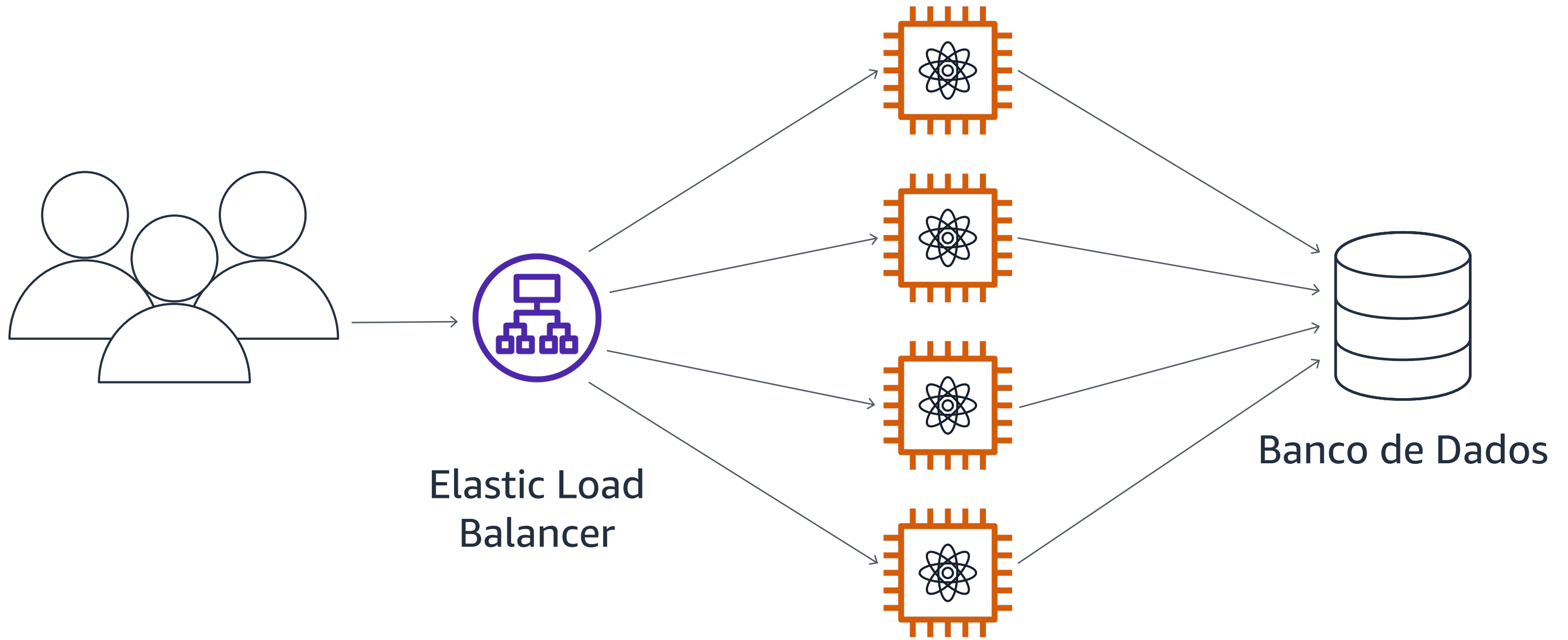
# Web Service na Nuvem



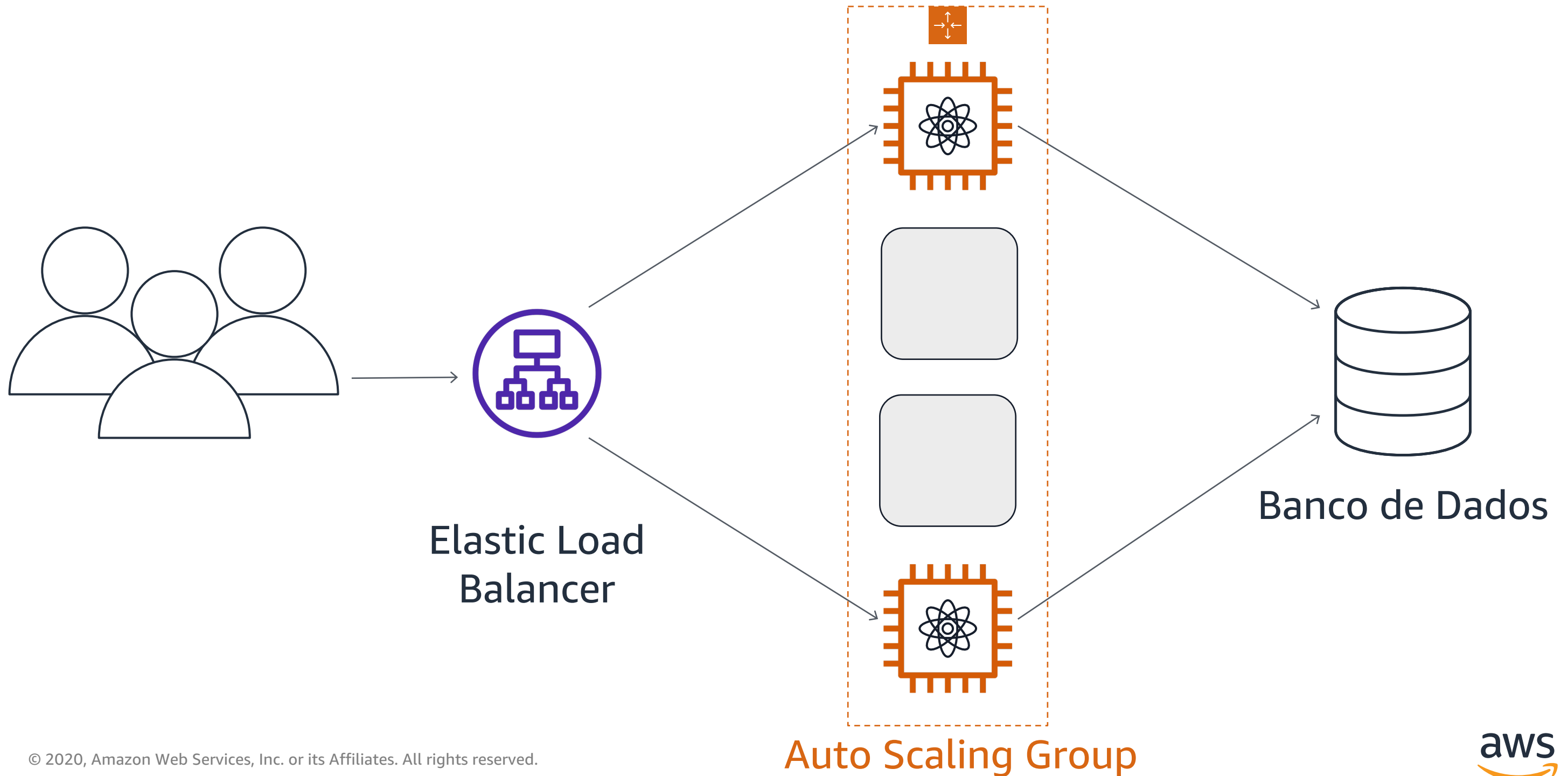
# Web Service na Nuvem – Escalando Verticalmente



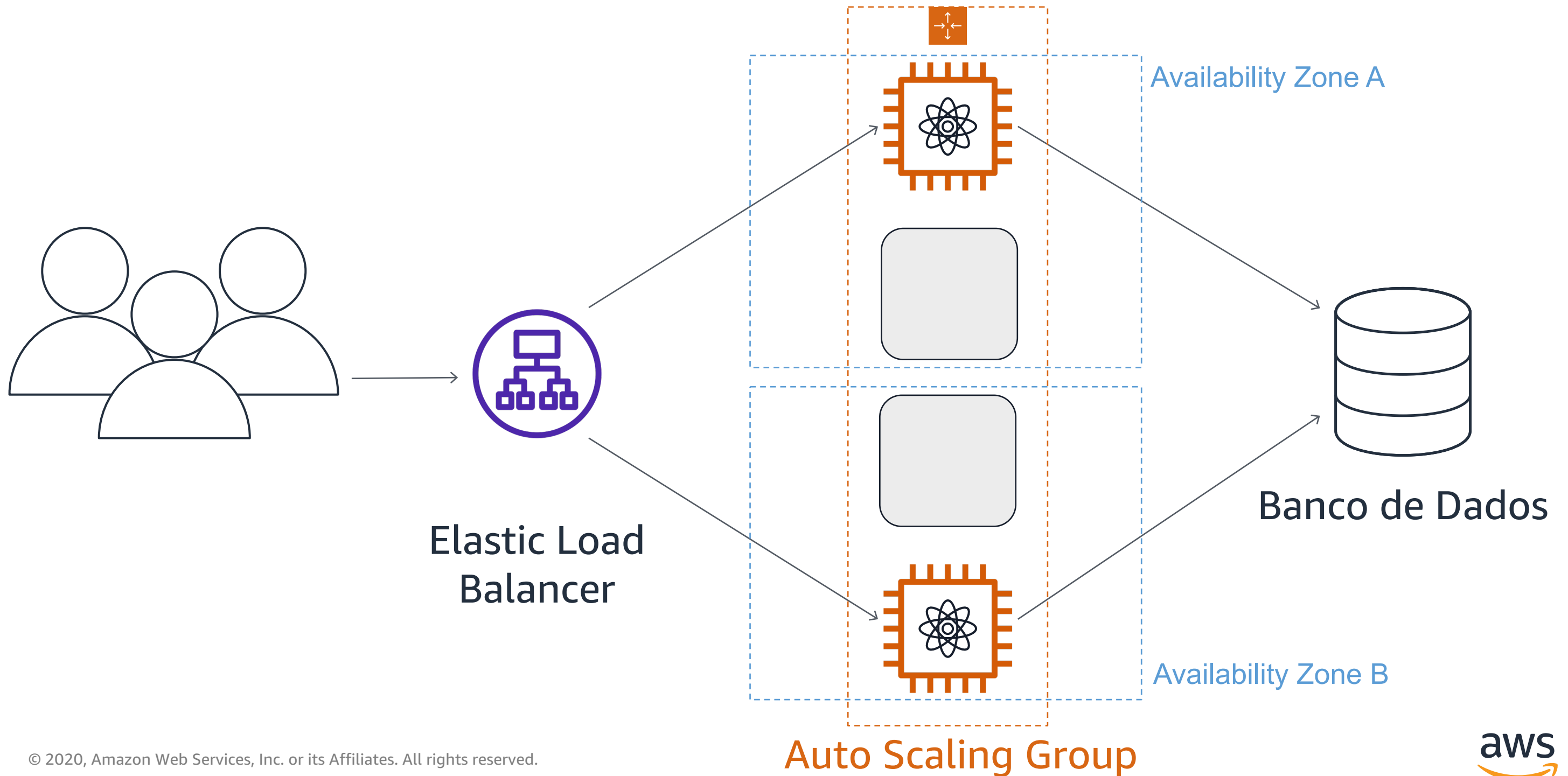
# Web Service na Nuvem – Escalando Horizontalmente



# Web Service na Nuvem – Auto Scaling



# Web Service na Nuvem – Alta Disponibilidade





# Gerenciamento da Aplicação

## Core Business

- Aplicação + Código
- Lógica de Negócio

## Infraestrutura

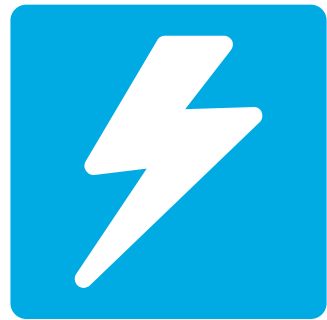
- Provisionamento de Servidores
- Sistemas Operacionais
- Patches
- Políticas de Escala
- Planejamento de Capacidade
- Disponibilidade

# Serverless

# Componentes Essenciais



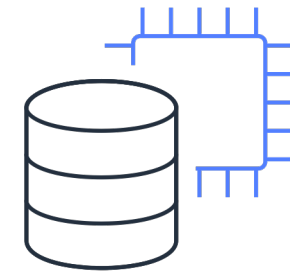
# Componentes Essenciais



Evento

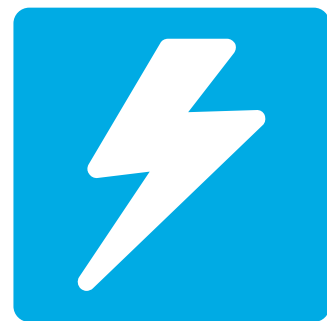


Código



Outros  
Serviços

# AWS Lambda – Componentes Serverless



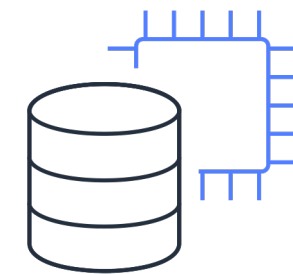
**Evento**

Event Source



**Handler**

AWS Lambda  
Function



Amazon SNS



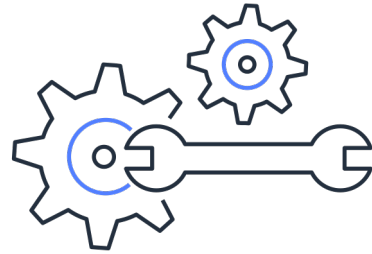
Amazon S3

**Outros  
Serviços**



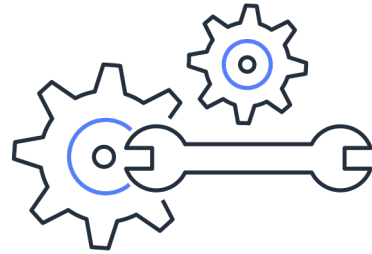
Amazon DynamoDB

# Serverless



- ✓ Sem gerenciamento de servidores

# Serverless

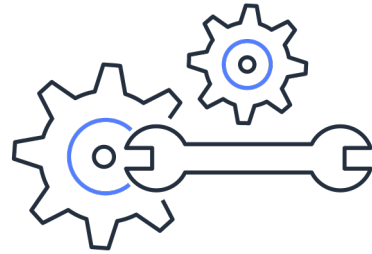


✓ Sem gerenciamento de servidores



✓ Escalabilidade flexível

# Serverless



✓ Sem gerenciamento de servidores



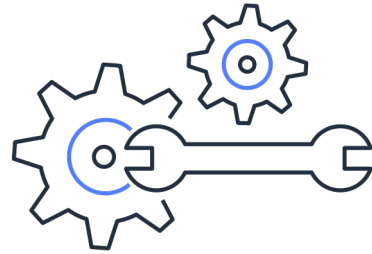
✓ Escalabilidade flexível



✓ Pagamento pelo valor



# Serverless



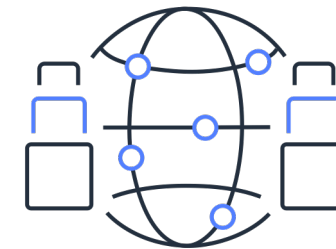
✓ Sem gerenciamento de servidores



✓ Escalabilidade flexível

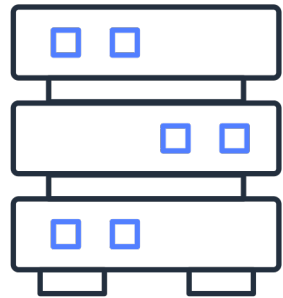


✓ Pagamento pelo valor



✓ Alta disponibilidade

# Serverless Amplia os Benefícios da Nuvem



Servidores físicos  
em data centers



Servidores virtuais  
na nuvem



Serverless

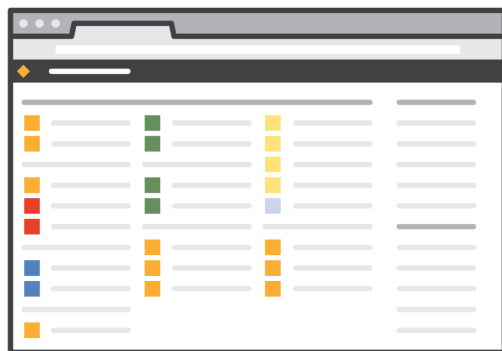
# Desenvolvimento Serverless

Boas Práticas

# Boas Práticas de Desenvolvimento São as Mesmas

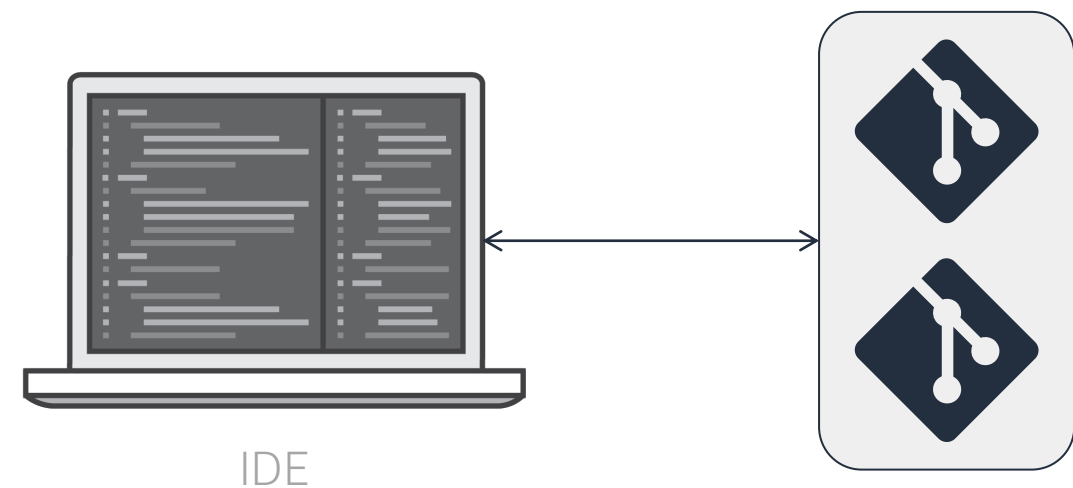
## Experimentação

Uso do Console de Gerenciamento da **AWS** para criação, teste e deploy.

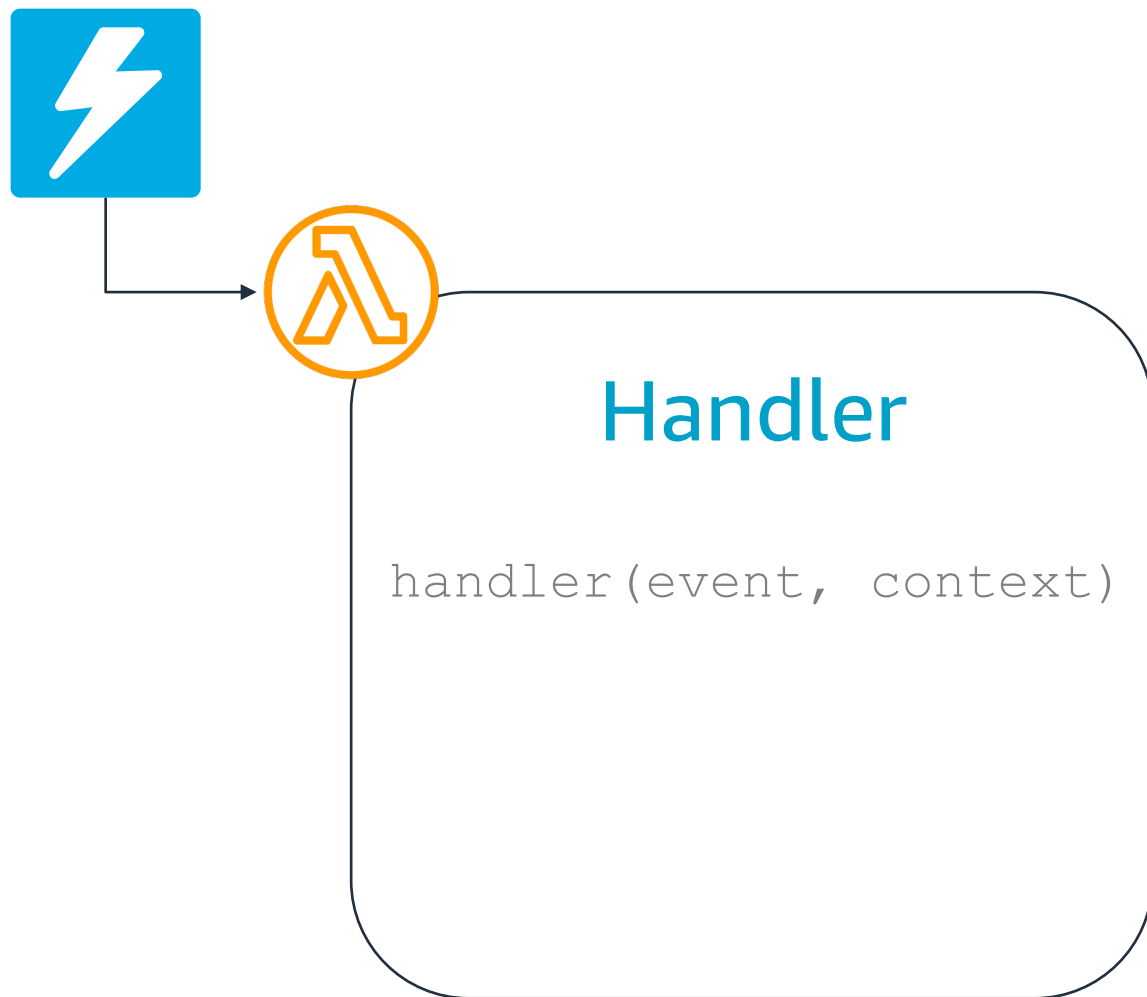


## Produção

Desenvolvimento e deploy seguindo boas práticas.

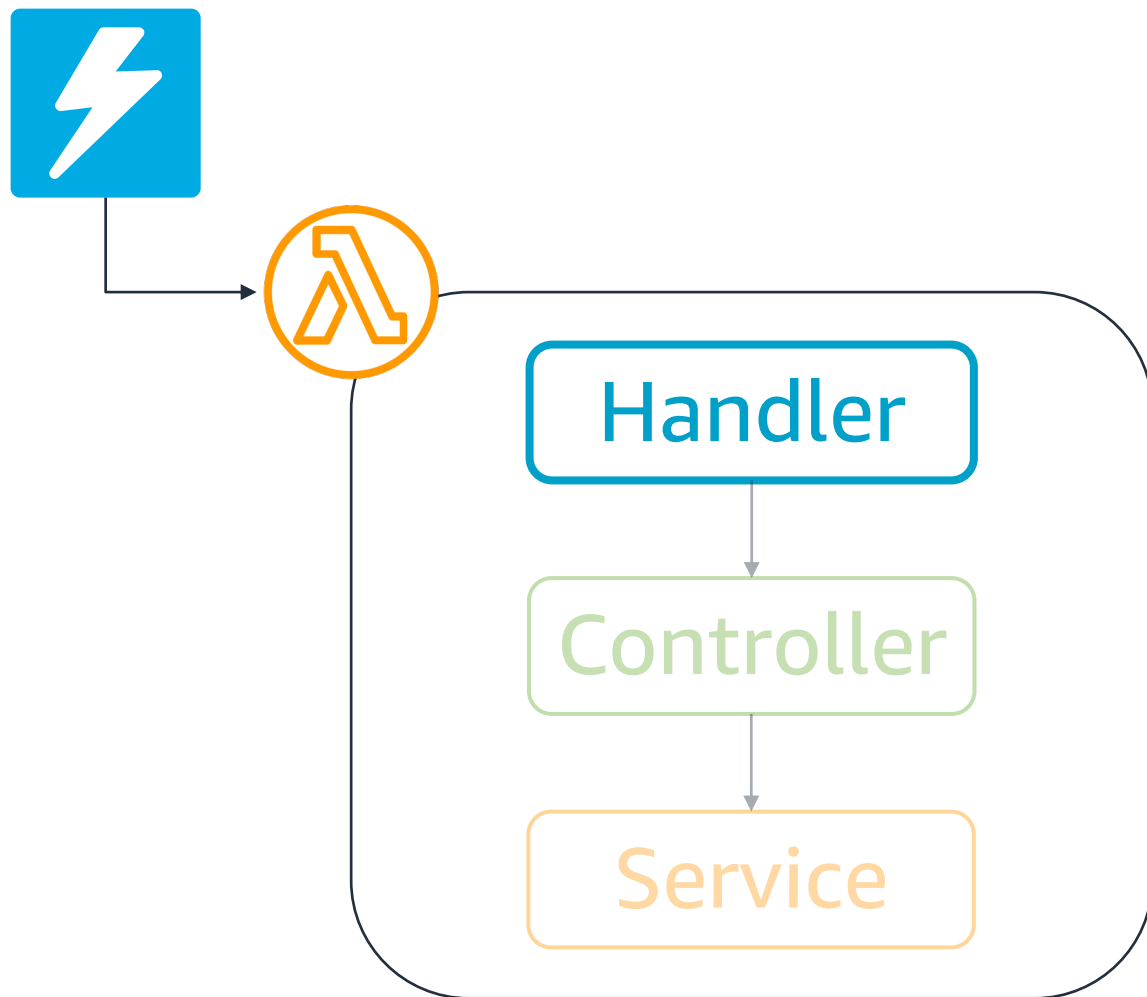


# Separação de Responsabilidades



- Toda função Lambda tem um método **handler**
- Processamento de eventos
- Funcionalidade separada em múltiplas classes e funções

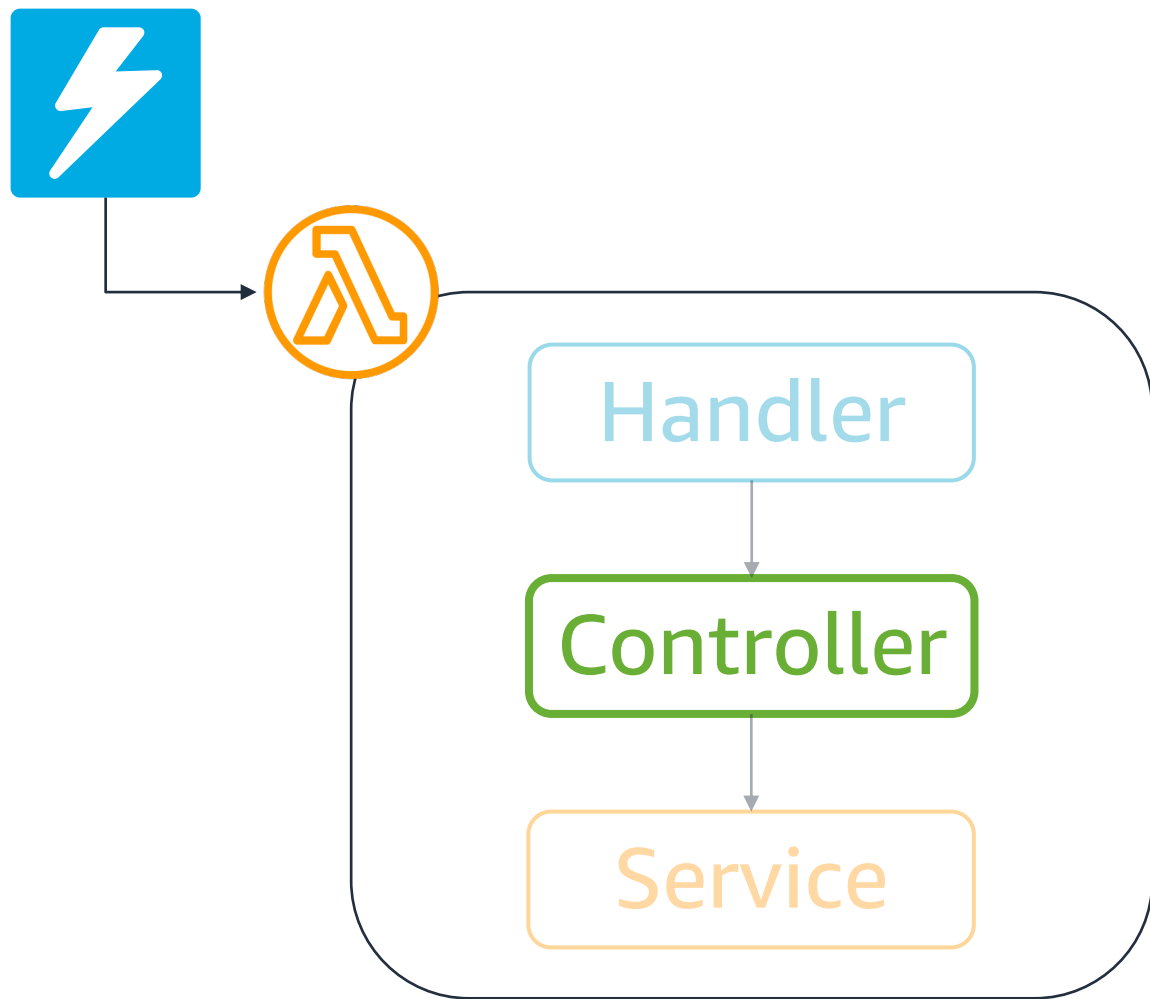
# Separação de Responsabilidades



## Handler

- Configuração da função
- Código específico para Lambda
- Sem lógica de negócio

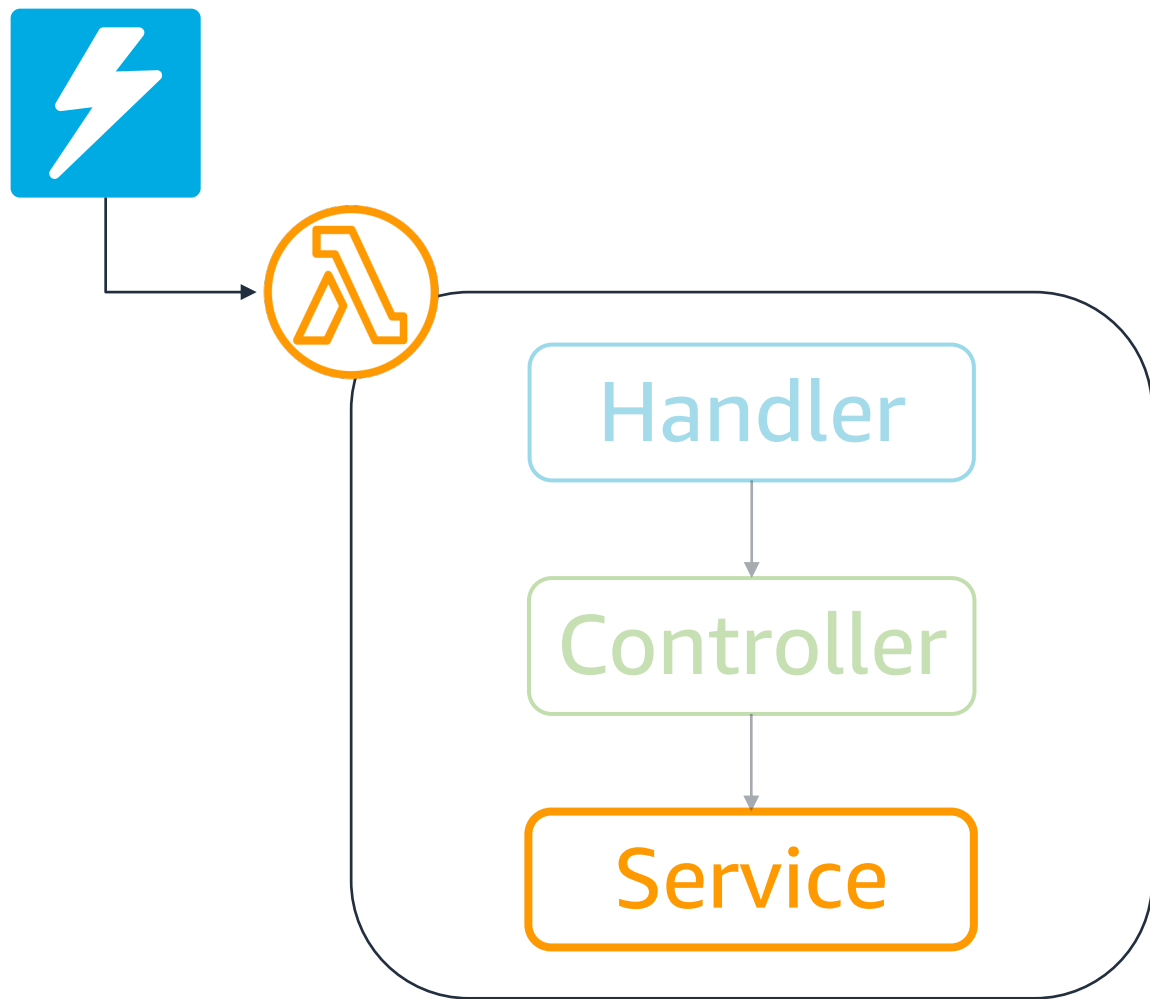
# Separação de Responsabilidades



## Controller

- Processamento do evento
- Lógica de Negócio

# Separação de Responsabilidades



## Service

- Abstrações
- Integrações externas



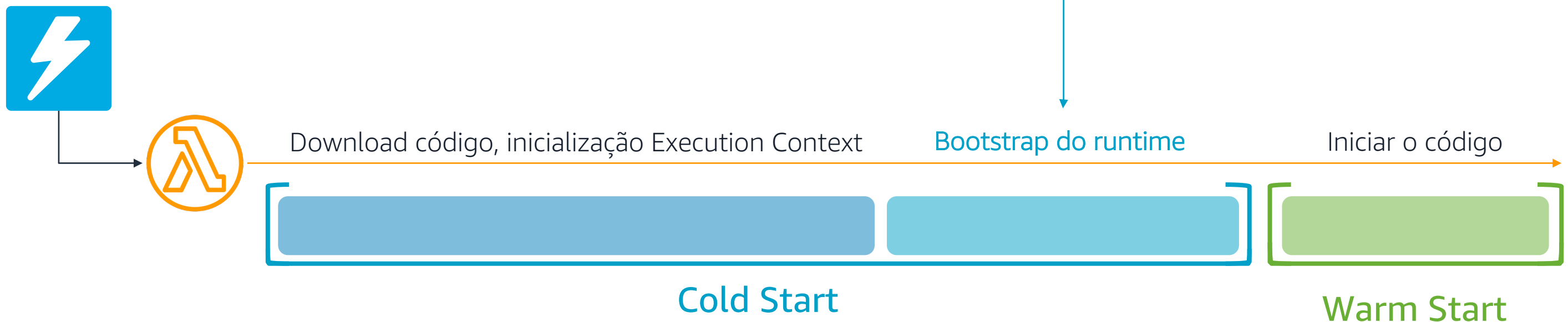
# Execution Context

- Ambiente de runtime temporário
- Baseado na configuração de memória e timeout definidas
- Recursos necessários para executar a função Lambda
- Carrega e inicializa as dependências da função
- “Bootstrapping” do runtime
- Reutilizado por algum tempo pelas próximas execuções da função

# Dependências

## Bootstrap

- Minimizar dependências
- Considerar tempo de inicialização
- Outras dependências e frameworks



# Suporte Serverless para Frameworks Conhecidos

AWS Serverless  
Java Container



AWS Serverless  
Express



Zappa



# Desenvolvimento Serverless

Ferramentas

# Ferramentas

## Conhecidas

- Sistema de Controle de Versão
- IDE
- Build
- Deploy

## Novas

- Framework para desenvolvimento de aplicações Serverless

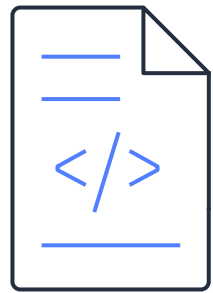
# Serverless Application Framework

Criação

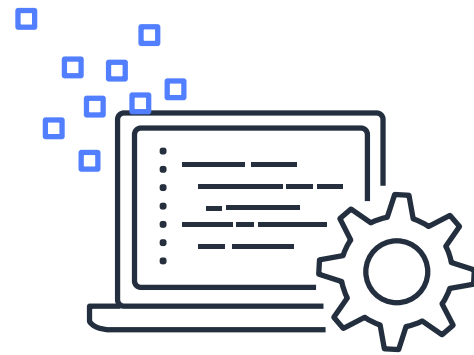
Build

Package

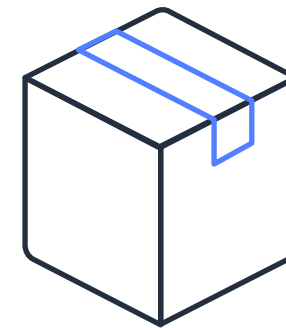
Deploy



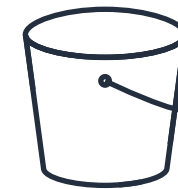
Código



Binário



Arquivo ZIP



S3 Bucket

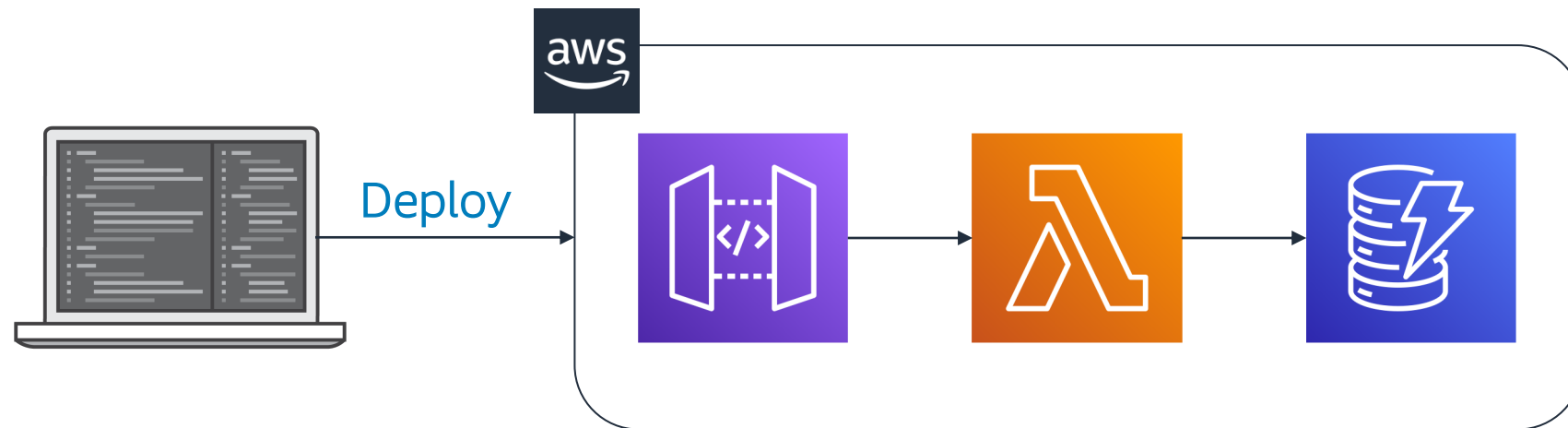


AWS Lambda

IDE + Ferramenta de Build

Serverless Application Framework

# Etapas de Deploy



1. Build
2. ZIP
3. Upload para Amazon S3
4. Criar atualizar roles
5. Criar função Lambda
6. Criar API REST
7. Criar recurso
8. Criar método
9. Criar tabela
10. Configurar integrações

# AWS CloudFormation

- Infraestrutura como código
- Definição dos recursos que precisam ser criados
- Provisionamento automatizado e replicável
- Código = versionamento + revisão + testes



# Serverless Application Model (SAM)

# SAM

- Framework para construir aplicações serverless
- Escreva com YAML
- Deploy com AWS CloudFormation
- Open Source
- <https://github.com/aws-labs/serverless-application-model>



# SAM

## SAM Templates

- Infraestrutura como código
- Aplicações Serverless
- Sintaxe abreviada
- Definir recursos e event sources

## SAM CLI

- Conjunto de ferramentas
- Desenvolvimento local
- Debugging
- Build, package, deploy

# SAM Template

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  GetHtmlFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: ./src
      Handler: index.gethtml
      Runtime: nodejs12.x
      Policies: AmazonDynamoDBReadOnlyAccess
    Events:
      Api:
        Type: Api
        Properties:
          Path: /{proxy+}
          Method: ANY
  ListTable:
    Type: AWS::Serverless::SimpleTable
```

# SAM Serverless Resources

`AWS::Serverless::Function`

`AWS::Serverless::Api`

`AWS::Serverless::HttpApi`

`AWS::Serverless::SimpleTable`

`AWS::Serverless::LayerVersion`

`AWS::Serverless::Application`

`AWS::Serverless::StateMachine`

# SAM CLI

```
sam init
```

# SAM CLI

`sam init`

`sam build`

# SAM CLI

`sam init`

`sam build`

`sam deploy --guided`



# SAM CLI

```
sam init
```

```
sam build
```

```
sam deploy --guided
```

```
# Host API locally
```

```
sam local start-api
```

# SAM CLI

**sam init**

**sam build**

**sam deploy --guided**

# Directly invoke function

**sam local invoke**

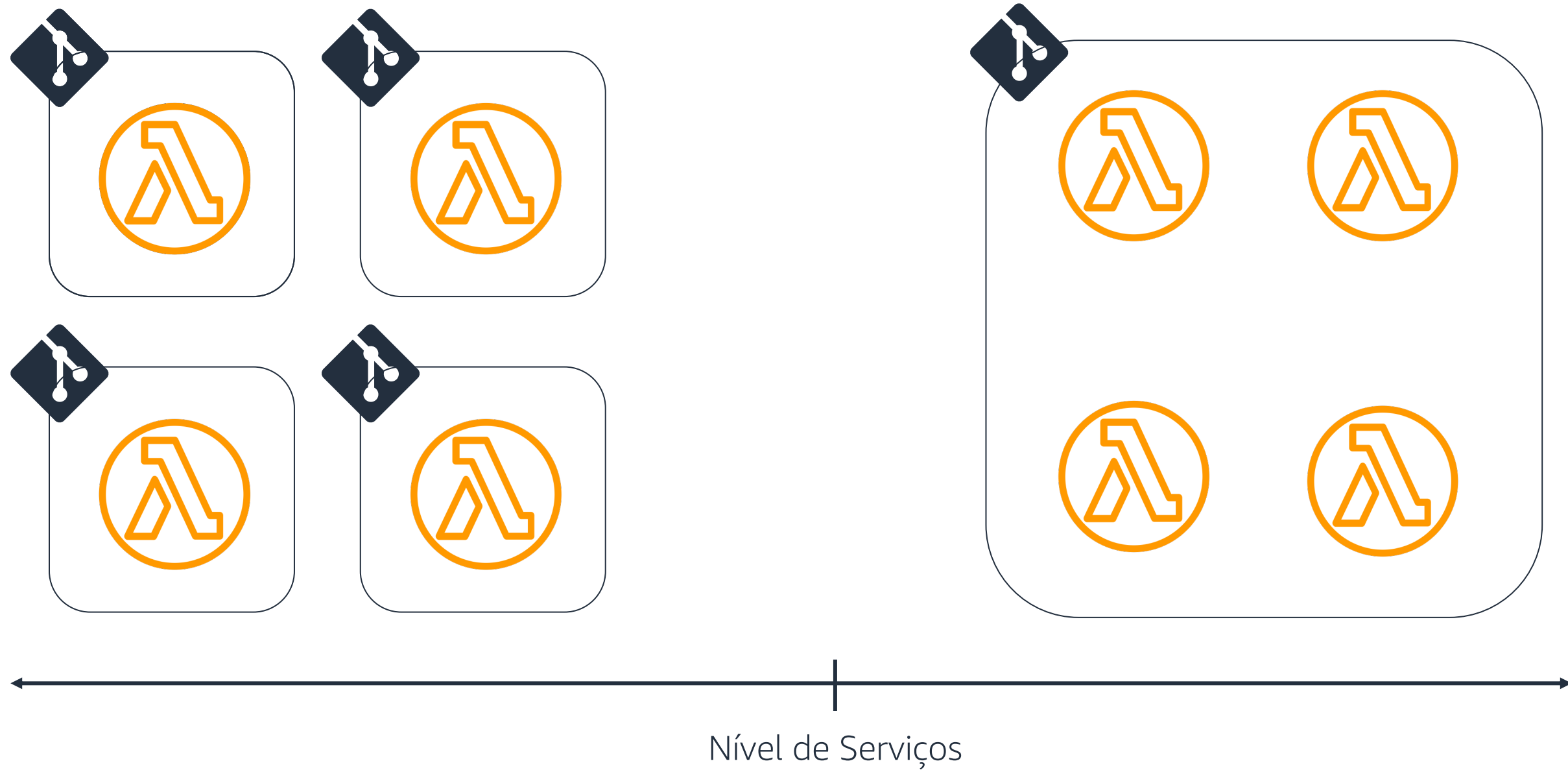
# Desenvolvimento Serverless

Sistemas de Controle de Versão

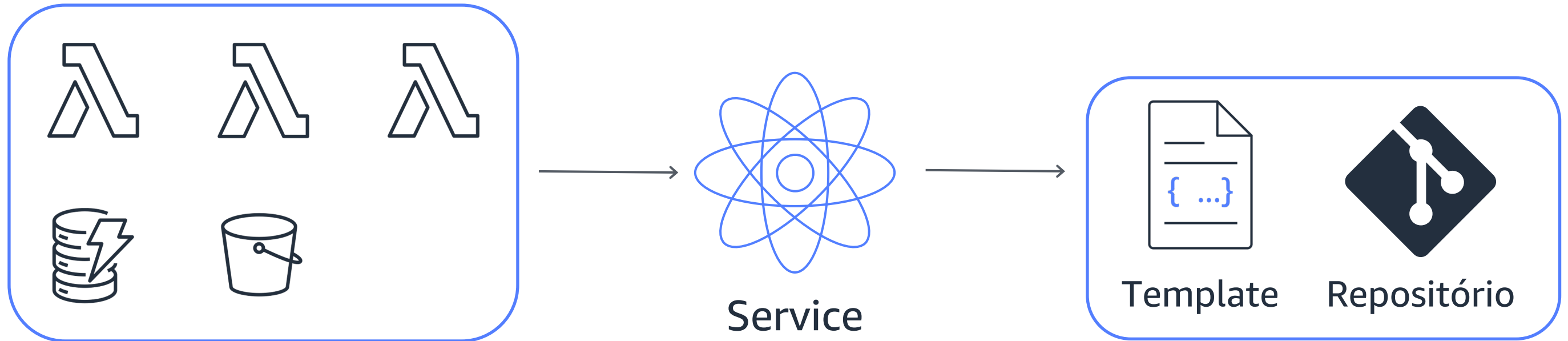
# Repositórios de Código



# Repositórios de Código



# Serviços



Uma ou mais funções,  
serviços e recursos AWS

# Desenvolvimento Serverless

Ambientes de Desenvolvimento Integrado (IDE)

# AWS Cloud9

- Criação
- Execução
- Debug
- Deploy





# Obrigada!

Ana Cunha

 @mcunhaana

 analuizacunha

