

---

# Serverless

— Prof. Raphael Rodrigues Pereira —

---

# O que é Serverless

É um modelo de computação em nuvem que permite que os desenvolvedores construam e executem aplicativos sem necessidade de gerenciamento dos servidores.

- O servidor é gerenciado pelo provedor de serviços em nuvem.
- Escalonamento automático
- Cliente paga somente pela utilização (requisições e tempo de execução)
- Tempo de execução: 15 minutos\*

# Comparando com arquitetura tradicional

## Arquitetura Tradicional

- Montar e Configurar Servidor da aplicação
- Configuração Infraestrutura física ou virtual (Instância, Recursos (memória, processamento))
- Servidor sempre em execução

## Serverless

- Sem contato com a infra
- DEV foca apenas no código
- Pagamento por execução
- Ideal para funções que são executadas esporadicamente
  - Resposta a algum evento
  - Rotinas Agendadas

# Principais serviços Serverless nos Provedores de Nuvem

**AWS**

Lambda



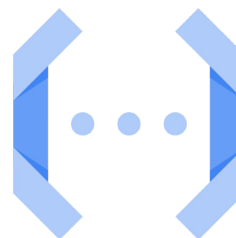
**Microsoft**

Azure Functions



**Google**

Google Cloud  
Functions



Cloud Functions

# Principais vantagens de se utilizar Serverless

## AWS Lambda

- Paga somente se a função é executada.
- Suporte a várias linguagens de programação nativamente (Python, Node, Java, C#, Go, PHP)
- Possibilita subir imagem de container até mesmo com outras linguagens
- Integração com outros serviços AWS (S3, SNS, etc)
- Integração com CloudWatch - logs da AWS
- Free Tier: 1 milhão de requisições por mês grátis \*



# Exemplos de Uso

Compactar imagens que foram subidas para um Bucket no S3

Monitorar um diretório do S3 e, quando um novo arquivo é adicionado, dispara uma ação para processá-lo ou enviá-lo para outro serviço.

Automatização de tarefas:

- Limpeza de cache de uma aplicação diariamente
- Gerar relatório semanal com dados de outros serviços AWS

# Mais Exemplos de Uso - segundo a AWS

## Quando usar Lambda

Lambda é um serviço de computação ideal para cenários de aplicativos que precisam aumentar rapidamente e diminuir para zero quando não há demanda. Por exemplo, você pode usar Lambda para:

- **Processamento de arquivos:** use o Amazon Simple Storage Service (Amazon S3) para acionar o processamento de dados do Lambda em tempo real após um upload.
- **Processamento de fluxo:** use o Lambda e o Amazon Kinesis para processar dados de streaming em tempo real para rastreamento de atividades de aplicativos, processamento de ordens de transações, análise de fluxo de cliques, limpeza de dados, filtragem de logs, indexação, análise de mídia social, telemetria de dados de dispositivos da Internet das Coisas (IoT) e medição.
- **Aplicativos Web:** combine o Lambda com outros serviços da AWS para criar aplicativos Web poderosos que aumentam e diminuem automaticamente e são executados em uma configuração de alta disponibilidade em vários data centers.
- **Backends de IoT:** crie backends sem servidor usando Lambda para lidar com solicitações de API da Web, dispositivos móveis, IoT e de terceiros.
- **Backends móveis:** crie backends usando Lambda e Amazon API Gateway para autenticar e processar solicitações de API. Use o AWS Amplify para integrar facilmente com seus frontends iOS, Android, Web e React Native.

# Alguns conceitos

**AWS Lambda:** serviço serverless da AWS

**Lambda Function:** código / função propriamente dita inserida no AWS Lambda

**Lambda Application:** é uma combinação de lambda functions, triggers e outros recursos que juntos executam uma determinada tarefa.



# Exemplo Prático AWS

## Criando primeira Lambda Function

**ATENÇÃO!** Os próximos slides servem apenas como guia informativo. Não é necessário que os alunos sigam esses passos durante a aula. No entanto, para quem desejar reproduzir as etapas, será necessário criar uma conta na AWS e cadastrar um cartão de crédito.

A AWS oferece recursos gratuitos através do Free Tier, mas é importante ficar atento aos limites de uso para evitar cobranças indesejadas. Recomendamos também que, ao final dos experimentos, todos os recursos criados na AWS sejam excluídos para evitar custos futuros.

# Criação da Lambda Function

Neste exemplo será criado um bucket S3 que, ao realizar um upload, acionará a lambda function que apenas exibirá o nome do arquivo e horário de importação.

Essa informação poderia ser enviada para um outro serviço AWS também, ou armazenar essas informações em um banco de dados como DynamoDB.

# Acesse o console AWS, busque por Lambda e clique em Criar Função

[Lambda](#) > [Funções](#) > Criar função

## Criar função [Informações](#)

Escolha uma das opções a seguir para criar a função.

☒ Criar do zero

Comece com um simples exemplo de Hello World.

☐ Usar um esquema

Crie um aplicativo do Lambda a partir do código de exemplo e de predefinições de configuração para casos de uso comum.

☐ Imagem de contêiner

Selecione uma imagem de contêiner a ser implantada para sua função.

Escolha uma das opções para iniciar

### Informações básicas

#### Nome da função

Insira um nome que descreva o propósito da função.

testFunction1

Dê o nome da função

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

#### Tempo de execução [Informações](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 20.x



Informe a linguagem para a AWS configurar o ambiente adequado para execução.

#### Arquitetura [Informações](#)

Escolha a arquitetura do conjunto de instruções desejada para o código da função.

☒ x86\_64

☐ arm64


## Mais abaixo temos:

### ▼ Alterar a função de execução padrão

#### Papel de execução

Escolha uma função que defina as permissões da sua função. Para criar uma função personalizada, acesse o [console do IAM](#).

- ☒ Criar uma função com permissões básicas do Lambda
- ☐ Usar uma função existente
- ☐ Criar uma função a partir da política da AWS templates

 A criação da função pode levar alguns minutos. Não exclua a função ou edite as políticas de confiança ou permissões dessa função.

O Lambda criará um papel de execução chamado myFunctionTest1-role-lpkfsj5l com a permissão de fazer upload de logs para o Amazon CloudWatch Logs.

**Você pode optar por manter as permissões básicas para a função Lambda, selecionar políticas predefinidas aplicadas a uma função já existente, ou escolher entre políticas padrão fornecidas pela AWS. Caso necessário, também é possível criar uma política totalmente personalizada através do IAM.**

## ▼ Configurações avançadas

☐ **Habilitar assinatura de código** [Informações](#)

Use as configurações de assinatura de código para garantir que o código tenha sido assinado por uma origem aprovada e não tenha sido alterado desde a assinatura.

☐ **Habilitar URL da função** [Informações](#)

Use URLs da função para atribuir endpoints HTTP(S) à função do Lambda.

☐ **Habilitar etiquetas** [Informações](#)

Uma etiqueta é um rótulo que você atribui a um recurso da AWS. Cada etiqueta consiste em uma chave e um valor opcional. Você pode usar etiquetas para pesquisar e filtrar seus recursos, rastrear seus custos da AWS e impor controle de acesso baseado em atributos.

☐ **Habilitar VPC** [Informações](#)


Conecte sua função a uma VPC para acessar recursos privados durante a invocação.

Cancelar

**Criar função**

Nas configurações avançadas é possível, por exemplo, habilitar a URL da função para que possa fazer chamadas HTTP para a função Lambda. Vamos deixar sem nada habilitado e **Criar Função**

Após criar a função a seguinte tela é exibida:  
Visão Geral da Função:

 A função **myFunctionTest1** foi criada com êxito. Agora é possível alterar o código e a configuração dela. Para invocar sua função com um evento de teste, selecione "Testar".

[Lambda](#) > [Funções](#) > myFunctionTest1


# myFunctionTest1


Controlar Copiar ARNAções

Visão geral da funçãoInformações

Exportar para o Application ComposerFazer download

DiagramaModelo

 myFunctionTest1

 Layers (0)

+ Adicionar gatilho

+ Adicionar destino


Descrição

-

Última modificação

há 22 segundos

ARN da função

 arn:aws:lambda:us-east-1:011528280878:function:myFunctionTest1

URL da função [Informações](#)

-

Código

Testar

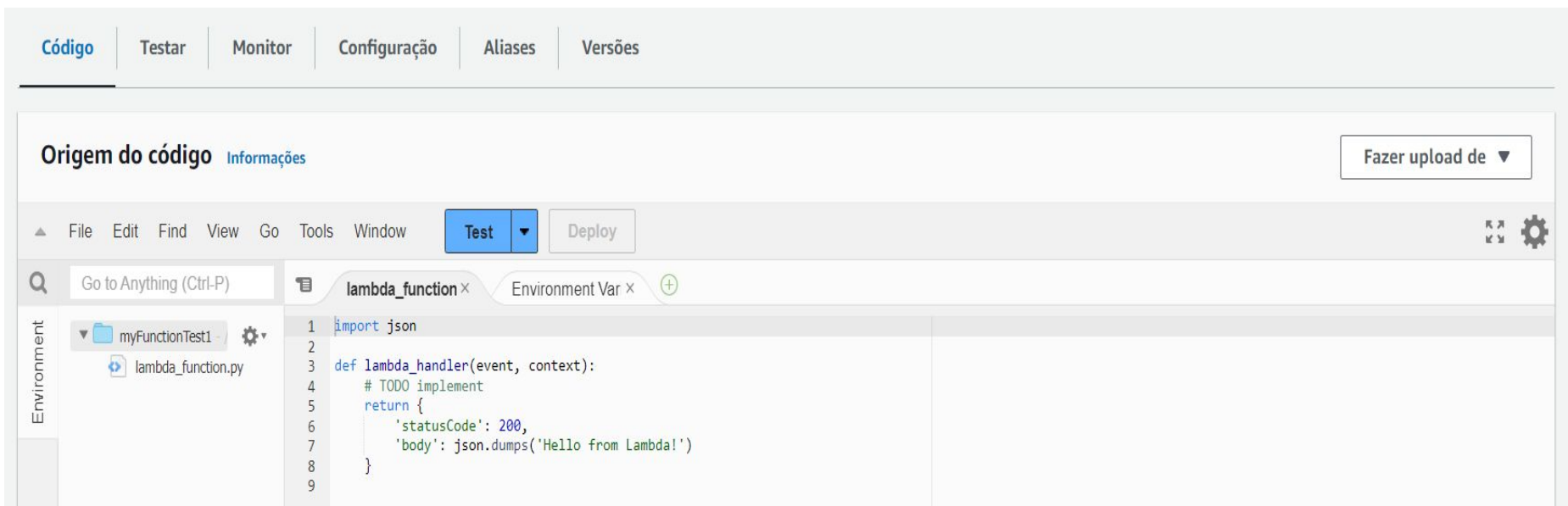
Monitor

Configuração

Aliases

Versões

Após criar a função a seguinte tela é exibida:  
Seção contendo código, testes e monitoramento da Função Lambda



Altere o código da aplicação para o seguinte:

[Clique para acessar o código no github](#)

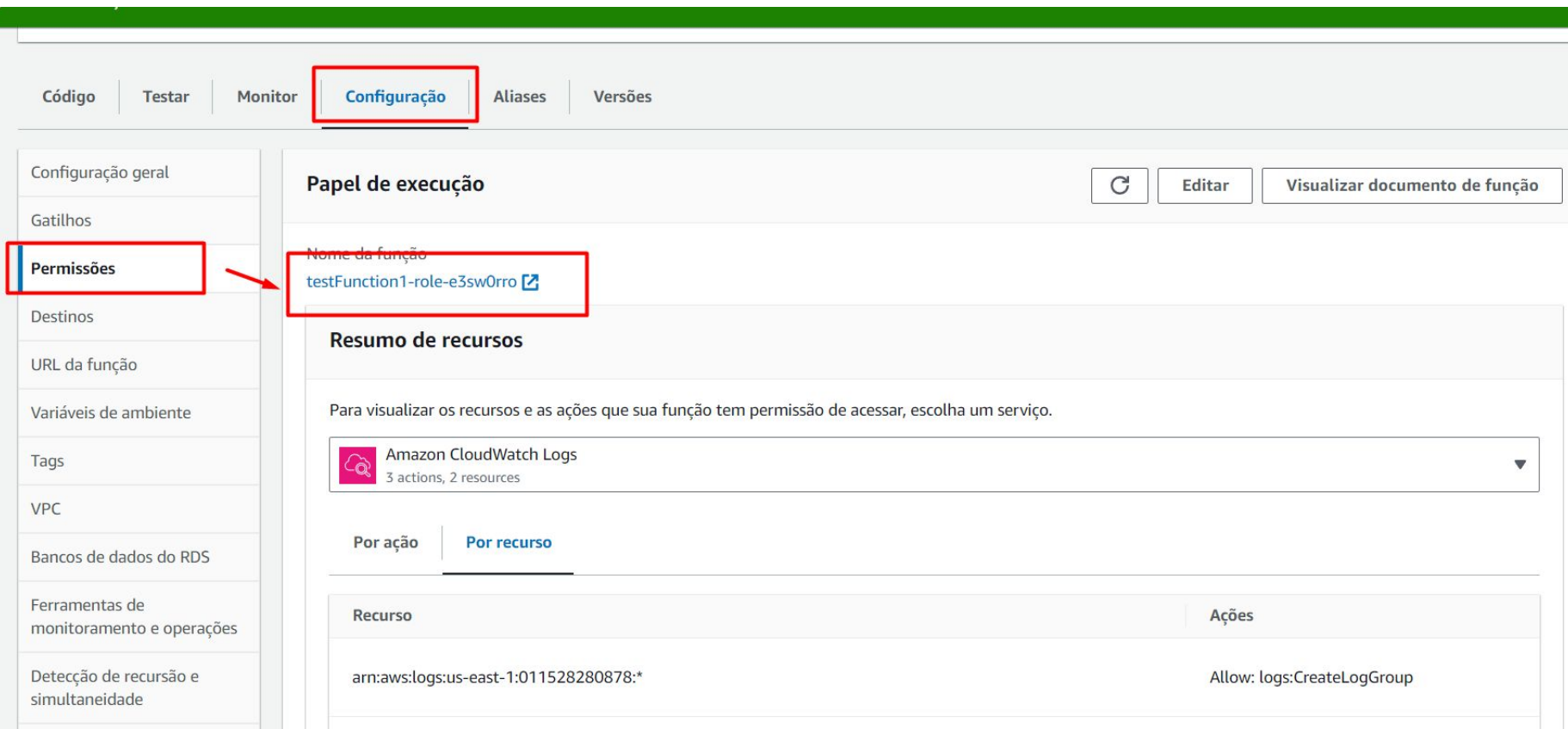
```
import json
from datetime import datetime
def lambda_handler(event, context):
    # Extrai informações do evento S3
    try:
        for record in event['Records']:
            bucket_name = record['s3']['bucket']['name']
            object_key = record['s3']['object']['key']
            # Obtém a data e hora atuais
            current_time = datetime.utcnow().strftime( '%Y-%m-%d %H:%M:%S' )

            print(
                json.dumps({
                    'message': 'Upload de arquivo realizado' ,
                    'bucket_name': bucket_name,
                    'file_name': object_key,
                    'timestamp': current_time
                })
            )
            # Retorna o nome do arquivo, data e hora
    return {
        'statusCode': 200,
        'body': json.dumps({
```



Alterando permissões para que a lambda tenha acesso ao S3

Acesse a guia Configuração > Permissões > clique sobre o nome da role que foi criada junto com a Lambda function



**Configuração** | Aliases | Versões


Configuração geral  
Gatilhos  
**Permissões**  
Destinos  
URL da função  
Variáveis de ambiente  
Tags  
VPC  
Bancos de dados do RDS  
Ferramentas de monitoramento e operações  
Detecção de recursão e simultaneidade

**Papel de execução** Recarregar Editar Visualizar documento de função

Nome da função  
testFunction1-role-e3sw0rro [Link](#)

**Resumo de recursos**

Para visualizar os recursos e as ações que sua função tem permissão de acessar, escolha um serviço.

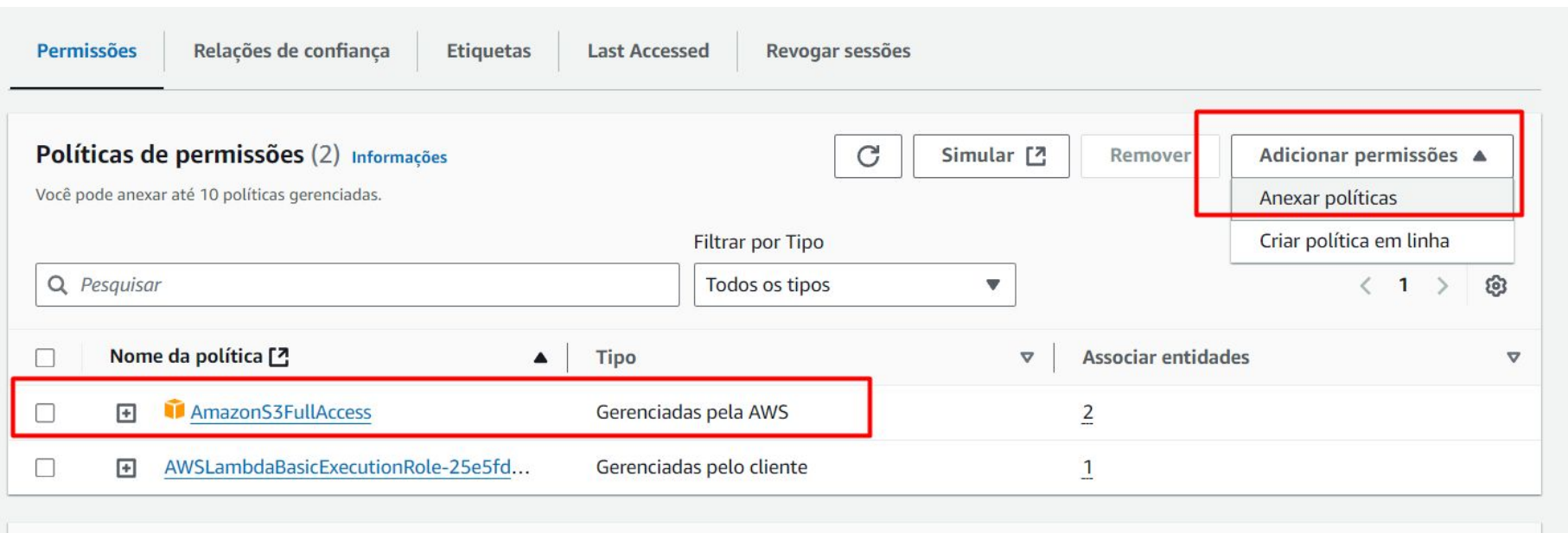
 Amazon CloudWatch Logs  
3 actions, 2 resources

Por ação | **Por recurso**

Recurso	Ações
arn:aws:logs:us-east-1:011528280878:*	Allow: logs:CreateLogGroup

Alterando permissões para que a lambda tenha acesso ao S3

Clique em Adicionar Permissões > Anexar Políticas > filtre por S3 e escolha a permissão mais conveniente.



The screenshot shows the AWS IAM console interface. At the top, there are tabs for 'Permissões', 'Relações de confiança', 'Etiquetas', 'Last Accessed', and 'Revogar sessões'. The 'Permissões' tab is selected. Below the tabs, there's a section titled 'Políticas de permissões (2) Informações'. To the right of this section are buttons for 'Recarregar', 'Simular', 'Remover', and 'Adicionar permissões'. The 'Adicionar permissões' button is highlighted with a red box, and its dropdown menu is open, showing options: 'Anexar políticas' (highlighted), 'Criar política em linha', and 'Criar política em linha'. Below the dropdown, there's a search bar labeled 'Pesquisar' and a filter dropdown labeled 'Filtrar por Tipo' with 'Todos os tipos' selected. The main content area is a table with columns: 'Nome da política', 'Tipo', and 'Associar entidades'. The first row is highlighted with a red box and contains the policy 'AmazonS3FullAccess' under the 'Gerenciadas pela AWS' type. The second row is 'AWSLambdaBasicExecutionRole-25e5fd...' under the 'Gerenciadas pelo cliente' type.

	Nome da política	Tipo	Associar entidades
<input type="checkbox"/>	<a href="#">AmazonS3FullAccess</a>	Gerenciadas pela AWS	<a href="#">2</a>
<input type="checkbox"/>	<a href="#">AWSLambdaBasicExecutionRole-25e5fd...</a>	Gerenciadas pelo cliente	<a href="#">1</a>

Neste exemplo selecionamos S3FullAccess apenas para demonstração - cuidado com isso em ambiente corporativo - o ideal é que seja mais restrito possível.

# Criando gatilho para eventos de criação de objetos no bucket S3

[Lambda](#) > Adicionar gatilhos

## Adicionar gatilho

### Configuração do gatilho [Informações](#)

 S3  
aws asynchronous storage

#### Bucket

Escolha ou insira o ARN de um bucket do S3 que serve como fonte do evento. O bucket deve estar na mesma região da função.

🔍 s3/bucket-test01-dev ✕ ↺

Região do bucket: us-east-1

#### Tipos de evento

Selecione os eventos para os quais você deseja acionar a função do Lambda. Também é possível configurar um prefixo ou sufixo para um evento. No entanto, para cada bucket, eventos individuais não podem ter várias configurações com prefixos ou sufixos sobrepostos que possam corresponder à mesma chave de objeto.

▼

Todos os eventos de criação de objeto ✕

#### Prefixo - *opcional*

Insira um único prefixo opcional para limitar as notificações a objetos com chaves que começam com caracteres correspondentes. Qualquer [caractere especial](#)  deve ser codificado no URL.

Por exemplo, imagens/

Primeiramente, Crie um bucket S3.

Na seção de Visão Geral da Lambda clique em Criar Gatilho  
Selecione o Bucket

Tipo de evento ficará o padrão  
(Todos os eventos e criação de objeto)

Pode ser informado um prefixo ou sufixo para que seja executado somente quando o arquivo ter essas informações.

# Cuidado com Invocação Recursiva

## Invocação recursiva

Se a sua função grava objetos em um bucket do S3, verifique se você está usando diferentes buckets do S3 para entrada e saída. Gravar no mesmo bucket aumenta o risco de criar uma invocação recursiva, o que pode resultar em aumento do uso do Lambda e em maiores custos.

Saiba mais [↗](#)

- ✓ Reconheço que o uso do mesmo bucket do S3 para entrada e saída não é recomendado e que essa configuração pode causar invocações recursivas, aumento do uso do Lambda e maiores custos.

O Lambda adicionará as permissões necessárias para que AWS S3 seja capaz de invocar a função do Lambda a partir deste acionador. Saiba mais [↗](#) sobre o modelo de permissões do Lambda.

Se sua função realiza a gravação de um novo arquivo é recomendável ter novo bucket só para salvar estes arquivos. Para não entrar em um processo de loop.

# Testando

Após configurar, basta fazer um upload para o bucket S3 e o mesmo acionará a trigger da lambda criada. A lambda será executada e o log ficará gravado no AWS CloudWatch

```
▼ 2024-10-03T20:32:49.478Z {"message": "Upload de arquivo realizado", "bucket_name": "bucket-test-lambda-upload", "file_name": "72000082766_tickets-M...

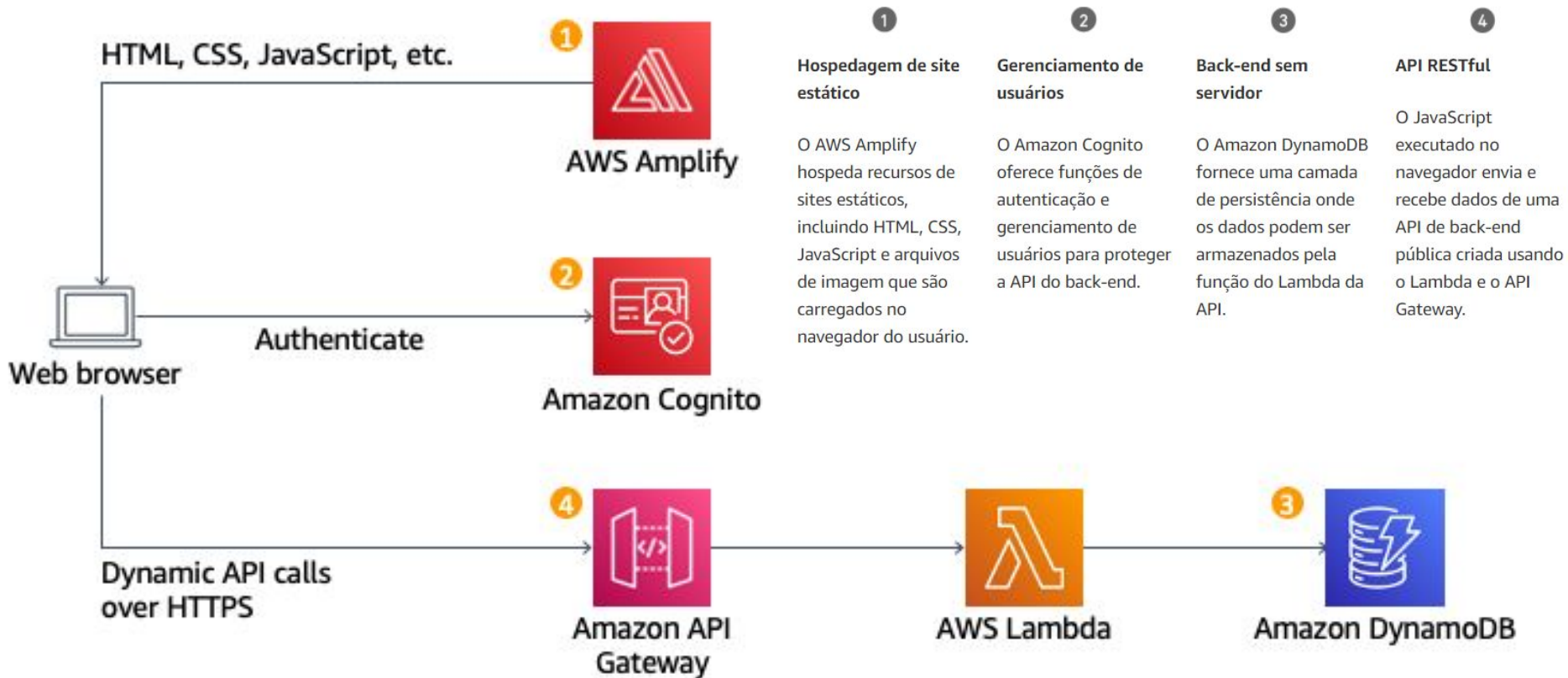
{
  "message": "Upload de arquivo realizado",
  "bucket_name": "bucket-test-lambda-upload",
  "file_name": "72000082766_tickets-May-02-2023-11_48.csv",
  "timestamp": "2024-10-03 20:32:49"
}

▶ 2024-10-03T20:32:49.480Z END RequestId: 5243452c-7e7a-476c-9a35-f2d0f986350b
```

# Arquitetura Serverless AWS

Exemplos de arquitetura Serverless na AWS

# Arquitetura de uma Aplicação Web Serverless na AWS



## Example serverless application architecture

