



# Desmistificando **Websockets** com AWS **Lambda** e API Gateway

Gabriel Prando

10/2024

# Quem é Gabriel Prando?



- ❖ Engenheiro de Plataforma no iFood;
- ❖ Experiências como Full stack, Infra, Backend, DevEx;
- ❖ Engenheiro de Computação (UTFPR-PB);
- ❖ Mestrando em Engenharia Elétrica e Computação (UTFPR-PB);
- ❖ Em qualquer rede como @prandogabriel.

...



# Agenda

- ❖ Websockets;
  - ❖ Serverless;
  - ❖ AWS Lambda;
  - ❖ API Gateway;
  - ❖ Integrando Lambda com Websockets.
- ...

## — Qual o foco de hoje?

- ❖ Websocket;
- ❖ Function as a Service da AWS (FaaS).



+



- Quem aqui já mexeu com websockets?



- Quem aqui já mexeu com websockets?

E com AWS Lambda?



— Quem aqui já mexeu com websockets?

E com AWS Lambda?

E com websockets + Lambda?



## — Disclaimer



## — Disclaimer

Nunca trabalhei com websocket em ambiente produtivo...

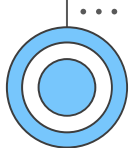




# WebSocket

O que é e como funciona?





# Websocket

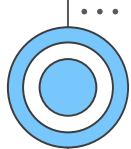
- ❖ RFC 6455 - Internet Engineering Task Force (IETF) e W3C standard (2011);
- ❖ Permite abrir uma sessão de comunicação interativa; bidirecional entre o navegador do usuário e um servidor;
- ❖ Conexão persistente;
- ❖ Início com HTTP e em seguida Websocket.



...



...



# Casos de uso

- ❖ Atualização em tempo real;
- ❖ Chats;
- ❖ Notificações;
- ❖ Jogos online;
- ❖ Aplicações colaborativas.

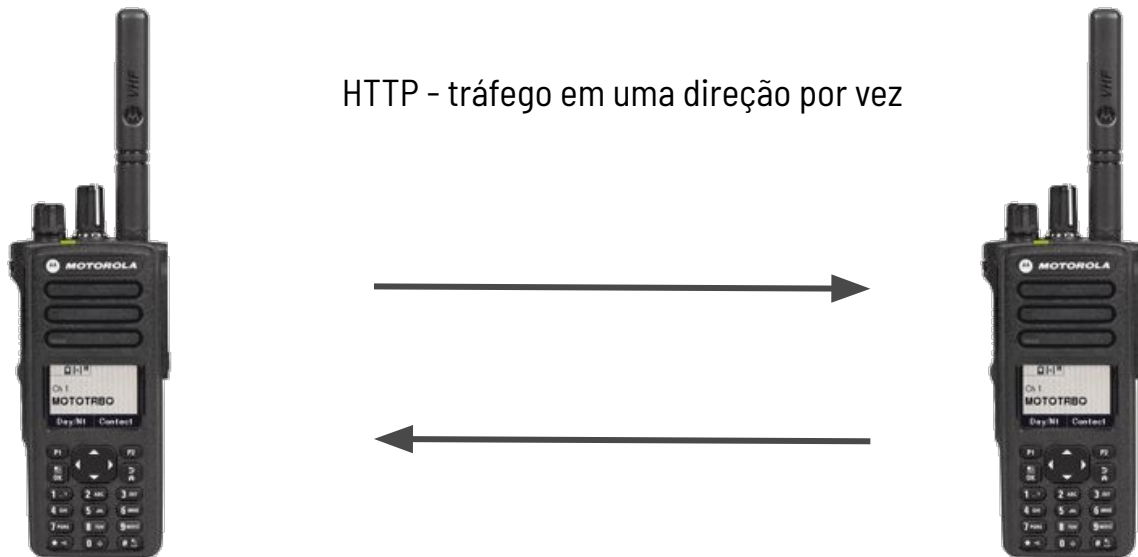


...



...

## — HTTP vs WebSocket



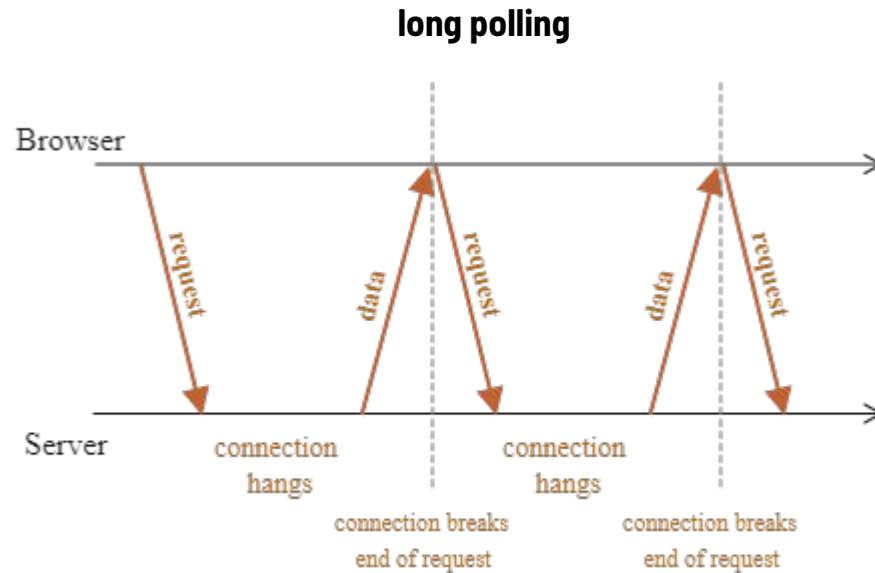
## — HTTP vs WebSocket



WebSocket - tráfego bidirecional

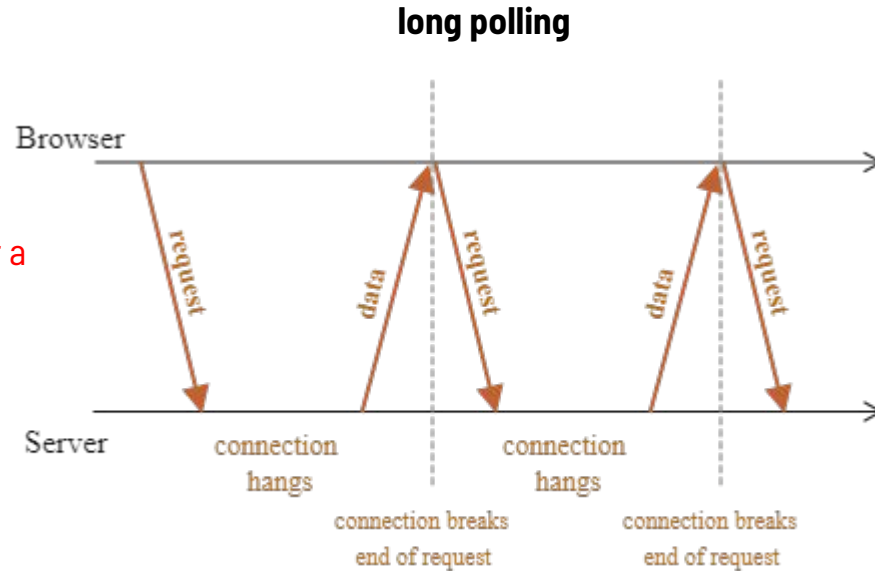


# Estratégias utilizadas antes do websocket



# Estratégias utilizadas antes do websocket

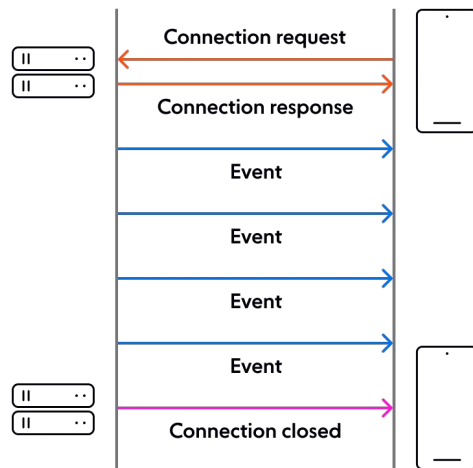
- ❖ Alto consumo recursos;
- ❖ Latência;
- ❖ Múltiplas conexões (reabrir a todo momento);
- ❖ ...





# Estratégias utilizadas antes do websocket

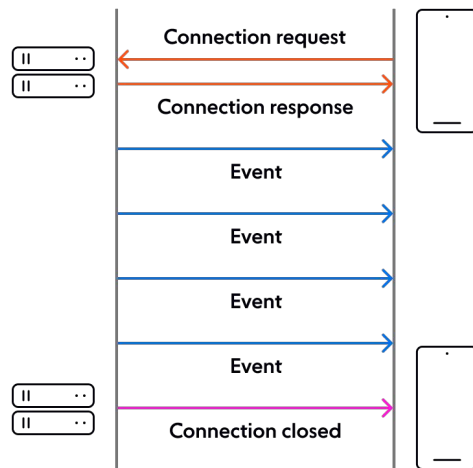
## Server-Sent Events



# Estratégias utilizadas antes do websocket

- ❖ Comunicação unidirecional;
- ❖ Compatibilidade reduzida (browsers antigos sem suporte);
- ❖ ...

## Server-Sent Events



## — HTTP vs WebSocket

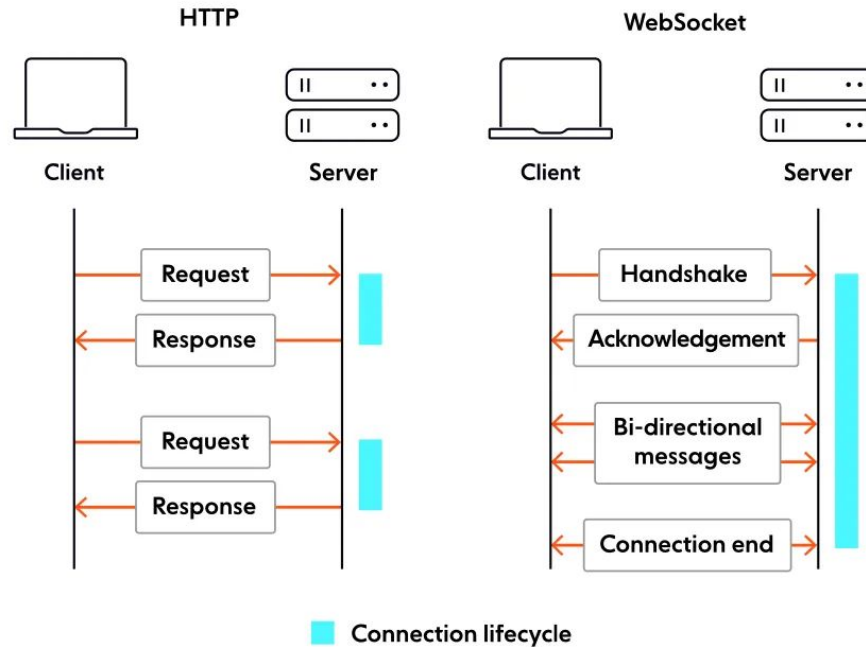


Figure 1.3: WebSockets vs. the traditional HTTP request/response model

Fonte: <https://websocket.org/>

# Handshake

- ❖ HTTP request;
- ❖ Solicitação de upgrade para WS.

## Request

```
GET /chat HTTP/1.1
Host: example.com:8000
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Version: 13
```

# Handshake

- ❖ Recebe sucesso para mudar para WS;
- ❖ Sec-WebSocket-Accept gerado a partir da chave  
258EAF5E-E914-47DA-95CA-C5AB0DC85B11 e Web Socket Key enviada no handshake.

## Response

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+x0o=
```



***Vamos deixar websocket de lado por um momento...***

# Serverless

**O que é serverless  
computing? (computação  
sem servidor)**





## — Ora ora...

'Sem servidor' descreve a experiência. Os servidores são invisíveis **para o desenvolvedor**, que não os vê, gerencia ou interage, mas eles **existem sim**.



# AWS Lambda

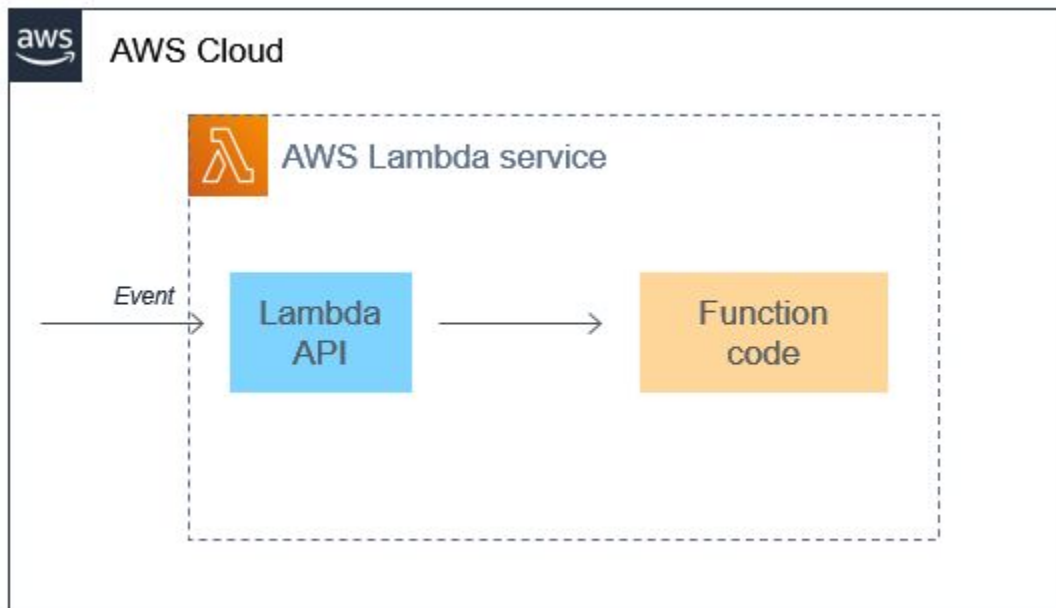


- ❖ Sem servidor para gerenciar;
- ❖ Escala com o uso;
- ❖ Orientado a eventos / triggers;
- ❖ Agnóstico a tecnologia;
- ❖ Mais tempo para pensar;
- ❖ Maior agilidade.

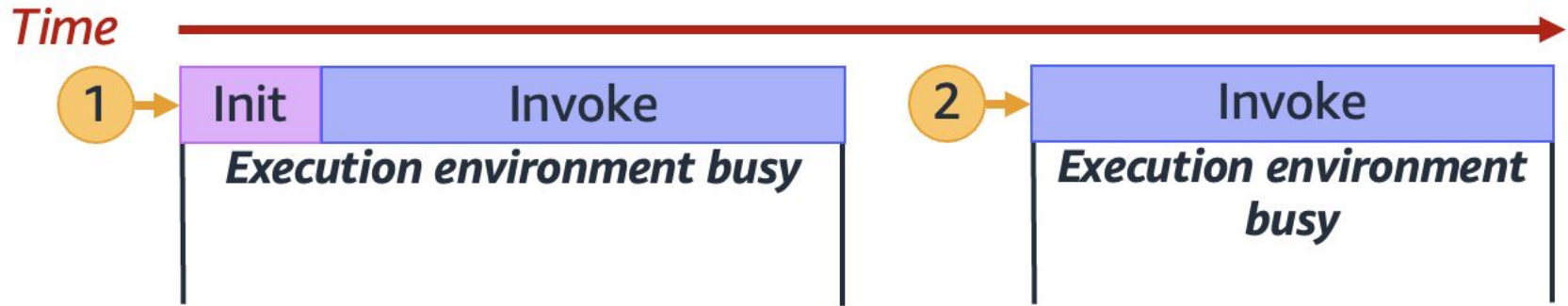
# Casos de uso

- ❖ Rest APIs
- ❖ Processamento assíncrono
- ❖ Processamento de dados (ETL)
- ❖ Startups, devido a economia de tempo, foco desenvolvimento, fácil desenvolvimento e manutenção, (Serverless first).

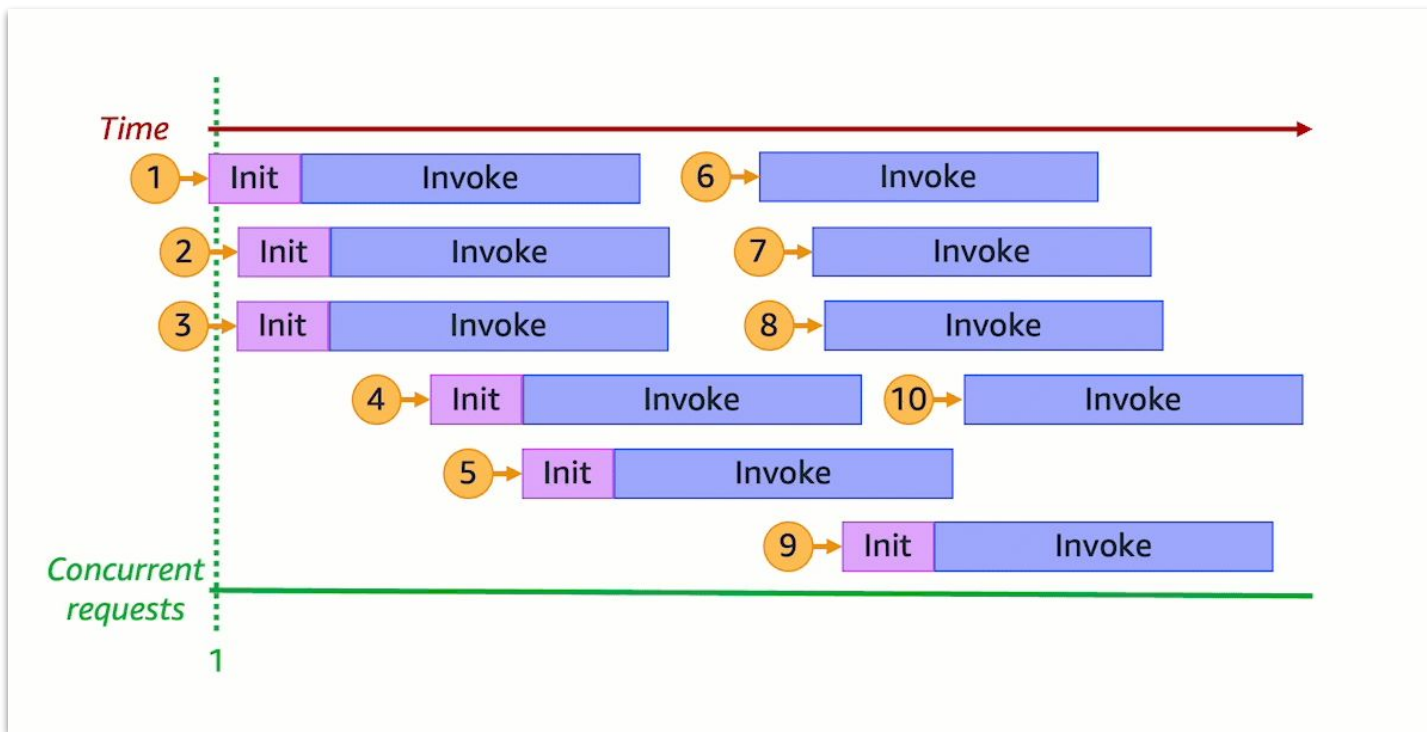
## — Como uma lambda é chamada?



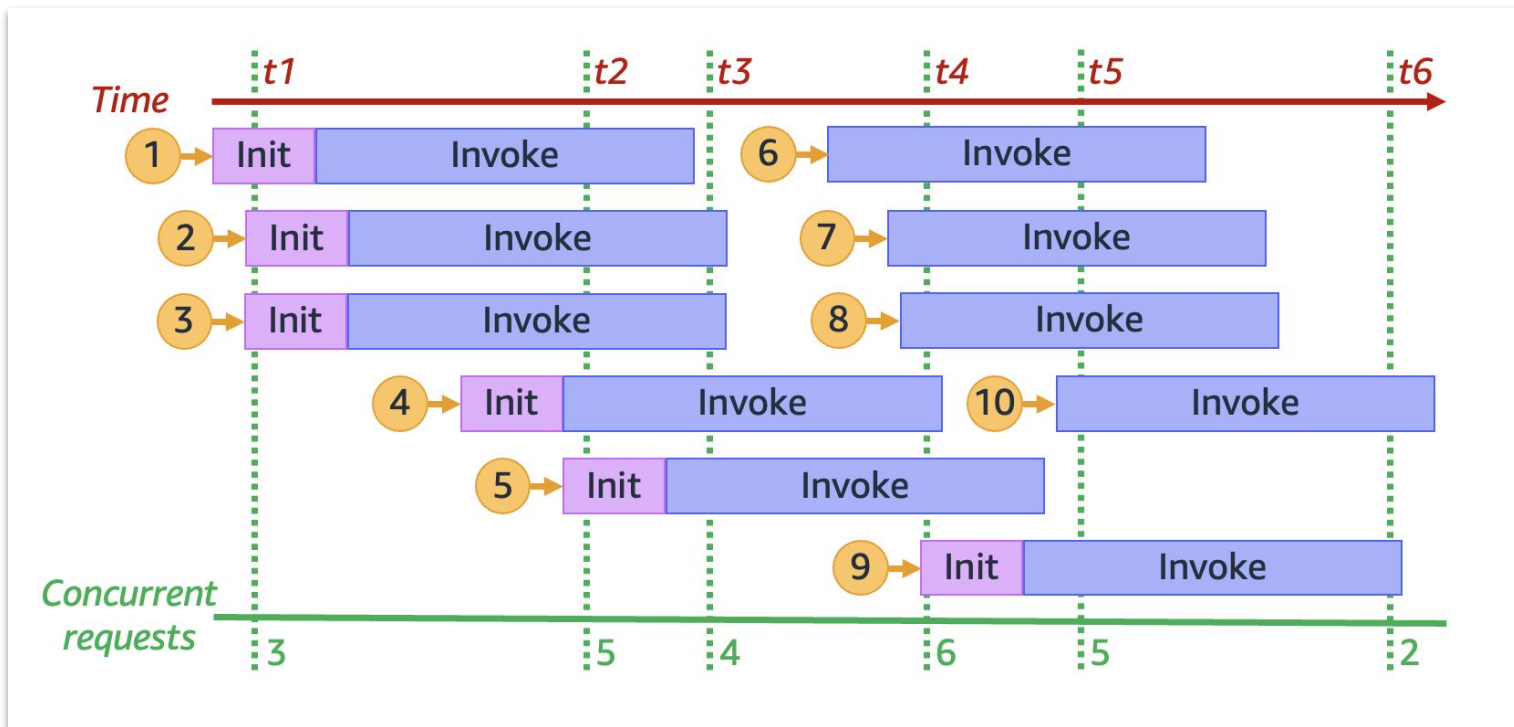
## — Como functions escalam?

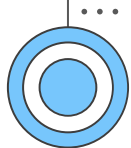


## — Como functions escalam?



## — Como functions escalam?

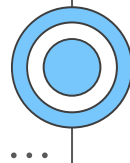
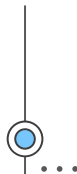
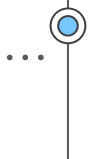




# Como lambdas escalam?

❖ Ponto de atenção:

➤ Limite de 1000 instâncias simultâneas, por default:





# Como lambdas escalam?

❖ Ponto de atenção:

➤ Limite de 1000 instâncias simultâneas, por default:

■  $1000\text{req} * 200\text{ms/req} * 60\text{s} * 60\text{min} * 24\text{h};$

- 5k req/s;
- 300.000 req/min;
- 18.000.000 req/hora;
- 432.000.000 req/dia;
- 12.960.000.000 req/mês.

# Como lambdas escalam?

## ❖ Ponto de atenção:

### ➤ Limite de 1000 instâncias simultâneas, por default:

■  $1000\text{req} * 200\text{ms/req} * 60\text{s} * 60\text{min} * 24\text{h};$

- 5k req/s;
- 300.000 req/min;
- 18.000.000 req/hora;
- 432.000.000 req/dia;
- 12.960.000.000 req/mês.

Acho que da pra começar  
uma startup com isso...

# E quanto custaria isso?

## Unit conversions

Amount of memory allocated:  $256 \text{ MB} \times 0.0009765625 \text{ GB in a MB} = 0.25 \text{ GB}$

Amount of ephemeral storage allocated:  $512 \text{ MB} \times 0.0009765625 \text{ GB in a MB} = 0.5 \text{ GB}$

## Pricing calculations

$12,960,000,000 \text{ requests} \times 200 \text{ ms} \times 0.001 \text{ ms to sec conversion factor} = 2,592,000,000.00 \text{ total compute (seconds)}$

$0.25 \text{ GB} \times 2,592,000,000.00 \text{ seconds} = 648,000,000.00 \text{ total compute (GB-s)}$

$648,000,000.00 \text{ GB-s} - 400,000 \text{ free tier GB-s} = 647,600,000.00 \text{ GB-s}$

$\text{Max}(647,600,000.00 \text{ GB-s}, 0) = 647,600,000.00 \text{ total billable GB-s}$

Tiered price for:  $647,600,000.00 \text{ GB-s}$

$647,600,000 \text{ GB-s} \times 0.0000166667 \text{ USD} = 10,793.35 \text{ USD}$

Total tier cost =  $10,793.3549 \text{ USD}$  (monthly compute charges)

**Monthly compute charges: 10,793.35 USD**

$12,960,000,000 \text{ requests} - 10,000,000 \text{ free tier requests} = 12,950,000,000 \text{ monthly billable requests}$

$\text{Max}(12,950,000,000 \text{ monthly billable requests}, 0) = 12,950,000,000.00 \text{ total monthly billable requests}$

$12,950,000,000.00 \text{ total monthly billable requests} \times 0.0000002 \text{ USD} = 2,591.80 \text{ USD}$  (monthly request charges)

**Monthly request charges: 2,591.80 USD**

$0.50 \text{ GB} - 0.5 \text{ GB (no additional charge)} = 0.00 \text{ GB billable ephemeral storage per function}$

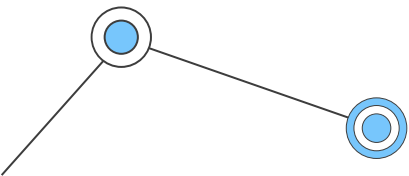
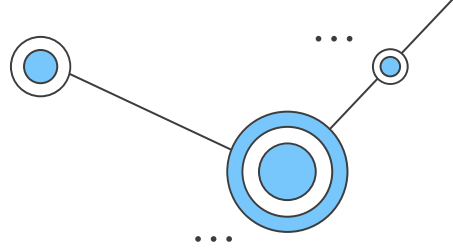
**Monthly ephemeral storage charges: 0 USD**

$10,793.35 \text{ USD} + 2,591.80 \text{ USD} = 13,385.15 \text{ USD}$

**Lambda costs - With Free Tier (monthly): 13,385.15 USD**

# Vamos montar um cenário

- ❖ Estamos em um contexto onde:
  - Precisamos de atualização em tempo real;
  - Não temos uma arquitetura de containers/VMs estabelecida;
  - Possuímos um ecossistema em serverless.

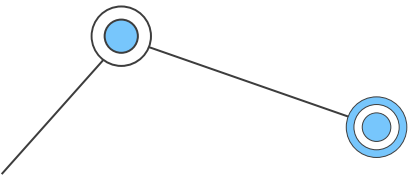
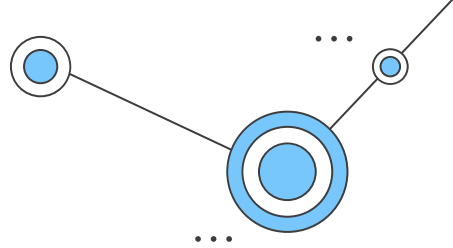


- Seria possível utilizar lambda para isso?



# WebSocket no AWS API Gateway

- ❖ Serviço para criar, publicar, manter, monitorar e proteger APIs em qualquer escala;
- ❖ API Gateway gerencia as conexões WS;
- ❖ Trabalha com rotas:
  - \$connect, \$disconnect, \$default, custom...
- ❖ Integra com lambda invocando sob demanda.



## — Rota connect

- ❖ Gera connectionId;

```
{
  "requestContext": {
    "routeKey": "$connect",
    "eventType": "CONNECT",
    "extendedRequestId": "ZJcXjH0joAMFJwA=",
    "requestTime": "12/Sep/2024:22:55:38 +0000",
    "messageDirection": "IN",
    "stage": "dev",
    "connectedAt": 1609459200000,
    "connectionId": "ZJcXjH0joAMFJwA=",
    "domainName": "example.com",
    "apiId": "abc123"
  },
  "headers": {
    "Host": "example.com",
    "Connection": "Upgrade",
    "Upgrade": "websocket",
    "Sec-WebSocket-Key": "dGhlIHhXbXZsBub25jZQ==",
    "Sec-WebSocket-Version": "13"
  }
}
```

## — Rota disconnect

- ❖ Info importante para limpar bases ou fazer lógicas.

```
{
  "requestContext": {
    "routeKey": "$disconnect",
    "eventType": "DISCONNECT",
    "extendedRequestId": "ZJcXjH0joAMFJwA=",
    "requestTime": "12/Sep/2024:22:55:38 +0000",
    "messageDirection": "IN",
    "stage": "dev",
    "connectedAt": 1609459200000,
    "connectionId": "ZJcXjH0joAMFJwA=",
    "domainName": "example.com",
    "apiId": "abc123"
  },
  "isBase64Encoded": false
}
```

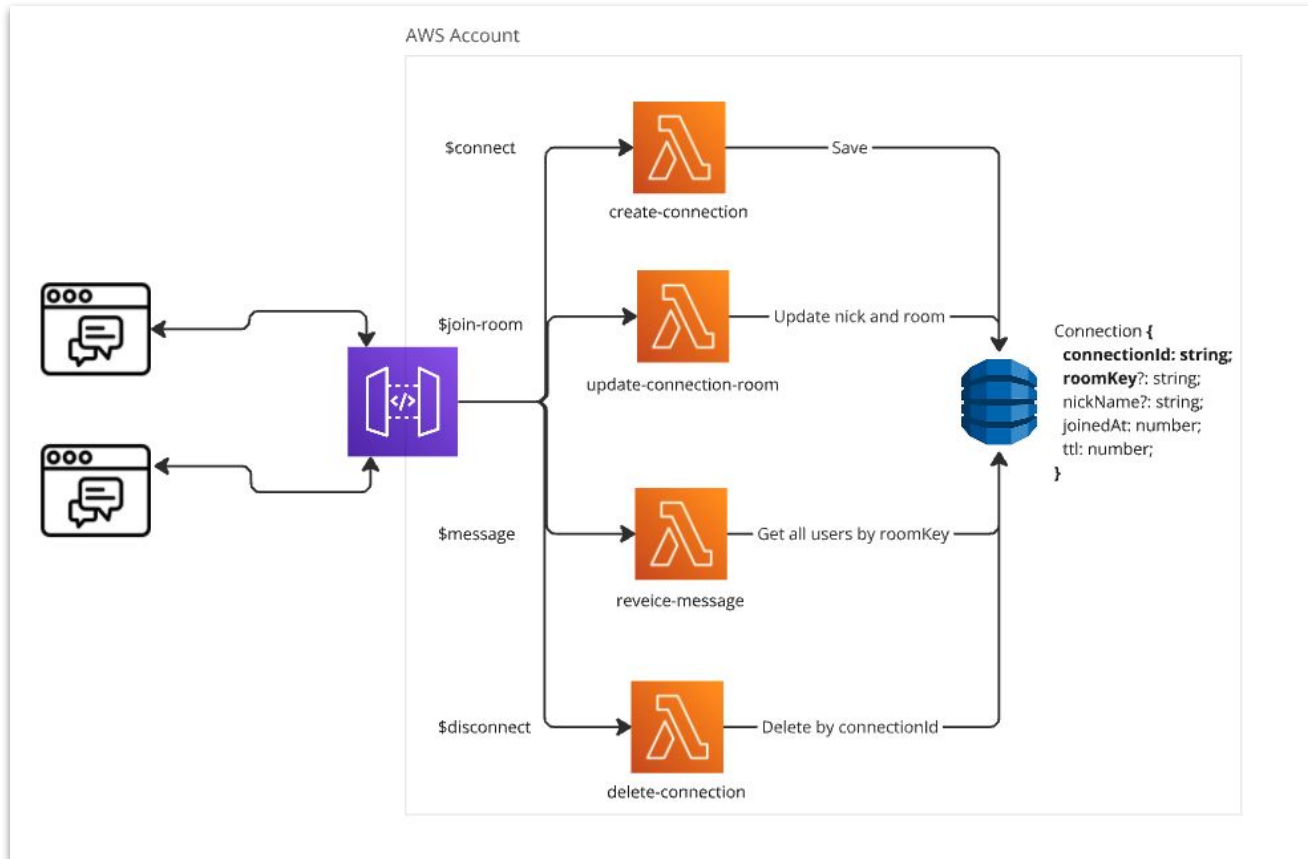


## — Rota custom

- ❖ Customizações;

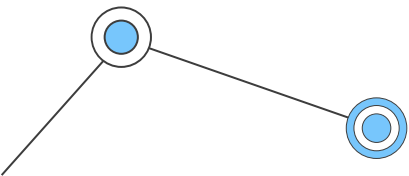
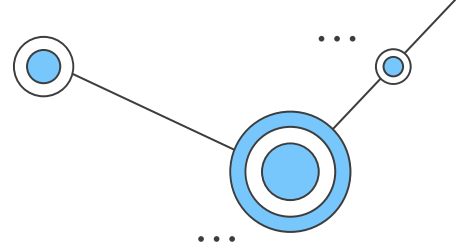
```
{
  "requestContext": {
    "routeKey": "sendMessage",
    "eventType": "MESSAGE",
    "extendedRequestId": "ZJcXjH0joAMFJwA=",
    "requestTime": "12/Sep/2024:22:55:38 +0000",
    "messageDirection": "IN",
    "stage": "dev",
    "connectedAt": 1609459200000,
    "connectionId": "ZJcXjH0joAMFJwA=",
    "domainName": "example.com",
    "apiId": "abc123"
  },
  "body": "{\"action\":\"sendMessage\",\"data\":\"Hello, world!\"}",
  "isBase64Encoded": false
}
```

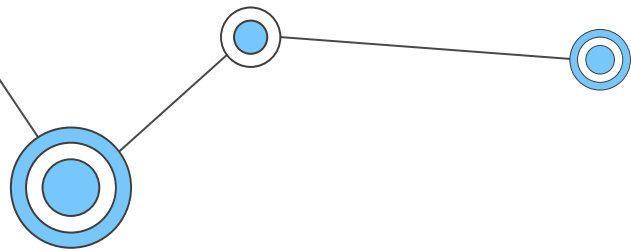
# Construindo um chat



***Será que isso funciona mesmo?***

# Construindo um chat



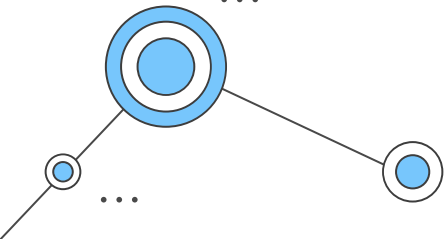


...

Perguntas?

Obrigado!

...



Aponte o celular e conecte-se :)



**SCAN ME!**