

Building a Serverless Cloud Service

John Chang <john@surfncode.io>

Python Porto #10

A Little Context



JSON
REST
AJAX

AWS EC2



2000

2005

2010

2015



A Little Context

PaaS Heroku
Google App Engine
AWS EB

IaaS AWS EC2



2000

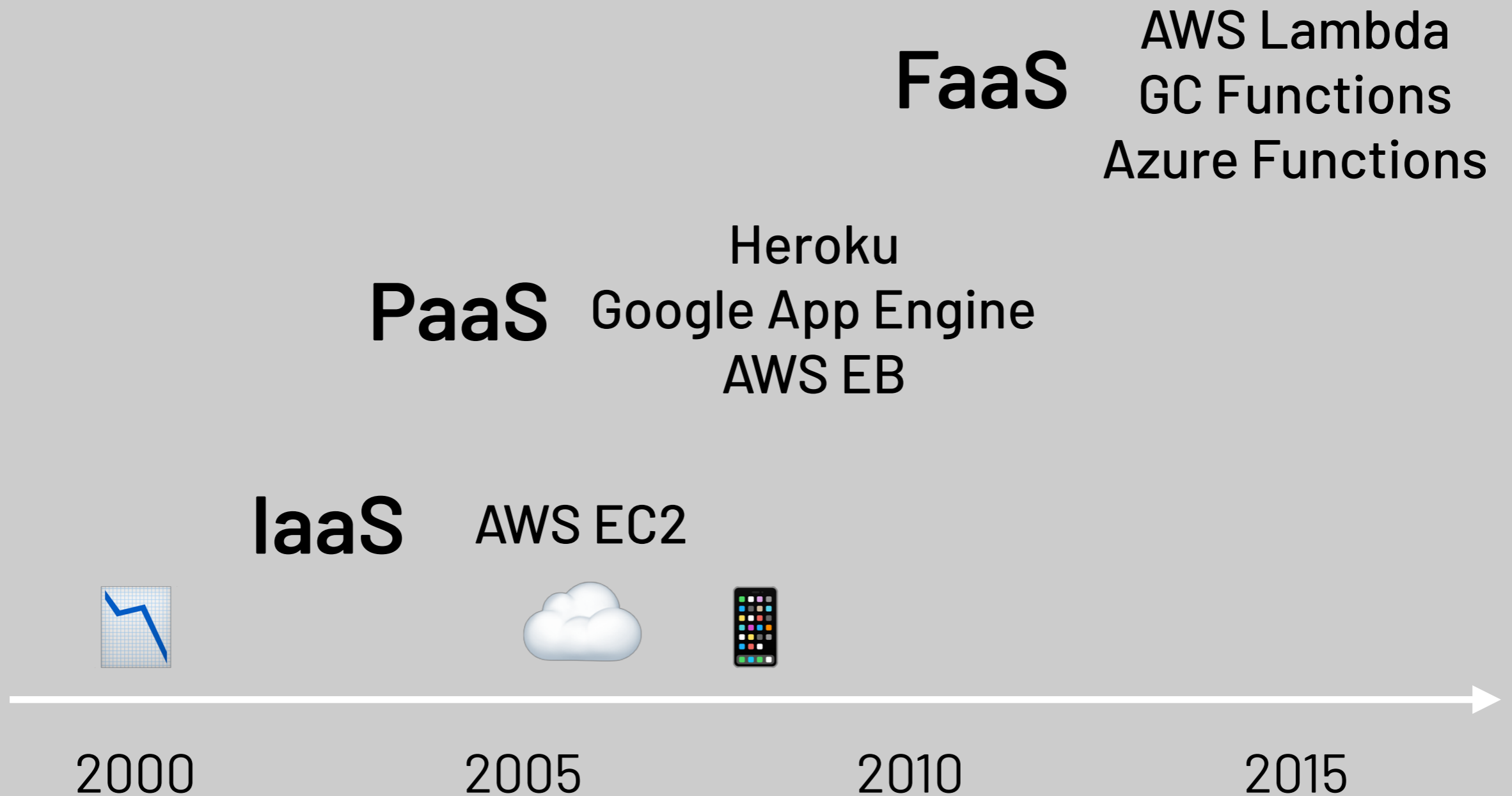
2005

2010

2015



A Little Context



What is Serverless?

FaaS

AWS Lambda
GC Functions
Azure Functions


- Serverless Computation == FaaS
- ... also Serverless Databases
- "Serverless" means transparent scaling

Example

“Hello World” on AWS Lambda


Author from scratch

Start with a simple "hello world" example.



Blueprints

Choose a preconfigured template as a starting point for your Lambda function.



AWS Serverless Application Repository

Find and deploy serverless applications published by AWS, AWS partners, and other developers.



Author from scratch [Info](#)

Name

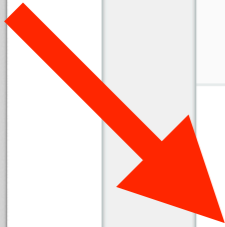
Runtime

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation.
[Learn more](#) about Lambda execution roles.

Existing role

You can use an existing role with this function. Lambda must be able to assume this role, and the role must have



py-porto-... Throttle Qualifiers Actions Select a test event.. Test Save

Function code [Info](#)

Code entry type

Edit code inline

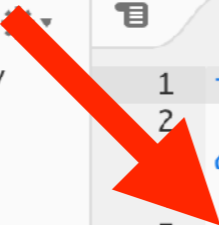
Runtime

Python 3.7

Handler [Info](#)

lambda_function.lambda_ha

```
File Edit Find View Goto Tools Window  
Environment  
py-porto-hello  
lambda_function.py  
lambda_function x  
1 import json  
2  
3 def lambda_handler(event, context):  
4     # TODO implement  
5     return {  
6         'statusCode': 200,  
7         'body': json.dumps('Hello from Lambda!')  
8     }  
9
```



aws Lambda Management Console

Services Resource Groups john @ surfncode Frankfurt Support

py-porto-... Throttle Qualifiers Actions pyportodemo Test Save

configuration. Choose Test to input a test event when you want to test your function.

Execution result: succeeded (logs)

Details

The section below shows the result returned by your function execution.

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

Summary

Code SHA-256	Request ID
ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08ml=	c7846e6a-ed7d-11e8-ad6b-c90b8c87d5fa
Duration	Billed duration
7.74 ms	100 ms
Resources configured	Max memory used
128 MB	20 MB

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

So What?



- Zero admin
- Truly transparent scaling
- No vendor lock-in
- Lower costs

FaaS Pricing

- Billed by 1) GB-second rate + 2) Cost per request

Summary	
Code SHA-256 ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08ml=	Request ID c7846e6a-ed7d-11e8-ad6b-c90b8c87d5fa
Duration 7.74 ms	Billed duration 100 ms
Resources configured 128 MB	Max memory used 20 MB

- $128 \text{ MB} \times (1 \text{ GB}/1024 \text{ MB}) \times 100 \text{ ms} \times (1 \text{ s}/1000 \text{ ms}) = 0.0125 \text{ GB-s}$
- 400,000 GB-s, 1M requests free per month

So What?



- Zero admin
- Transparent scaling
- No vendor lock-in
- Lower costs

- More abstraction
- More systems complexity
- Latency
- Vendor lock-in (sort of...)

FaaS Platforms

AWS Lambda

Supported
Python 2.7, 3.6, 3.7



**Google Cloud
Functions**

Beta
Python 3.7.1



Azure Functions

Experimental (1.x),
N/A (2.x)

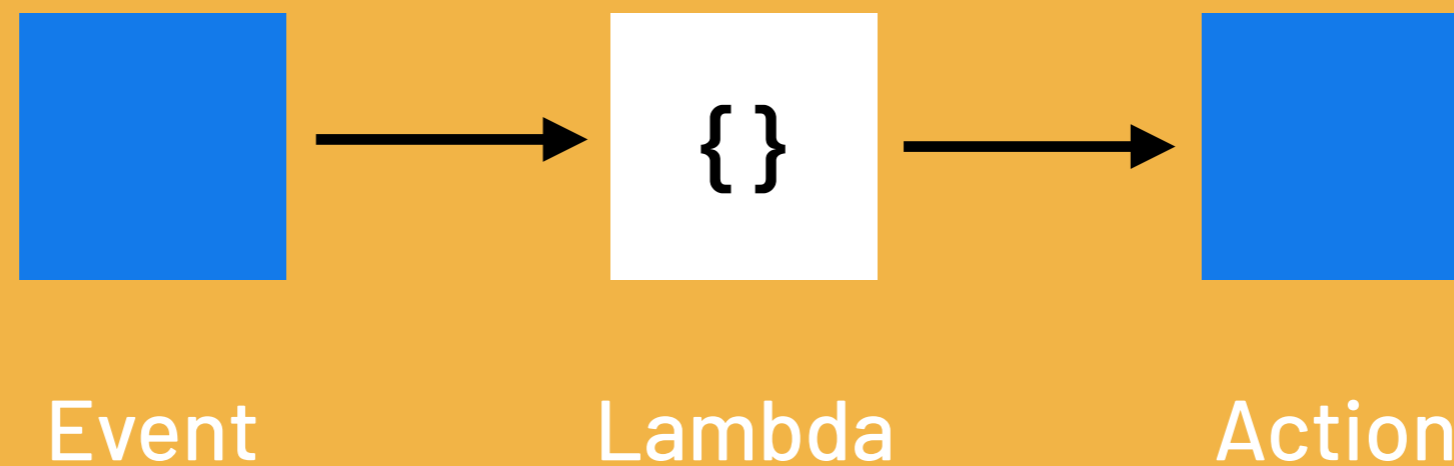


FaaS Use Cases

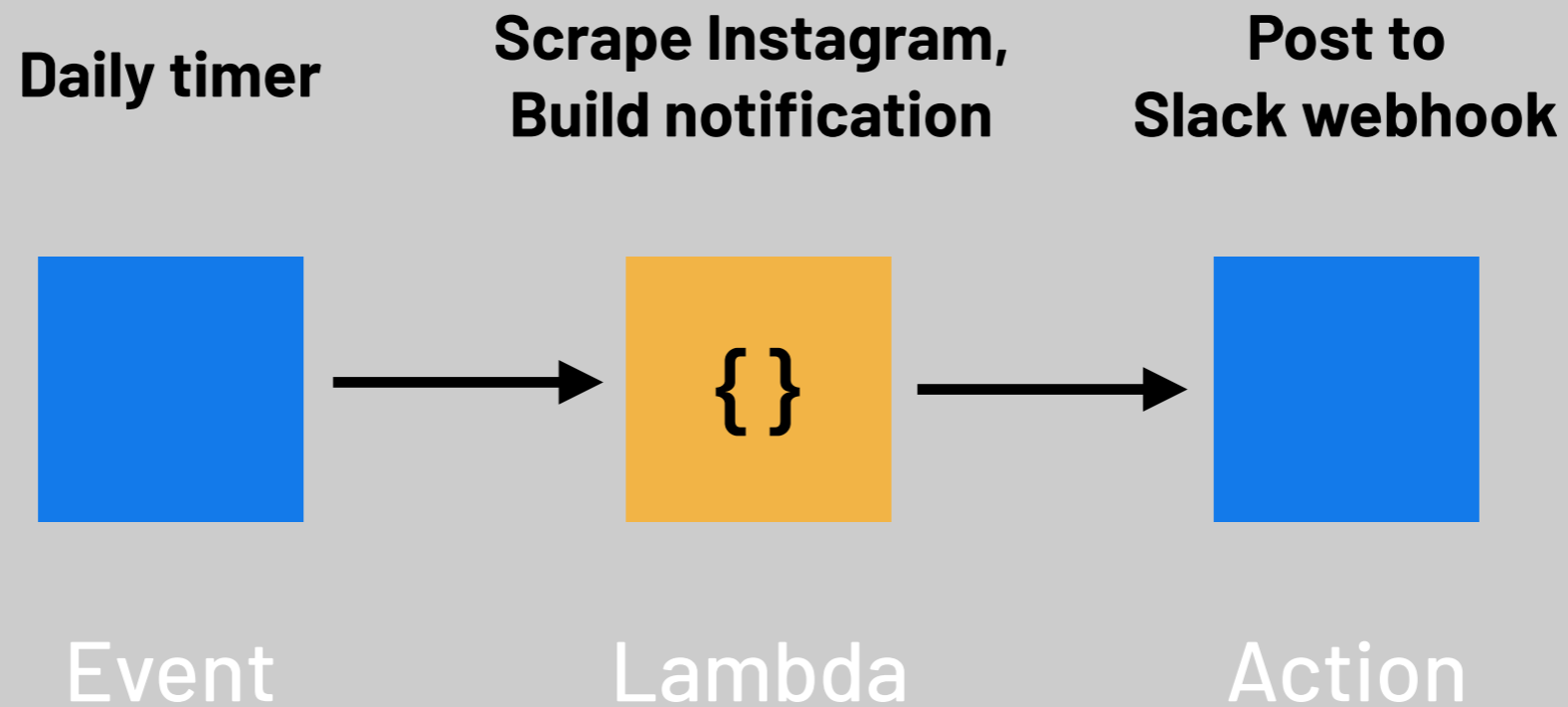
- Event-oriented, short-lived (minutes) execution
 1. Trigger
 2. Data pipeline
 3. API service

Trigger

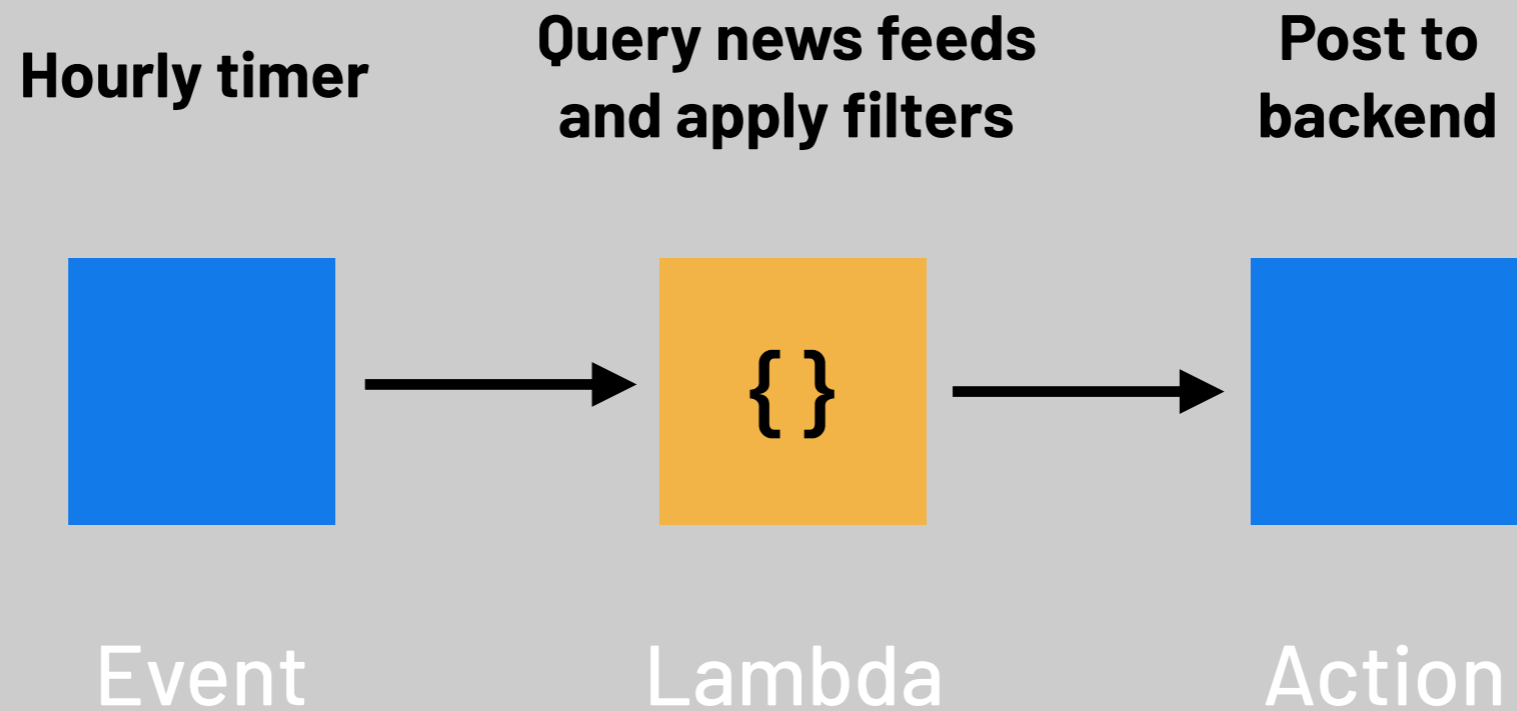
- On-demand or scheduled tasks



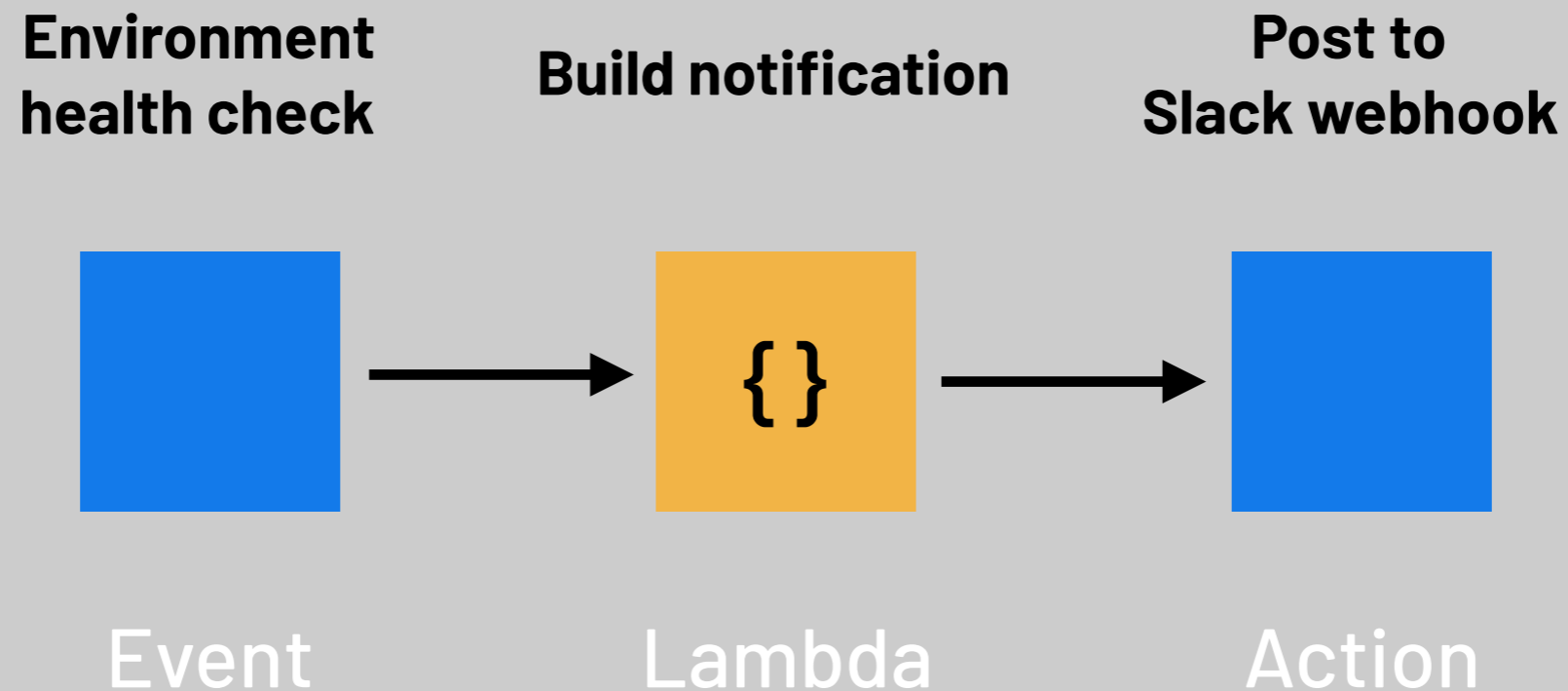
Ex: Lunchbot



Ex: Newsbot

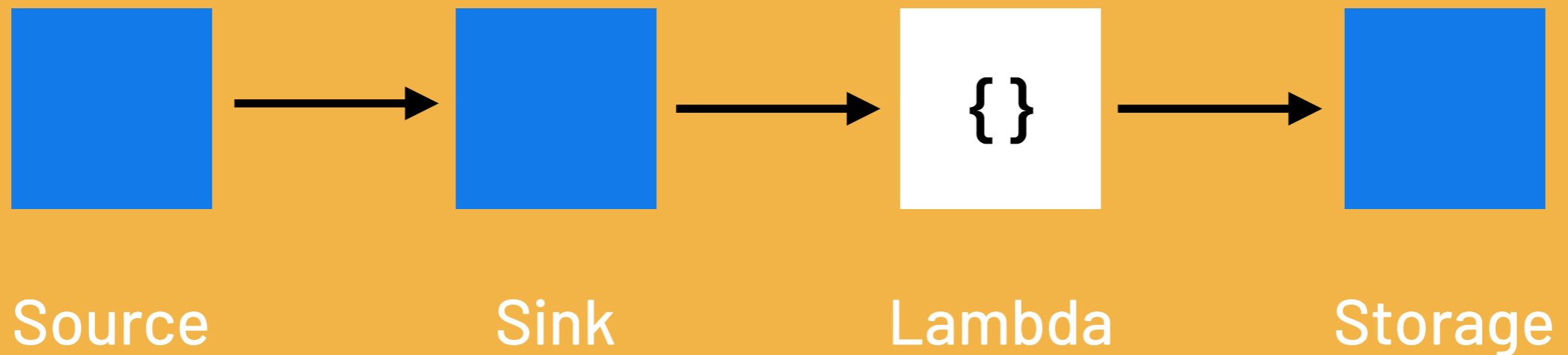


Ex: Server Health Alarms



Data Pipeline

- Real-time log/file/media processing



- e.g. resize and compress image upon upload

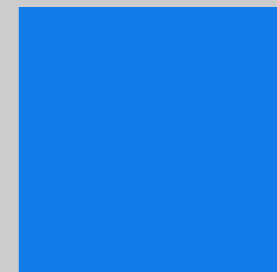
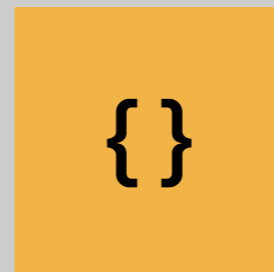
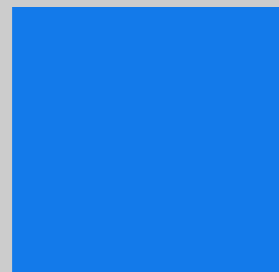
Ex: Server Log Analytics

Web server
access logs

Post to
Kinesis Firehose

Parse user agent
strings

Write to
S3



Source

Sink

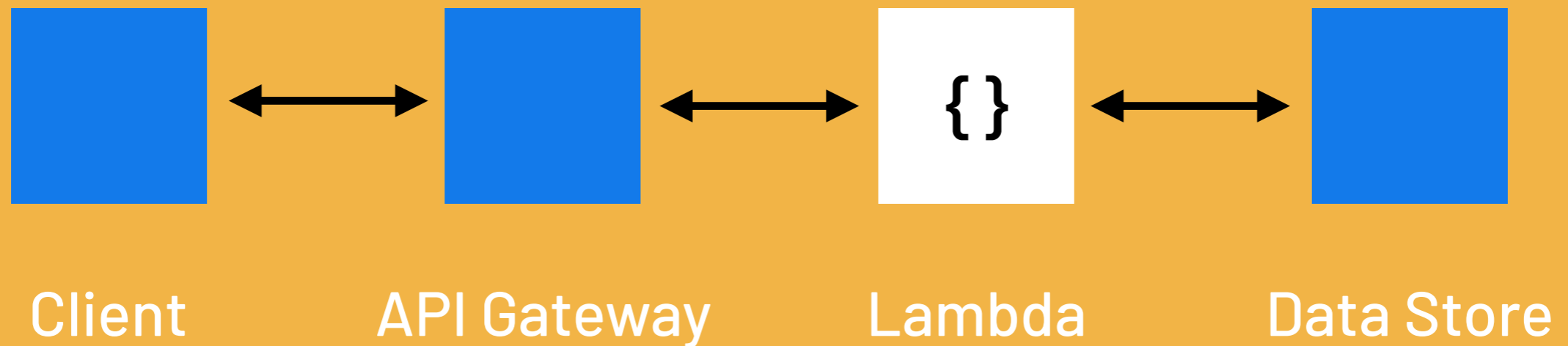
Lambda

Storage

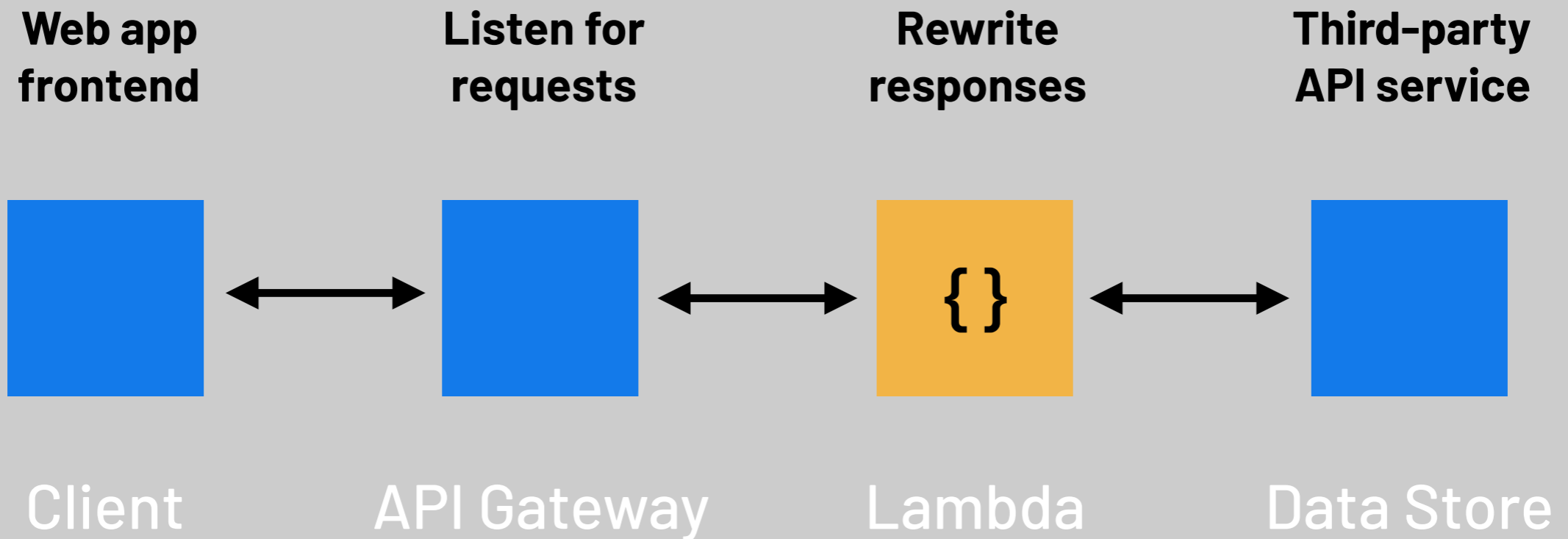
- Can be used with AWS Athena ("serverless" SQL)

API Service

- Microservices, Proxy backends



Ex: Proxy Backend



Example

SURFnCODE Lunchbot



mafaldasmercado • Follow
Mafalda's

mafaldasmercado Bom dia ❤️ hoje poderá optar por:

- Sumo Natural de Manga e Laranja;
- Creme de Couve-flor, Pêra, Mel e Parmesão;
- Empadão de Pescada e Camarão com Maionese de Coentros;
- Linguine de Cogumelos e Bacon;
- Tarte de Batata Doce e Hummus;
- Tarte de Amêndoa;
- Bolo Mars;

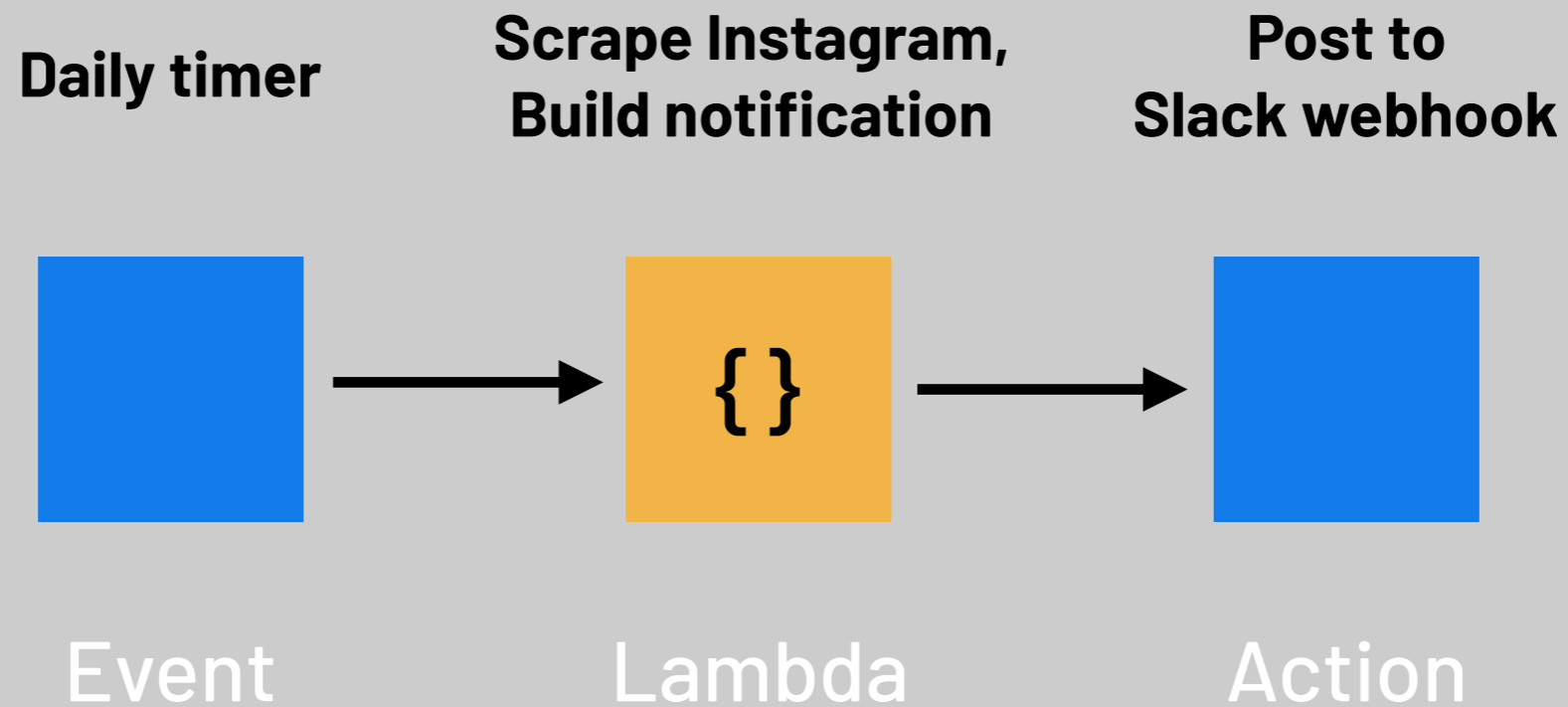
Até já! Faça a sua reserva 💕

mafaldasmercado #pornfood #sharefood #gourmetfood #foodstories #food #burguer #instafood #foodlike #foodpics #mercadomatosinhos #marketfood #mercado #matosinhos #realfood #foodlover #nowporto #p3top



Log in to like or comment.

Ex: Lunchbot



- Project
- py-porto-demo
 - beautifulsoup4-4.6.3.dist-info
 - bin
 - bs4
 - certifi
 - certifi-2018.10.15.dist-info
 - chardet
 - chardet-3.0.4.dist-info
 - idna
 - idna-2.7.dist-info
 - requests
 - requests-2.20.1.dist-info
 - urllib3
 - urllib3-1.24.1.dist-info
 - lambda_function.py
 - lunchbot.py

```
1 import json
2
3 import lunchbot
4
5
6
7 def lambda_handler(event, context):
8     lunchbot.mafaldas_slack_notify()
9
10    return {
11        'statusCode': 200,
12        'body': json.dumps('Hello from Lambda!')}
13
```



```
Project
├── py-porto-demo
│   ├── beautifulsoup4-4.6.3.dist-info
│   ├── bin
│   ├── bs4
│   ├── certifi
│   ├── certifi-2018.10.15.dist-info
│   ├── chardet
│   ├── chardet-3.0.4.dist-info
│   ├── idna
│   ├── idna-2.7.dist-info
│   ├── requests
│   ├── requests-2.20.1.dist-info
│   ├── urllib3
│   ├── urllib3-1.24.1.dist-info
│   ├── lambda_function.py
│   └── lunchbot.py
└── lunchbot.py

1 import datetime
2 import json
3 import re
4
5 from bs4 import BeautifulSoup
6 import requests
7
8
9 def mafaldas_slack_notify():
10     # Fetch Instagram HTML
11     headers = {'User-Agent': 'Lunchbot/0.0 (+mailto:hello@surfnocode.io)'}
12     url = 'https://www.instagram.com/mafaldasmercado/'
13
14     r = requests.get(url, headers=headers)
15
16     r.raise_for_status()
17
18     # Parse Instagram HTML
19     soup = BeautifulSoup(r.text, 'html.parser')
20
21     script_tag = soup.find('script', string=re.compile('window._sharedData'))
22     script_string = script_tag.string
23     m = re.match('window._sharedData = (.+);', script_tag.string)
24     data_json = m.group(1)
25     obj = json.loads(data_json)
26     edges = obj['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges']
27
28     # Examine the first post
29     node = edges[0]['node']
30
31     # Make sure it's from today
32     taken_at_dt = datetime.datetime.fromtimestamp(node['taken_at_timestamp'])
33     if taken_at_dt.date() == datetime.datetime.today().date():
34         caption_text = node['edge_media_to_caption']['edges'][0]['node']['text']
35
36     # Construct a Slack message
37     slack_message_dict = {
38         "attachments": [
39             {
40                 "pretext": "Getting hungry? Here's the latest from Mafalda's.",
41                 "title": "Mafalda's on Instagram",
42                 "title_link": 'https://www.instagram.com/p/{}/'.format(node['shortcode']),
43                 "text": caption_text,
44                 "thumb_url": node['thumbnail_src']
```

```
Project
├── py-porto-demo
│   ├── beautifulsoup4-4.6.3.dist-info
│   ├── bin
│   ├── bs4
│   ├── certifi
│   ├── certifi-2018.10.15.dist-info
│   ├── chardet
│   ├── chardet-3.0.4.dist-info
│   ├── idna
│   ├── idna-2.7.dist-info
│   ├── requests
│   ├── requests-2.20.1.dist-info
│   ├── urllib3
│   ├── urllib3-1.24.1.dist-info
│   ├── lambda_function.py
│   └── lunchbot.py
└── lunchbot.py

1 import datetime
2 import json
3 import re
4
5 from bs4 import BeautifulSoup
6 import requests
7
8
9 def mafaldas_slack_notify():
10     # Fetch Instagram HTML
11     headers = {'User-Agent': 'Lunchbot/0.0 (+mailto:hello@surfncode.io)'}
12     url = 'https://www.instagram.com/mafaldasmercado/'
13
14     r = requests.get(url, headers=headers)
15
16     r.raise_for_status()
17
18     # Parse Instagram HTML
19     soup = BeautifulSoup(r.text, 'html.parser')
20
21     script_tag = soup.find('script', string=re.compile('window._sharedData'))
22     script_string = script_tag.string
23     m = re.match('window._sharedData = (.+);', script_tag.string)
24     data_json = m.group(1)
25     obj = json.loads(data_json)
26     edges = obj['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']['edges']
27
28     # Examine the first post
29     node = edges[0]['node']
30
31     # Make sure it's from today
32     taken_at_dt = datetime.datetime.fromtimestamp(node['taken_at_timestamp'])
33     if taken_at_dt.date() == datetime.datetime.today().date():
34         caption_text = node['edge_media_to_caption']['edges'][0]['node']['text']
35
36     # Construct a Slack message
37     slack_message_dict = {
38         "attachments": [
39             {
40                 "pretext": "Getting hungry? Here's the latest from Mafalda's.",
41                 "title": "Mafalda's on Instagram",
42                 "title_link": 'https://www.instagram.com/p/{}/'.format(node['shortcode']),
43                 "text": caption_text,
44                 "thumb_url": node['edge_media_to_image_media']['edges'][0]['node']['image_versions2']['images'][0]['url']
45             }
46         ]
47     }
```


- Overview
- Authentication
- Login Permissions
- Permissions Review
- Sandbox Mode
- Secure Requests
- Endpoints
- Embedding
- Mobile Sharing
- Libraries
- Support
- Changelog
- Platform Policy



To continuously improve Instagram users' privacy and security, we are accelerating the deprecation of Instagram API Platform, making the following changes effective immediately. We understand that this may affect your business or services, and we appreciate your support in keeping our platform secure.

These **capabilities** will be disabled immediately (previously set for July 31, 2018 or December 11, 2018 deprecation). The following will be deprecated according to the timeline we **shared previously**:

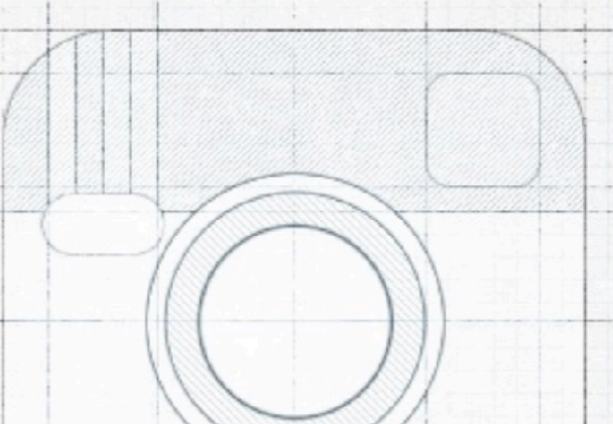
- Public Content - all remaining capabilities to read public media on a user's behalf on December 11, 2018
- Basic - to read a user's own profile info and media in early 2020

For your reference, information on the **new Instagram Graph API**.

Hello Developers.

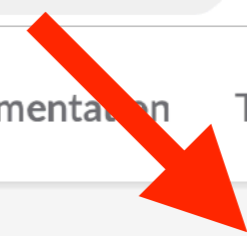
The Instagram API Platform can be used to build non-automated, authentic, high-quality apps and services that:

- Help **individuals share their own content** with 3rd party apps.
- Help **brands and advertisers** understand, manage their audience and



```
Project
├── py-porto-demo
│   ├── beautifulsoup4-4.6.3.dist-info
│   ├── bin
│   ├── bs4
│   ├── certifi
│   ├── certifi-2018.10.15.dist-info
│   ├── chardet
│   ├── chardet-3.0.4.dist-info
│   ├── idna
│   ├── idna-2.7.dist-info
│   ├── requests
│   ├── requests-2.20.1.dist-info
│   ├── urllib3
│   ├── urllib3-1.24.1.dist-info
│   ├── lambda_function.py
│   └── lunchbot.py
└── lunchbot.py

18 # Parse Instagram HTML
19 soup = BeautifulSoup(r.text, 'html.parser')
20
21 script_tag = soup.find('script', string=re.compile('window._sharedData'))
22 script_string = script_tag.string
23 m = re.match('window._sharedData = (.+);', script_tag.string)
24 data_json = m.group(1)
25 obj = json.loads(data_json)
26 edges = obj['entry_data']['ProfilePage'][0]['graphql']['user']['edge_owner_to_timeline_media']
27
28 # Examine the first post
29 node = edges[0]['node']
30
31 # Make sure it's from today
32 taken_at_dt = datetime.datetime.fromtimestamp(node['taken_at_timestamp'])
33 if taken_at_dt.date() == datetime.datetime.today().date():
34     caption_text = node['edge_media_to_caption']['edges'][0]['node']['text']
35
36 # Construct a Slack message
37 slack_message_dict = {
38     "attachments": [
39         {
40             "pretext": "Getting hungry? Here's the latest from Mafalda's.",
41             "title": "Mafalda's on Instagram",
42             "title_link": 'https://www.instagram.com/p/{}/'.format(node['shortcode']),
43             "text": caption_text,
44             "thumb_url": node['thumbnail_src'],
45             "ts": node['taken_at_timestamp']
46         },
47         {
48             "author_name": "✖ Translate to English",
49             "author_link": "https://translate.google.com/#pt/en/{}".format(caption_text),
50         }
51     ]
52 }
53
54 slackbot_url = 'https://hooks.slack.com/services/T7BEXL952/BDZH7QMCM/8s0qtqWtzUXS0qBhGKFZpc2'
55 requests.post(slackbot_url, json=slack_message_dict)
56
57
58 if __name__ == '__main__':
59     mafaldas_slack_notify()
60
```

Create New App

Start here

- Building Slack apps
- Recent updates
- Best practices
- App blueprints

App features

- Internal integrations
- Incoming webhooks
- Slash commands
- Building bots
- Dialogs
- Shared Channels
- Enterprise Grid
- Legacy: Workspace apps

Messages

- Introduction
- Basic formatting
- Attaching content
- Unfurling links

Your Apps

i If your app is (or will be) listed in the Slack App Directory, please review our new [Slack App Directory Agreement](#). These terms are in addition to the existing Developer Policy, API TOS, and Brand Guidelines.

By keeping your app in the App Directory or review process, you're confirming your agreement to the Slack App Directory Agreement and to providing additional information for security review, if requested. If you don't agree with this Agreement, please send an email to feedback@slack.com, and we'll remove your app from the App Directory or the review process.

I Agree

Filter apps by name or workspace

App Name	Workspace	Distribution Status
AWS SNS	SURFnCODE	Not distributed

Settings

- Basic Information
- Collaborators
- Install App
- Manage Distribution

Features

Incoming Webhooks

- Interactive Components
- Slash Commands
- OAuth & Permissions
- Event Subscriptions
- Bot Users
- User ID Translation

- Slack ♥
- Help
- Contact
- Policies
- Our Blog

Activate Incoming Webhooks

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details. You can include [message attachments](#) to display richly-formatted messages.

Each time your app is installed, a new Webhook URL will be generated.

If you deactivate incoming webhooks, new Webhook URLs will not be generated when your app is installed to your team. If you'd like to remove access to existing Webhook URLs, you will need to [Revoke All OAuth Tokens](#).

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}'  
https://hooks.slack.com/services/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
```

Copy



lunch_... Throttle Qualifiers Actions Test Test Save

- ### Add triggers
- Choose a trigger from the list below to add it to your function.
- API Gateway
 - AWS IoT
 - CloudWatch Events
 - CloudWatch Logs
 - CodeCommit
 - Cognito Sync Trigger

lunch_mafaldas_slack_notify

py-porto-demo

Name	Date Modified	Size
▶ beautifulsoup4-4.6.3.dist-info	Today at 12:26	
▶ bin	Today at 12:26	
▶ bs4	Today at 12:26	
▶ certifi	Today at 12:26	
▶ certifi-2018.10.15.dist-info	Today at 12:26	
▶ chardet	Today at 12:26	
▶ chardet-3.0.4.dist-info	Today at 12:26	
▶ env	Today at 12:26	
▶ idna	Today at 12:26	
▶ idna-2.7.dist-info	Today at 12:26	
▶ lambda_function.py	Today at 12:13	200 b
▶ lunchbot.py	Today at 12:18	
▶ requests	Today at 12:26	
▶ requests-2.20.1.dist-info	Today at 12:26	
▶ urllib3	Today at 12:26	
▶ urllib3-1.24.1.dist-info	Today at 12:26	

15 of 16 selected, 267,01 GB available





lunch_...

Throttle

Qualifiers

Actions

Test

Test

Save

Add triggers

Choose a trigger from the list below to add it to your function.

API Gateway

AWS IoT

CloudWatch Events

CloudWatch Logs

CodeCommit

Cognito Sync Trigger



lunch_mafaldas_slack_notify



CloudWatch Events



Amazon Cloud

Add triggers from the list on the left

Resources that the function appears here



Function code [Info](#)

Code entry type

Upload a .zip file

Runtime

Python 3.6

Handler [Info](#)

lambda_function.lambda_

Function package*

Upload

For files larger than 10 MB, consider uploading using Amazon S3.

- CloudWatch
- Dashboards
- Alarms
 - ALARM 0
 - INSUFFICIENT 0
 - OK 3
- Billing
- Events
- Rules**
- Event Buses
- Logs
- Metrics
- Favorites
- + Add a dashboard

Rules > Lunchtime Actions

Summary

ARN `arn:aws:events:eu-central-1:403040752257:rule/Lunchtime`

Schedule Cron expression `0 12 ? * TUE-FRI *`

- Next 10 Trigger Date(s)**
1. Thu, 22 Nov 2018 12:00:00 GMT
 2. Fri, 23 Nov 2018 12:00:00 GMT
 3. Tue, 27 Nov 2018 12:00:00 GMT
 4. Wed, 28 Nov 2018 12:00:00 GMT
 5. Thu, 29 Nov 2018 12:00:00 GMT
 6. Fri, 30 Nov 2018 12:00:00 GMT
 7. Tue, 04 Dec 2018 12:00:00 GMT
 8. Wed, 05 Dec 2018 12:00:00 GMT
 9. Thu, 06 Dec 2018 12:00:00 GMT
 10. Fri, 07 Dec 2018 12:00:00 GMT

Status Enabled

Description

Monitoring [Show metrics for the rule](#)

Targets

Filter: « < Viewing 1 to 1 of 1 Targets > »

Type	Name	Input	Role	Additional parameters
Lambda function	lunch_mafaldas_slack_notify	Matched event		



#porto ▾



AWS SNS APP 12:00

Getting hungry? Here's the latest from Mafalda's.

Mafalda's on Instagram

Bom dia ❤️ hoje poderá optar por:



- Sumo Natural de Manga e Laranja;
- Creme de Couve-flor, Pêra, Mel e Parmesão;
- Empadão de Pescada e Camarão com Maionese de Coentros;
- Linguine de Cogumelos e Bacon;
- Tarte de Batata Doce e Hummus;
- Tarte de Amêndoa;
- Bolo Mars;

Até já! Faça a sua reserva

Message #porto



Send

What is Serverless?

FaaS

AWS Lambda
GC Functions
Azure Functions

- Serverless Computation == FaaS
- ... also Serverless Databases
- "Serverless" means transparent scaling

What is SURFnCODE?

- Startup-that-makes-startups
- Based in Matosinhos



- We're hiring! 
careers.surfncode.io

(even Python newbies)

