



Serverless Python

And a tangent about the secret origins of the Power Rangers

By Shy Ruparel



Shy Ruparel

Developer Evangelist | Contentful

@ShyRuparel

He/Him

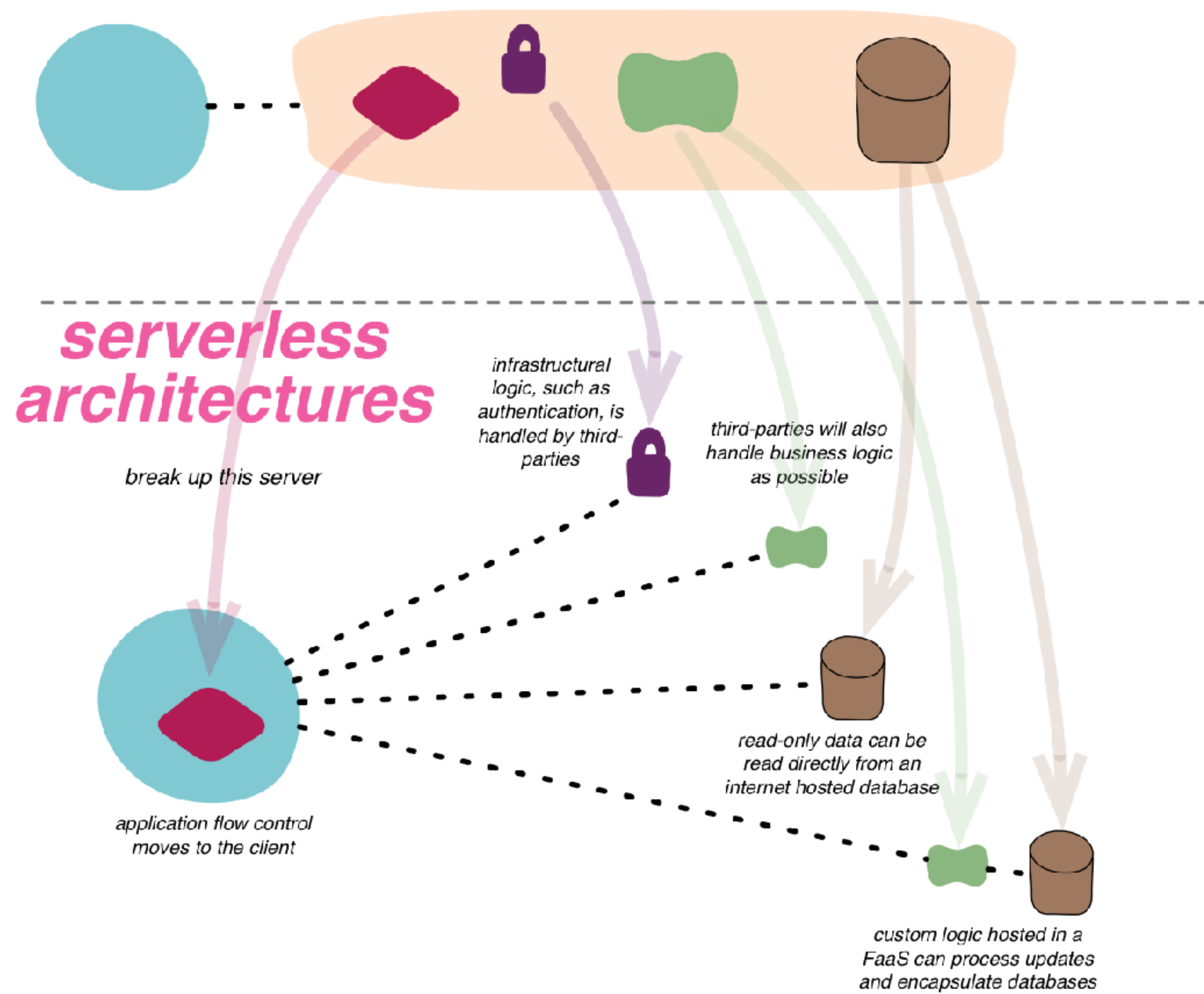
serverless vs Serverless



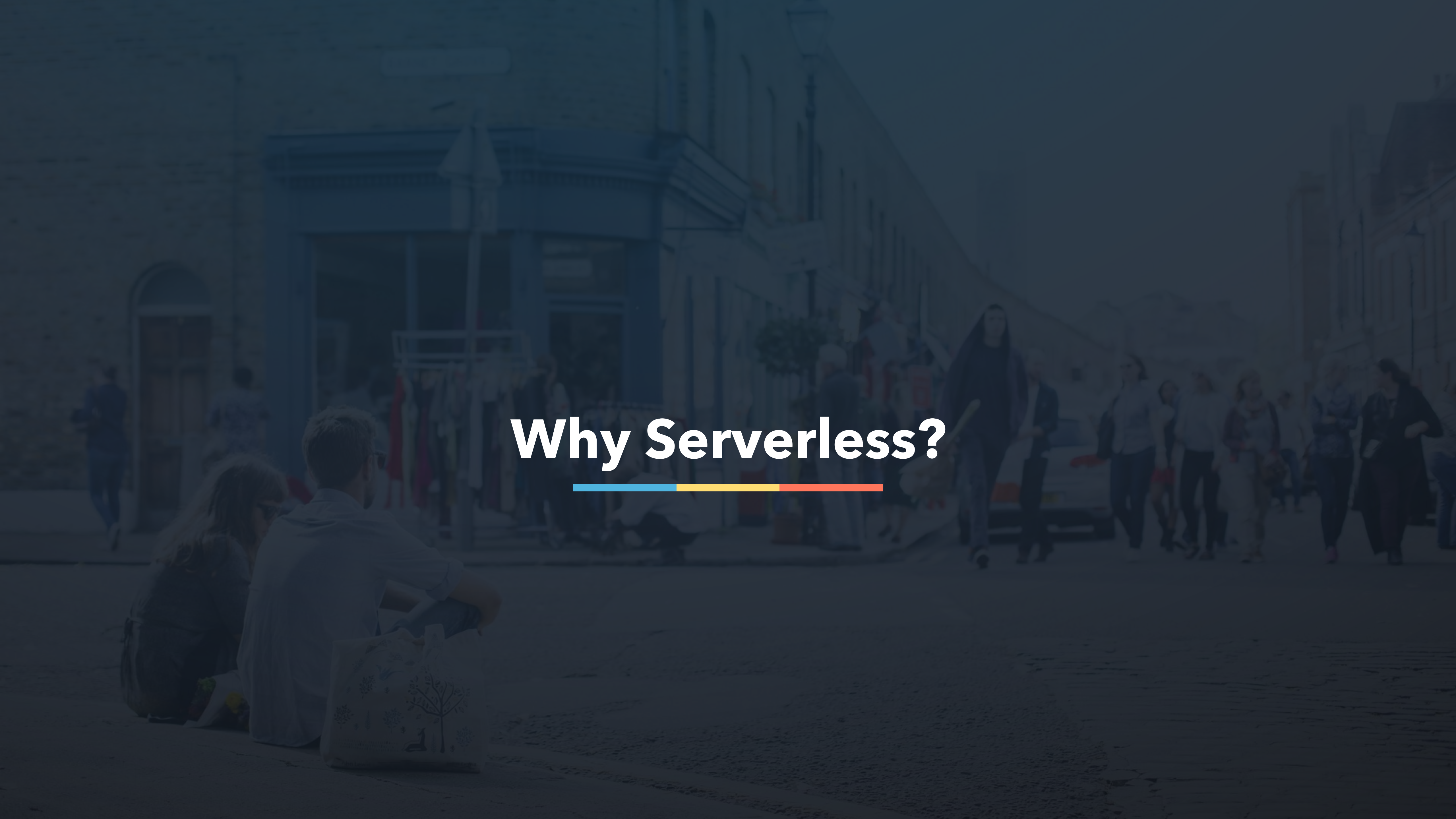
Serverless Architectures



A traditional internet delivered app has a client communicating with a long-lived server process that handles most aspects of the application's logic




Why Serverless?



Pay for What you Use

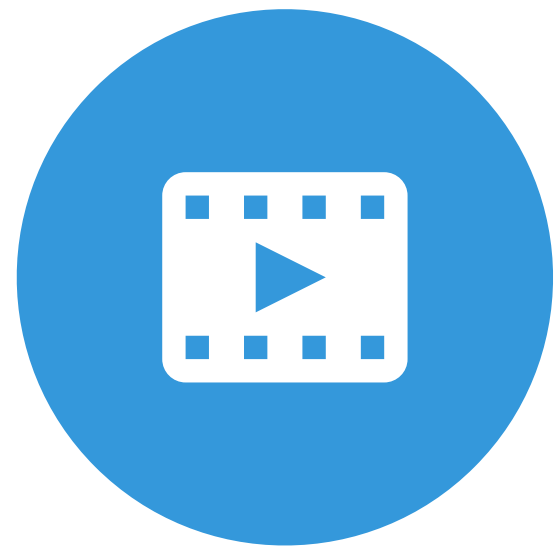


1 Request = 1 Server



SERVERLESS

Going serverless allows you to create and run applications and services without having to worry about servers - at least not servers that you have to maintain.



Zero Downtime




Zero Maintenance




Pay for what you use



Infinite Scaling



**Please don't fire your
ops team**

A dimly lit office scene with two women working on laptops. The woman on the left has long braids and wears glasses, while the woman on the right has long blonde hair and wears a red and black plaid shirt. They are both looking at their screens. The background is dark and out of focus.

Let's look at the state of serverless python



Serverless



Chalice



Zappa



WSGI



Microservice



WSGI

Serverless
Zappa



Microservice

Serverless
AWS Chalice
Zappa

Vendor Lock-in





Zappa

Let's make a Microsite





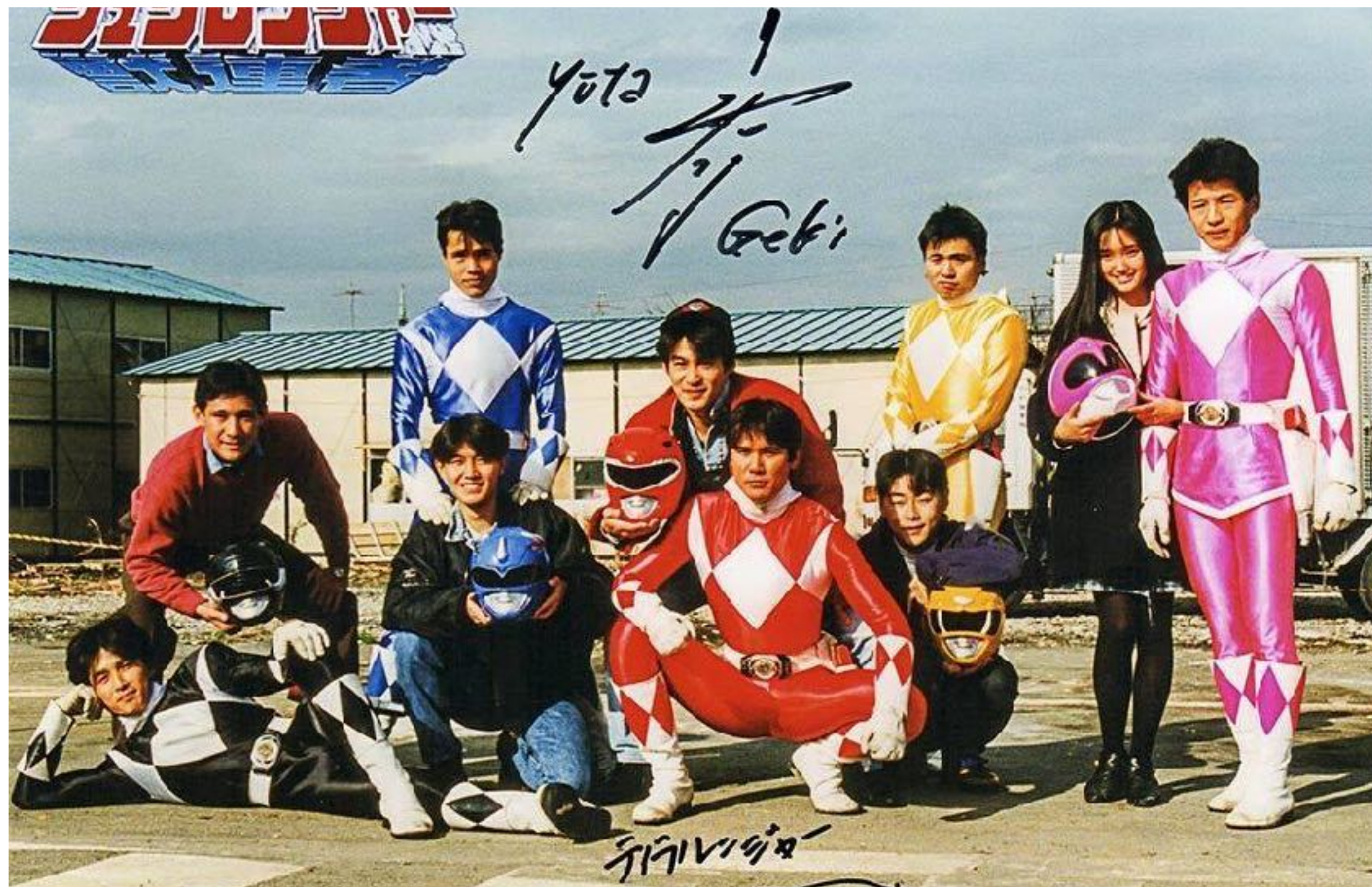
**But first it's time for a
small tangent**



The Origin of Power Rangers







KYŌRYŪ SENTAI ZYURANGER

1992-1993



MIGHTY MORPHIN POWER RANGERS

1993-1996



AK RANGER



AK RANGER



AK RANGER



AK RANGER



AK RANGER



HOW
BUILT
THIS
WITH GUY RAZ





**Ok. So I built a
dumb Website about this.**




HENSHIN.RUPAREL.CO

www.contentful.com

Henshin!


"Henshin!" (変身) is the Japanese word for "transformation/transform". It refers to a subset of Japanese Super Hero that transforms between a super-powered form and a normal civilian mode. A [Henshin Hero](#) has distinct normal and powered forms, and needs to actively switch between the two. In essence, the character's powers are all turned off while they are in their Secret Identity.

Kyōryū Sentai Zyuranger
Original release: February 21, 1992 - February 12, 1993



AK RANGER

Gosei Sentai Dairanger
Original release: February 19, 1993 - February 11, 1994

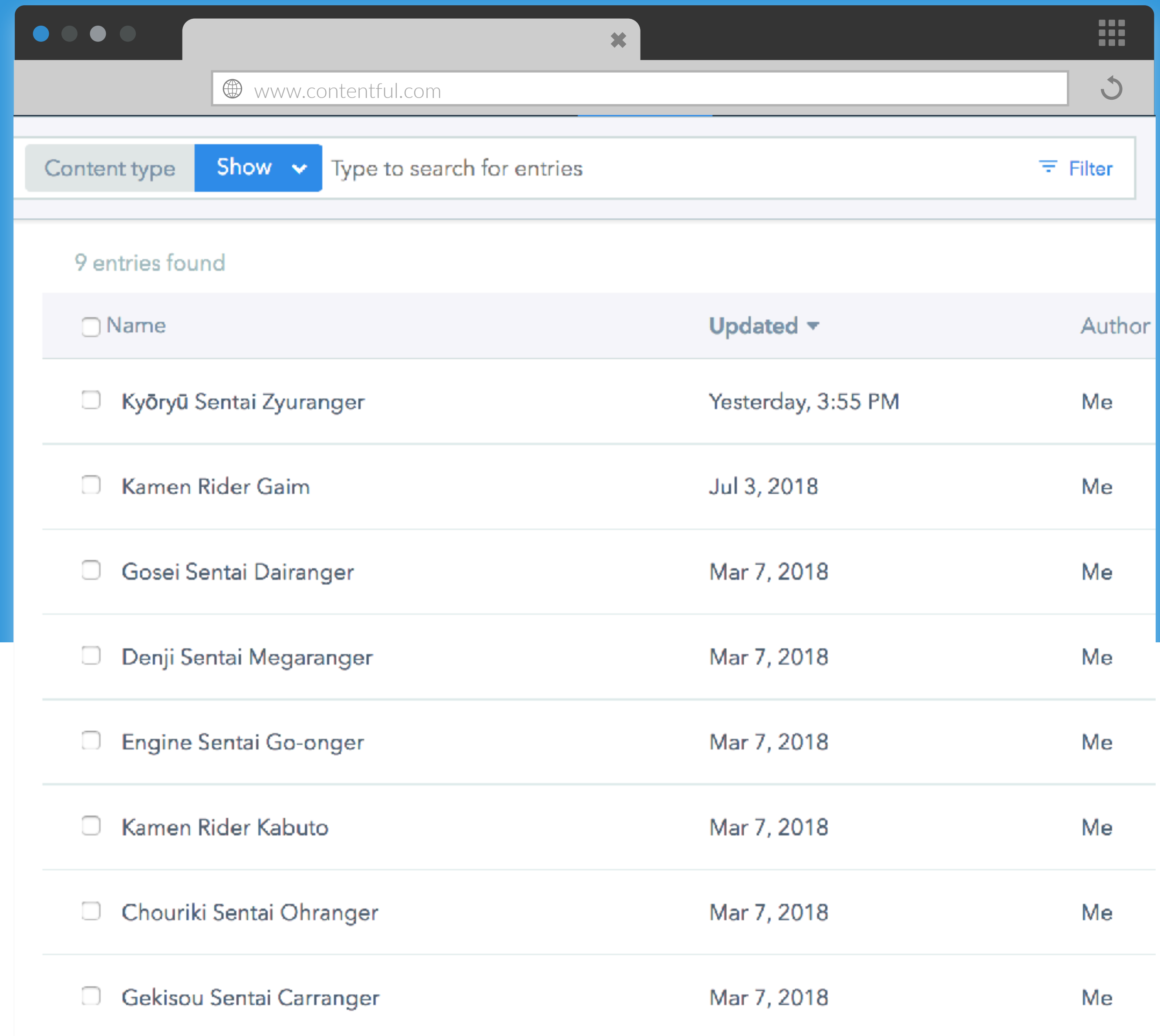


80 LINE FLASK APP + CONTENTFUL

```
app.py
1 from flask import Flask, render_template, url_for, abort
2 from flaskext.markdown import Markdown
3 import contentful
4
5 SPACE_ID = 'Something'
6 DELIVERY_API_KEY = 'Something Else'
7
8 client = contentful.Client(
9     SPACE_ID,
10    DELIVERY_API_KEY)
11
12 app = Flask(__name__)
13 Markdown(app)
14
15
16 def format_datetime(value):
17     """Format date time object using jinja filters"""
18     return (value.strftime('%B %-d, %Y'))
19
20
21 app.jinja_env.filters['datetime'] = format_datetime
22
23
24 @app.route('/')
25 @app.route('/home')
26 def index():
27     """index route. Gathers information from contentful and renders page"""
28     shows = client.entries(
29         {'content_type': 'show',
30          'order': 'fields.first_episode_date'})
31
32     entry_id = '7AmisHpntSSY0ku0cueecw'
33     intro_string = client.entry(entry_id)
```

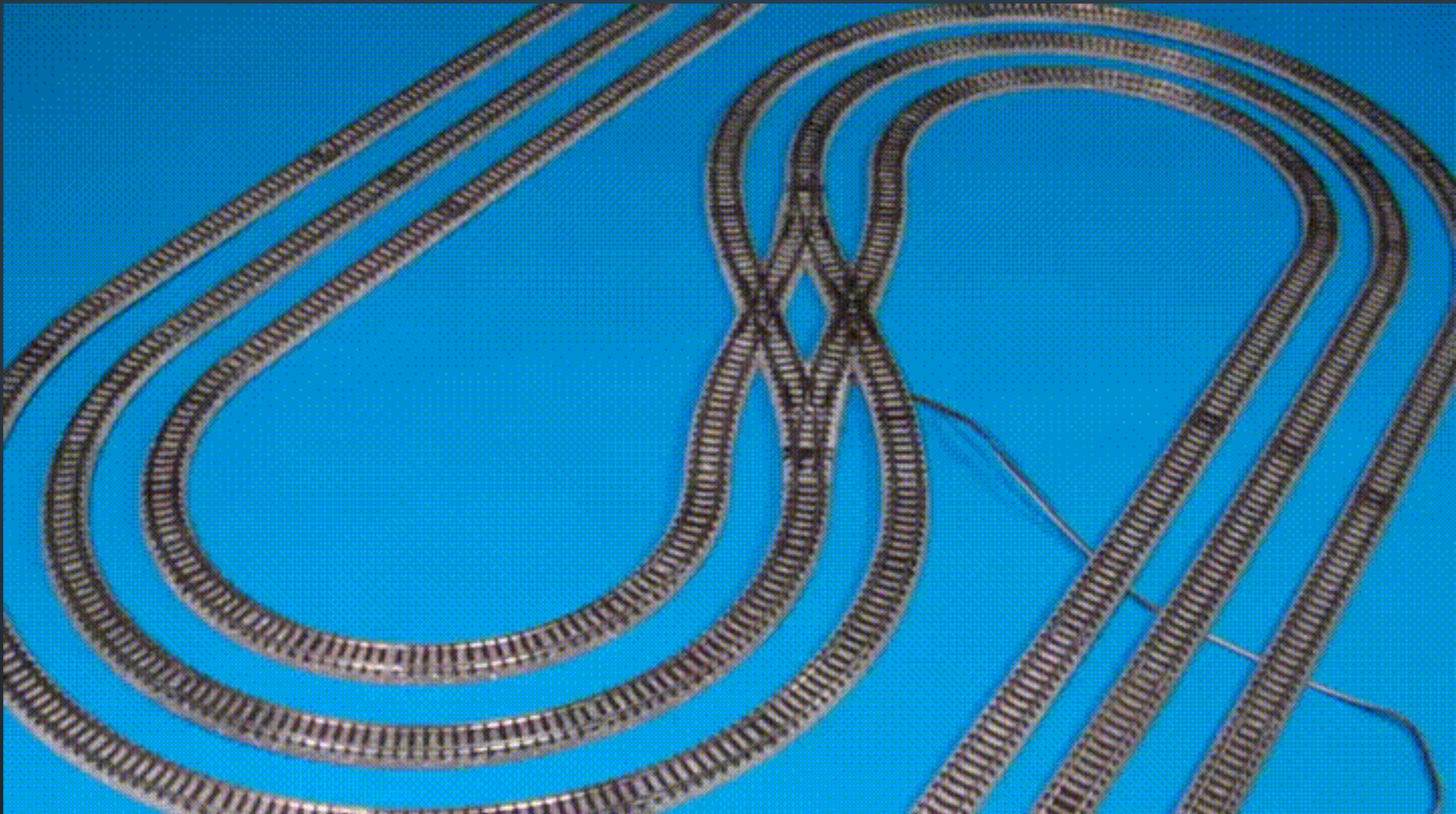
MacBook

CONTENTFUL



The screenshot shows a web browser window with the URL `www.contentful.com`. The interface includes a search bar with the placeholder text "Type to search for entries" and a "Filter" button. Below the search bar, it states "9 entries found". A table displays the following data:

<input type="checkbox"/> Name	Updated ▾	Author
<input type="checkbox"/> Kyōryū Sentai Zyuranger	Yesterday, 3:55 PM	Me
<input type="checkbox"/> Kamen Rider Gaim	Jul 3, 2018	Me
<input type="checkbox"/> Gosei Sentai Dairanger	Mar 7, 2018	Me
<input type="checkbox"/> Denji Sentai Megaranger	Mar 7, 2018	Me
<input type="checkbox"/> Engine Sentai Go-onger	Mar 7, 2018	Me
<input type="checkbox"/> Kamen Rider Kabuto	Mar 7, 2018	Me
<input type="checkbox"/> Chouriki Sentai Ohranger	Mar 7, 2018	Me
<input type="checkbox"/> Gekisou Sentai Carranger	Mar 7, 2018	Me



A woman with long blonde hair, wearing a light blue blazer, is standing and writing on a whiteboard with a blue marker. She is in profile, facing right. In the foreground, the back of a person's head with long dark hair is visible, looking towards the whiteboard. The whiteboard has some faint, illegible writing on it. The background is slightly out of focus, showing what appears to be a meeting room with a window.

Let's make it serverless



Zappa



AWS Lambda

+

API Gateway

OPEN FILES

app.py

FOLDERS

zappa-henshin

contentful-migrations

env

import_export

static

Temp

templates

.gitignore

app.py

README.md

requirements.txt

zappa_settings.json

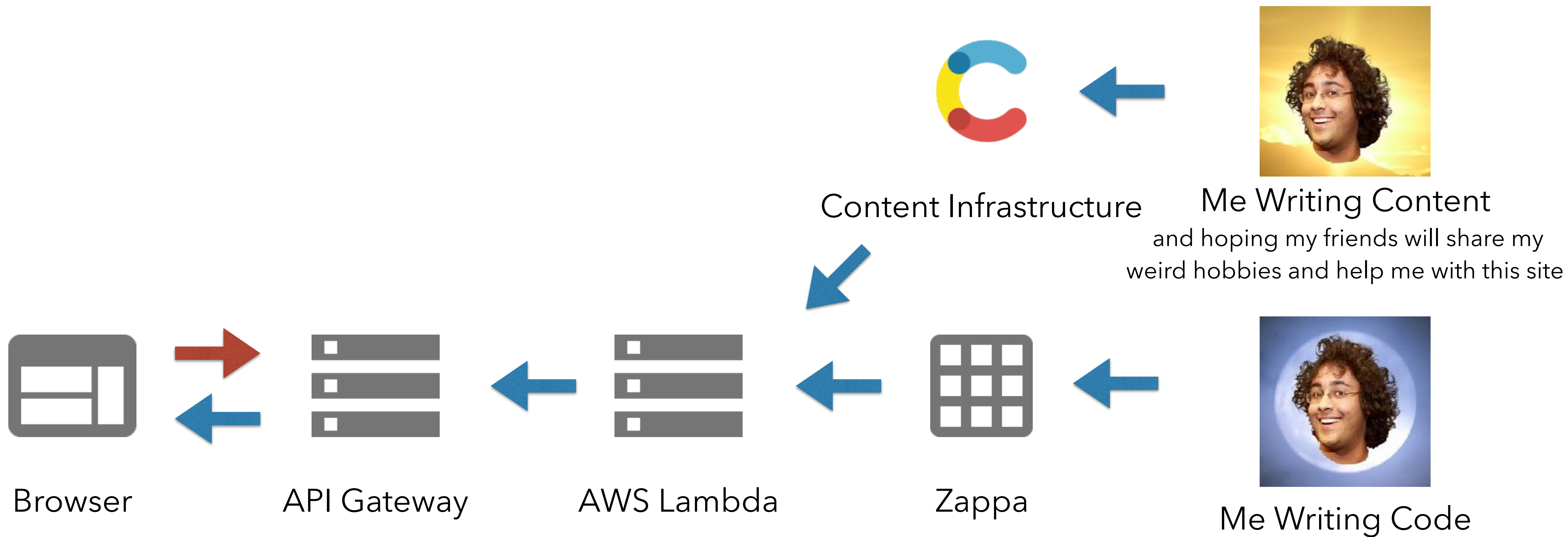
app.py

```
1  ~/Documents/Contentful/toku_lambda/zappa-henshin/app.py url_for, abort
2  from flaskext.markdown import Markdown
3  import contentful
4
5  SPACE_ID = 'Something'
6  DELIVERY_API_KEY = 'Something Else'
7
8  client = contentful.Client(
9      SPACE_ID,
10     DELIVERY_API_KEY)
11
12  app = Flask(__name__)
13  Markdown(app)
14
15
16  def format_datetime(value):
17      """Format date time object using jinja filters"""
18      return (value.strftime('%B %-d, %Y'))
19
20
21  app.jinja_env.filters['datetime'] = format_datetime
22
23
24  @app.route('/')
25  @app.route('/home')
26  def index():
27      """index route. Gathers information from contentful and renders page"""
28      shows = client.entries(
29          {'content_type': 'show',
30           'order': 'fields.first_episode_date'})
31
32      entry_id = '7AmisHpntSSY0ku0cueecw'
33      intro_string = client.entry(entry_id)
34
35      return render_template('index.html',
36                             shows=shows,
37                             intro_string=intro_string.intro_string,
38                             title=intro_string.title)
39
40
41  @app.route('/show/<string:entry_id>')
42  def show(entry_id):$
43      """Take a Slug and return a Show."""
44      show = client.entries(
45          {'content_type': 'show',
46           'fields.slug': entry_id})
47      show = show[0]
48
49      return render_template('show.html',
50                             show=show
```

zappa-henshin/master*, File is modified, 3*, + Shy Ruparel (6 months ago), Line 9, Column 14

Spaces: 4

Python



SERVERLESS FRAMEWORK

```
10:53:17 AM serverless-henshin venv 8s $ pip freeze > requirements.txt
10:53:23 AM serverless-henshin venv $ sls deploy
serverless: Generated requirements from /Users/shy/Documents/Contentful/toku_lambda/serverless-henshin/requirements.txt in /Users/shy/Documents/Contentful/toku_lambda/serverless-henshin/.serverless/requirements.txt...
serverless: Installing requirements from /Users/shy/Documents/Contentful/toku_lambda/serverless-henshin/.serverless/requirements/requirements.txt ...
serverless: Docker Image: lambci/lambda:build-python3.6
serverless: Using Python specified in "runtime": python3.6
serverless: Packaging Python WSGI handler...
serverless: Packaging service...
serverless: Excluding development dependencies...
serverless: Injecting required Python packages to package...
serverless: Using Python specified in "runtime": python3.6
serverless: Uploading CloudFormation file to S3...
serverless: Uploading artifacts...
serverless: Uploading service .zip file to S3 (17.14 MB)...
serverless: Validating template...
serverless: Updating Stack...
serverless: Checking Stack update progress...
.....
serverless: Stack update finished...
Service Information
service: serverless-flask
stage: dev
region: us-east-1
stack: serverless-flask-dev
api keys:
None
endpoints:
ANY - https://oolqxr76le.execute-api.us-east-1.amazonaws.com/dev
ANY - https://oolqxr76le.execute-api.us-east-1.amazonaws.com/dev/{proxy+}
functions:
app: serverless-flask-dev-app
10:54:53 AM serverless-henshin venv 1m0s $ |
```

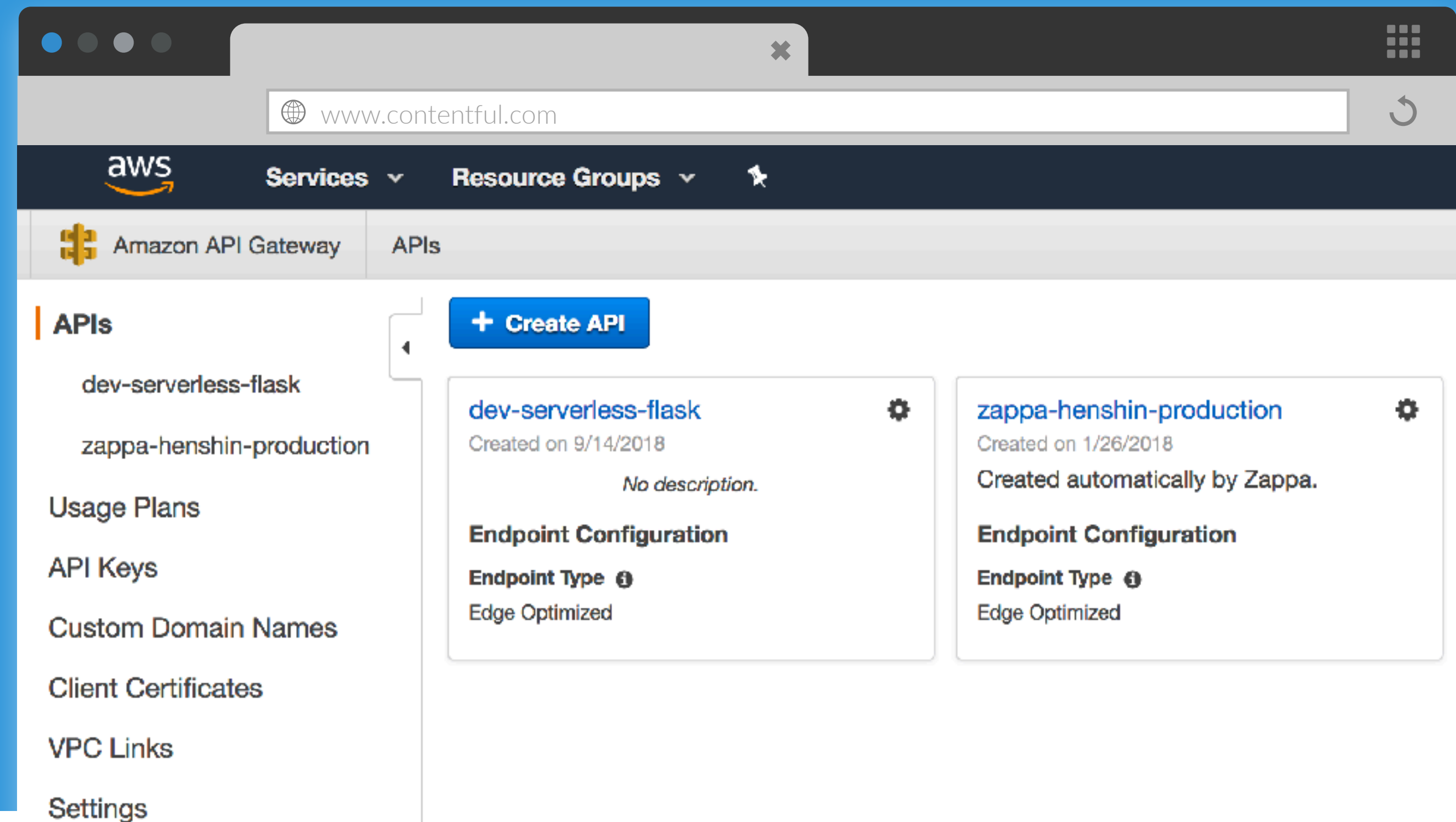
MacBook

SERVERLESS FRAMEWORK CONFIG

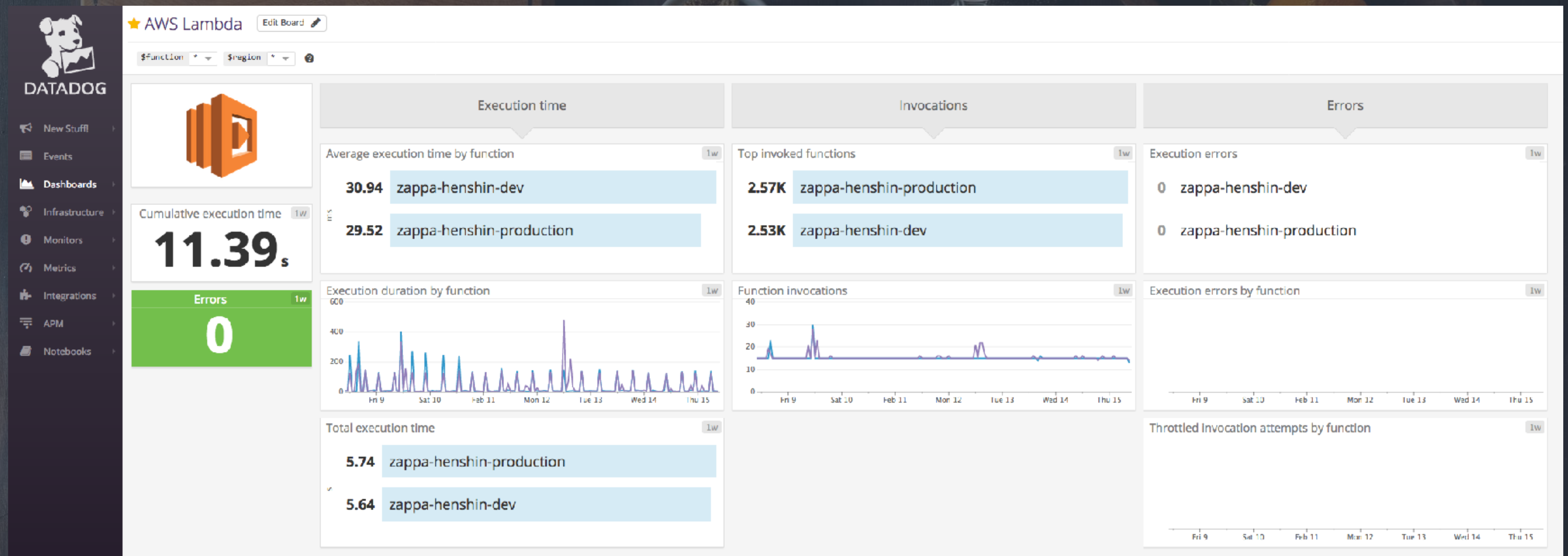
```
app.py  serverless.yml
1  # serverless.yml
2
3  service: serverless-flask
4
5  plugins:
6    - serverless-python-requirements
7    - serverless-wsgi
8
9  custom:
10   wsgi:
11     app: app.app
12     packRequirements: false
13     pythonRequirements:
14       dockerizePip: non-linux
15
16   provider:
17     name: aws
18     runtime: python3.6
19     stage: dev
20     region: us-east-1
21
22   functions:
23     app:
24       handler: wsgi.handler
25       events:
26         - http: ANY /
27         - http: 'ANY {proxy+}'
```

MacBook

BOTH DEPLOYMENT FRAMEWORKS HANDLE AWS LAMBDA AND API GATEWAY



Let's talk Metrics





30 ms

RENDER TIME

\$0.00000002/ms

COST ON LAMBDA



3,200,000

FREE FUNCTION CALLS ON LAMBDA

\$3.50

PER MILLION CALLS ON API GATEWAY



Basically no time

TO GET UP AND RUNNING



Shy Ruparel

Developer Evangelist | Contentful

@ShyRuparel

He/Him

Slides -> [GitHub.com/shy/talks](https://github.com/shy/talks)

Background Images -> wocintechchat.com