

A white "AWS" logo inside an orange cloud shape.

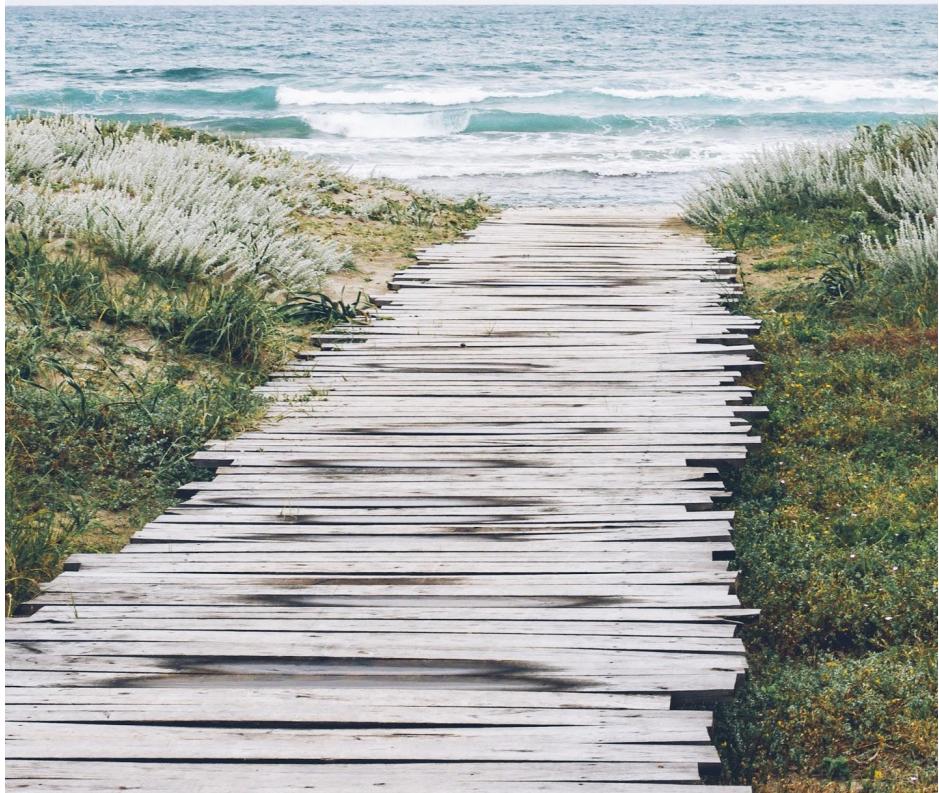
AWS



# Micro Service using AWS API Gateway and Lambda

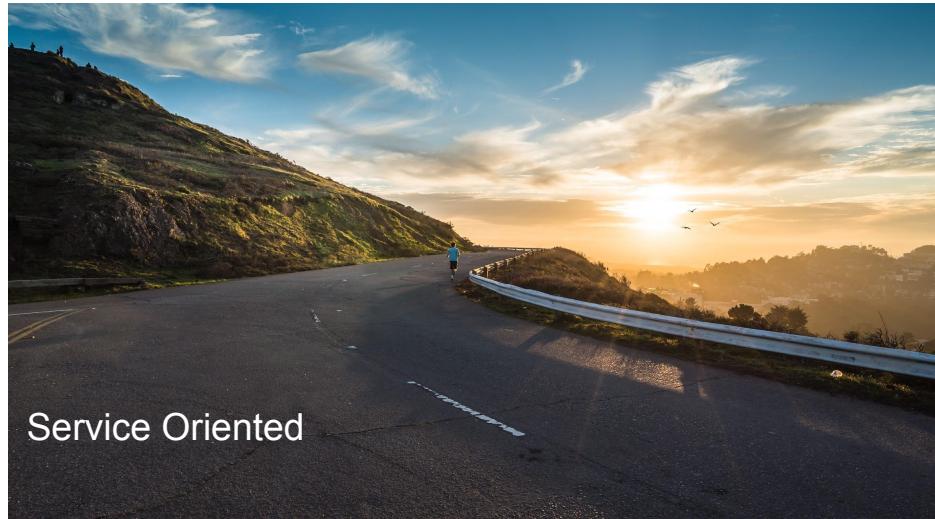
- Antonio

Micro Service



Providing Services

I need a service.  
Simple and easy. I  
just need a path get  
right to my  
destination.



Service Oriented

Micro Service

# AWS Account.

AWS

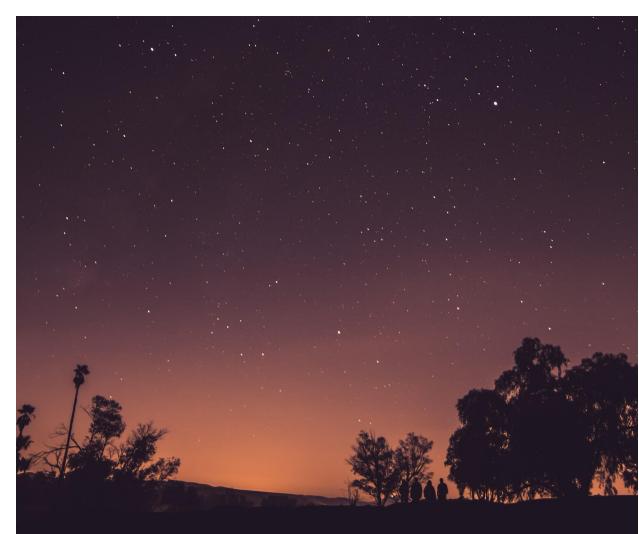
Serverless



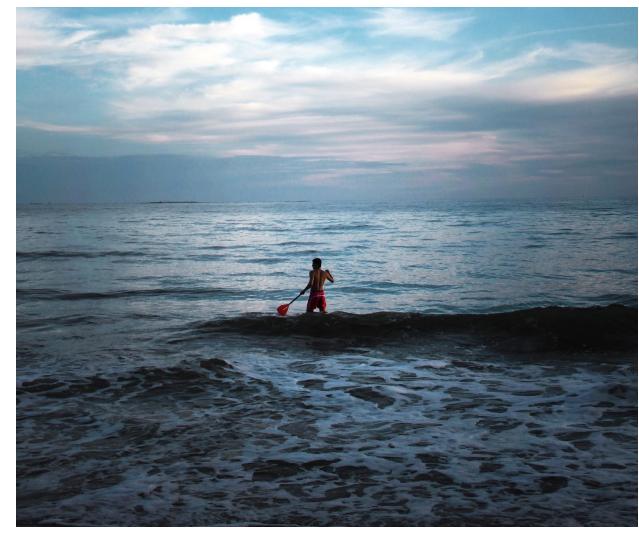


# Serverless architectures

- build and run applications and services
- Server management is done by AWS.
- You no longer have to provision, scale, and maintain servers to run your applications, databases, and storage systems



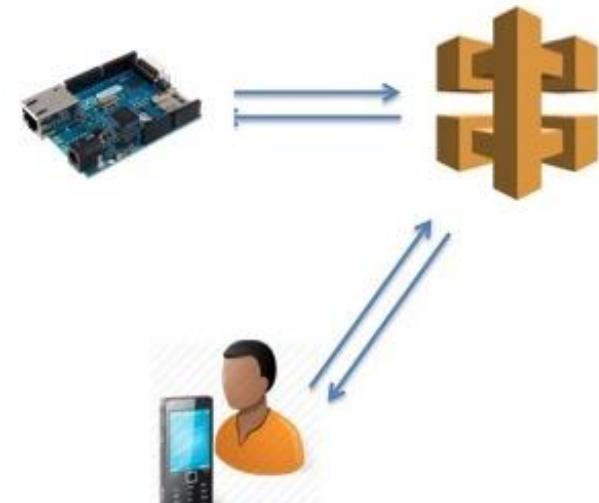
Less time managing and  
setup of servers



More time for whatever else  
you want to do



# API Gateway.



API Gateway tightly integrates with AWS Lambda to allow you to create completely server-less APIs. With Amazon API Gateway, you create REST APIs that your mobile and web applications can use to call publicly available AWS services through your code running in AWS Lambda.



## APIs

Community Reporting

LambdaMicroservice

wix-wrapper

Usage Plans

API Keys

Custom Domain Names

Client Certificates

Settings

Create API

Community Reporting

*Community Incident Reports*

LambdaMicroservice

*Created by AWS Lambda*

wix-wrapper

*access wrapper lambda function for WIX*

## RESOURCE ACTIONS

[Create Method](#)[Create Resource](#)[Enable CORS](#)[Edit Resource Documentation](#)

## API ACTIONS

[Deploy API](#)[Import API](#)[Edit API Documentation](#)[Delete API](#)

No methods defined for the resource.

Resources

Actions ▾

## / Methods

/

✓ ✖

- ANY
- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

No methods defined for the resource.

## / - POST - Setup

---

Choose the integration point for your new method.

Integration type  Lambda Function [i](#)

HTTP [i](#)

Mock [i](#)

AWS Service [i](#)

Use Lambda Proxy integration  [i](#)

Lambda Region

▼

# Manage API concerns here easily

- Authentication
- Transformation
- Routes
- Cross Origin Resource Sharing(CORS)

Lambda.



Lambda functions

## Configure function

A Lambda function consists of the custom code you want to execute. [Learn more about Lambda functions.](#)

Name\*

myFunctionName

Description

A starter AWS Lambda function.

Runtime\*

Node.js 4.3

### Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. [Learn more](#) about deploying Lambda functions.

Code entry type

Edit code inline

```
1 'use strict';
2
3 console.log('Loading function');
4
5 exports.handler = (event, context, callback) => {
6   //console.log('Received event:', JSON.stringify(event, null, 2));
7   console.log('value1 =', event.key1);
8   console.log('value2 =', event.key2);
9   console.log('value3 =', event.key3);
10  callback(null, event.key1); // Echo back the first key value
11  //callback('Something went wrong');
12 };
13
```

### Code in:

- Java
- Node.js
- Python
- C#

Set memory use and time outs.

Connect to many many or the many many AWS services available.

Use live console on left or upload to S3.

npm packages for local  
<https://www.npmjs.com/package/lambda-local>

<https://www.npmjs.com/package/node-lambda>

## Input test event



Use the editor below to enter an event to test your function with. You can edit the event again by choosing **Configure test event** in the Actions list. Note that changes to the event will only be saved locally.

Sample event template

Hello World



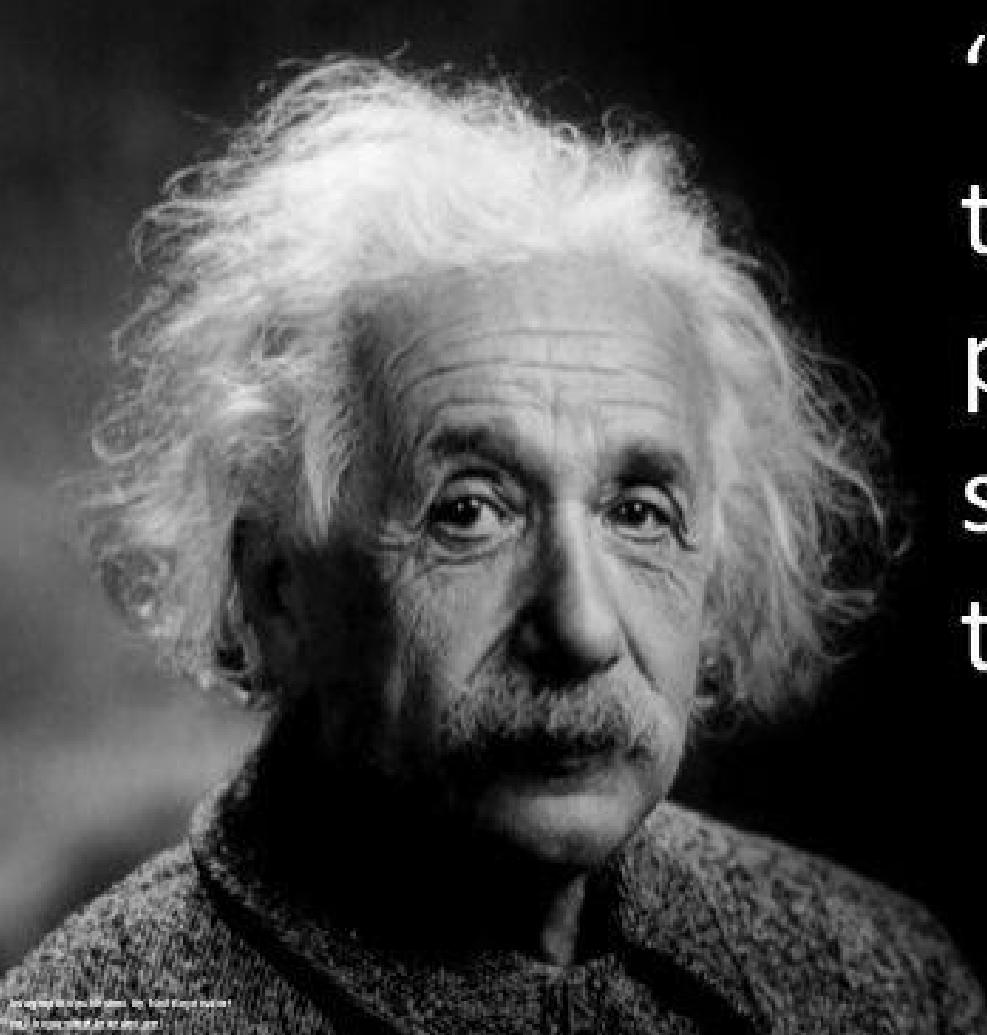
```
1 {  
2   "key3": "Meow meow meow meow",  
3   "key2": "Meow",  
4   "key1": "Hellow world"  
5 }
```

## Micro Service



I have my service.  
Simple and easy. I  
just get right to my  
destination.

POST to API endpoint:  
<https://aws.apigateaway.com/holla>

A black and white portrait of Albert Einstein, showing him from the chest up. He has his characteristic wild, white hair and a well-groomed, light-colored beard and mustache. He is looking slightly to the right of the camera with a thoughtful expression. The background is dark and out of focus.

“In theory,  
theory and  
practice are the  
same. In practice,  
they are not.”

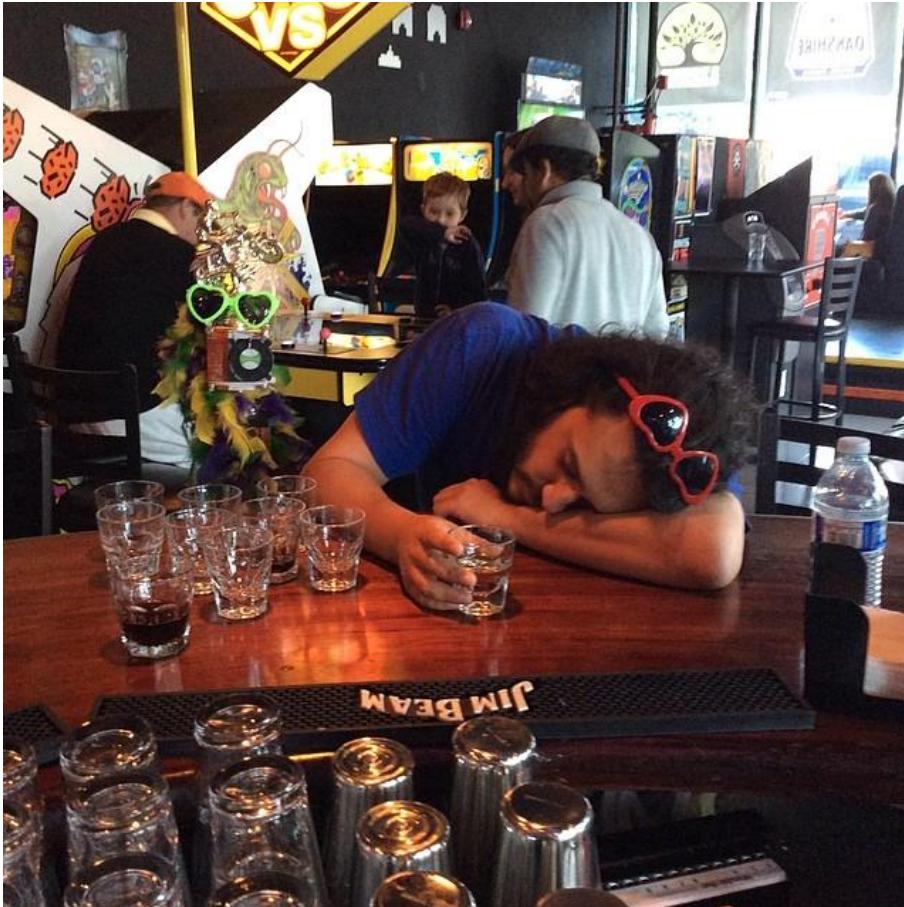
*Albert Einstein*

# Example 1.

# MY NEED

Show the next upcoming meetup for Eugene Web Devs on the website, but browser cross origin resource sharing (CORS) restrictions...

I am just too busy to get CORS working properly per the meetup docs that I didn't even read. Because I am busy.



# Eugene Web Developers

[Home](#) [Members](#) [Sponsors](#) [Photos](#) [Pages](#) [Discussions](#) [More](#)

[Group tools](#)  [My profile](#)

eug  
web  
devs

Change photo

Eugene, OR  
Founded Jan 26, 2012

About us...

Invite friends

Developers 336

Group reviews 7

## Web and Mobile Development in Eugene's Growing Tech Community

+ Schedule a new Meetup

Upcoming (3) Suggested (0) Past Calendar

### Modern Web Services: How we serve up our content.

IDX Broker  
100 East Broadway, Eugene, OR [\(map\)](#)



Ryan Olds will discuss which modern services we all use today Apache, NGINX, etc.. and how we are serving up our content.  
[Learn more](#)

Hosted by: Antonio Ortega Jr (Organizer), and Ryan Olds

## What's new



Thu Mar 30  
6:30 PM

I'm going

26 going

2 comments

MORE

✓ NEW RSVP

MeetUp.com

API

EugeneWebDevs.com

Next Meet up: Modern Web Services:  
How we serve up our content.

Meeting at 100 East Broadway

RSVP Now



Discuss development

Eugene Web Developers is all about web and mobile app development in the age of



Dev Community Focused

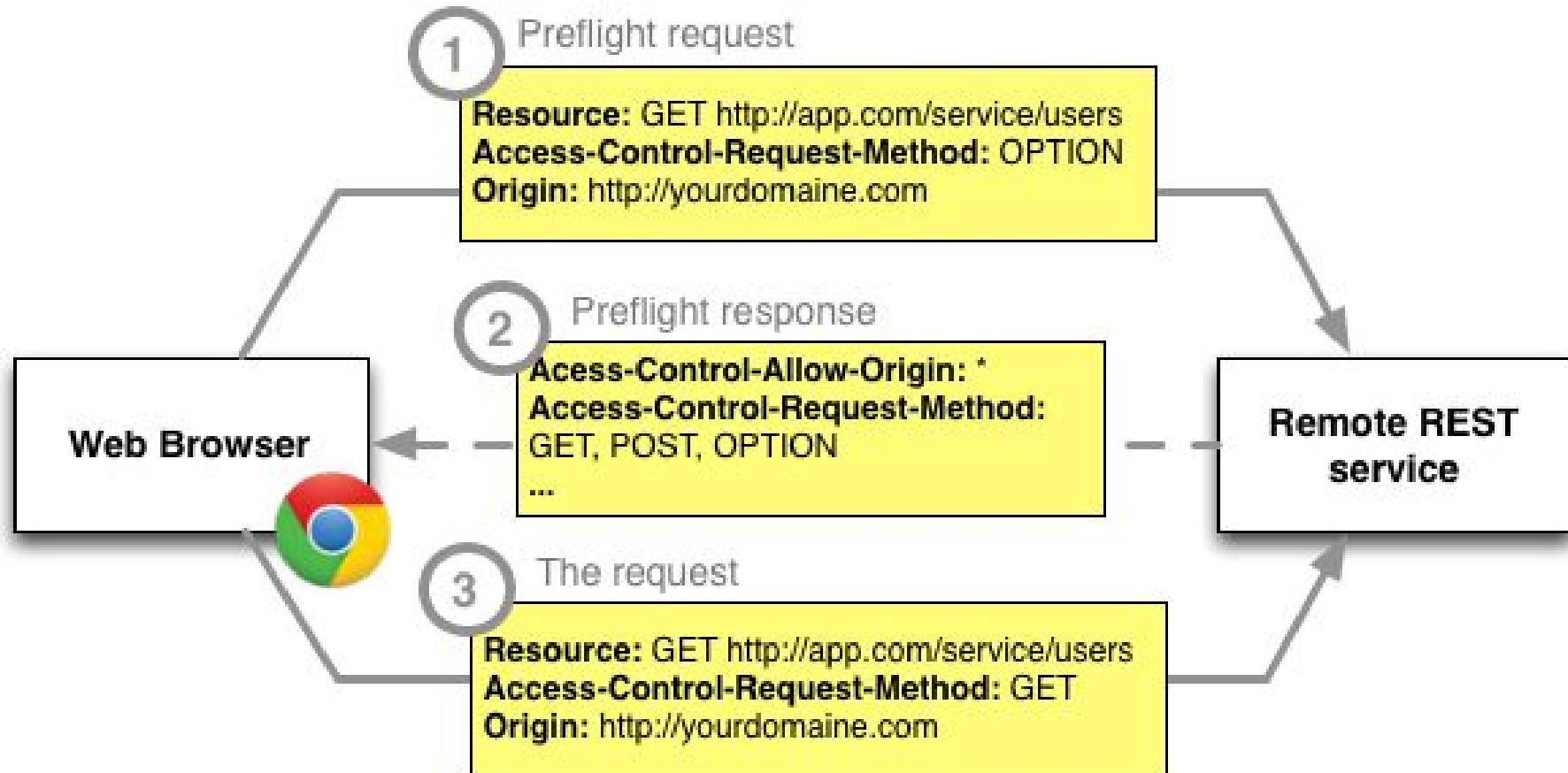
Eugene's web development community has been quietly growing and thriving.



Web Technology

We are into: Web development languages such as Javascript, Python, Ruby, .NET,

# "Cross-Origin Resource Sharing"



GET

OPTIONS

TEST  
⚡

Client

### Method Request

Auth: NONE

ARN: arn:aws:execute-api:us-west-2:529123413029:bhd6emvIk2/\*/\*

### Integration Request

Type: LAMBDA

Region: us-west-2

Lambda nextEugeneWebDevsMeetup

### Method Response

HTTP Status: 200

### Integration Response

HTTP status pattern: -

Output passthrough: Yes

[Qualifiers](#)[Test](#)[Actions](#)

Code

Configuration

Triggers

Monitoring



Code entry type

Edit code inline

```
1 #meetup endpoint https://api.meetup.com/self/events?photo-host=public&page=1&sig_id=137579212&status=upcoming&sig=1
2 import json
3 import urllib2
4 import ssl
5
6 def lambda_handler(event, context):
7     meetup = 'https://api.meetup.com/eugenewebdevs/events?photo-host=public&page=1&sig_id=137579212&status=upcoming'
8     req = urllib2.Request(meetup)
9     response = urllib2.urlopen(req)
10    the_page = response.read()
11    return the_page
```



# Eugene Web Developers

[Home](#) Members Sponsors Photos Pages Discussions More

Group tools My profile

eug  
web  
devs

Change photo

Eugene, OR  
Founded Jan 26, 2012

About us...

Invite friends

Developers 336

Group reviews 7

## Web and Mobile Development in Eugene's Growing Tech Community

+ Schedule a new Meetup

Upcoming (3) Suggested (0) Past Calendar

### Modern Web Services: How we serve up our content.

IDX Broker  
100 East Broadway, Eugene, OR (map)



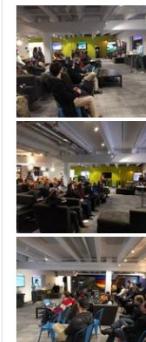
Ryan Olds will discuss which modern services we all use today Apache, NGINX, etc.. and how we are serving up our content.  
[Learn more](#)

Hosted by: Antonio Ortega Jr (Organizer), and Ryan Olds

Thu Mar 30  
6:30 PM

I'm going  
26 going  
2 comments

## What's new



MORE

✓ NEW RSVP

# AWS

# MeetUp.com

# API

# EugeneWebDevs.com

Next Meet up: Modern Web Services:  
How we serve up our content.

Meeting at 100 East Broadway

## RSVP Now



Discuss development

Eugene Web Developers is all about web and mobile app development in the age of

Dev Community Focused

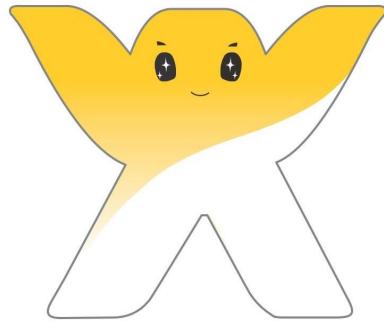
Eugene's web development community has been quietly growing and thriving.

Web Technology

We are into: Web development languages such as Javascript, Python, Ruby, .NET,

# Example 2

**NEED**



**AND**

IDX Broker dynamic wrappers  
to work together

# IDX Wrappers

```
<html>
  <head></head>
    <body>
      <h1>I'm a Header</h1>
      <div id="idxStart"></div>
      <div id="idxStop"></div>
        <footer>
<marquee>Hi Josh!!</marquee>
        </footer>
      </body>
    </html>
```

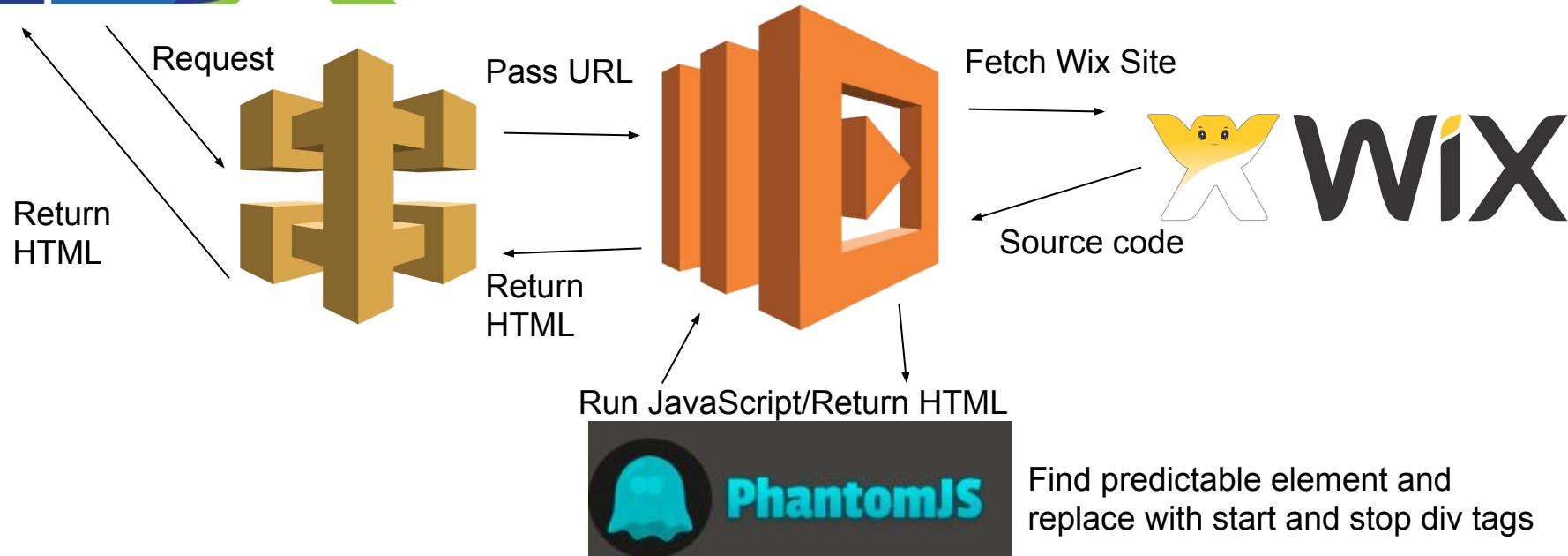
WIX

```
<html>
  <head>
    <script src="script1.js"></script>
    <script src="script2.js"></script>
    <script src="script3.js"></script>
    <script src="script4.js"></script>
    <script src="script5.js"></script>
    <script src="script6.js"></script>
    <script src="script7.js"></script>
  </head>
  <div id="content"></div>
  —
  </body>
<html>
```

# Micro Service to create a Dynamic Wrapper URL



cURL API Gateway passing the WIX site URL  
<https://aws.apigatewayurl.com?site=http://wixsite.com>



# Source Available

On  Stash

Happy to DEMO for anyone

Pain points.



## THINGS ABOUT THIS THAT SUCK

- Your favorite language might be supported
- Newest versions of languages or packages might not work
- Testing on local not so straightforward
- Testing via the web interface takes longer
- Larger amounts of code require upload every time
- API gateway might transform your returns by default
- You might actually need more than a micro service

Dollar Bills.



What about the Price?????????????????

## API Calls

\$3.50 per million API calls received, plus the cost of data transfer out, in gigabytes.

---

## Data Transfer Costs

If you use Amazon API Gateway, you will be charged for API calls and out-to-Internet data transfer as described below.

### Amazon API Gateway Data-Transfer-Out Rates

\$0.09/GB for the first 10 TB

\$0.085/GB for the next 40 TB

\$0.07/GB for the next 100 TB

\$0.05/GB for the next 350 TB

If you allocated 128MB of memory to your function, executed it 30 million times in one month, and it ran for 200ms each time, your charges would be calculated as follows:

#### **Monthly compute charges**

The monthly compute price is \$0.00001667 per GB-s and the free tier provides 400,000 GB-s.

Total compute (seconds) =  $30M * (0.2sec) = 6,000,000$  seconds

Total compute (GB-s) =  $6,000,000 * 128MB/1024 = 750,000$  GB-s

Total Compute – Free tier compute = Monthly billable compute seconds

$750,000$  GB-s –  $400,000$  free tier GB-s =  $350,000$  GB-s

**Monthly compute charges =  $350,000 * \$0.00001667 = \$5.83$**

#### **Monthly request charges**

The monthly request price is \$0.20 per 1 million requests and the free tier provides 1M requests per month.

Total requests – Free tier request = Monthly billable requests

$30M$  requests –  $1M$  free tier requests =  $29M$  Monthly billable requests

**Monthly request charges =  $29M * \$0.2/M = \$5.80$**

#### **Total compute charges**

**Total charges = Compute charges + Request charges =  $\$5.83 + \$5.80 = \$11.63$  per month**

# My Bill



## Month-to-Date Spend by Service

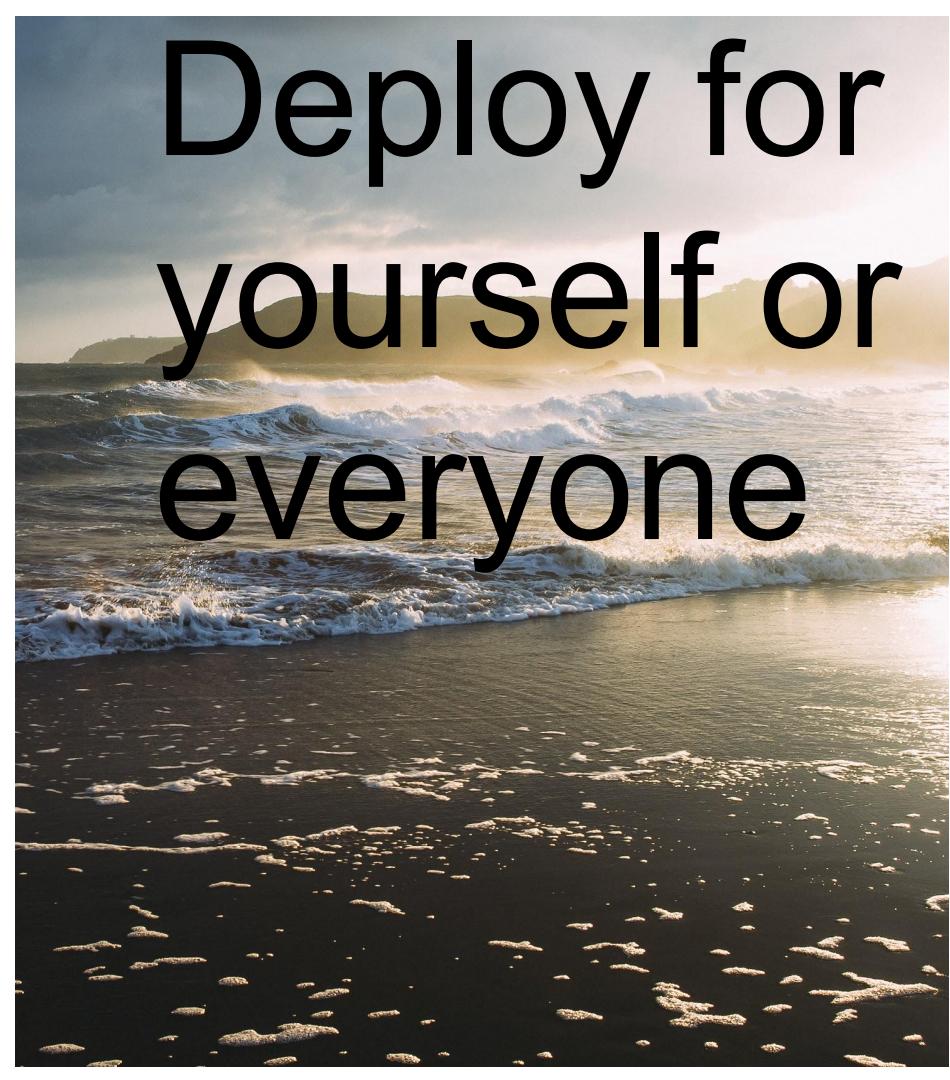
[Bill Details](#)

The chart below shows the proportion of costs spent for each service you use.

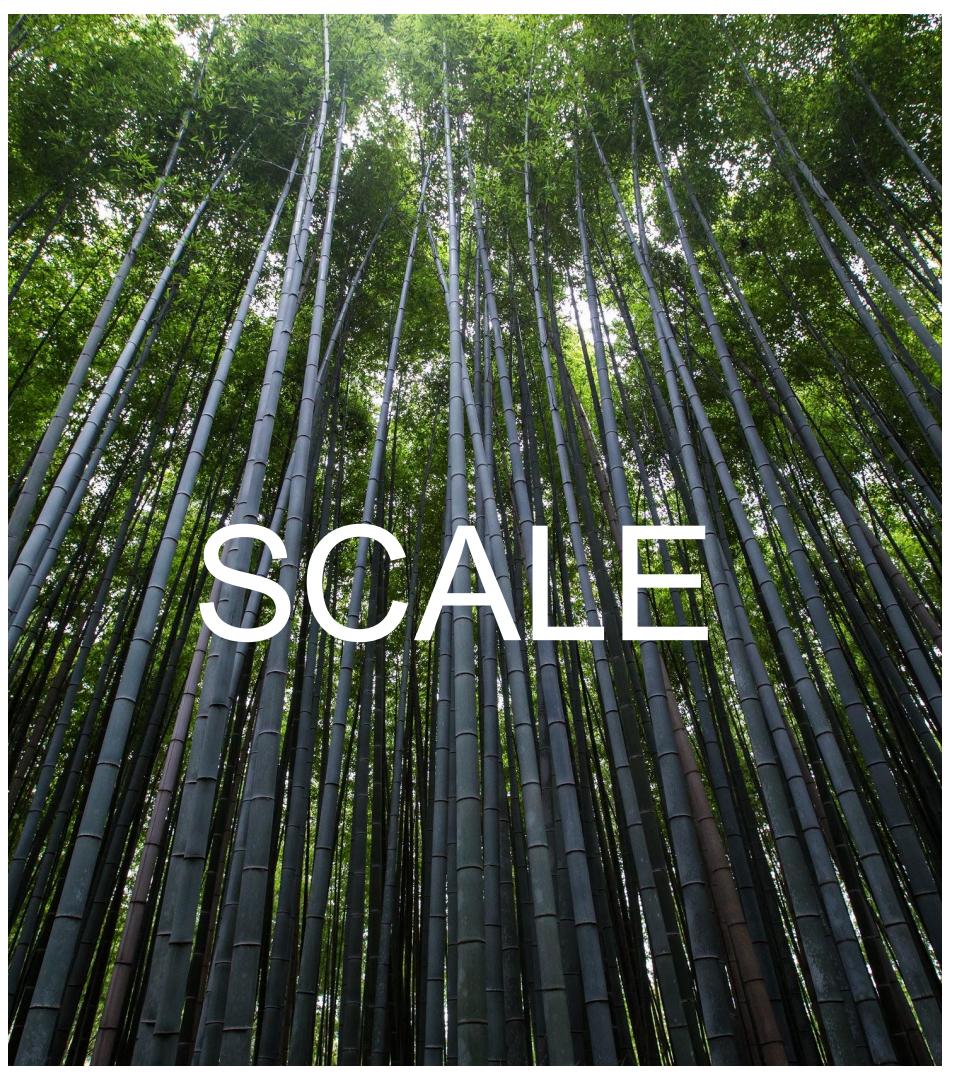


EC2	\$3.34
S3	\$0.01
IoT	\$0.00
CloudWatch	\$0.00
Other Services	\$0.00
Tax	\$0.00
<b>Total</b>	<b>\$3.35</b>

Deploy for  
yourself or  
everyone



SCALE





AWS

GAME OVER



Micro Service  
Using AWS

Amazon API Gateway  
and  
Lambda

- Antonio