

# Going Serverless



@jagthedrummer - OctoLabs.com/railsconf2016

A large, grey elephant is the central figure. It has a globe resting on its back, which features the words "GLOBE", "CHARGE", and "SUE". A woman in a blue top and black skirt is leaning against the elephant's trunk, while a man in a grey suit is crouching near its front legs.

# Serverless?

A photograph of Steve Carell as Michael Scott from the TV show "The Office". He is wearing a red pinstripe suit, a white shirt, and a striped tie. He is standing behind a microphone, gesturing with his hands as if he is speaking or performing. The background is dark and out of focus.

# Serverless



# **SERVERLESS** **FRAMEWORK**

*THE SERVERLESS APPLICATION FRAMEWORK*



framework for building  
web, mobile and IoT applications **exclusively** on  
AWS Lambda, API Gateway, and related services

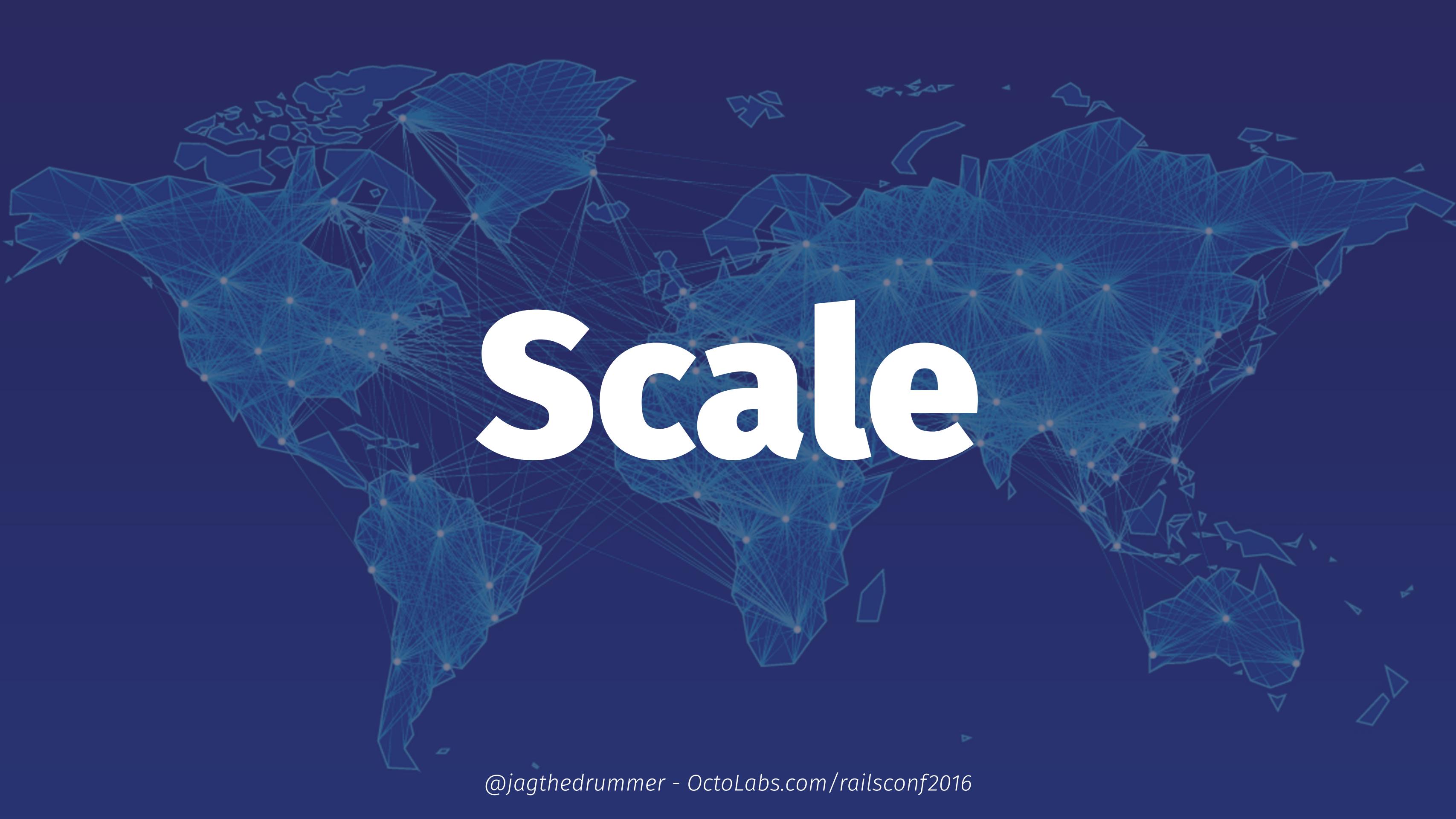


@jagthedrummer - OctoLabs.com/railsconf2016

# Vendor Lock-in 😞



# OPS



# Scale

# Money

# Jeremy Green

Consultant, Author, SaaSer



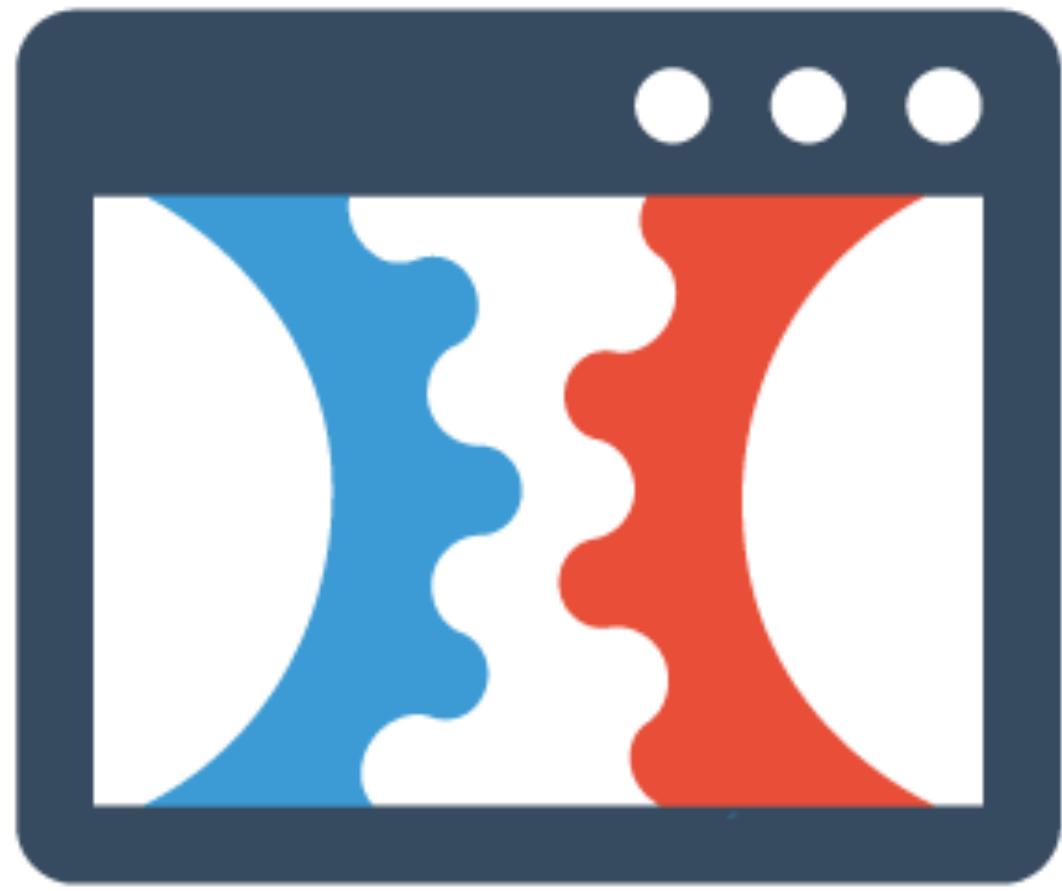
@jagthedrummer  
jeremy@octolabs.com

**IndependentConsultingManual.com**

**Remarq.io**

Things I Enjoy:  
Dopamine, Serotonin

Other Interests:  
Drumming, Photography, and Brewing



# click funnels



# The Pieces

# Lambda

# API Gateway

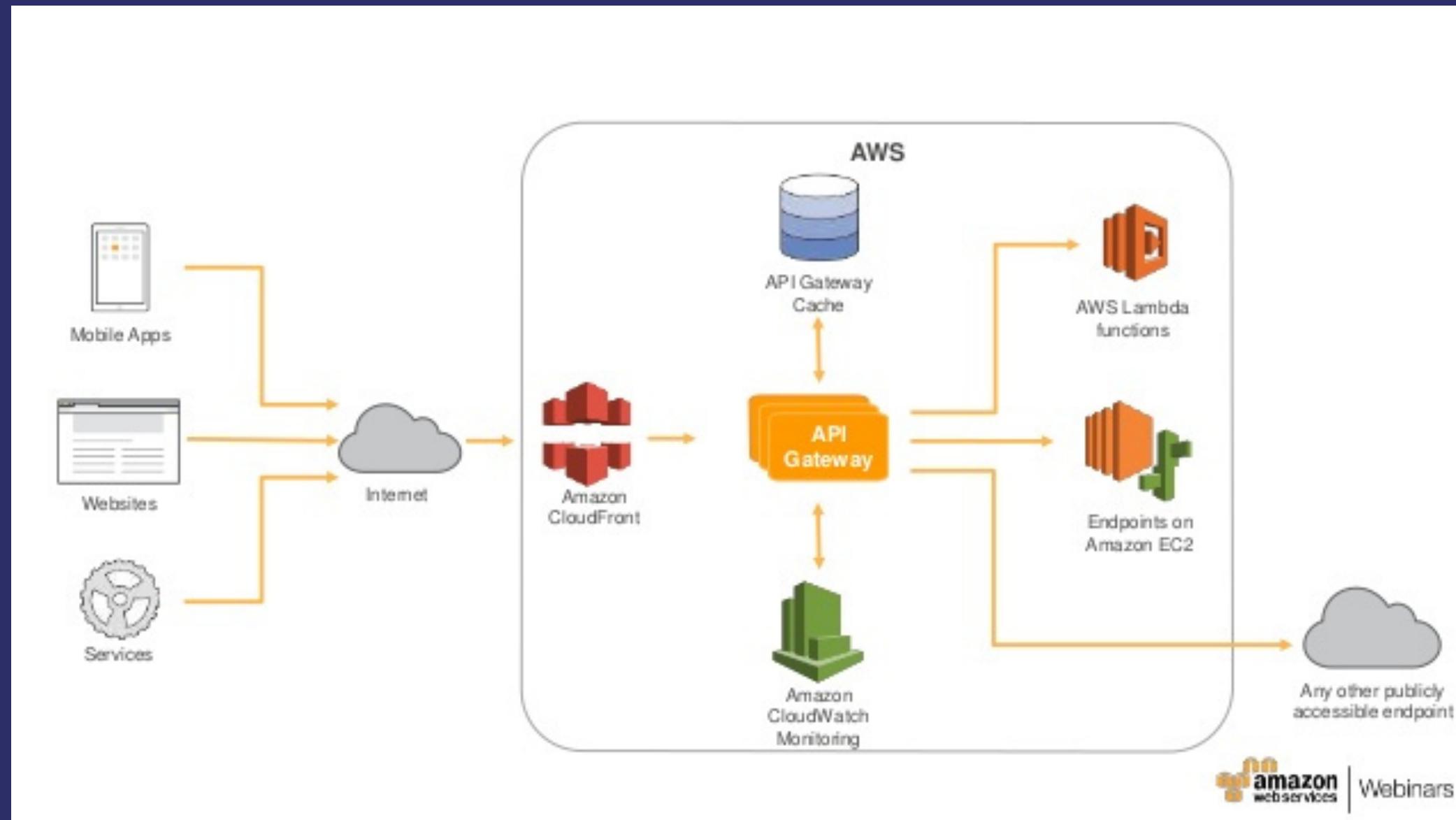
# DynamoDB

# RDS

@jagthedrummer - OctoLabs.com/railsconf2016

# CloudFormation

# Putting it all together



Deckset for Mac: Turn your x Google Image Result for ht Google Image Result for ht Lambda Management Con Backstory Jeremy Green

<https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions>

AWS Services Edit Jeremy Green N. Virginia Support

Lambda > Functions ?

You have 16 Lambda function(s) using 59.1 MB of code storage. Choose any Lambda function to view details on invocation requests, duration, and errors (metrics may take up to 60 seconds to appear).

Create a Lambda function Actions

Filter Viewing 1-12 of 16

	Function name	Description	Code size	Memory (MB)	Timeout (s)
○	serverlessDynamoCrudExample-u...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessDynamoCrudExample-c...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessDynamoCrudExample-c...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessDynamoCrudExample-d...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessDynamoCrudExample-in...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessDynamoCrudExample-s...	Serverless Lambda function for project: serverlessD...	665.9 kB	1024	6
○	serverlessTest2-createeasy	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6
○	serverlessTest2-indexeasy	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6
○	serverlessTest2-index	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6
○	serverlessTest2-hellopost	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6
○	serverlessTest2-hello	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6
○	serverlessTest2-deletetable	Serverless Lambda function for project: serverlessT...	805.7 kB	1024	6

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Deckset for Mac: Turn your ... aws dynamodb - Google S... Google Image Result for ht... Google Image Result for ht... Lambda Management Con... Backstory Jeremy Green

https://console.aws.amazon.com/lambda/home?region=us-east-1#/create?step=1

AWS Services Edit Jeremy Green N. Virginia Support

Lambda > New function

**Step 1: Select blueprint**

### Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under CC0.

Filter All languages Viewing 1-9 of 42

<b>s3-get-object-python</b> An Amazon S3 trigger that retrieves metadata for the object that has been updated. python2.7 · s3	<b>config-rule-change-triggered</b> An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types. nodejs · config	<b>dynamodb-process-stream</b> An Amazon DynamoDB trigger that logs the updates made to a table. nodejs · dynamodb
<b>microservice-http-endpoint</b> A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway. nodejs · api-gateway	<b>node-exec</b> Demonstrates running an external process using the Node.js child_process module. nodejs	<b>slack-echo-command-python</b> A function that handles a Slack slash command and echoes the details back to the user. python2.7 · api-gateway · slack
<b>simple-mobile-backend</b> A simple mobile backend (read/write to DynamoDB). nodejs · mobile	<b>kinesis-process-record-python</b> An Amazon Kinesis stream processor that logs the data being published. python2.7 · kinesis	<b>splunk-kinesis-logging</b> Demonstrates logging events streamed from AWS Kinesis to Splunk's HTTP Event Collector. nodejs · splunk · kinesis

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

### Configure function

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

Name\*

Description

Runtime\*

### 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  Upload a .ZIP file  Upload a file from Amazon S3

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

### Lambda function handler and role

Handler\*

Role\*

Ensure that popups are enabled to create a new role. Suggested role: Basic execution role

### Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)\*

Timeout\*

All AWS Lambda functions run securely inside a default system-managed VPC. However, you can optionally configure Lambda to access resources, such as databases, within your custom VPC. [Learn more](#) about accessing VPCs within Lambda. Please ensure your role has appropriate permissions to configure VPC. Select "Basic with VPC" in the role dropdown above to add these permissions.

VPC

\* These fields are required.

Cancel Previous Next

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

<https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions>HelloWorld?tab=endpoints>

**Add API endpoint**

Please go to the [IAM console](#) to configure the security for your API endpoint.

Configure your Lambda function to be invoked on requests made to an API endpoint.

**API endpoint type**: API Gateway

**API name**: LambdaMicroservice

**Resource name**: /HelloWorld

**Method**: GET

**Deployment stage**: prod

**Security**: AWS IAM

**Cancel** **Submit**

ARN - arn:aws:lambda:us-east-1:852612687751:function:HelloWorld

Lambda > Functions > HelloWorld

AWS Services Edit Jeremy Green N. Virginia Support

Feedback English © 2008 – 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

<https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions>HelloWorld?tab=eventSources>

**Add event source**

Configure your Lambda function to respond to events from the event sources listed below. You may also call your Lambda function directly using the AWS mobile SDK for [Android](#) and [iOS](#).

**Event source type**: **AWS IoT**

AWS IoT  
Alexa Skills Kit  
Alexa Smart Home  
CloudWatch Events - Schedule  
CloudWatch Logs  
Cognito Sync Trigger  
DynamoDB

**Cancel** **Submit**

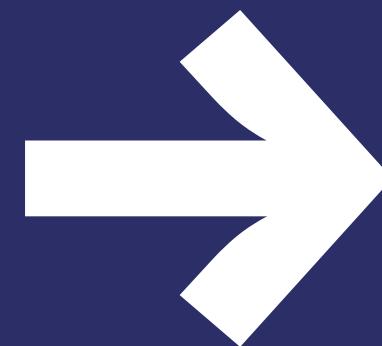
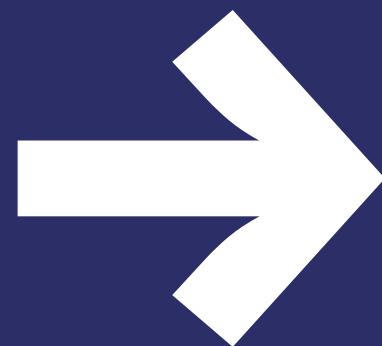
ARN - arn:aws:lambda:us-east-1:852612687751:function:HelloWorld

Lambda > Functions > HelloWorld

AWS Services Edit Jeremy Green N. Virginia Support

Feedback English © 2008 – 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

# Coding in the browser?



# Serverless to the rescue

[serverless.com](http://serverless.com)

Manage Lambda, API Gateway and CloudFormation  
via code and CLI instead of via GUI.

[docs.serverless.com](http://docs.serverless.com)

# Getting started

# First step

- Create a new AWS account!
- Srsly!

```
$ npm install -g serverless
```

```
$ sls project create
```



# The Serverless Application Framework

serverless.com, v0.5.5

```
Serverless: Initializing Serverless Project...
Serverless: Enter a name for this project: (serverless-bkqhpg) going-serverless-demo
Serverless: Enter a new stage name for this project: (dev)
Serverless: For the "dev" stage, do you want to use an existing Amazon Web Services
           profile or create a new one?
> Existing Profile
  Create A New Profile
Serverless: Select a profile for your project:
> lambdatest
Serverless: Creating stage "dev"...
Serverless: Select a new region for your stage:
> us-east-1
  us-west-2
  eu-west-1
  eu-central-1
  ap-northeast-1
Serverless: Creating region "us-east-1" in stage "dev"...
Serverless: Deploying resources to stage "dev" in region "us-east-1" via Cloudformation
           (~3 minutes)...
Serverless: /
```

```
Serverless: Successfully deployed "dev" resources to "us-east-1"
Serverless: Successfully created region "us-east-1" within stage "dev"
Serverless: Successfully created stage "dev"
Serverless: Successfully initialized project "going-serverless-demo"
```

```
$ cd going-serverless-demo
```

```
$ tree
```

```
•
├── admin.env # AWS Profiles - gitignored
├── package.json # npm package file
├── s-project.json # project and author data
└── s-resources-cf.json # CloudFormation template
    └── meta # meta data for stage/regions config and variables - gitignored
        ├── resources
        │   └── s-resources-cf-dev-useast1.json
        └── variables
            ├── s-variables-common.json
            ├── s-variables-dev-useast1.json
            └── s-variables-dev.json
```

3 directories, 8 files

**What should  
we build?**

Let's build  
something real\*

# l-paas

**left-pad as a service\***

\* OK, "real" 😈

```
$ sls function create left-pad
```

Serverless: Please, select a runtime for this new Function  
> nodejs4.3  
python2.7  
nodejs (v0.10, soon to be deprecated)

Serverless: For this new Function, would you like to create an Endpoint, Event, or just the Function?

> Create Endpoint

Create Event

Just the Function...

Serverless: Successfully created function: "left-pad"

```
$ tree left-pad/  
left-pad/  
└── event.json # sample event for testing function locally  
└── handler.js # function handler  
└── s-function.json # data for your lambda function, endpoints and event sources  
  
0 directories, 3 files
```



it!

```
$ sls dash deploy
```

```
$ sls dash deploy
```



## Serverless: Select the assets you wish to deploy:

# left-pad

## \* function - left-pad

# \* endpoint - left-pad - GET

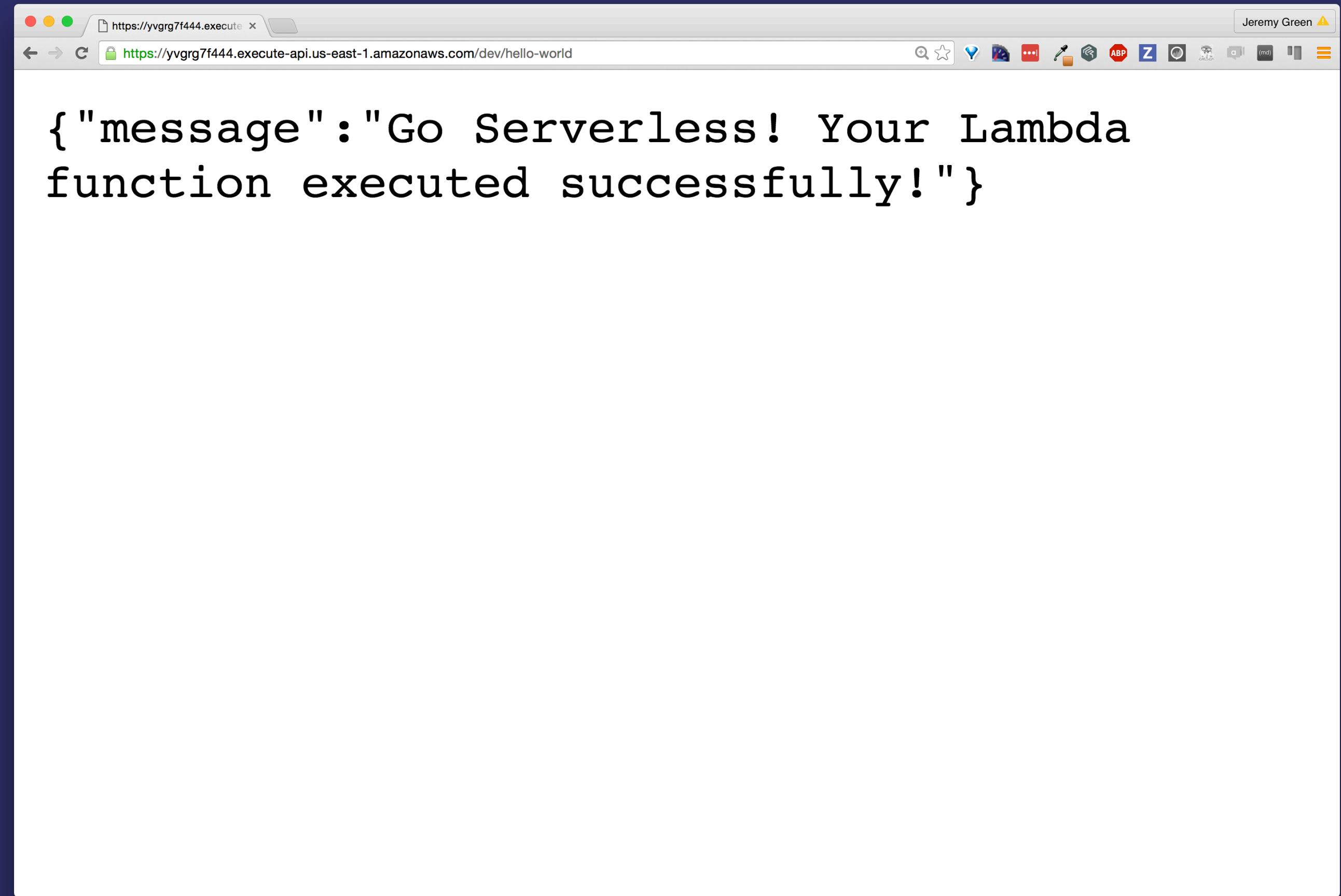
— — — — —

# > Deploy

Cancel

```
Serverless: Deploying the specified functions in "dev" to the following regions: us-east-1
Serverless: -----
Serverless: Successfully deployed the following functions in "dev" to the following regions:
Serverless: us-east-1 -----
Serverless:   left-pad (going-serverless-demo-left-pad):
  arn:aws:lambda:us-east-1:852612687751:function:going-serverless-demo-left-pad:dev

Serverless: Deploying endpoints in "dev" to the following regions: us-east-1
Serverless: Successfully deployed endpoints in "dev" to the following regions:
Serverless: us-east-1 -----
Serverless:   GET - left-pad -
  http://serverless.octolabs.com/left-pad
```



# Anatomy of a Lambda Function

# left-pad/handler.js

```
'use strict';

module.exports.handler = function(event, context, cb) {
  return cb(null, {
    message: 'Go Serverless! Your Lambda function executed successfully!'
  });
};
```

# event

used to pass data to the function

```
{  
  string: 'test',  
  padding: 10  
}
```

# context

provides runtime information

```
{  
  getRemainingTimeInMillis: function(){},  
  functionName: 'handler',  
  functionVersion: 'xyz',  
  awsRequestId: '12345',  
  identity: {...},  
  clientContext: {...}  
}
```

# callback

used to return data or an error

```
// signature  
callback(error,data);  
// call with error  
callback("some error message");  
//or call with data  
callback(null, someData);
```

```
$ cd left-pad  
  
$ npm init  
...  
$ npm install left-pad --save
```

# left-pad/handler.js

```
let letpad = require('left-pad');

module.exports.handler = function(event, context, cb) {

};

};
```

# left-pad/handler.js

```
let letpad = require('left-pad');

module.exports.handler = function(event, context, cb) {
  var string = event.string || "",
    padding = event.padding || 0,
};

};
```

# left-pad/handler.js

```
let letpad = require('left-pad');

module.exports.handler = function(event, context, cb) {
  var string = event.string || "",
    padding = event.padding || 0,
    paddedString = letpad(string, padding),
    payload = { paddedString };

};

};
```

# left-pad/handler.js

```
let letpad = require('left-pad');

module.exports.handler = function(event, context, cb) {
  var string = event.string || "",
    padding = event.padding || 0,
    paddedString = letpad(string, padding),
    payload = { paddedString };
  return cb(null, payload);
};
```

# left-pad/s-function.json (simplified)

```
{  
  "runtime": "nodejs4.3",  
  "handler": "handler.handler",  
  "memorySize": 1024  
  "endpoints": [  
  
  ]  
}
```

# left-pad/s-function.json (simplified)

```
{  
  "runtime": "nodejs4.3",  
  "handler": "handler.handler",  
  "memorySize": 1024  
  "endpoints": [  
    {  
      "path": "left-pad",  
      "method": "GET",  
      "requestTemplates": {  
        "application/json": {  
          }  
      }  
    }  
  ]  
}
```

# left-pad/s-function.json (simplified)

```
{  
  "runtime": "nodejs4.3",  
  "handler": "handler.handler",  
  "memorySize": 1024  
  "endpoints": [  
    {  
      "path": "left-pad",  
      "method": "GET",  
      "requestTemplates": {  
        "application/json": {  
          "string": "$input.params('string')",  
          "padding": "$input.params('padding')"  
        }  
      }  
    }  
  ]  
}
```

```
$ sls dash deploy  
...  
Serverless:    GET - left-pad -  
http://serverless.octolabs.com/left-pad
```

[http://serverless.octolabs.com/left-pad?  
string=test&padding=10](http://serverless.octolabs.com/left-pad?string=test&padding=10)

```
{  
  "paddedString": "      test"  
}
```

# Testing

```
$ npm install -g mocha
```

```
$ npm install --save-dev chai
```

# left-pad/handler\_test.js

```
var expect = require("chai").expect;
var handler = require("./handler.js");

describe('handler', function(){
  it('returns the right thing', function(){
    });
  });
});
```

# left-pad/handler\_test.js

```
var expect = require("chai").expect;
var handler = require("./handler.js");

describe('handler', function(){
  it('returns the right thing', function(){
    var event = {
      string: 'test',
      padding: 10
    };
    expect(handler(event)).to.equal("  test");
  });
});
```

# left-pad/handler\_test.js

```
var expect = require("chai").expect;
var handler = require("./handler.js");

describe('handler', function(){
  it('returns the right thing', function(){
    var event = {
      string: 'test',
      padding: 10
    };
    var context = {};
    });
  });
});
```

# left-pad/handler\_test.js

```
var expect = require("chai").expect;
var handler = require("./handler.js");

describe('handler', function(){
  it('returns the right thing', function(){
    var event = {
      string: 'test',
      padding: 10
    };
    var context = {};
    var cb = function(error, response){
      expect(error).to.be.null;
      expect(response.paddedString).to.equal("      test");
    };
    });
  });
});
```

# left-pad/handler\_test.js

```
var expect = require("chai").expect;
var handler = require("./handler.js");

describe('handler', function(){
  it('returns the right thing', function(){
    var event = {
      string: 'test',
      padding: 10
    };
    var context = {};
    var cb = function(error, response){
      expect(error).to.be.null;
      expect(response.paddedString).to.equal("      test");
    };
    handler.handler(event, context, cb);
  });
});
```

```
$ mocha handler_test.js
```

handler

✓ returns the right thing

1 passing (9ms)

**what about  
Rails?**

```
mruby-hello-world$ tree .
```

```
+
├── event.json
├── handler.js
├── handler.rb          # A ruby script! :D
├── mruby                # The mruby executable
└── s-function.json
```

```
0 directories, 5 files
```

# mruby-hello-world/handler.js

```
var spawn = require('child_process').spawn;

module.exports.handler = function(event, context, callback) {
  var child = spawn('./mruby', ['handler.rb', JSON.stringify(event, null, 2)]);
  var rubyOutput = [];
  child.stdout.on('data', function (data) { rubyOutput.push(data.toString()); });
  child.stderr.on('data', function (data) { rubyOutput.push(data.toString()); });
  child.on('close', function (code) {
    callback(null, {
      message: "We're back from ruby land",
      rubyOutput: rubyOutput
    });
  });
}
```

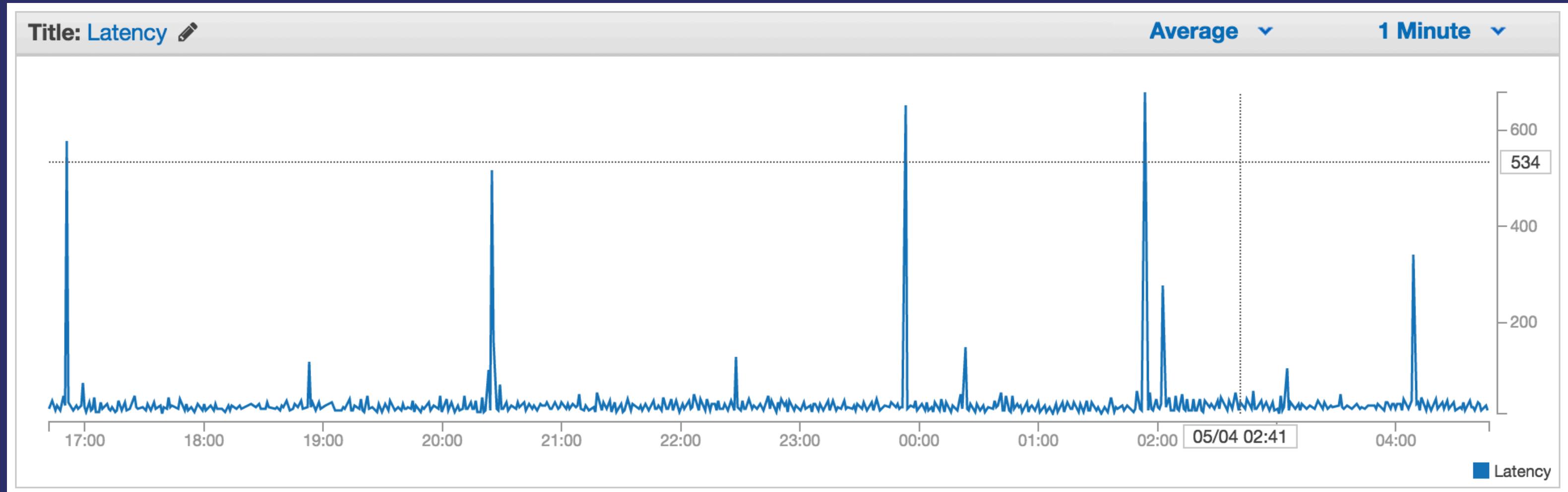
# mruby-hello-world/handler.rb

```
puts 'Hello, Lambda from Ruby!'
puts ARGV
```

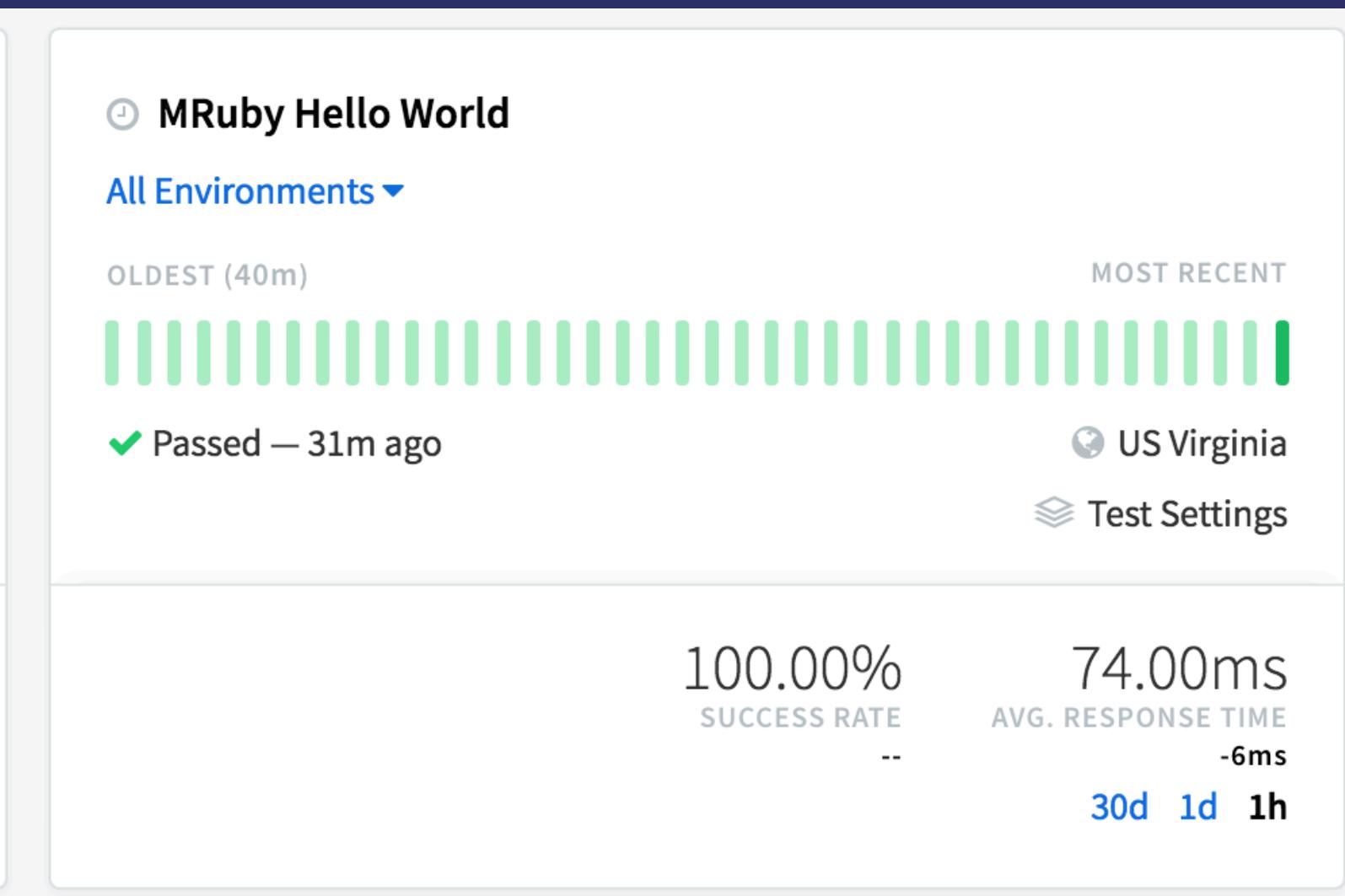
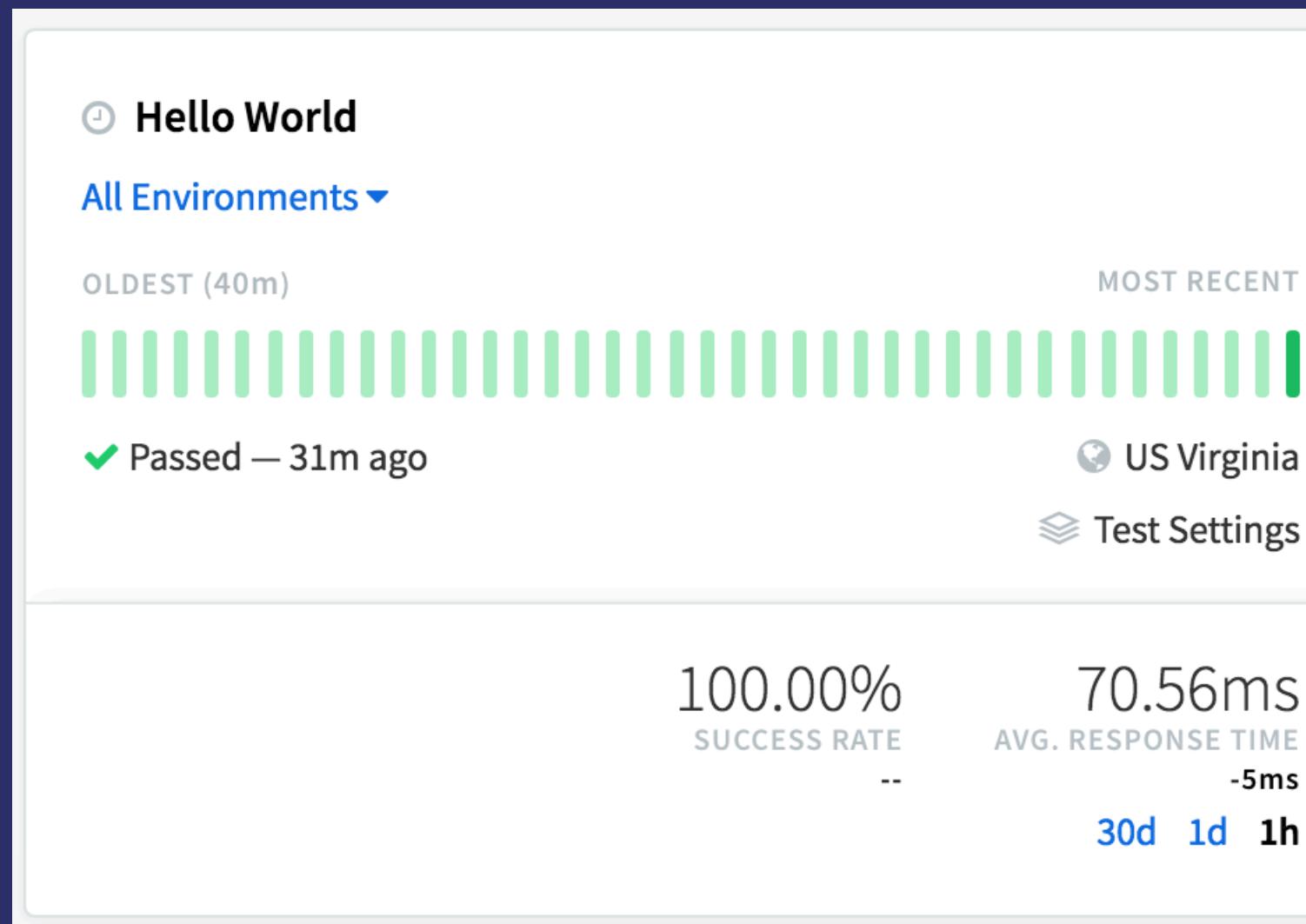
<http://serverless.octolabs.com/mruby-hello-world>

```
{  
  "message": "We're back from ruby land",  
  "rubyOutput": ["Hello, Lambda from Ruby!\n[\"{}\"]\n"]  
}
```

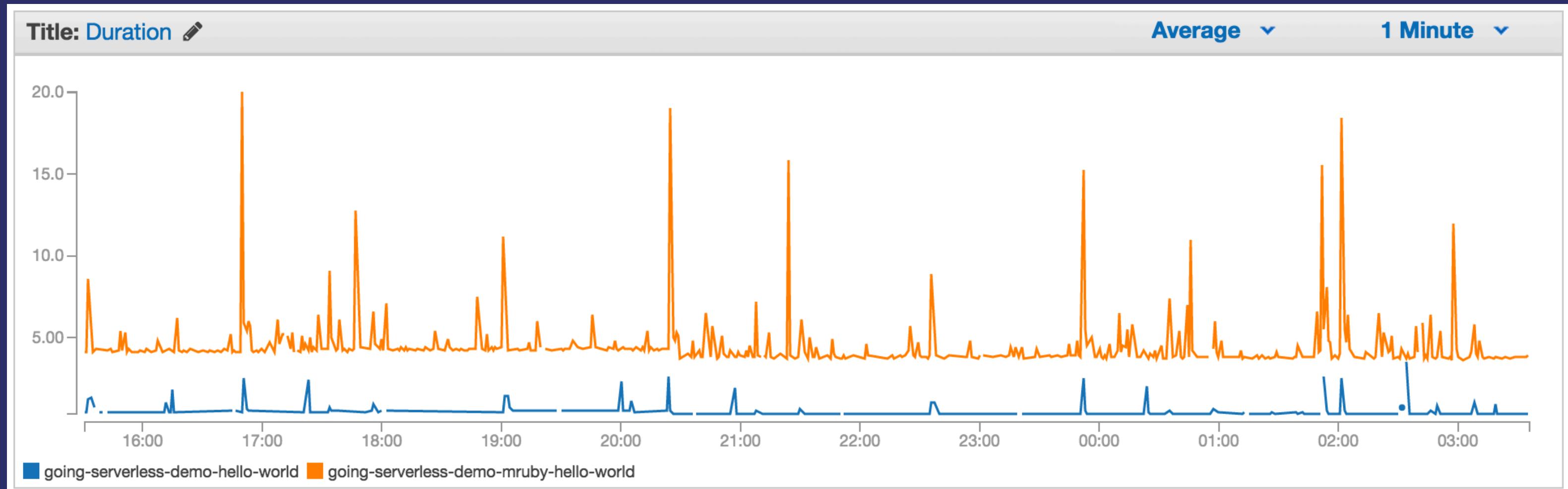
# API Gateway Latency



# Remote API call timing comparison



# Lambda timing comparison



# Wrapping Up

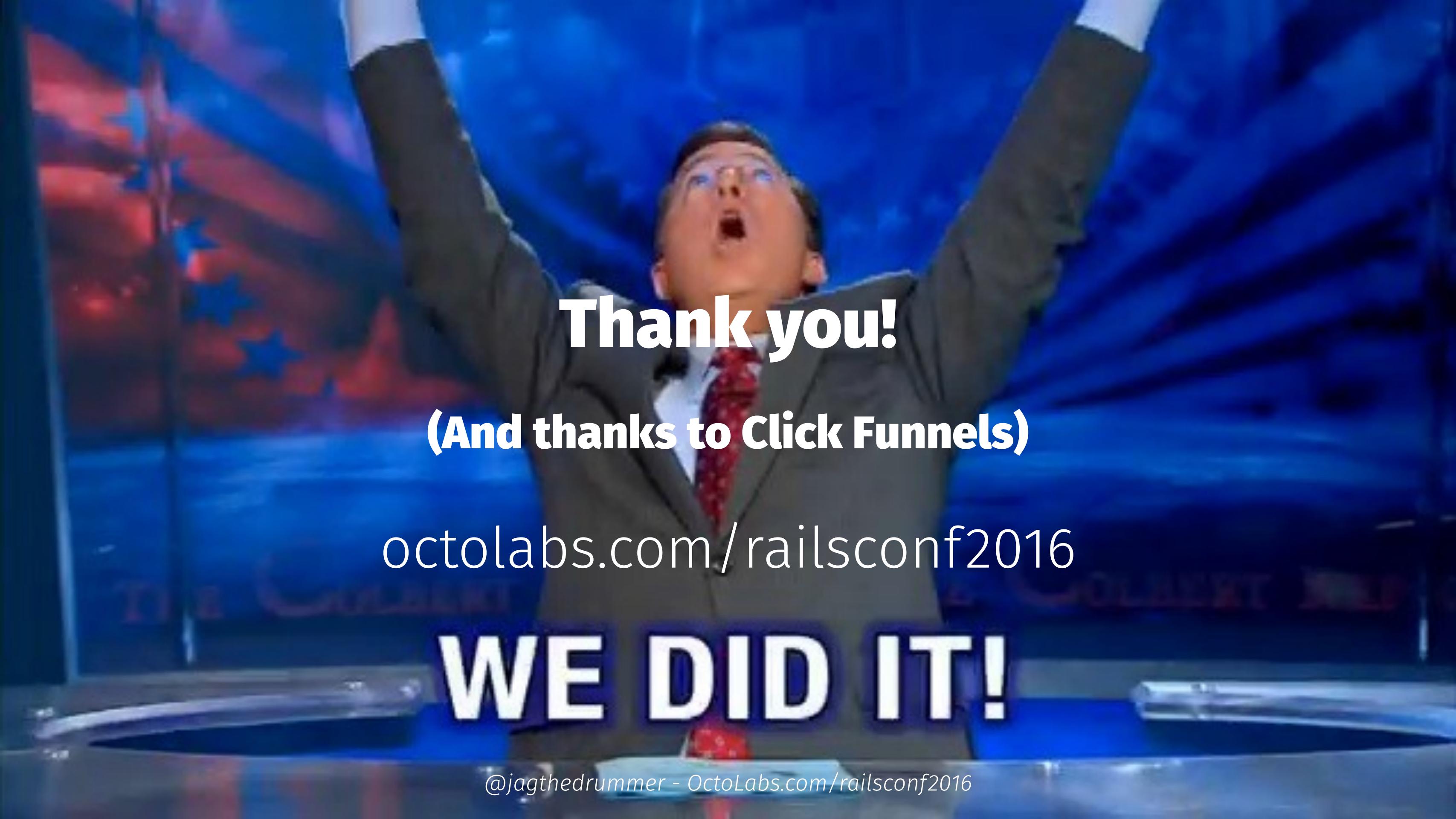
# AWS Provides the Building Blocks

# **Serverless Provides Structure and Process**

# You provide the



@jagthedrummer - OctoLabs.com/railsconf2016



**Thank you!**  
**(And thanks to Click Funnels)**

[octolabs.com/railsconf2016](http://octolabs.com/railsconf2016)

**WE DID IT!**

# Bonus slides

# Lambda Pricing 💰

Charged by:

- # of requests
- length of execution

# Lambda Request Pricing

First 1 million requests per month are free

\$0.20 per 1 million requests thereafter  
(\$0.0000002 per request)

# Lambda Execution Pricing

First 400,000 GB-seconds per month are free  
\$0.00001667 for per GB-second thereafter

GB-second

=

LambdaMemoryInGigabytes \* ExecutionTime

1GB \* 1sec = 1 GB-sec

0.5GB \* 2sec = 1 GB-sec

0.5GB \* 0.5sec \* 4executions = 1 GB-sec

# Lambda Pricing Example

3 million executions per month

512MB of memory

1 second execution time

\$18.74 per month

# Lambda Pricing Example

3 million executions per month

512MB of memory

**0.5 second execution time**

\$6.24 per month

# Lambda Pricing Example

3 million executions per month

512MB of memory

**0.2 second execution time**

\$0.40 per month

# Lambda Lifecycle



1. You upload your code
2. Amazon doesn't do anything

# Lambda Cold Start ❄

1. AWS Receives Execution Request
2. Container is provisioned
3. Container is loaded with your code
4. Your code begins execution
5. Your code returns a result

# Time Passes



(But not too much)

# Lambda Container Reuse

1. AWS Receives Execution Request
2. Your code begins execution
3. Your code returns a result

# Possible Architectures

- Monolithic
- Microservices
- Nanoservices

# Monolithic Architecture

- A single Lambda handles multiple concerns
- Multiple API Gateway endpoints map to one Lambda
- Cold start will be slow

# Microservices Architecture

- A single Lambda function for each concern / resource
- Multiple API Gateway endpoints map to multiple Lambdas
- Cold start not as slow

# Nanoservices Architecture

- A single Lambda function for each logical function
- One-to-one mapping between API Gateway & Lambda
  - Fastest cold start