

Going Serverless



@jagthedrummer - OctoLabs.com



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



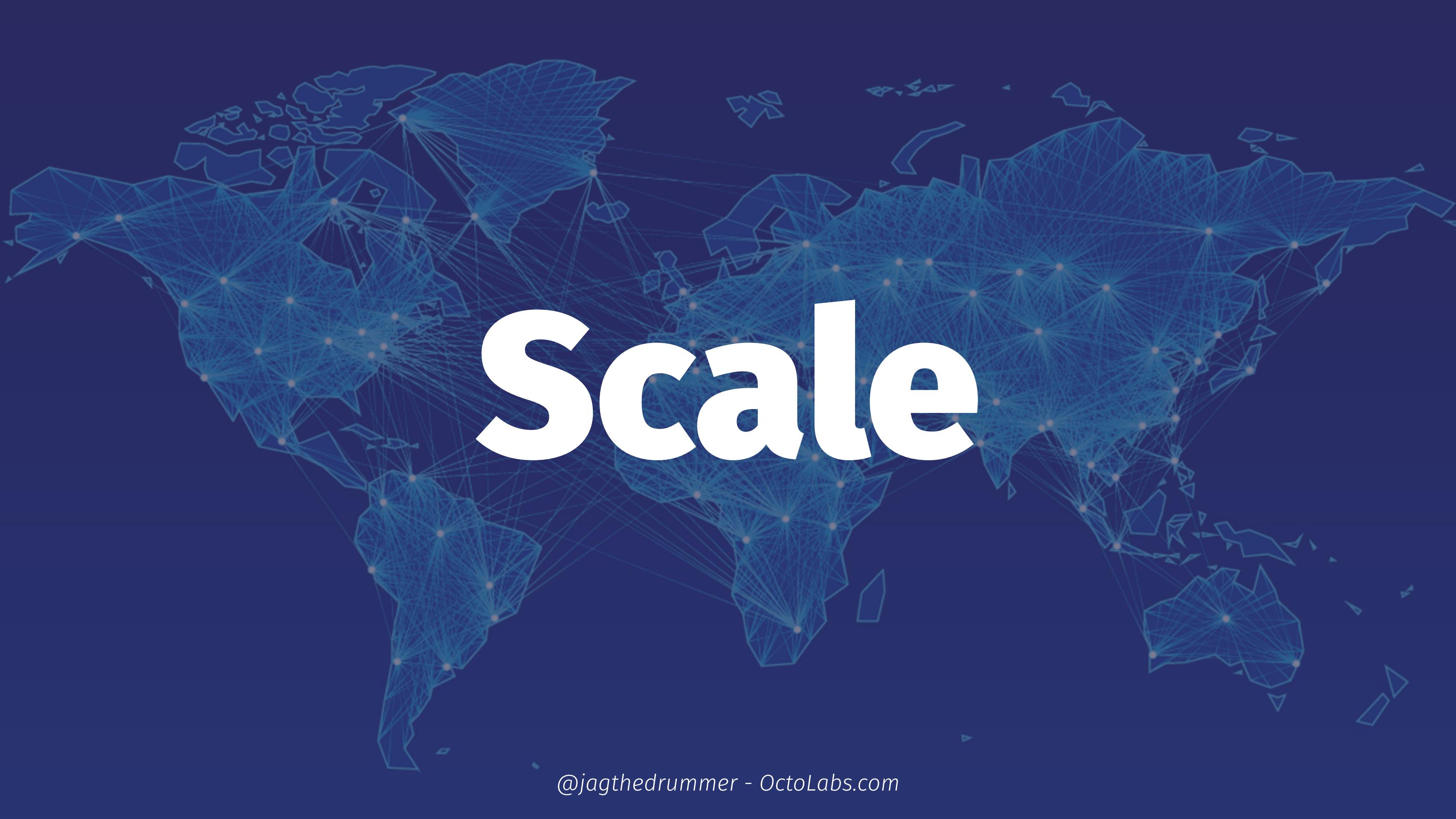
@jagthedrummer - OctoLabs.com

Vendor Lock-in 😟

A photograph of two hands, belonging to different people, joined together by a pair of silver handcuffs. The hands are positioned with the palms facing each other. The background is a clear blue sky with some wispy white clouds. The overall composition suggests a sense of confinement or dependency.



OPS



Scale



Money

@jagthedrummer - OctoLabs.com

Jeremy Green

Consultant, Author, SaaSer



@jagthedrummer
jeremy@octolabs.com

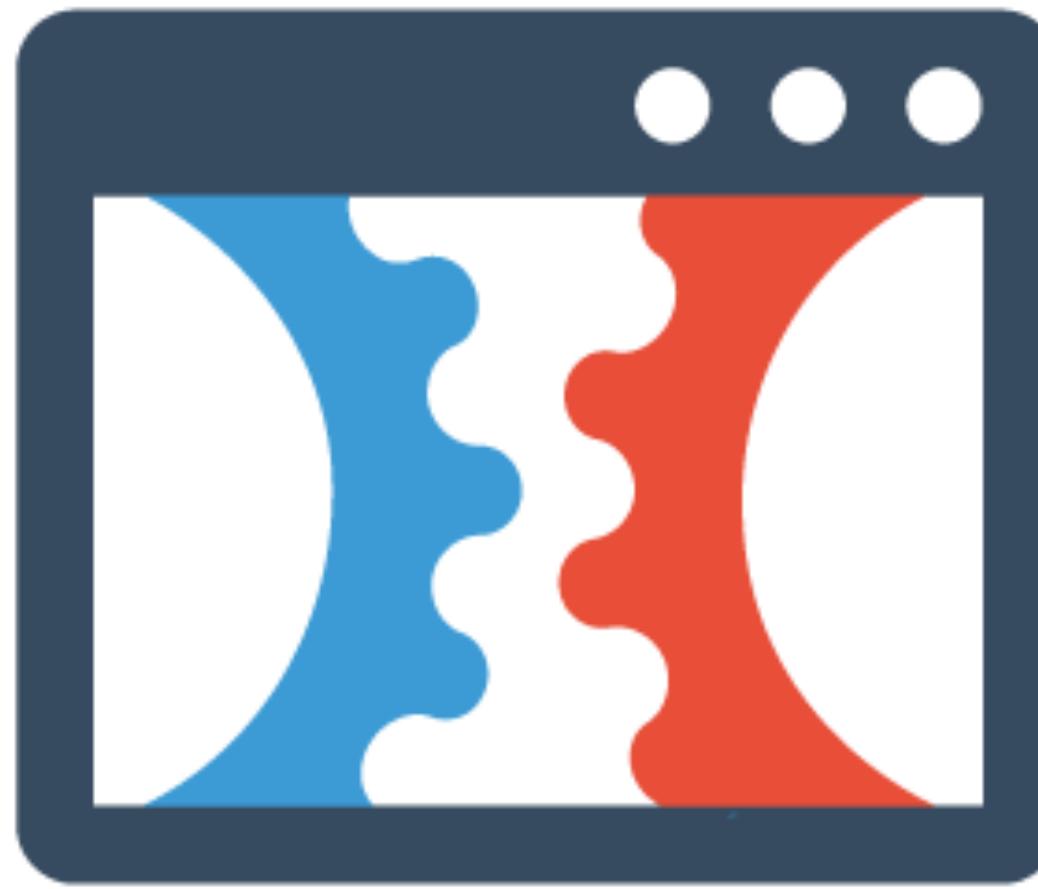
IndependentConsultingManual.com

Remarq.io

CloudHdr.com

Things I Enjoy:
Dopamine, Serotonin

Other Interests:
Drumming, Photography, and Brewing



click funnels



The Pieces

Lambda

API Gateway

DynamoDB

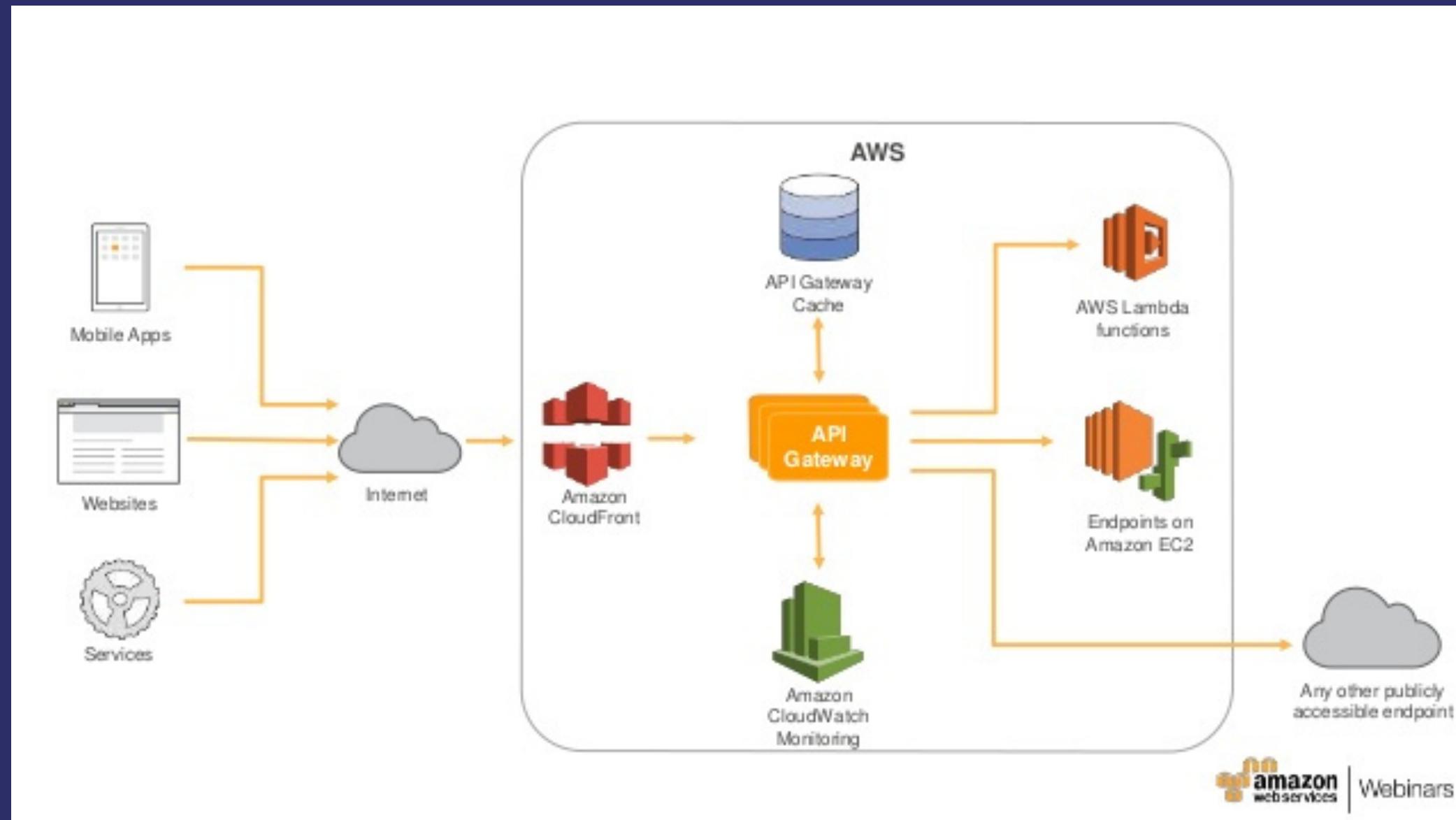
RDS

@jagthedrummer - OctoLabs.com

CloudFormation

@jagthedrummer - OctoLabs.com

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 x Google 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

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

The screenshot shows the 'Configure function' step of creating a new Lambda function. The 'Name*' field is set to 'myFunctionName'. The 'Description' field contains 'A starter AWS Lambda function.'. The 'Runtime*' dropdown is set to 'Node.js 4.3'. In the 'Lambda function code' section, the 'Code entry type' is set to 'Edit code inline'. The code editor displays the following Node.js code:

```
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};
```

The screenshot shows the 'Lambda function handler and role' configuration step. The 'Handler*' field is set to 'index.handler'. The 'Role*' dropdown is open, with a note below it stating 'Ensure that popups are enabled to create a new role. Suggested role: Basic execution role'. Under 'Advanced settings', the 'Memory (MB)*' is set to 128, and the 'Timeout*' is set to 0 min 3 sec. The 'VPC' dropdown is set to 'No VPC'. A note at the bottom states: '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.' At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

<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**

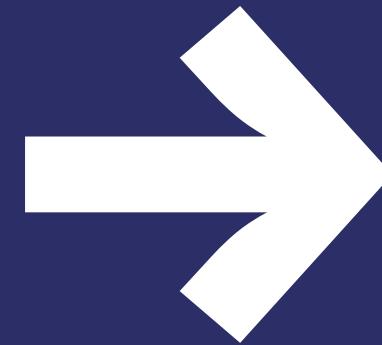
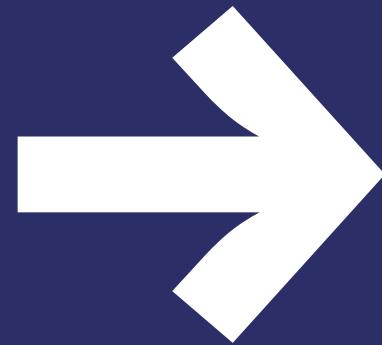
ARN - arn:aws:lambda:us-east-1:852612687751:function:HelloWorld

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?



@jagthedrummer - OctoLabs.com

Serverless to the rescue

serverless.com

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

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
Serverless: Enter a new function name to be created in the CWD: 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

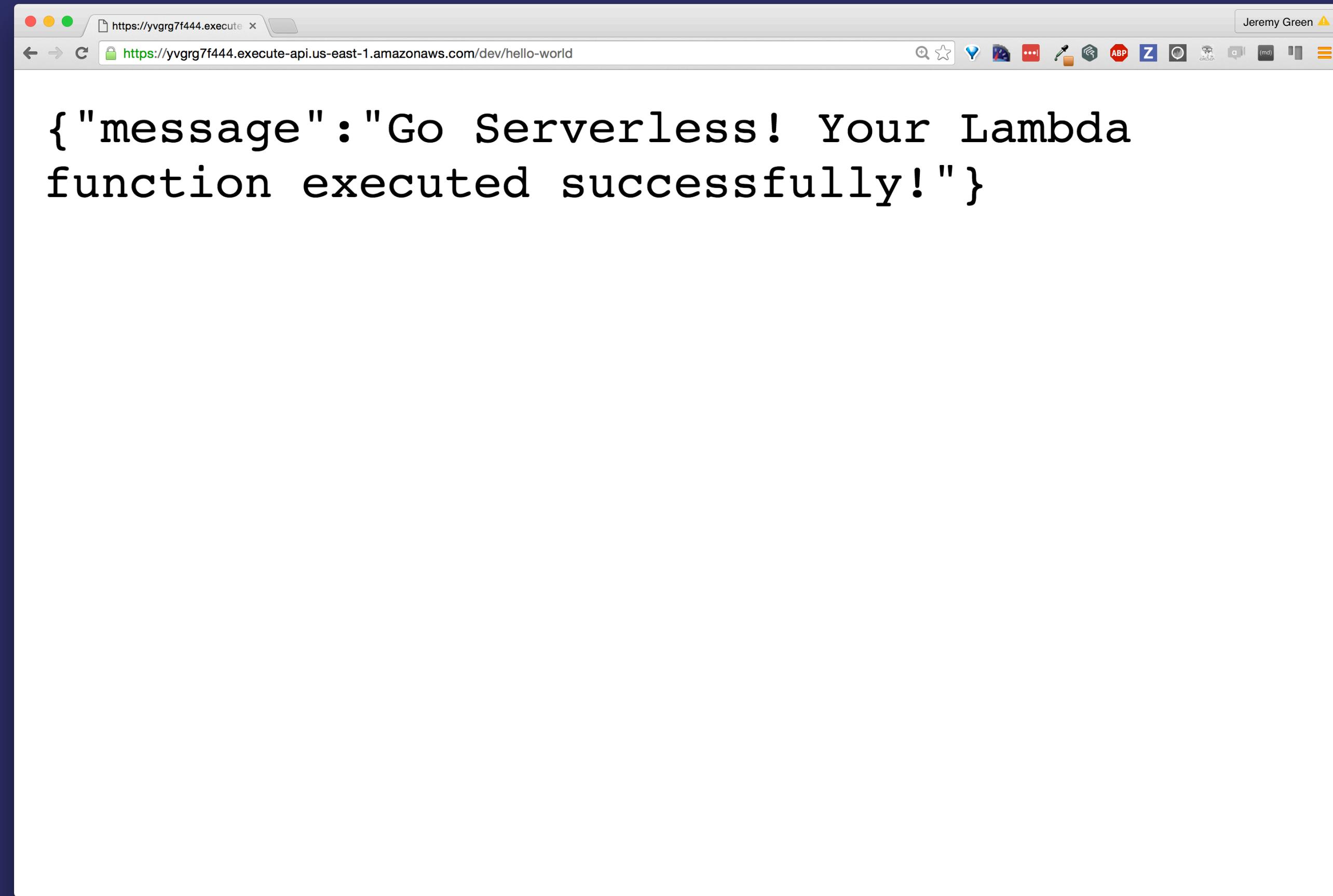
— — — — —

> 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 -
  https://yvgrg7f444.execute-api.us-east-1.amazonaws.com/dev/left-pad
```



left-pad/handler.js

```
'use strict';

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

Anatomy of a Lambda Function

event

used to pass data to the function

```
{  
  key1: 'value1',  
  key2: 'value2',  
  key3: 'value3'  
}
```

context

provides runtime information

```
{  
  getRemainingTimeInMillis: function(){},  
  functionName: 'handler',  
  functionVersion: 'xyz',  
  awsRequestId: '12345',  
  logGroupName: 'abcde',  
  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);
```

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

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

left-pad/handler.js

```
'use strict';

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": [  
    {  
      "path": "left-pad",  
      "method": "GET",  
      "requestTemplates": {  
        "application/json": {  
          "string": "$input.params('string')",  
          "padding": "$input.params('padding')"  
        }  
      }  
    }  
  ]  
}
```

```
$ sls dash deploy  
...  
Serverless:    GET - left-pad -  
https://yvgrg7f444.execute-api.us-east-1.amazonaws.com/dev/left-pad
```

[https://yvgrg7f444.execute-api.us-east-1.amazonaws.com/dev/left-pad?
string=test&padding=10](https://yvgrg7f444.execute-api.us-east-1.amazonaws.com/dev/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(){
    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)

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

Wrapping Up

AWS Provides the Building Blocks

Serverless Provides Structure and Process

You provide the



@jagthedrummer - OctoLabs.com

A man in a dark suit and red patterned tie stands on a stage with his arms raised in triumph. He is shouting into a microphone. The background shows a large stadium with blue lights and a banner that reads "WE DID IT!".

Thank you!

octolabs.com/goingserverless-okcjs

WE DID IT!