THE DEVELOPER'S CONFERENCE

A Glance Over the Serverless Framework

Rafael Zotto
Senior Software Architect, HP Inc.

Short Bio

THE DEVELOPER'S CONFERENCE

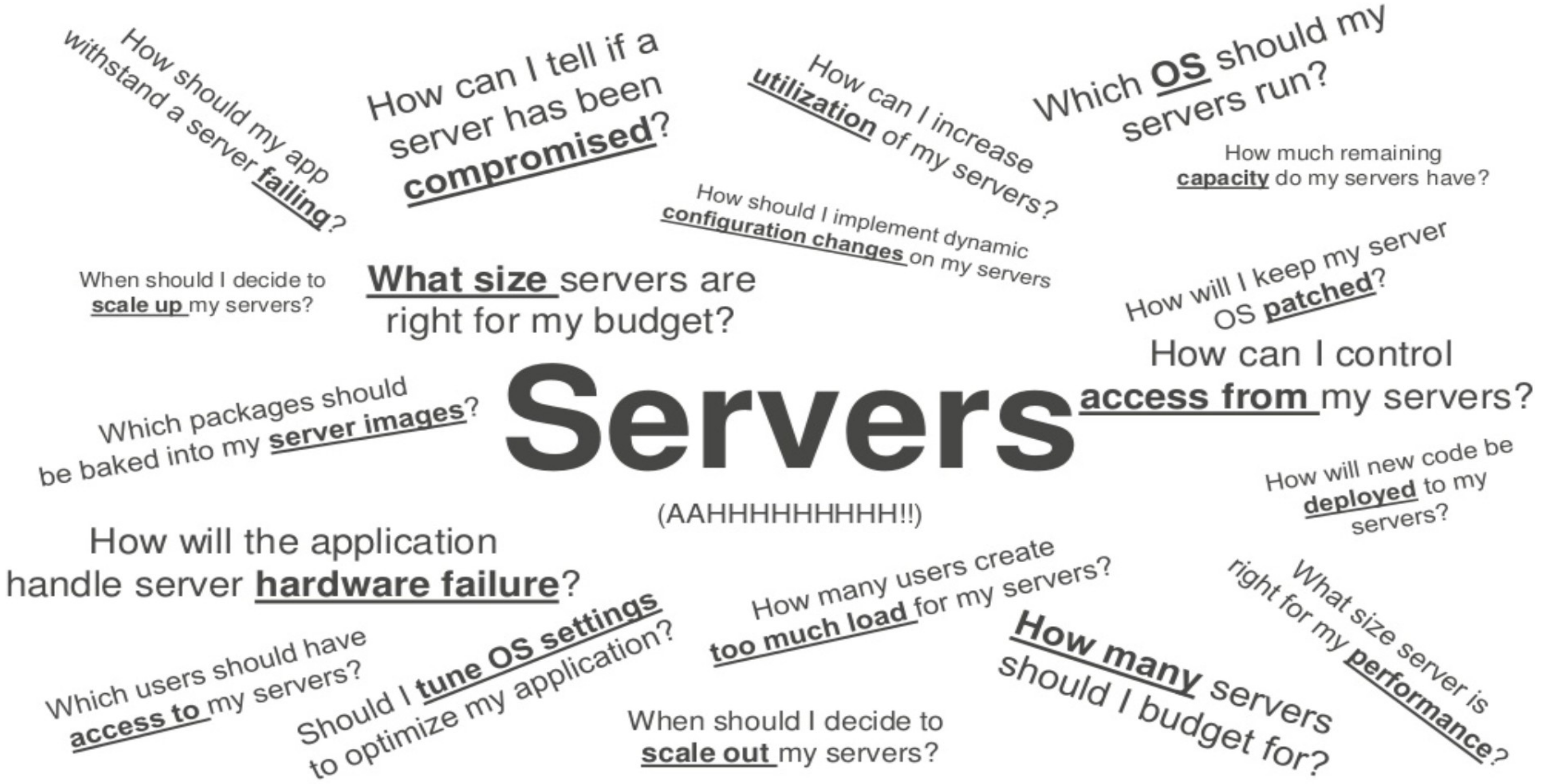
Rafael Zotto

Holds a master degree in Computer Science focused in high performance computing. Specialized in parallel and distributed computing with special interest in mobile and web technologies. Works for HP Inc. for the past decade acting as senior software architect for print firmware and wearable technologies. Recently joined the Data Science research team in Porto Alegre, Brazil.

Agenda

THE DEVELOPER'S CONFERENCE

- Background
- Serverless Framework: 10,000 Foot Overview
- Installation
- Demo



How should my app withstand a server **failing**?

How can I tell if a server has been **compromised**?

How can I increase **utilization** of my servers?

Which **OS** should my servers run?

How much remaining **capacity** do my servers have?

How should I implement dynamic **configuration changes** on my servers

When should I decide to **scale up** my servers?

What size servers are right for my budget?

How will I keep my server OS **patched**?

How can I control **access from** my servers?

Which packages should be baked into my **server images**?

Servers

(AAHHHHHHHHH!!)

How will new code be **deployed** to my servers?

How will the application handle server **hardware failure**?

How many users create **too much load** for my servers?

What size server is right for my **performance**?

Which users should have **access to** my servers?

Should I **tune OS settings** to optimize my application?

When should I decide to **scale out** my servers?

How many servers should I budget for?

* Image from "*Getting Started with Serverless Architecture*" presentation. Amazon Web Services.

Serverless Definition

- Platform to develop, run and manage applications without the complexity of building and maintaining infrastructure.

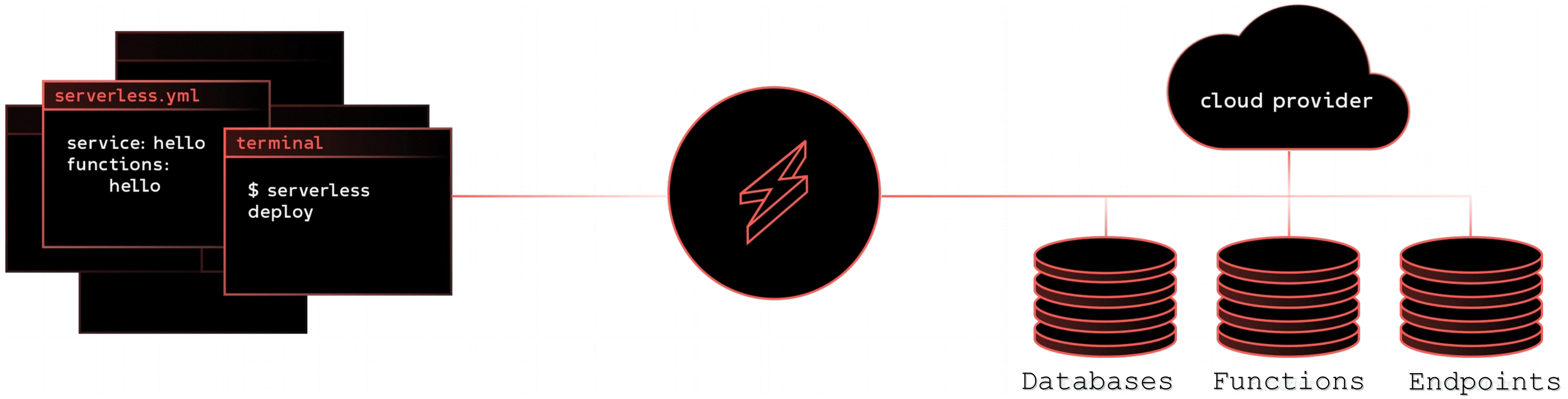
- No free lunch!
  - You will pay for it.
  - Sub-second billing

Architect to be Serverless

- Fully Managed
  - No provisioning, zero administration, high-available
- Developer Productivity
  - Focus on what matters, innovate quickly
- Continuous Scaling
  - Up and Down automatically

Serverless Framework

- *"The easy, open way to build serverless applications"*



serverless.yml

```
service: hello
functions:
    hello
```

terminal

```
$ serverless
deploy
```

Framework Pillars

- Infrastructure as Code
  - .yml file for definitions
- Simple Serverless Development
  - Intuitive CLI experience
- Provider Agnostic
  - Main Cloud Providers supported

Main Features

THE DEVELOPER'S CONFERENCE

- Multi Lingual
  - Pick your poison: python, node.js, java, go, scala, C#, ...
- Robust Ecosystem
  - Hundred of plugins
- Cloud Agnostic
  - AWS, Azure, IBM, Google Cloud,...
- Streaming Logs
  - Easy troubleshoot
- Lifecycle Management
  - Local development, stages, rollback, ...

THE DEVELOPER'S CONFERENCE

Getting Started

```
npm install -g serverless
```

```
serverless login
```

Choose your provider:

amazon web services™

Microsoft Azure

Google Cloud Platform

IBM OpenWhisk

Kubeless

fn

spotinst

CLOUDFLARE

Create a new service

- `sls create --template %template_id%`

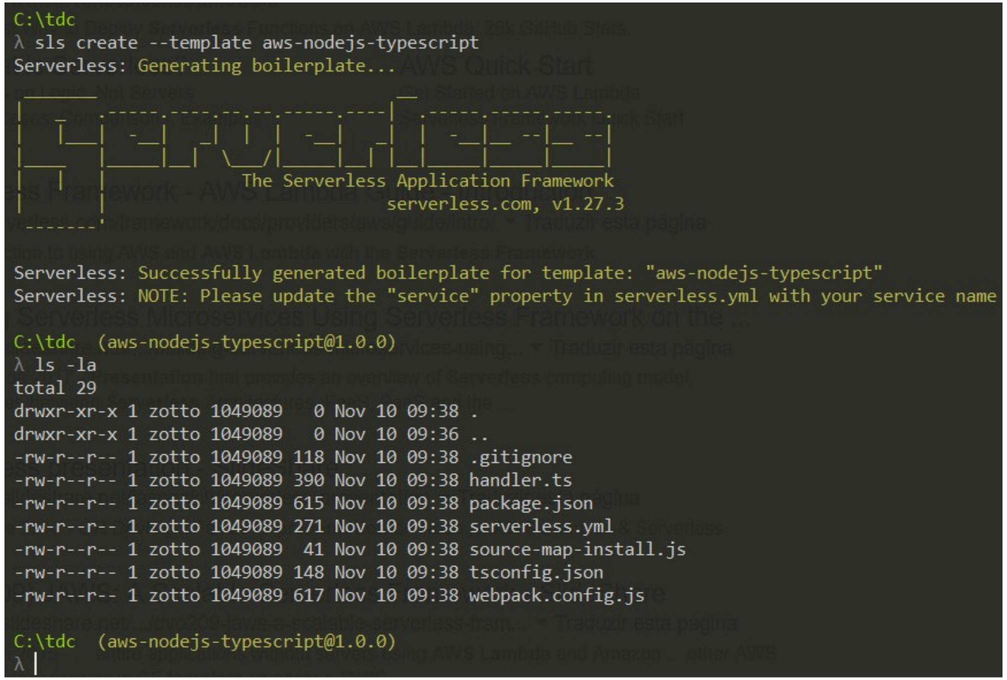
- Multiple templates available

  "aws-nodejs", "aws-nodejs-typescript", "aws-nodejs-ecma-script", "aws-python", "aws-python3", "aws-groovy-gradle", "aws-java-maven", "aws-java-gradle", "aws-kotlin-jvm-maven", "aws-kotlin-jvm-gradle", "aws-kotlin-nodejs-gradle", "aws-scala-sbt", "aws-csharp", "aws-fsharp", "aws-go", "aws-go-dep", "azure-nodejs", "fn-nodejs", "fn-go", "google-nodejs", "kubeless-python", "kubeless-nodejs", "openwhisk-java-maven", "openwhisk-nodejs", "openwhisk-php", "openwhisk-python", "openwhisk-swift", "spotinst-nodejs", "spotinst-python", "spotinst-ruby", "spotinst-java8", "webtasks-nodejs", "plugin" and "hello-world"

What is Created?

- A “ready-to-go” service!

- Lambda Function
- API Gateway

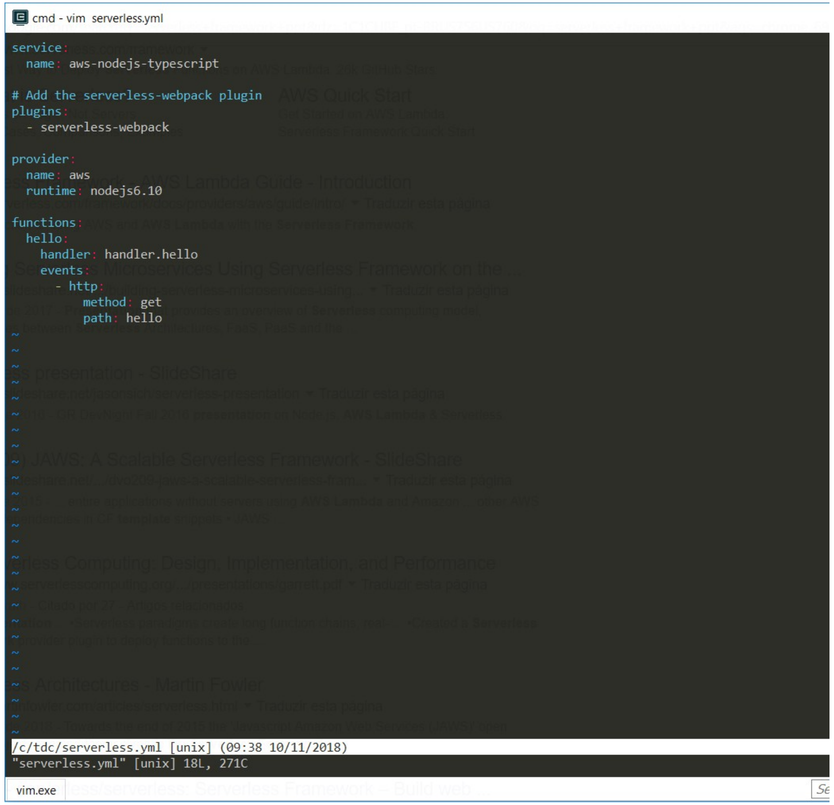


```
C:\tdc
λ sls create --template aws-nodejs-typescript
Serverless: Generating boilerplate...
 _______                             __
|   _   .-----.----.--.--.-----.----|  .-----.-----.-----.
|   |___|  -__|   _|  |  |  -__|   _|  |  -__|__ --|__ --|
|____   |_____|__|  \___/|_____|__| |__|_____|_____|_____|
|   |   |             The Serverless Application Framework
|       |                           serverless.com, v1.27.3
 -------'

Serverless: Successfully generated boilerplate for template: "aws-nodejs-typescript"
Serverless: NOTE: Please update the "service" property in serverless.yml with your service name

C:\tdc  (aws-nodejs-typescript@1.0.0)
λ ls -la
total 29
drwxr-xr-x 1 zotto 1049089   0 Nov 10 09:38 .
drwxr-xr-x 1 zotto 1049089   0 Nov 10 09:36 ..
-rw-r--r-- 1 zotto 1049089 118 Nov 10 09:38 .gitignore
-rw-r--r-- 1 zotto 1049089 390 Nov 10 09:38 handler.ts
-rw-r--r-- 1 zotto 1049089 615 Nov 10 09:38 package.json
-rw-r--r-- 1 zotto 1049089 271 Nov 10 09:38 serverless.yml
-rw-r--r-- 1 zotto 1049089  41 Nov 10 09:38 source-map-install.js
-rw-r--r-- 1 zotto 1049089 148 Nov 10 09:38 tsconfig.json
-rw-r--r-- 1 zotto 1049089 617 Nov 10 09:38 webpack.config.js

C:\tdc  (aws-nodejs-typescript@1.0.0)
λ
```

.yml Quick Peek

THE DEVELOPER'S CONFERENCE

- Service Stack Name
- Cloud Provider
- Default Runtime
- Functions
- Events



cmd - vim serverless.yml

```
service:
  name: aws-nodejs-typescript

# Add the serverless-webpack plugin
plugins:
  - serverless-webpack

provider:
  name: aws
  runtime: nodejs6.10

functions:
  hello:
    handler: handler.hello
    events:
      - http:
          method: get
          path: hello
~
/c/tdc/serverless.yml [unix] (09:38 10/11/2018)
"serverless.yml" [unix] 18L, 271C
```

Deploy, Test and Diagnose

- Service
  - `sls deploy -v`

- Function
  - `sls deploy function -f %function_name%`

Deploy, **Test** and Diagnose

THE DEVELOPER'S CONFERENCE

- Remote Invoke
  - `sls invoke -f %function_name%`
- Local Invoke
  - `sls invoke local -f %function_name%`
- *Option to pass input data*

Deploy, Test and **Diagnose**

THE DEVELOPER'S CONFERENCE

- Retrieve remote logs
  - `sls logs -f %function_name%`

- *Options to tail, filter and pooling.*

Cleanup

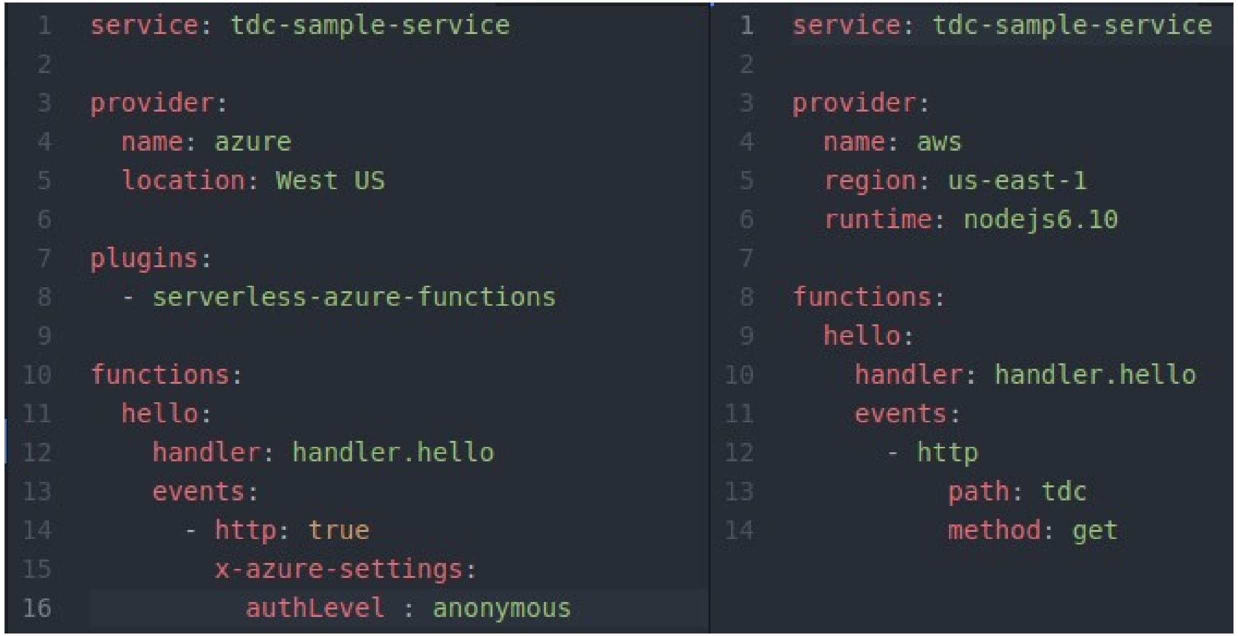
THE DEVELOPER'S CONFERENCE

- Remove the stack completely
  - `sls remove`

- Heads Up!
  - Removing and re-deploying cause the cloud IDs to change!

Changing Provider

- Adjust .yml file

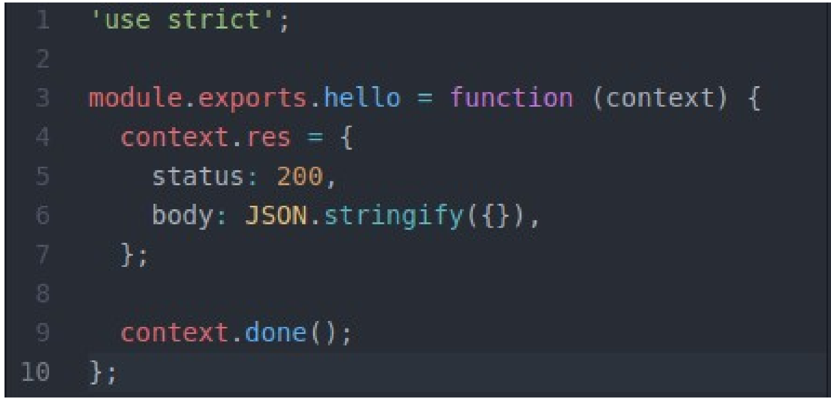


```yaml
service: tdc-sample-service

provider:
  name: azure
  location: West US

plugins:
  - serverless-azure-functions

functions:
  hello:
    handler: handler.hello
    events:
      - http: true
        x-azure-settings:
          authLevel : anonymous
```

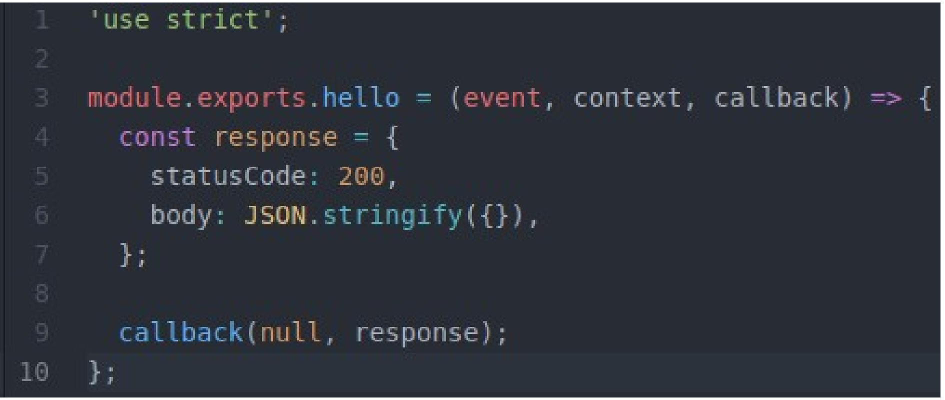
```yaml
service: tdc-sample-service

provider:
  name: aws
  region: us-east-1
  runtime: nodejs6.10

functions:
  hello:
    handler: handler.hello
    events:
      - http
          path: tdc
          method: get
```

Changing Provider

- Adjust entry point (handler)



```
'use strict';

module.exports.hello = function (context) {
  context.res = {
    status: 200,
    body: JSON.stringify({}),
  };

  context.done();
};
```



```
'use strict';

module.exports.hello = (event, context, callback) => {
  const response = {
    statusCode: 200,
    body: JSON.stringify({}),
  };

  callback(null, response);
};
```

Useful Resources

THE DEVELOPER'S CONFERENCE

- https://serverless.com/framework/
- https://serverless.com/framework/docs/getting-started/
- https://github.com/serverless/examples

THE DEVELOPER'S CONFERENCE