Building serverless apps with Node.js

Julien Simon, Principal Technical Evangelist, AWS
julsimon@amazon.fr
@julsimon

amazon web services

Evolution of Computing

Weeks

On-premise

Minutes

Virtual Machines

Amazon EC2

Seconds

Containers

Amazon ECS

Werner Vogels, CTO, Amazon.com
AWS re:Invent 2015

amazon web services

AWS Lambda

- Announced at re:Invent 2014
- Deploy pure functions in Java, Python, Node.js and C#
- Just code, without the infrastructure drama
- Built-in scalability and high availability
- Integrated with many AWS services
- Pay as you go
  - Combination of execution time (100ms slots) & memory used
  - Starts at $0.000000208 per 100ms
  - Free tier available: first 1 million requests per month are free

http://aws.amazon.com/lambda
AWS re:Invent 2014 | (MBL202) NEW LAUNCH: Getting Started with AWS Lambda https://www.youtube.com/watch?v=UFj27laTWQA

amazon web services

What can you do with AWS Lambda?

- Grow ‘connective tissue’ in your AWS infrastructure
  - Example: http://www.slideshare.net/JulienSIMON5/building-a-serverless-pipeline
- Build event-driven applications
- Build APIs together with Amazon API Gateway
  - RESTful APIs
  - Resources, methods
  - Stages

http://aws.amazon.com/apigateway

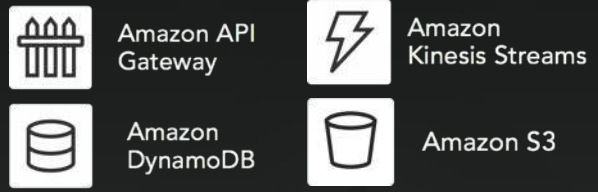
Serverless architecture

=

Managed services

+

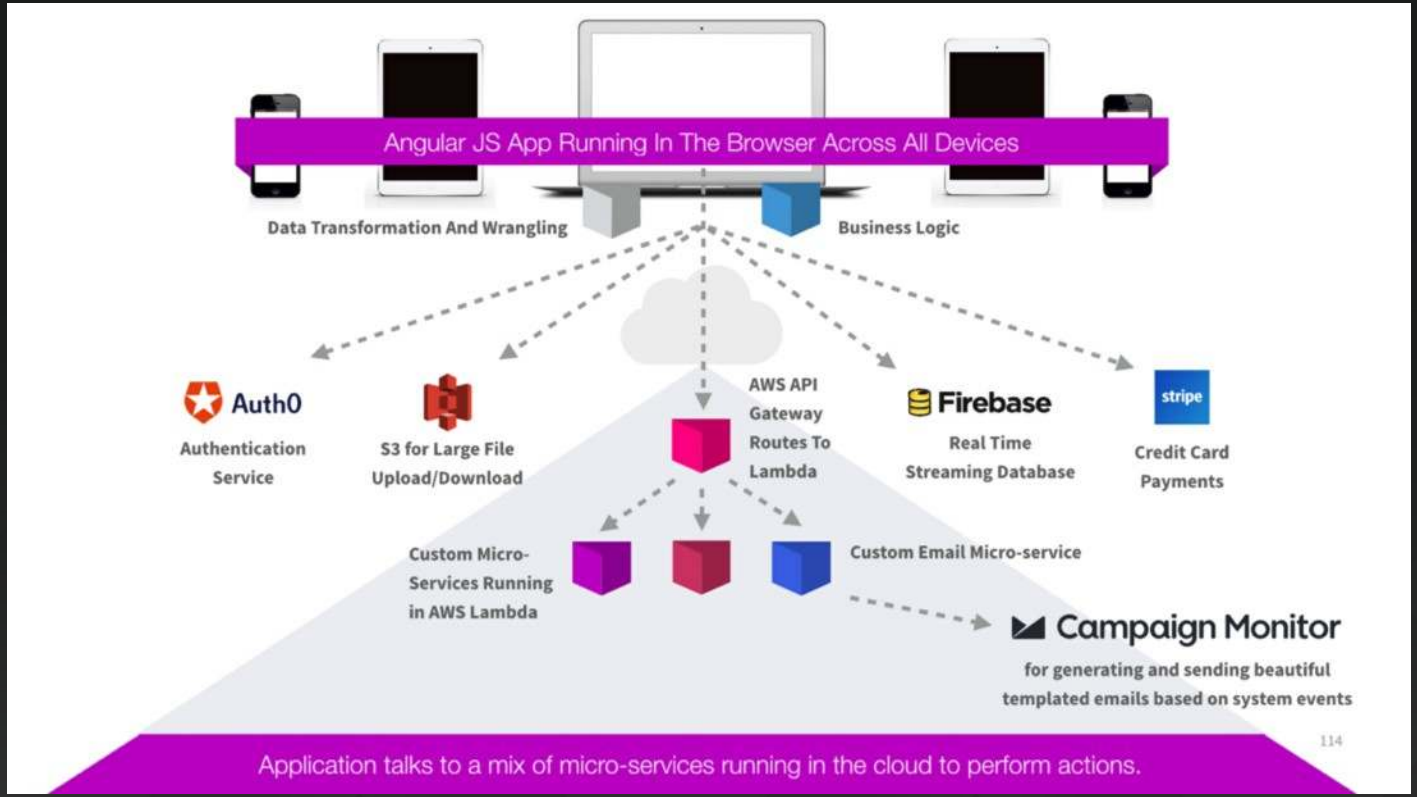
AWS Lambda



Reference architectures: http://www.allthingsdistributed.com/2016/06/aws-lambda-serverless-reference-architectures.html

amazon web services

A Cloud Guru: 100% Serverless



https://read.acloud.guru/serverless-the-future-of-software-architecture-d4473ffed864

Typical development workflow

1. Write and deploy a Lambda function
2. Create and deploy a REST API with API Gateway
3. Connect the API to the Lambda function
4. Invoke the API
5. Test, debug and repeat ;)

Simplifying Development

Code samples available at https://github.com/juliensimon/aws/tree/master/lambda_frameworks

The Serverless framework

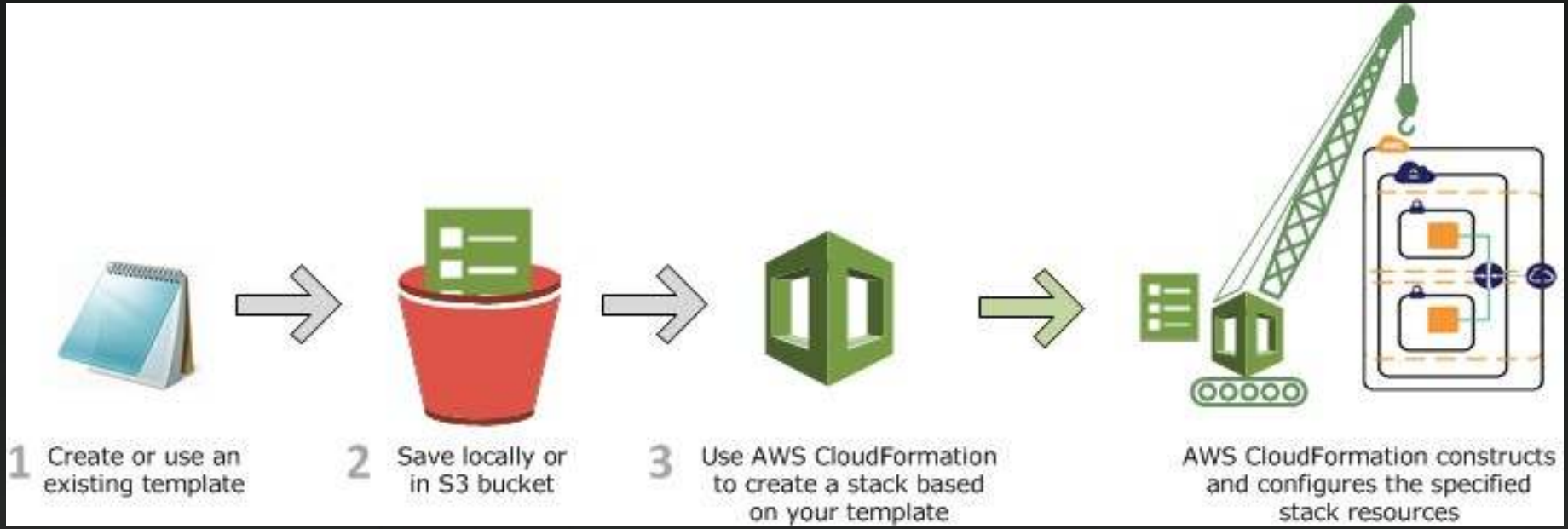
formerly known as JAWS: Just AWS Without Servers

- Announced at re:Invent 2015 by Austen Collins and Ryan Pendergast
- Supports Node.js, as well as Python, Java and C#
- Auto-deploys and runs Lambda functions, locally or remotely
- Auto-deploys your Lambda event sources: API Gateway, S3, DynamoDB, etc.
- Creates all required infrastructure with CloudFormation
- Simple configuration in YML

http://github.com/serverless/serverless
https://serverless.com
AWS re:Invent 2015 | (DVO209) https://www.youtube.com/watch?v=D_U6luQ6l90 & https://vimeo.com/141132756

amazon web services

AWS CloudFormation



Serverless: “Hello World” function

```
$ serverless create --template aws-nodejs

Edit handler.js, serverless.yml and event.json

$ serverless deploy [--stage stage_name]

$ serverless invoke [local] --function function_name

$ serverless remove
```

Serverless: “Hello World” API

Update serverless.yml:

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

```
$ serverless deploy
$ serverless info
$ http $URL
```

Gordon

- Released in Oct'15 by Jorge Batista
- Supports Python, Javascript, Golang, Java, Scala, Kotlin (including in the same project)
- Auto-deploys and runs Lambda functions, locally or remotely
- Auto-deploys your Lambda event sources: API Gateway, CloudWatch Events, DynamoDB Streams, Kinesis Streams, S3
- Creates all required infrastructure with CloudFormation
- Simple configuration in YML

https://github.com/jorgebastida/gordon
https://news.ycombinator.com/item?id=11821295

Gordon: “Hello World” API

```
$ gordon startproject hellonode

$ gordon startapp helloapp --runtime=js

Write function

$ gordon build

$ echo '{"name":"Julien"}' | gordon run hello.helloworld

$ gordon apply [--stage stage_name]

$ http post $URL name=‘Wellington’

$ gordon delete --confirm
```

More Lambda frameworks

- Apex https://github.com/apex/apex
  - Released in Dec'15 by TJ Holowaychuk
  - Python, Javascript, Java, Golang
  - Terraform integration to manage infrastructure for event sources
- Zappa https://github.com/Miserlou/Zappa
  - Released in Feb'16 by Rich Jones
  - Python web applications on AWS Lambda + API Gateway
- AWS Chalice https://github.com/awslabs/chalice
  - Released in Jul'16, still in beta
  - Python web applications, aka "Flask for Lambda"
- Docker-lambda https://github.com/lambci/docker-lambda
  - Released in May'16 by Michael Hart
  - Run functions in Docker images that "replicate" the live Lambda environment

Simplifying Deployment

AWS Serverless Application Model (SAM)

formerly known as Project Flourish

- CloudFormation extension released in Nov'16 to bundle Lambda functions, APIs & events
- 3 new CloudFormation resource types
  - *AWS::Serverless::Function*
  - *AWS::Serverless::Api*
  - *AWS::Serverless::SimpleTable*
- 2 new CloudFormation CLI commands
  - '*aws cloudformation package*'
  - '*aws cloudformation deploy*'
- Integration with CodeBuild and CodePipeline for CI/CD

https://aws.amazon.com/fr/blogs/compute/introducing-simplified-serverless-application-deployment-and-management
https://github.com/awslabs/serverless-application-model/

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: Get items from a DynamoDB table.
Resources:
  GetFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs4.3
      Policies: AmazonDynamoDBReadOnlyAccess
      Environment:
        Variables:
          TABLE_NAME: !Ref Table
      Events:
        GetResource:
          Type: Api
          Properties:
            Path: /resource/{resourceId}
            Method: get
  Table:
    Type: AWS::Serverless::SimpleTable
```

Sample SAM template for:

- Lambda function
- HTTP GET API
- DynamoDB table

Demo: simple CRUD service for DynamoDB

```
$ aws s3 mb s3://jsimon-samdemo-sydney
--region ap-southeast-2

$aws cloudformation package
--template-file template.yaml
--output-template-file output.yaml
--s3-bucket jsimon-samdemo-sydney

$ aws cloudformation deploy
--template-file output.yaml --stack-name samdemo
--region ap-southeast-2
--capabilities CAPABILITY_IAM
```

https://github.com/awslabs/serverless-application-model/tree/master/examples/2016-10-31/api_backend

Lambda videos from re:Invent 2016

AWS re:Invent 2016: What's New with AWS Lambda (SVR202)https://www.youtube.com/watch?v=CwxWhyGteNc

AWS re:Invent 2016: Serverless Apps with AWS Step Functions (SVR201)
https://www.youtube.com/watch?v=75MRve4nv8s

AWS re:Invent 2016: Real-time Data Processing Using AWS Lambda (SVR301)
https://www.youtube.com/watch?v=VFLKOy4GKXQ

AWS re:Invent 2016: Serverless Architectural Patterns and Best Practices (ARC402)
https://www.youtube.com/watch?v=b7UMoc1iUYw

AWS re:Invent 2016: Bringing AWS Lambda to the Edge (CTD206)
https://www.youtube.com/watch?v=j26novaqF6M

AWS re:Invent 2016: Ubiquitous Computing with Greengrass (IOT201)
https://www.youtube.com/watch?v=XQQjX8GTEko

Thank you!

Julien Simon, Principal Technical Evangelist, AWS
julsimon@amazon.fr
@julsimon

amazon web services