

Go serverless: Building an interactive Web Application on AWS

Jason Woodward
Principal @ State & Plain
edUi 2016 / Charlottesville, VA
#edui_aws

<http://www.woodwardjd.com/edui2016/>

AWS Service Overview

- S3
- DynamoDB
- Lambda
- API Gateway
- CloudFront
- Lots of other stuff (notably IAM and EC2)

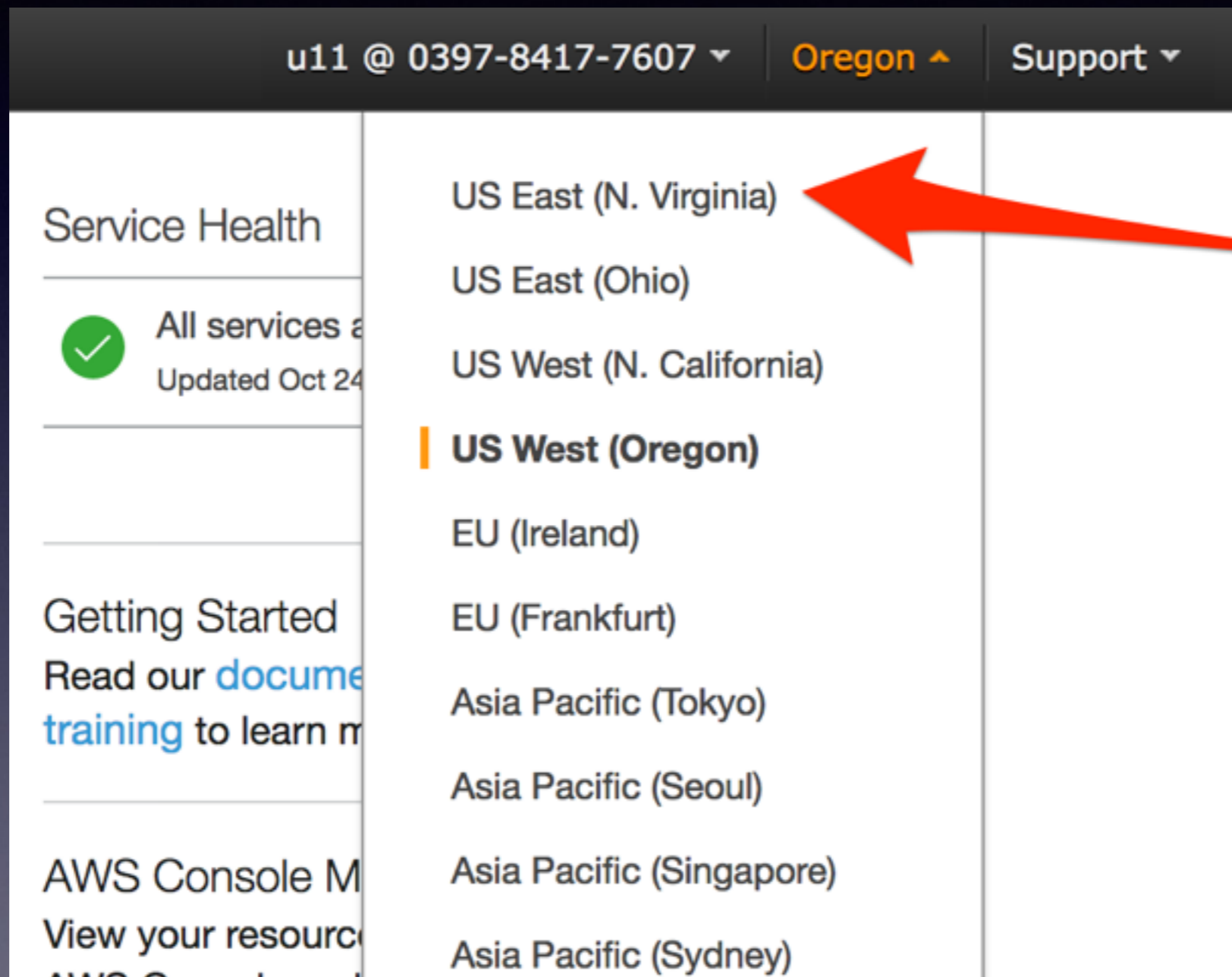
<http://www.woodwardjd.com/edui2016/>

Let's get logged in

- You installed AWS CLI, right?
- `$ aws configure`
region: `us-east-1`
output format: none / hit enter / accept default
- `$ aws s3 ls`
- For today: ALWAYS US East 1 (N Virginia) !!

<http://www.woodwardjd.com/edui2016/>

Let's get logged in



The screenshot shows the AWS console interface. At the top, there is a header bar with the following elements:

- Account ID and email: u11 @ 0397-8417-7607
- Region: Oregon (highlighted in orange)
- Support link: Support

Below the header, the main content area is divided into two columns:

- Left Column:**
 - Service Health: A green checkmark icon and the text "All services a" and "Updated Oct 24".
 - Getting Started: "Read our [documents](#) and [training](#) to learn more."
 - AWS Console M: "View your resources" and "AWS CloudTrail".
- Right Column:** A list of regions with "Oregon" selected and highlighted in orange:
 - US East (N. Virginia)
 - US East (Ohio)
 - US West (N. California)
 - US West (Oregon)**
 - EU (Ireland)
 - EU (Frankfurt)
 - Asia Pacific (Tokyo)
 - Asia Pacific (Seoul)
 - Asia Pacific (Singapore)
 - Asia Pacific (Sydney)

A red arrow points from the right side of the image towards the "Oregon" region in the list.

<http://www.woodwardjd.com/edui2016/>

What to create?

- What should we create today?
- Email Address collection form!

<http://www.woodwardjd.com/edui2016/>

Steps

- Static HTML page / form
- Identify Actions and Design and set up API
- Add jQuery to static HTML page which talks to API
- Design and set up database
- Update API functions to read / write to database
- Test API in API Gateway control panel
- Test in browser
- Secure / Speed up

<http://www.woodwardjd.com/edui2016/>

Get some static up there

- Create an S3 bucket named edui-u<number>
- Download App HTML (static) from workshop page. Windows folks: watch out for hidden file extensions!
- ```
$ aws s3 cp edui-app-static.html
s3://edui-uNN/index.html --acl
public-read
```

<http://www.woodwardjd.com/edui2016/>

# Identify Actions

- Get Emails (REST GET)
- Add an Email (REST POST)
- Create Lambda functions named uN-getEmail and uN-postEmail, creating a new role from a template (use S3 object read-only), using the GET and POST Lambda JS Placeholder code from workshop webpage

<http://www.woodwardjd.com/edui2016/>



# Design API

- API is pre-created due to AWS per-minute creation limit
- Create a resource “emails” in API Gateway
- Add GET and POST methods to the resource
- Add CORS to what we created
- Deploy to “prod” stage (note: also rate limited... we'll bear with this one)

<http://www.woodwardjd.com/edui2016/>

# Design API

This screenshot shows the 'Resources' section of an API design tool. A tree view on the left shows a root resource '/' and a sub-resource '/emails'. The '/emails' resource is expanded, showing its methods: GET, OPTIONS, and POST. A dropdown menu for 'Actions' is open, showing two categories: 'RESOURCE ACTIONS' and 'API ACTIONS'. The 'Delete API' option is highlighted in red.

Resources

Actions ▾ / Met

▼ /

▼ /emails

GET

OPTIONS

POST

RESOURCE ACTIONS

Create Method

Create Resource

Enable CORS

API ACTIONS

Deploy API

Import API

Delete API

This screenshot shows the 'Stages' section of the API design tool. A tree view on the left shows a stage named 'prod' with a sub-resource '/'. The '/emails' resource is expanded, showing its methods: POST, GET, and OPTIONS. A blue 'Create' button is visible in the top right corner.

Stages

Create

▼ prod

▼ /

▼ /emails

POST

GET

OPTIONS

# Add API calls to HTML

- We'll use jQuery ajax to make calls to the GET and POST methods we just created
- Download App HTML (dynamic) from workshop page, **update with your API endpoint URL**
- ```
$ aws s3 cp edui-app-dynamic.html  
s3://edui-uNN/index.html --acl  
public-read
```

<http://www.woodwardjd.com/edui2016/>

Design Database Model

- Holds Email Address and Name
- Create DynamoDB table named u<NUMBER>-emails
- Primary partition key: email
- Additional field is name

<http://www.woodwardjd.com/edui2016/>

Build Functions

- Updating the body of the Lambda functions we already created
- Download Lambda Function Bodies (GET and POST) from workshop page
- Try your Lambda
- Update policy so the Lambda function can read/write from/to DynamoDB (and remove the S3 one while we're at it)

<http://www.woodwardjd.com/edui2016/>

Test!

- Test in API Gateway
- Test in Browser

<http://www.woodwardjd.com/edui2016/>

Secure / Speed

- What security problems do you see?
- How can we make this faster?

<http://www.woodwardjd.com/edui2016/>