Innovator Island – Day 3

Build a serverless web app for a theme park

James Beswick
Sr. Developer Advocate, AWS Serverless

© 2020, Amazon Web Services, Inc. or its Affiliates.

About me

- James Beswick
  - Email: jbeswick@amazon.com
  - Twitter: @jbesw
- Senior Developer Advocate – AWS Serverless
- Self-confessed serverless geek
- Software Developer
- Product Manager
- Previously:
  - Multiple start-up tech guy
  - Rackspace, USAA, Morgan Stanley, J P Morgan
  - Enjoys travel, coffee, comedy and theme parks…

Day 3 agenda

Chromakey and image compositing

The services and architecture

Coding and discussion

Full 5-day agenda at
http://bit.ly/islandworkshop

© 2020, Amazon Web Services, Inc. or its Affiliates.

Prerequisites

- AWS account – personal account with full admin access
- Use one of the supported regions:
  - us-west-2 (US West - Oregon), us-east-2 (US East - Ohio), us-east-1 (US East - Northern Virginia)
  - ap-southeast-2 (Sydney, Australia)
  - eu-central-1 (Frankfurt, Germany)
- Cost:
  - Attending workshop is free BUT
  - Running the workshop is might incur small charges
- Workshop <u>cannot</u> be completed with a tablet
- Compatible with Chrome/Firefox (not Safari)
- Familiarity with the AWS Management Console
- Continues from everything built in Day 1

© 2020, Amazon Web Services, Inc. or its Affiliates.

This week's adventure

Day 1: Set up the app

- Deploy the frontend and backend applications

Day 2: Ride wait times

- Integrate your app with the ride systems so guests can see wait times

Day 3: Ride photos

- Build a photo-processing flow so guests can take selfies around the park

Day 4: Analyzing visitor stats

- Gain insights from all the wristband data collected from visitors

Day 5: Processing events

- Use an event bus to handle maintenance and ride malfunctions to improve operations

Full agenda at
http://bit.ly/islandworkshop

© 2020, Amazon Web Services, Inc. or its Affiliates.

aws

Module 3: Ride photos

© 2020, Amazon Web Services, Inc. or its Affiliates.

Module 3: Ride photos

© 2020, Amazon Web Services, Inc. or its Affiliates.

S3 upload process

Ideal for many mobile/web apps for media or file uploads

Backend:

1. Set up S3 bucket with CORS config
2. Create Lambda function with **PutObject** IAM policy

Frontend:

1. Get signed URL from API endpoint
2. Upload directly to signed URL from frontend application

S3 bucket CORS policy

© 2020, Amazon Web Services, Inc. or its Affiliates.

Deploy a new version of a Lambda function via SAM

```
// const uuidv4 = require('uuid/v4') // BROKEN!
const { v4: uuidv4 } = require('uuid')  // New code
```

What's the problem?

- The NPM package UUID has a breaking change.
- If you deployed the backend on Monday, you need to update.
- If you deployed after Monday, you have working code.

How will we fix this?

1. Demo the broken code.
2. Update the code in the function.
3. Redeploy the new function.
4. Test the working function.

© 2020, Amazon Web Services, Inc. or its Affiliates.

Go to
http://bit.ly/innovatorisland

© 2020, Amazon Web Services, Inc. or its Affiliates.

#InnovatorIsland

© 2020, Amazon Web Services, Inc. or its Affiliates.

Bonus challenge

Application translation

- Help international guests understand the app by adding language translation

Go to:

- https://bit.ly/innovatorisland
- Module 4

© 2020, Amazon Web Services, Inc. or its Affiliates.

Thank you - see you tomorrow!

James Beswick, AWS Serverless
jbeswick@amazon.com
@jbesw

© 2020, Amazon Web Services, Inc. or its Affiliates.