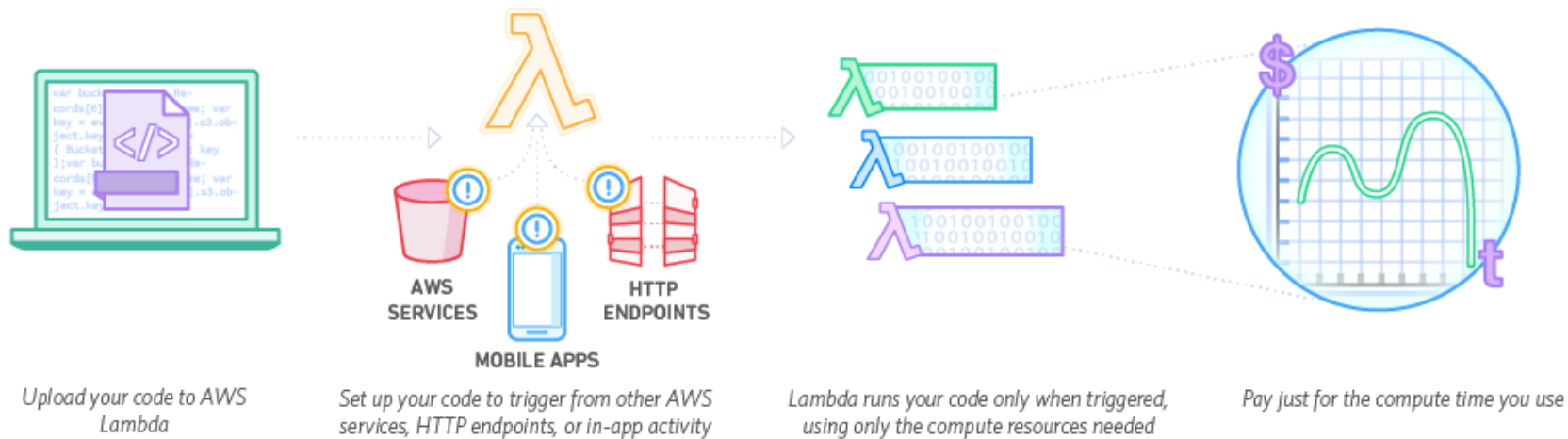


# AWS Lambda Services

# What is AWS Lambda?

- AWS Lambda is a service that is offered by Amazon where it's possible to run a coding project that works in tandem with other Amazon Cloud Services such as S3, DynamoDB, etc. It also allows for the running of code without the need for a server



# What can it do?

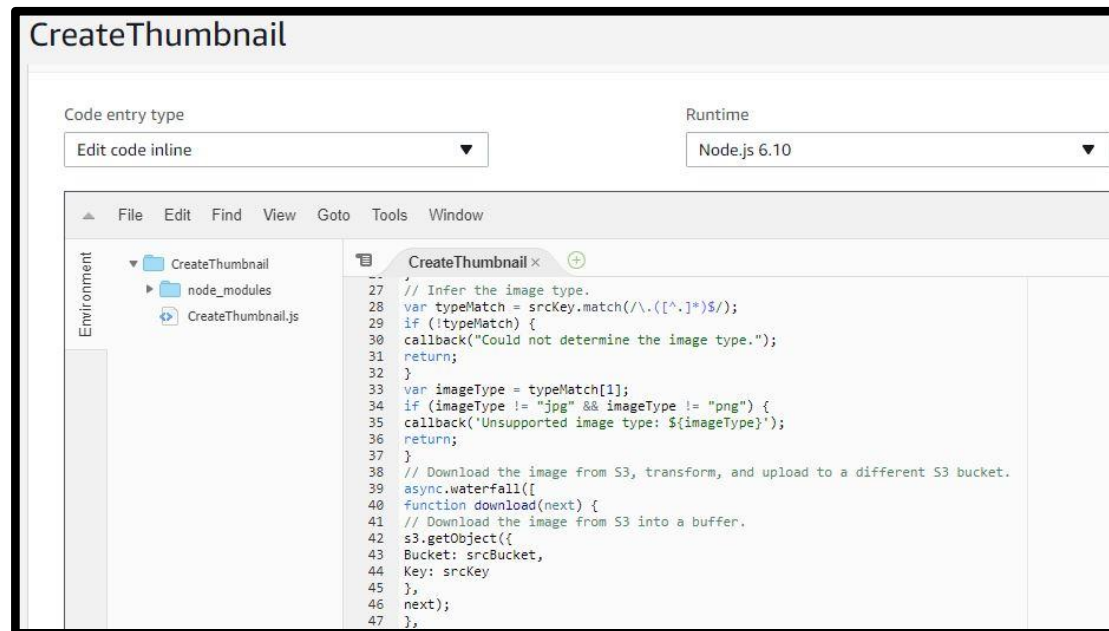
- ▶ There's all sorts of practical applications for Lambda
  - ▶ Real-Time File Processing
    - ▶ Thumbnailing images, transcoding videos, indexing files, processing logs, etc.
  - ▶ Stream Processing
    - ▶ Social Media Trend data, transaction order processing, metrics generation, etc.
  - ▶ Data Warehousing
  - ▶ Back-End Systems
  - ▶ Web Applications
  - ▶ Serverless Applications

# What goes into a Lambda function?

- ▶ Lambda functions can be made in the 3 following languages
  - ▶ NODE.JS(JavaScript)
  - ▶ Java 8
  - ▶ Python
- ▶ A Node.Js function just requires the source code and any Node.Js libraries that the function depends on all wrapped up in a zip file
- ▶ A Java function is similar, but requires a bit more time and effort to actually build as you need to build a JAR file or a ZIP file through Maven
- ▶ Python is relatively simple, since Lambda already includes AWS SDK for Python you only need to include your source code

# Is Any One Language Necessarily Better Than The Others?

- ▶ This comes down entirely to personal preference and what you can handle.
  - ▶ Java Lambda functions can be rather large and unwieldy compared to Node.js and Python in my opinion. Not to mention Node.JS and Python functions can be edited while logged into the AWS management console allowing for quick code changes on the fly



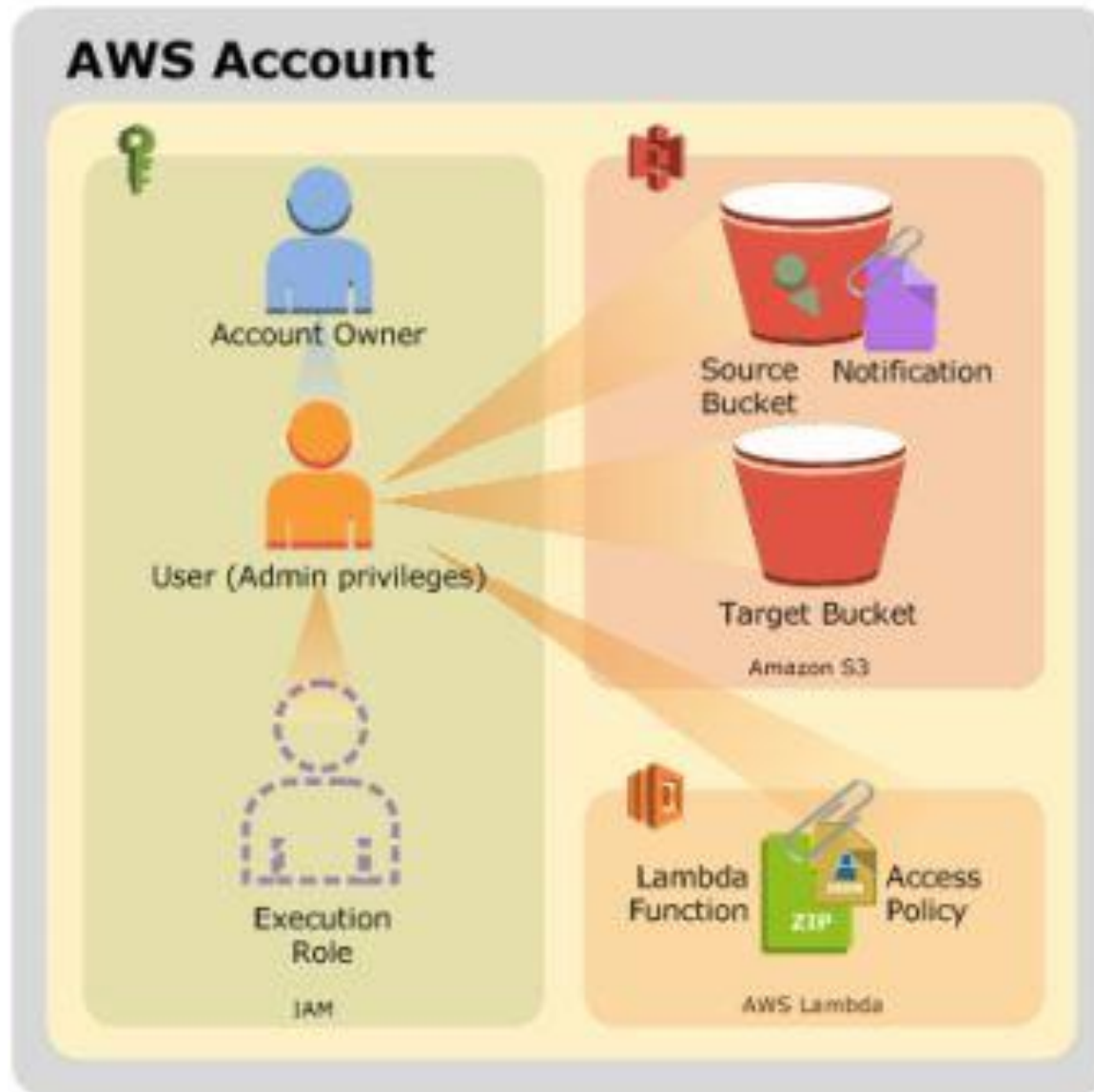
The screenshot shows the AWS Lambda console interface for a function named 'CreateThumbnail'. At the top, there are two dropdown menus: 'Code entry type' set to 'Edit code inline' and 'Runtime' set to 'Node.js 6.10'. Below these, a file explorer on the left shows the function's environment with folders for 'CreateThumbnail' and 'node\_modules', and a file 'CreateThumbnail.js'. The main area displays the JavaScript code for the function, which is a Node.js module. The code includes a comment '// Infer the image type.', a regular expression to match image file extensions, a function to handle unsupported image types, and an asynchronous function 'download' that uses 's3' to download and upload images. The code is wrapped in an IIFE (Immediately Invoked Function Expression) and ends with a call to 'module.exports'.

```
27 // Infer the image type.
28 var typeMatch = srcKey.match(/\.(.*$)/);
29 if (!typeMatch) {
30   callback("Could not determine the image type.");
31   return;
32 }
33 var imageType = typeMatch[1];
34 if (imageType != "jpg" && imageType != "png") {
35   callback('Unsupported image type: ${imageType}');
36   return;
37 }
38 // Download the image from S3, transform, and upload to a different S3 bucket.
39 async.waterfall([
40   function download(next) {
41     // Download the image from S3 into a buffer.
42     s3.getObject({
43       Bucket: srcBucket,
44       Key: srcKey
45     },
46     next);
47   },
```

# The Project

- ▶ For this project, I was asked to create a Lambda function which converted images in a S3 bucket into thumbnail-sized images.
- ▶ To accomplish this, I created a Node.JS JavaScript function which creates a new image that is a resized version of any JPG or PNG that is added to the bucket. The bucket is hosted on my account, which a user with admin privileges is given access to.
- ▶ The S3 bucket that contains the images to be transferred has an event handler attached to it, which runs the Lambda function whenever an object is added to the account

# The Project Pt. 2



# Comparing the image sizes



Original Image



Resized Image



# Problems with the Project

- ▶ For some reason at the start, the Lambda function couldn't access the S3 bucket whenever the ObjectCreated event fired.
- ▶ To fix this, I added the AmazonS3FullAccess policy to the execution role

Summary

Role ARN: arn:aws:iam::188796408411:role/cscd467-project-lambda-execution-role

Role description

Instance Profile ARNs

Path: /

Creation time: 2017-12-05 02:13 PST

Permissions | Trust relationships | Access Advisor | Revoke sessions

[Attach policy](#) Attached policies: 3

Policy name ▾	Policy type ▾
▶ <a href="#">AmazonS3FullAccess</a>	AWS managed policy
▶ <a href="#">AWSLambdaTracerAccessExecutio...</a>	Managed policy
▶ <a href="#">AWSLambdaBasicExecutionRole</a>	AWS managed policy

[+ Add inline](#)

# Applications of the Project

- This function is very similar to how websites handle the creation of profile pictures, taking an image and resizing it so it can be used by the specific needs for the website



# Applications Part 2

- ▶ Not to mention, a program like this would make it easier to generate image sizes for various screen sizes (like tablets and mobile phones)