

The background features a vibrant, multi-colored gradient. It starts with a dark blue on the left, transitions through purple and magenta, and then into bright orange and yellow towards the right. A diagonal line separates the darker blue on the left from the lighter colors on the right.

AWS
re:Invent

SVS214-R

Amazon S3 to Lambda: A flexible pattern at the core of serverless applications

James Beswick

Senior Developer Advocate, AWS Serverless
Amazon Web Services

About me



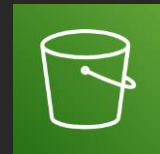
- James Beswick
- Senior Developer Advocate – AWS Serverless
- Serverless geek
- Software developer and product manager
- Previously:
 - Multiple start-up tech guy
 - Rackspace, USAA, Morgan Stanley, JPMorganChase
 - AWS customer since 2012

Agenda

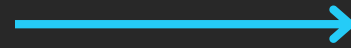
- Benefits of Amazon S3 and AWS Lambda interaction
- Using AWS Serverless Application Model for deployments
- Example use cases
- Live demo
- Wrap-up

S3 + Lambda

Triggering Lambda from S3



Amazon S3



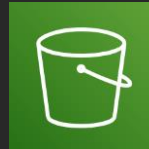
Lambda

Service characteristics

AWS Service

Characteristics

Storage



Amazon S3

- Object-based
- 11 9s of durability
- Virtually limitless storage
- Lifecycle management
- Supports encryption
- Fine-grained permission control

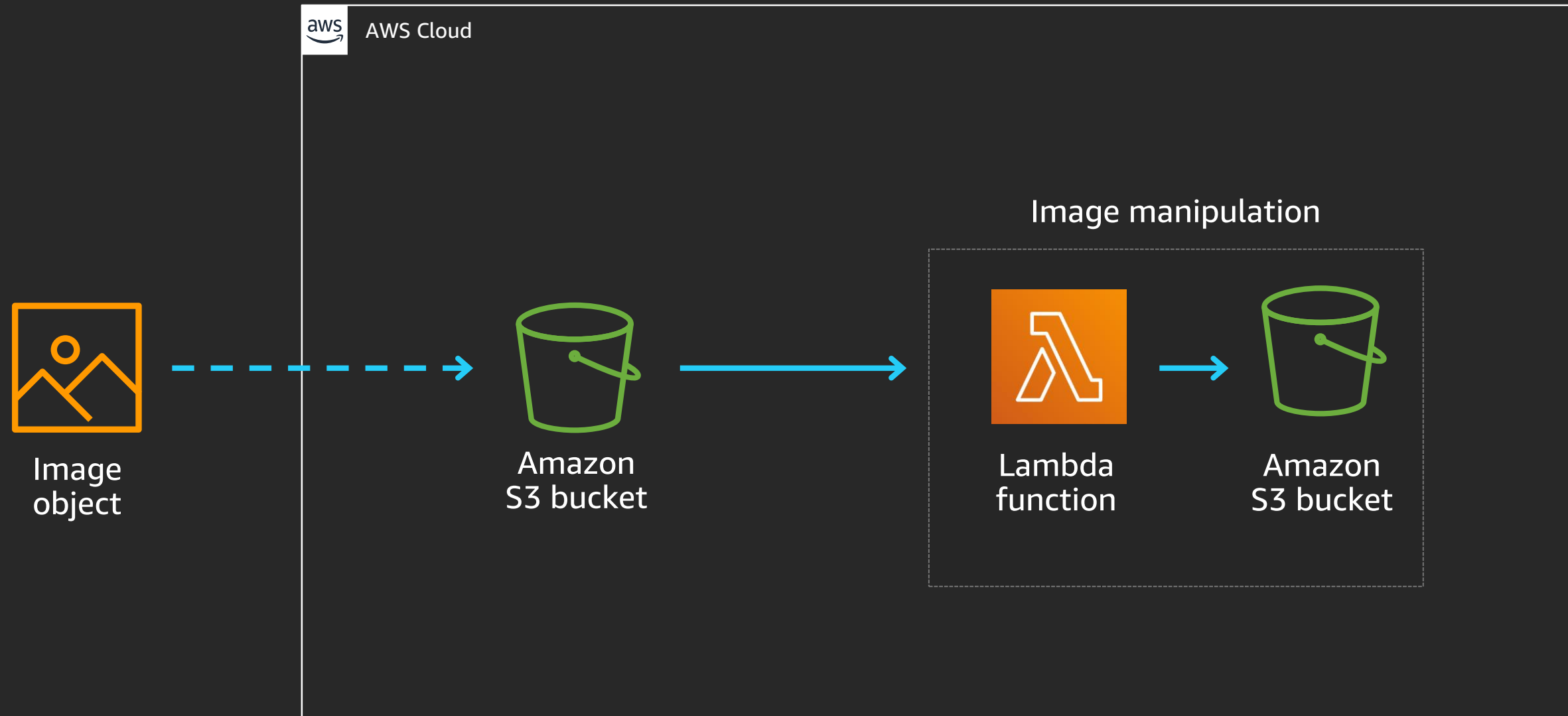
Compute



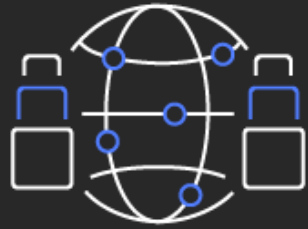
Lambda

- Functions as a service
- Rounded to 100ms billing
- Configurable memory
- Flexible runtimes
- Stateless
- Automated scaling
- Event-driven

Classic example—resizing images



The value of S3 + Lambda



Scalable

Brings compute
to data



Event-driven

Invoked in response
to S3 activity



Pay for value

Measurable,
attributable



Secure

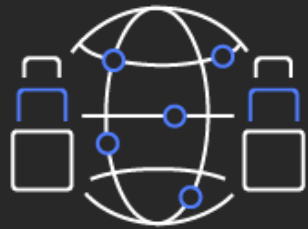
Granular permissions
via AWS Identity and
Access Management (IAM)



Serverless

No infrastructure
to manage

The value of S3 + Lambda



Scalable

Brings compute
to data



Event-driven

Invoked in response
to S3 activity



Pay for value

Measurable,
attributable



Secure

Granular permissions
via IAM



Serverless

No infrastructure
to manage

Introducing SAM



AWS Serverless Application Model (SAM)

- AWS CloudFormation extension optimized for serverless
- New serverless resource types: functions, APIs, tables
- Supports anything AWS CloudFormation supports
- Open specification (Apache 2.0)



<https://github.com/awslabs/serverless-application-model>

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31
```

Tells AWS CloudFormation this is a SAM template it needs to “transform”

Parameters:

```
InputBucketName:  
  Type: String  
  Default: 's3-auto-translator'
```

Specifies an input parameter

Resources:

```
InputS3Bucket:  
  Type: AWS::S3::Bucket  
  Properties:  
    BucketName: !Ref InputBucketName
```

Creates an S3 bucket

```
TranslatorFunction:  
  Type: AWS::Serverless::Function  
  Properties:  
    CodeUri: translatorFunction/  
    Handler: app.handler  
    Runtime: nodejs10.x  
    MemorySize: 128  
    Environment:  
      Variables:  
        targetLanguage: "es fr de"  
    Policies:  
      - S3CrudPolicy:  
        BucketName: !Ref InputBucketName
```

Creates a Lambda function with:

- Referenced managed IAM policy
- Language runtime/memory
- Code at the referenced zip location

Events:

```
FileUpload:  
  Type: S3  
  Properties:  
    Bucket: !Ref InputS3Bucket  
    Events: s3:ObjectCreated:*  
    Filter:  
      S3Key:  
        Rules:  
          - Name: suffix  
            Value: '.txt'
```

Defines the event triggering the Lambda function:

- New objects
- Specifies rule (ends in .txt)

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
  
Parameters:  
  InputBucketName:  
    Type: String  
    Default: 's3-auto-translator'  
  
Resources:  
  InputS3Bucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      BucketName: !Ref InputBucketName  
  TranslatorFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: translatorFunction/  
      Handler: app.handler  
      Runtime: nodejs10.x  
      MemorySize: 128  
      Environment:  
        Variables:  
          targetLanguage: "es fr de"  
      Policies:  
        - S3CrudPolicy:  
          BucketName: !Ref InputBucketName  
      Events:  
        FileUpload:  
          Type: S3  
          Properties:  
            Bucket: !Ref InputS3Bucket  
            Events: s3:ObjectCreated:*  
            Filter:  
              S3Key:  
                Rules:  
                  - Name: suffix  
                    Value: '.txt'
```

Tells CloudFormation this is a SAM template it needs to “transform”

Specifies an input parameter

Creates an S3 bucket

Creates a Lambda function with:

- Referenced managed IAM policy
- Language runtime/memory
- Code at the referenced zip location

Defines the event triggering the Lambda function:

- New objects
- Specifies rule (ends in .txt)

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
  
Parameters:  
  InputBucketName:  
    Type: String  
    Default: 's3-auto-translator'  
  
Resources:  
  InputS3Bucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      BucketName: !Ref InputBucketName  
  TranslatorFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: translatorFunction/  
      Handler: app.handler  
      Runtime: nodejs10.x  
      MemorySize: 128  
      Environment:  
        Variables:  
          targetLanguage: "es fr de"  
      Policies:  
        - S3CrudPolicy:  
          BucketName: !Ref InputBucketName  
      Events:  
        FileUpload:  
          Type: S3  
          Properties:  
            Bucket: !Ref InputS3Bucket  
            Events: s3:ObjectCreated:*  
            Filter:  
              S3Key:  
                Rules:  
                  - Name: suffix  
                    Value: '.txt'
```

Tells CloudFormation this is a SAM template it needs to “transform”

Specifies an input parameter

Creates an S3 bucket

Creates a Lambda function with:

- Referenced managed IAM policy
- Language runtime/memory
- Code at the referenced zip location

Defines the event triggering the Lambda function:

- New objects
- Specifies rule (ends in .txt)

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31

Parameters:
  InputBucketName:
    Type: String
    Default: 's3-auto-translator'
Resources:
  InputS3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref InputBucketName
  TranslatorFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: translatorFunction/
      Handler: app.handler
      Runtime: nodejs10.x
      MemorySize: 128
      Environment:
        Variables:
          targetLanguage: "es fr de"
      Policies:
        - S3CrudPolicy:
            BucketName: !Ref InputBucketName
      Events:
        FileUpload:
          Type: S3
          Properties:
            Bucket: !Ref InputS3Bucket
            Events: s3:ObjectCreated:*
            Filter:
              S3Key:
                Rules:
                  - Name: suffix
                    Value: '.txt'
```

Tells CloudFormation this is a SAM template it needs to “transform”

Specifies an input parameter

Creates an S3 bucket

Creates a Lambda function with:

- Referenced managed IAM policy
- Language runtime/memory
- Code at the referenced zip location

Defines the event triggering the Lambda function:

- New objects
- Specifies rule (ends in .txt)

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
  
Parameters:  
  InputBucketName:  
    Type: String  
    Default: 's3-auto-translator'  
  
Resources:  
  InputS3Bucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      BucketName: !Ref InputBucketName  
  TranslatorFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: translatorFunction/  
      Handler: app.handler  
      Runtime: nodejs10.x  
      MemorySize: 128  
      Environment:  
        Variables:  
          targetLanguage: "es fr de"  
      Policies:  
        - S3CrudPolicy:  
          BucketName: !Ref InputBucketName  
      Events:  
        FileUpload:  
          Type: S3  
          Properties:  
            Bucket: !Ref InputS3Bucket  
            Events: s3:ObjectCreated:*  
            Filter:  
              S3Key:  
                Rules:  
                  - Name: suffix  
                    Value: '.txt'
```

Tells CloudFormation this is a SAM template it needs to “transform”

Specifies an input parameter

Creates an S3 bucket

Creates a Lambda function with:

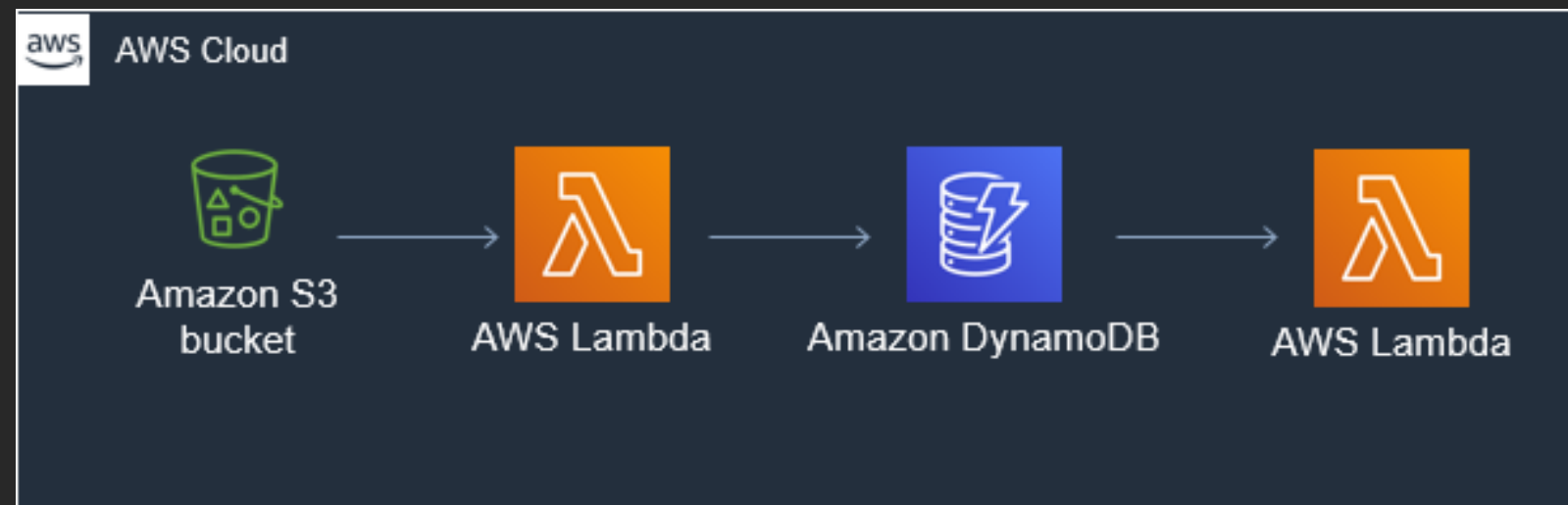
- Referenced managed IAM policy
- Language runtime/memory
- Code at the referenced zip location

Defines the event triggering the Lambda function:

- New objects
- Specifies rule (ends in .txt)

SAM transforms YAML into infrastructure

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
  
Parameters:  
  InputBucketName:  
    Type: String  
    Default: 's3-auto-translator'  
Resources:  
  InputS3Bucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      BucketName: !Ref InputBucketName  
  TranslatorFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: translatorFunction/  
      Handler: app.handler  
      Runtime: nodejs10.x  
      MemorySize: 128  
      Environment:  
        Variables:  
          targetLanguage: "es fr de"  
      Policies:  
        - S3CrudPolicy:  
          BucketName: !Ref InputBucketName  
      Events:  
        FileUpload:  
          Type: S3  
          Properties:  
            Bucket: !Ref InputS3Bucket  
            Events: s3:ObjectCreated:*  
            Filter:  
              S3Key:  
                Rules:  
                  - Name: suffix  
                    Value: '.txt'
```



The examples

1. Automated translation

Automatically translate objects written into an S3 bucket into 25 languages

Examples:

- Provide translations for your public website assets
- Translate resource files for mobile applications
- Translate customer communications to enable analytics
- Automate customer support across languages

2. Converting speech to text

Convert MP3 audio files into text using Amazon Transcribe

Examples:

- Transcribe 911/emergency calls
- Analyze customer support calls
- Provide captions on videos
- Provide transcriptions for webinars

3. DynamoDB importer

Use S3 as a staging area to automatically upload data into DynamoDB

Examples:

- General purpose import for DynamoDB tables
- Smoothing import load to manage WCUs in DynamoDB
- CSV import to DynamoDB using on-demand billing

4. Workflow management

Use S3 to initiate workflows using AWS Step Functions

Examples:

- Saving a signed customer contract from an email to start a work-order process
- Receiving patient medical files to initiate scheduling an appointment
- Receiving invoices to extract key information using Amazon Textract

5. Serving your static website or SPA

Use S3 with Amazon CloudFront to deploy a highly scalable, secure website with low latency for global customers

Examples:

- Serving static websites and single-page applications (SPAs)
- Restricting content based on geography
- Uploading files securely to S3 from a web or mobile application

6. Facial recognition web app

Use S3 with Amazon Rekognition to create a web application to recognize faces

Examples:

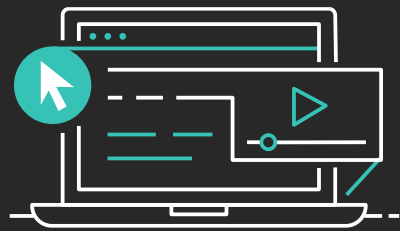
- In membership-based retail, clubs, or gyms, using faces to identify members
- For building entrances in workplaces, identifying employees to grant access

Demo: Web application powered by S3 + Lambda

To download the examples, visit:
<http://rebrand.ly/s3lambda>

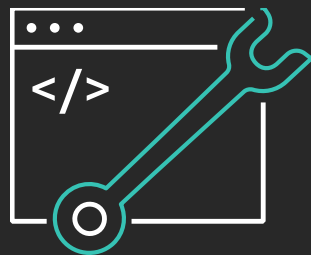
Learn serverless with AWS Training and Certification

Resources created by the experts at AWS to help you learn modern application development



Free, on-demand courses on serverless, including

- Introduction to Serverless Development
- Getting into the Serverless Mindset
- AWS Lambda Foundations
- Amazon API Gateway for Serverless Applications
- Amazon DynamoDB for Serverless Architectures



Additional digital and classroom trainings cover modern application development and computing

Visit the Learning Library at <https://aws.training>

Thank you!

James Beswick

jbeswick@amazon.com
@jbesw



Please complete the session survey in the mobile app.