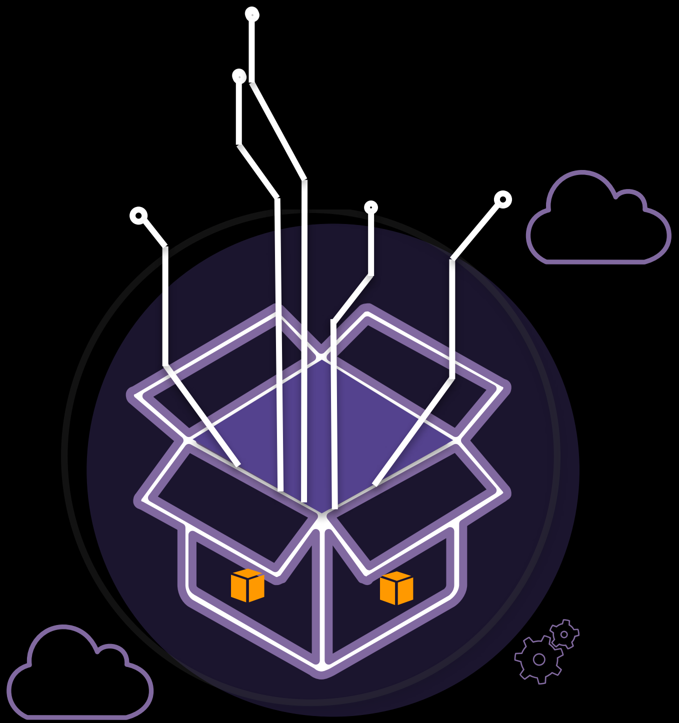
AWS re:Invent

What's New with AWS Lambda

Julien Simon, Principal Technical Evangelist, AWS

julsimon@amazon.fr
@julsimon

Capabilities of a serverless platform



Cloud Logic Layer

Orchestration and State Management

Responsive Data Sources

Application Modeling Framework

Developer Ecosystem

Integrations Library

Security and Access Control

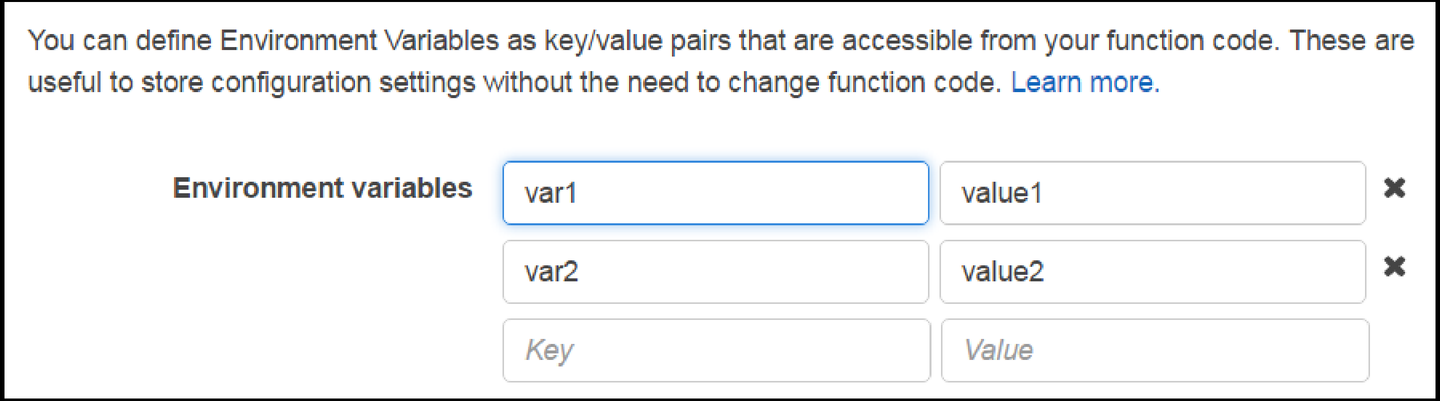
Reliability and Performance

Global Scale

CI/CD for serverless apps

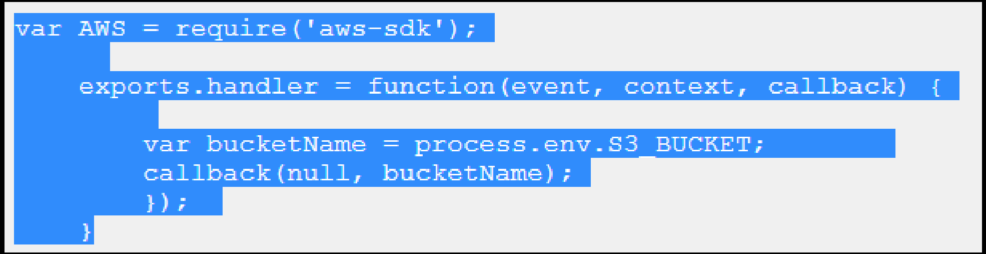
- New features
- AWS SAM
- SAM in AWS CloudFormation
- Serverless CI/CD pipelines
  - with AWS CodePipeline and AWS CodeBuild

Environment variables for Lambda functions New



You can define Environment Variables as key/value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more.

Environment variables		
	var1	value1
	var2	value2
	Key	Value



```
var AWS = require('aws-sdk');

    exports.handler = function(event, context, callback) {

        var bucketName = process.env.S3_BUCKET;
        callback(null, bucketName);
        });
    }
```

AWS Serverless Application Model ("SAM") New

- A common language for describing the contents of a serverless app.
- CloudFormation now "speaks serverless" with native support for SAM.
- New CloudFormation tools to package and deploy Lambda-based apps.
- Export Lambda blueprints and functions in SAM from the AWS Lambda console.

AWS Serverless Application Model New

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources: GetHtmlFunction:
Type: AWS::Serverless::Function
Properties:
CodeUri: s3://flourish-demo-bucket/todo_list.zip
Handler: index.gethtml
Runtime: nodejs4.3
Policies: AmazonDynamoDBReadOnlyAccess
Events:
GetHtml: Type: Api
Properties: Path: /{proxy+} Method: ANY
ListTable: Type: AWS::Serverless::SimpleTable
```

Functions

APIs

Storage

AWS Serverless Application Model New

```
AWSTemplateFormatVersion: '2010-09-09’
Transform: AWS::Serverless-2016-10-31
Resources: GetHtmlFunction:
Type: AWS::Serverless::Function
Properties:
CodeUri: s3://flourish-demo-bucket/todo_list.zip
Handler: index.gethtml
Runtime: nodejs4.3
Policies: AmazonDynamoDBReadOnlyAccess
Events:
GetHtml: Type: Api
Properties: Path: /{proxy+} Method: ANY
ListTable: Type: AWS::Serverless::SimpleTable
```

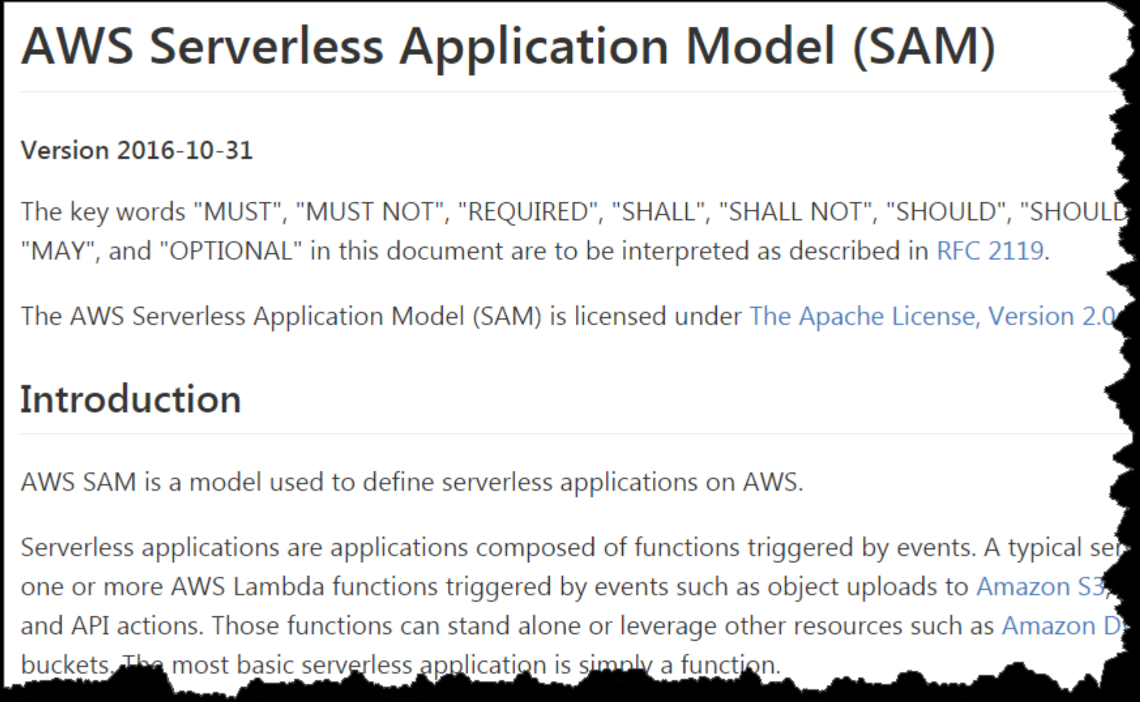
REPLACES:

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
GetHtmlFunctionGetHtmlPermissionProd:
Type: AWS::Lambda::Permission
Properties:
Action: lambda:invokeFunction
Principal: apigateway.amazonaws.com
FunctionName:
Ref: GetHtmlFunction
SourceArn:
Fn::Sub: arn:aws:execute-api:$
{AWS::Region}:${AWS::AccountId}:$
{ServerlessRestApi}/Prod/ANY/*
ServerlessRestApiProdStage:
Type: AWS::ApiGateway::Stage
Properties:
DeploymentId:
Ref: ServerlessRestApiDeployment
RestApiId:
Ref: ServerlessRestApi
StageName: Prod
ListTable:
Type: AWS::DynamoDB::Table
Properties:
ProvisionedThroughput:
WriteCapacityUnits: 5
ReadCapacityUnits: 5
AttributeDefinitions:
- AttributeName: id
AttributeType: S
KeySchema:
- KeyType: HASH
AttributeName: id
GetHtmlFunction:
Type: AWS::Lambda::Function
Properties:
Handler: index.gethtml
Code:
S3Bucket: flourish-demo-bucket
S3Key: todo_list.zip

Role:
Fn::GetAtt:
- GetHtmlFunctionRole
- Arn
Runtime: nodejs4.3
GetHtmlFunctionRole:
Type: AWS::IAM::Role
Properties:
ManagedPolicyArns:
-
arn:aws:iam::aws:policy/AmazonDynamoDB
ReadOnlyAccess
-
arn:aws:iam::aws:policy/service-role/AWSLa
mbdaBasicExecutionRole
AssumeRolePolicyDocument:
Version: '2012-10-17'
Statement:
- Action:
- sts:AssumeRole
Effect: Allow
Principal:
Service:
- lambda.amazonaws.com
ServerlessRestApiDeployment:
Type: AWS::ApiGateway::Deployment
Properties:
RestApiId:
Ref: ServerlessRestApi
Description: 'RestApi deployment id:
127e3fb91142ab1ddc5f5446adb094442581a
90d'
StageName: Stage
GetHtmlFunctionGetHtmlPermissionTest:
Type: AWS::Lambda::Permission
Properties:
Action: lambda:invokeFunction
Principal: apigateway.amazonaws.com
FunctionName:
Ref: GetHtmlFunction
SourceArn:
Fn::Sub: arn:aws:execute-api:$
{AWS::Region}:${AWS::AccountId}:$
{ServerlessRestApi}/*/ANY/*
ServerlessRestApi:
Type: AWS::ApiGateway::RestApi
Properties:
Body:
info:
version: '1.0'
title:
Ref: AWS::StackName
paths:
"/{proxy+}":
x-amazon-apigateway-any-method:
x-amazon-apigateway-integration:
httpMethod: ANY
type: aws_proxy
uri:
Fn::Sub: arn:aws:apigateway:$
{AWS::Region}:lambda:path/2015-03-
31/functions/${GetHtmlFunction.Arn}/
invocations
responses: {}
swagger: '2.0'
```

SAM: Open Specification New

A common language to describe the content of a serverless application *across the ecosystem*.

Apache 2.0 licensed GitHub project



AWS Serverless Application Model (SAM)

Version 2016-10-31

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
"MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

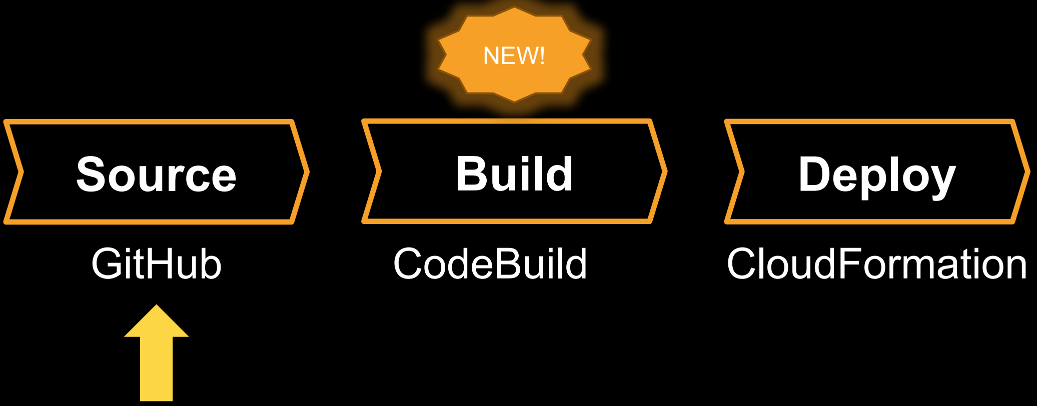
The AWS Serverless Application Model (SAM) is licensed under The Apache License, Version 2.0

Introduction

AWS SAM is a model used to define serverless applications on AWS.

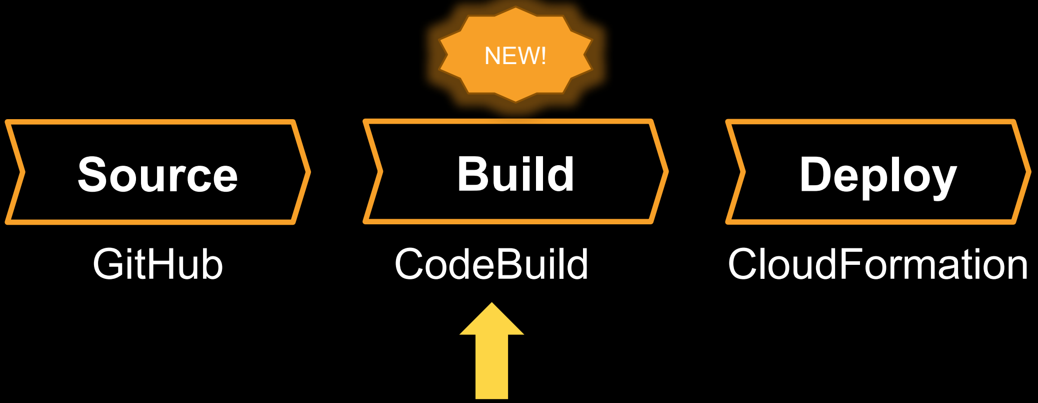
Serverless applications are applications composed of functions triggered by events. A typical ser
one or more AWS Lambda functions triggered by events such as object uploads to Amazon S3
and API actions. Those functions can stand alone or leverage other resources such as Amazon D
buckets. The most basic serverless application is simply a function.

Serverless CI/CD pipeline



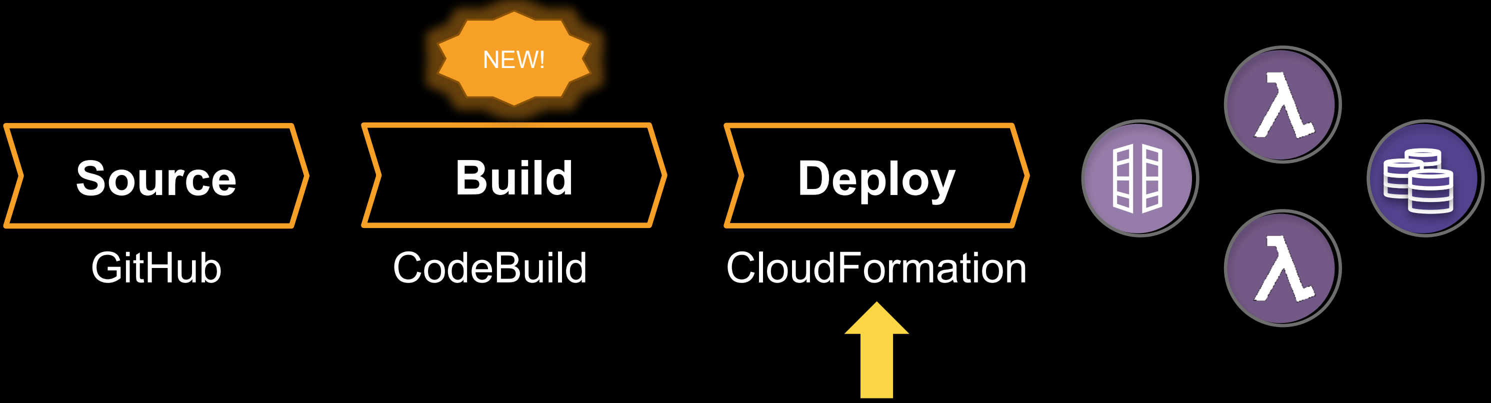
- Pull source directly from GitHub or CodeCommit using CodePipeline

Serverless CI/CD pipeline



- Pull source directly from GitHub or AWS CodeCommit using AWS CodePipeline
- Build and package serverless apps with AWS CodeBuild
  - npm, pip, Java compilation, BYO Docker…

Serverless CI/CD pipeline



- Pull source directly from GitHub or AWS CodeCommit using AWS CodePipeline
- Build and package serverless apps with AWS CodeBuild
- Deploy your completed Lambda app with AWS CloudFormation

How do I diagnose Lambda apps?

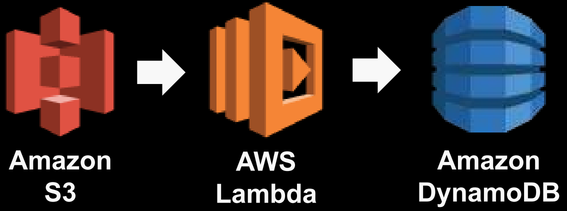
Introducing X-Ray Preview

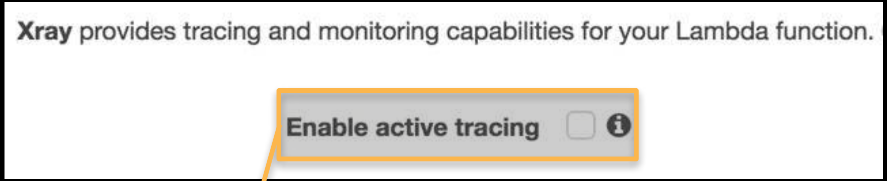
Gain visibility into events traveling through services

Trace calls and timing from Lambda functions to other AWS services

Easy configuration

Lambda support coming soon

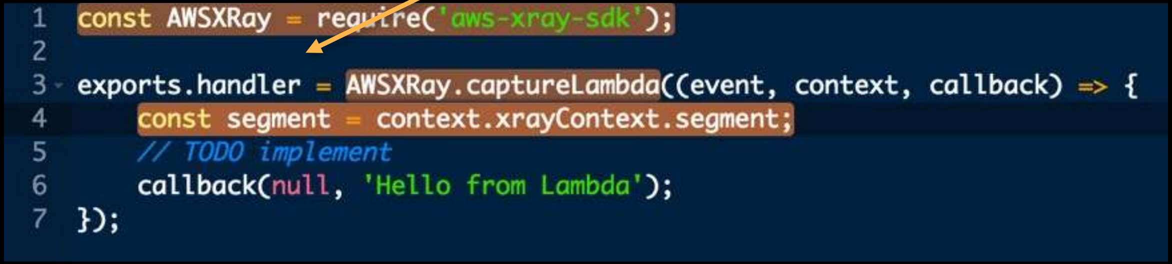




Xray provides tracing and monitoring capabilities for your Lambda function.

Enable active tracing ☐

Easy setup



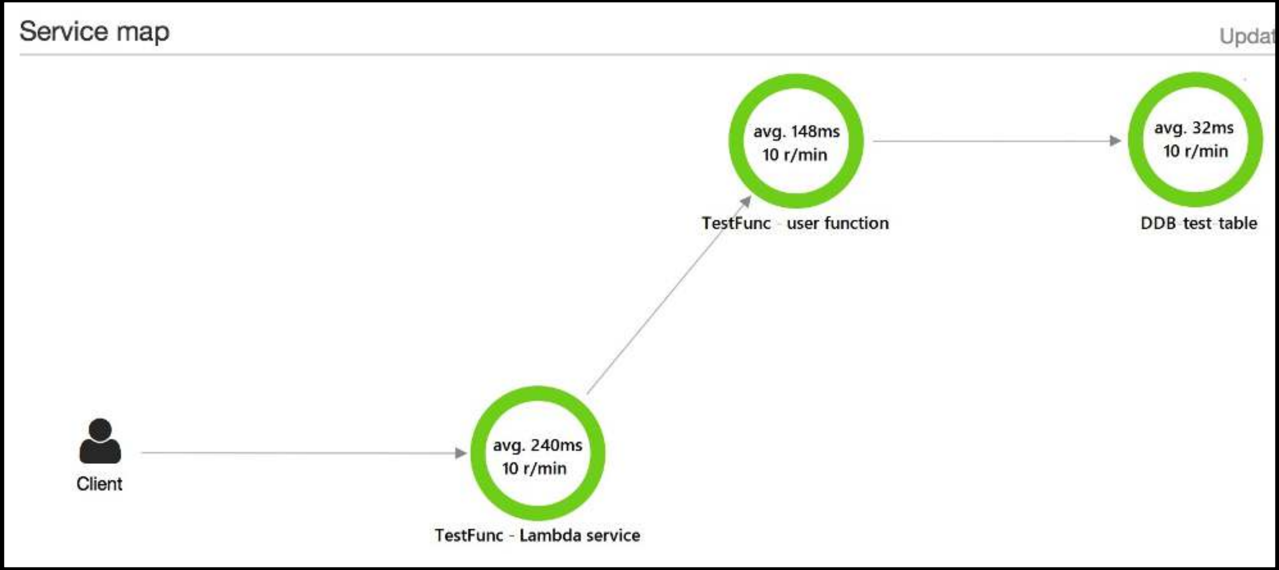
```
const AWSXRay = require('aws-xray-sdk');

exports.handler = AWSXRay.captureLambda((event, context, callback) => {
    const segment = context.xrayContext.segment;
    // TODO implement
    callback(null, 'Hello from Lambda');
});
```

Introducing X-Ray Preview

View the dynamic topology of your application

See actual dependencies among microservice components

Easily detect and diagnose missing events and throttles

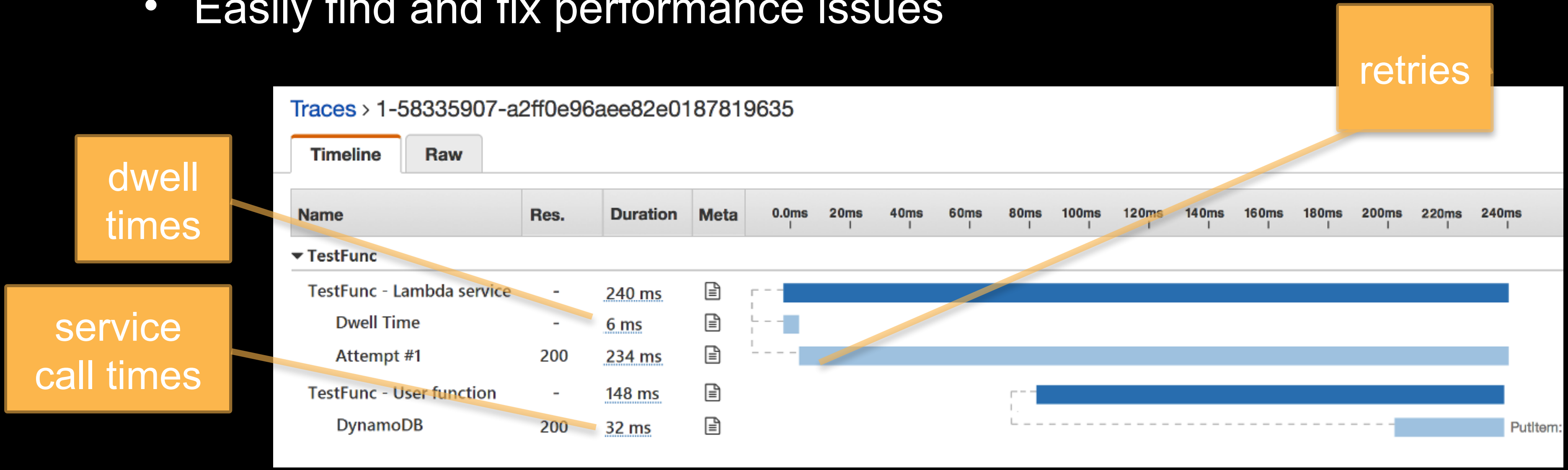


Introducing X-Ray Preview

See dwell time and retries for async invokes

Profile performance of calls your code makes to other AWS services

- Detect failures in event processing
- Easily find and fix performance issues



Traces > 1-58335907-a2ff0e96aee82e0187819635

Timeline | Raw

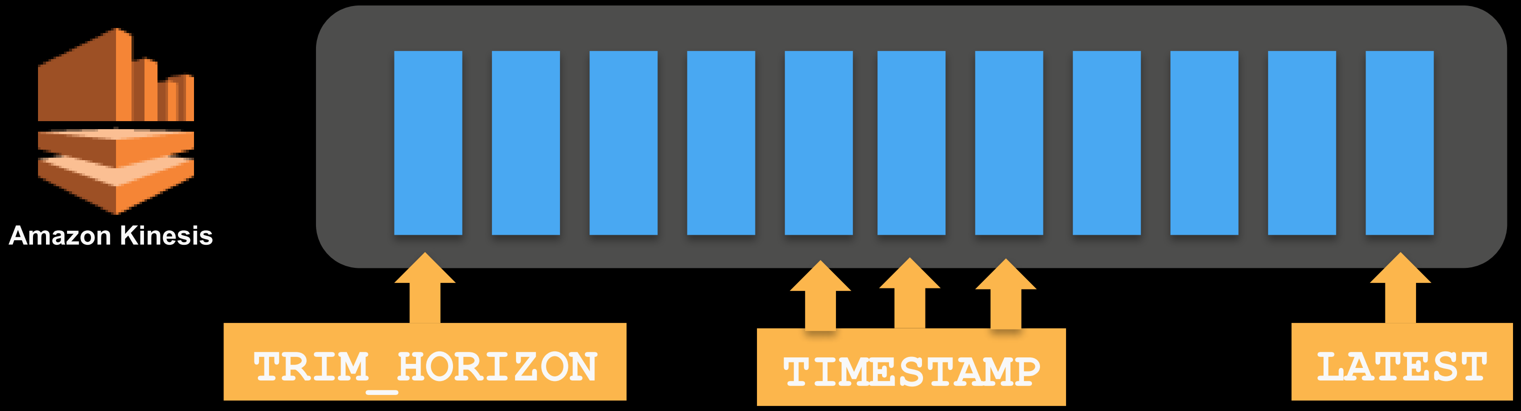
Name	Res.	Duration	Meta
▾ TestFunc			
TestFunc - Lambda service	-	240 ms	🗎
Dwell Time	-	6 ms	🗎
Attempt #1	200	234 ms	🗎
TestFunc - User function	-	148 ms	🗎
DynamoDB	200	32 ms	🗎

New Lambda features

- `AT_TIMESTAMP` Amazon Kinesis iterator
- C# with .NET Core
- Dead letter queue

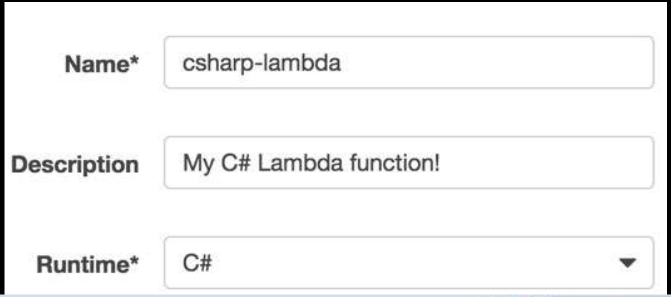
AT_TIMESTAMP Amazon Kinesis iterator New

- Process streaming data in Amazon Kinesis at any point in time
- Stop and start processing without rewinding or losing data

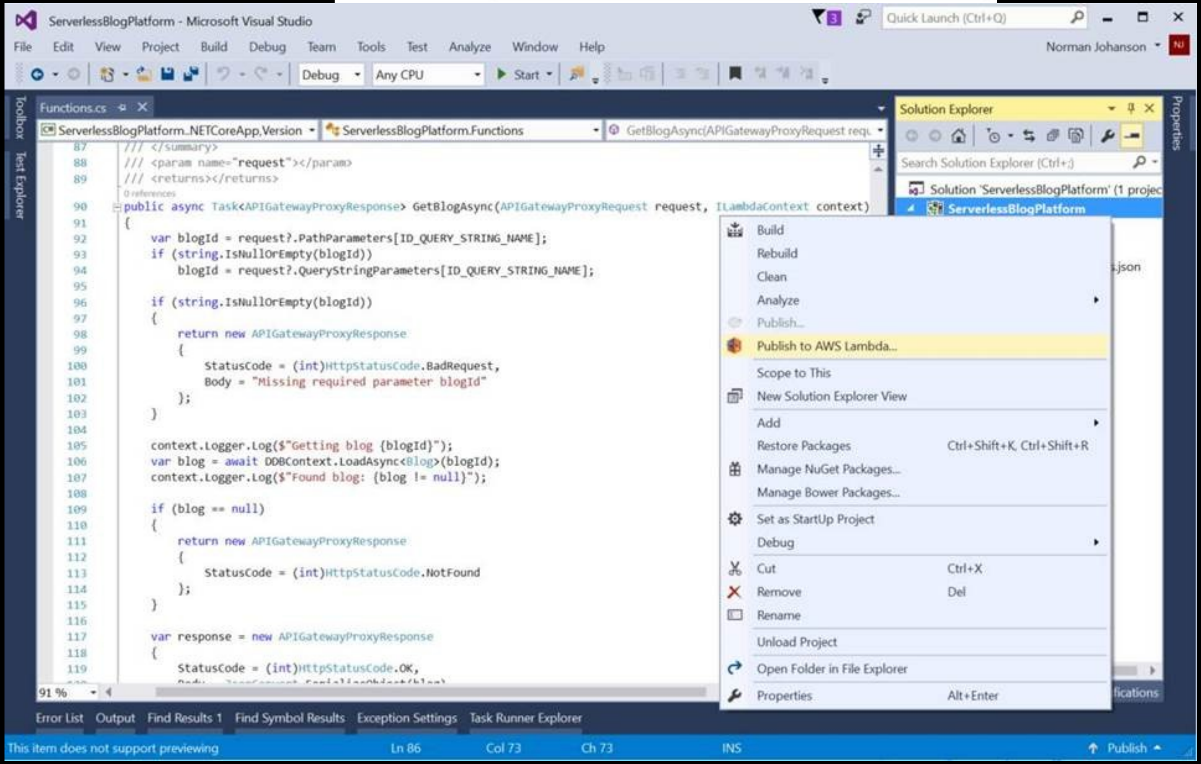


C# and .NET Core New

- Write Lambda functions in C#
- netcoreapp 1.0 on Amazon Linux
- Built-in logging and metrics
- Supports common AWS event types (S3, SNS)



Name*	csharp-lambda
Description	My C# Lambda function!
Runtime*	C#

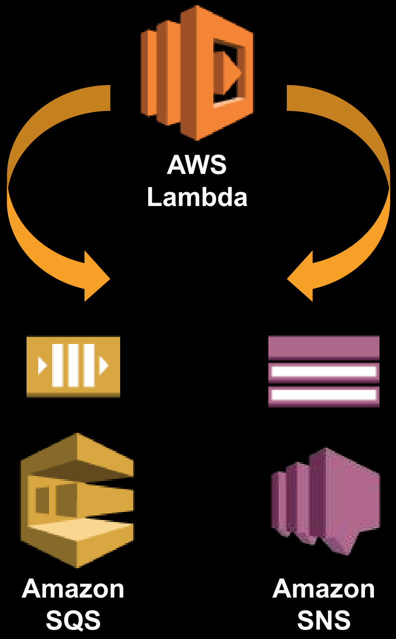

ServerlessBlogPlatform - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Norman Johanson
Debug
Any CPU
Start
Toolbox
Test Explorer
Functions.cs
ServerlessBlogPlatform..NETCoreApp,Version
ServerlessBlogPlatform.Functions
GetBlogAsync(APIGatewayProxyRequest requ
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ServerlessBlogPlatform' (1 projec
ServerlessBlogPlatform
Build
Rebuild
Clean
Analyze
Publish...
Publish to AWS Lambda...
Scope to This
New Solution Explorer View
Add
Restore Packages Ctrl+Shift+K, Ctrl+Shift+R
Manage NuGet Packages...
Manage Bower Packages...
Set as StartUp Project
Debug
Cut Ctrl+X
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter
Properties
91 %
Error List Output Find Results 1 Find Symbol Results Exception Settings Task Runner Explorer
This item does not support previewing
Ln 86 Col 73 Ch 73 INS
Publish

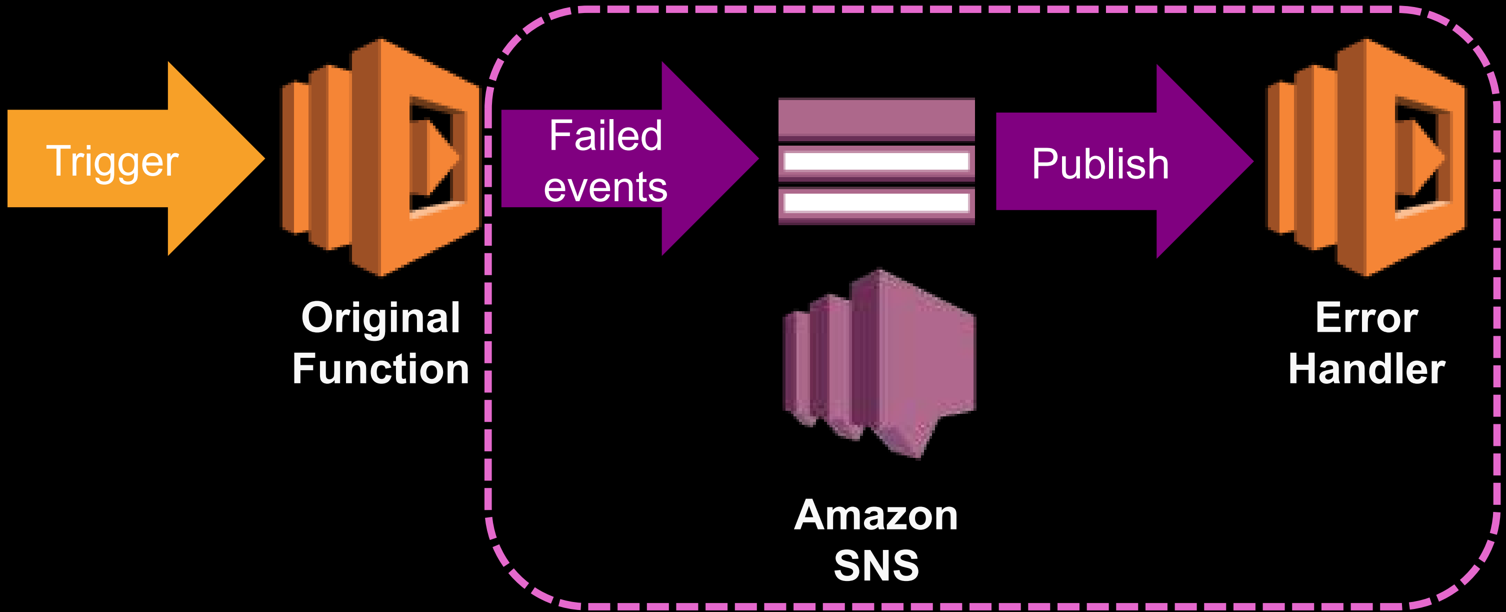
```csharp
/// </summary>
/// <param name="request"></param>
/// <returns></returns>
public async Task<APIGatewayProxyResponse> GetBlogAsync(APIGatewayProxyRequest request, ILambdaContext context)
{
    var blogId = request?.PathParameters[ID_QUERY_STRING_NAME];
    if (string.IsNullOrEmpty(blogId))
        blogId = request?.QueryStringParameters[ID_QUERY_STRING_NAME];

    if (string.IsNullOrEmpty(blogId))
    {
        return new APIGatewayProxyResponse
        {
            StatusCode = (int)HttpStatusCode.BadRequest,
            Body = "Missing required parameter blogId"
        };
    }

    context.Logger.Log($"Getting blog {blogId}");
    var blog = await DDBContext.LoadAsync<Blog>(blogId);
    context.Logger.Log($"Found blog: {blog != null}");

    if (blog == null)
    {
        return new APIGatewayProxyResponse
        {
            StatusCode = (int)HttpStatusCode.NotFound
        };
    }

    var response = new APIGatewayProxyResponse
    {
        StatusCode = (int)HttpStatusCode.OK,
```

Dead-letter queue for events New

Easily create reliable end-to-end event processing solutions

- Sends all unprocessed events to your SQS queue or SNS topic: 3 strikes rule
- Preserves events *even if your code has an issue* or the call was throttled
- Per-function
- Works for all async invokes, *including S3 and SNS events*





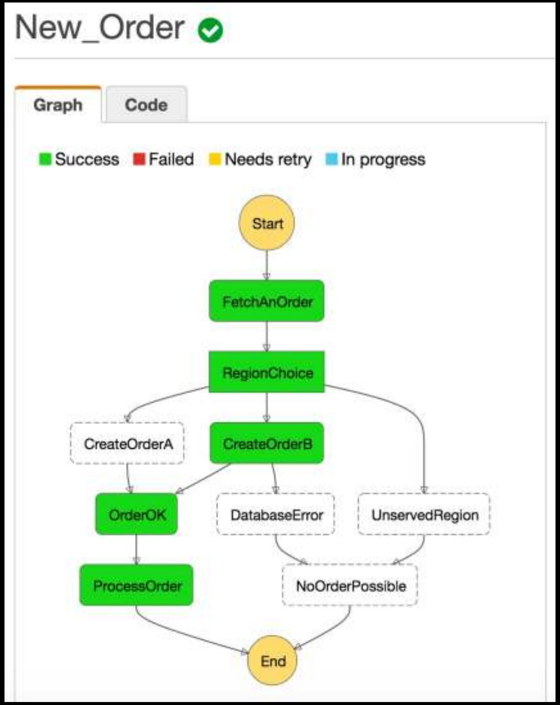
Lambda DLQ in action

Orchestrating Lambda functions

AWS Step Functions New

Reliably orchestrate multiple Lambda functions

Attempt a function more than 3X
Add callbacks to asynchronous functions
Handle situations that require waiting
Chain function execution (A→B→C)
Supports long-running workflows

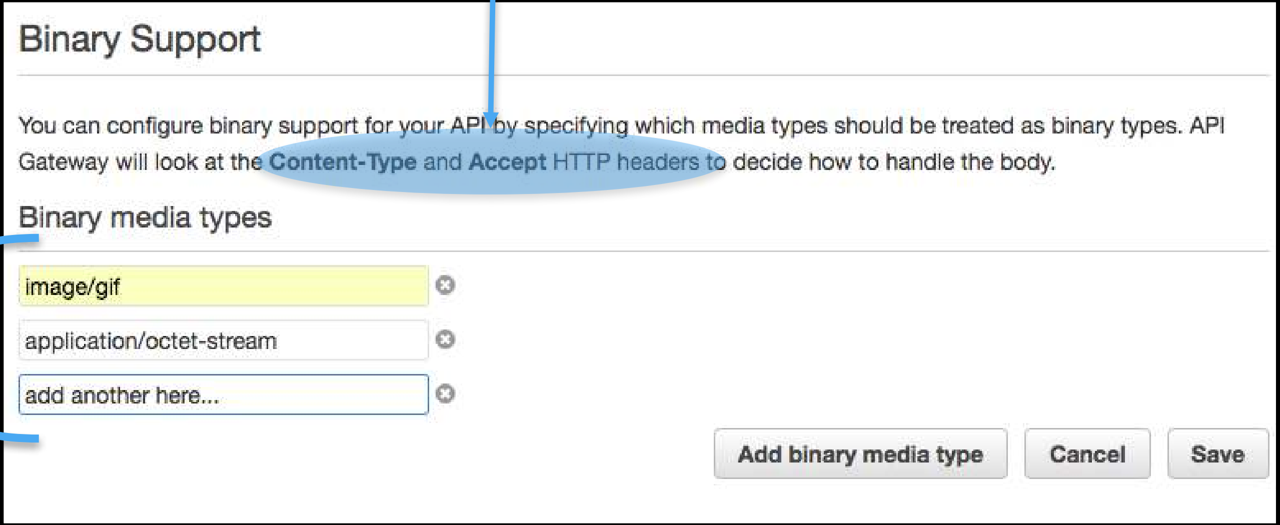


New API Gateway features

- Binary encoding
- Documentation support
- AWS Marketplace SaaS integration
- Developer Portal Reference Implementation

Binary encoding New

Uses Content-Type and Accept headers



Binary Support

You can configure binary support for your API by specifying which media types should be treated as binary types. API Gateway will look at the **Content-Type** and **Accept** HTTP headers to decide how to handle the body.

Binary media types

image/gif

application/octet-stream

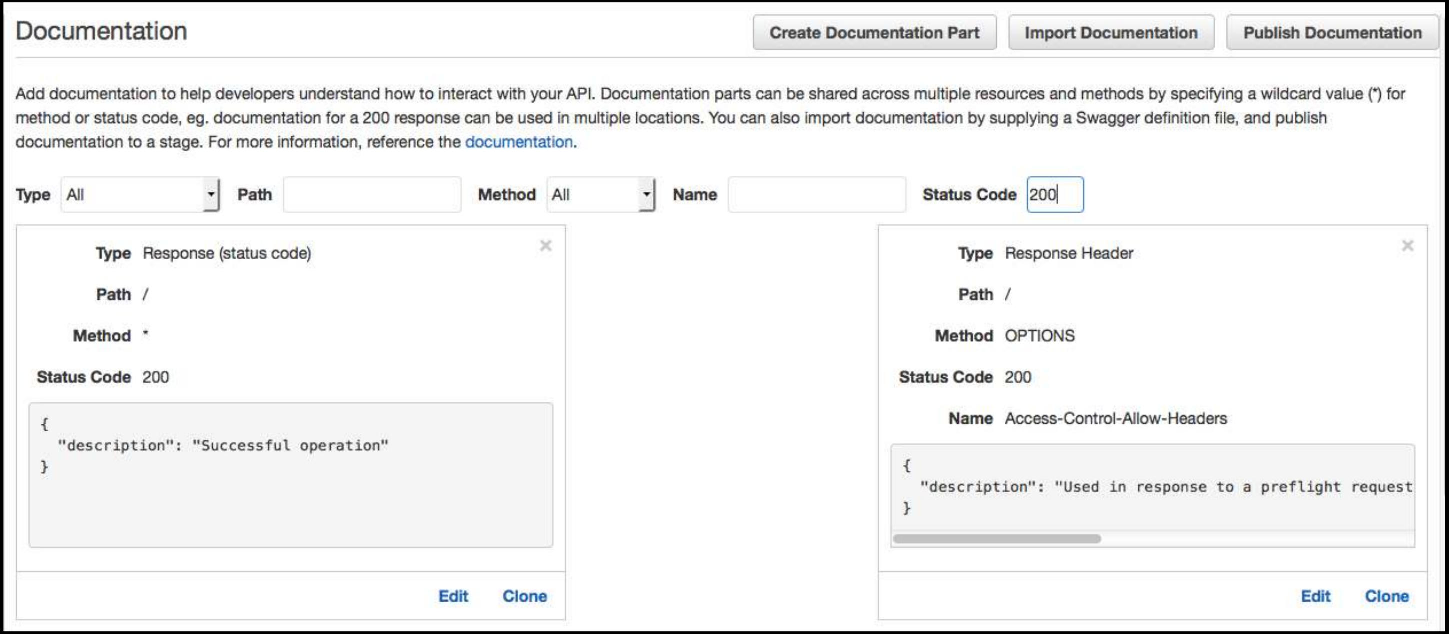
add another here...

Serve images, audio, and other binary content

Automatically base64-encodes Lambda integrations

API documentation New

- Document your APIs – edit doc parts directly in the API Gateway console
- Swagger import/export – fully round-trip-able
- Supports tech writers – independent update and publish flow



Documentation

Add documentation to help developers understand how to interact with your API. Documentation parts can be shared across multiple resources and methods by specifying a wildcard value (*) for method or status code, eg. documentation for a 200 response can be used in multiple locations. You can also import documentation by supplying a Swagger definition file, and publish documentation to a stage. For more information, reference the documentation.

Type: All Path: Method: All Name: Status Code: 200

Type	Response (status code)
Path	/
Method	*
Status Code	200

```
{
  "description": "Successful operation"
}
```

Type	Response Header
Path	/
Method	OPTIONS
Status Code	200
Name	Access-Control-Allow-Headers

```
{
  "description": "Used in response to a preflight request
}
```

API Gateway and AWS Marketplace integration New

- Use API Gateway to simplify building and operating APIs
- Sell your APIs on the AWS Marketplace
- Easy discovery and procurement for your API's consumers
- Track API usage by consumer / key
- Automated billing through AWS

F-Secure. URL Reputation APIs

NTT docomo Speech understanding APIs

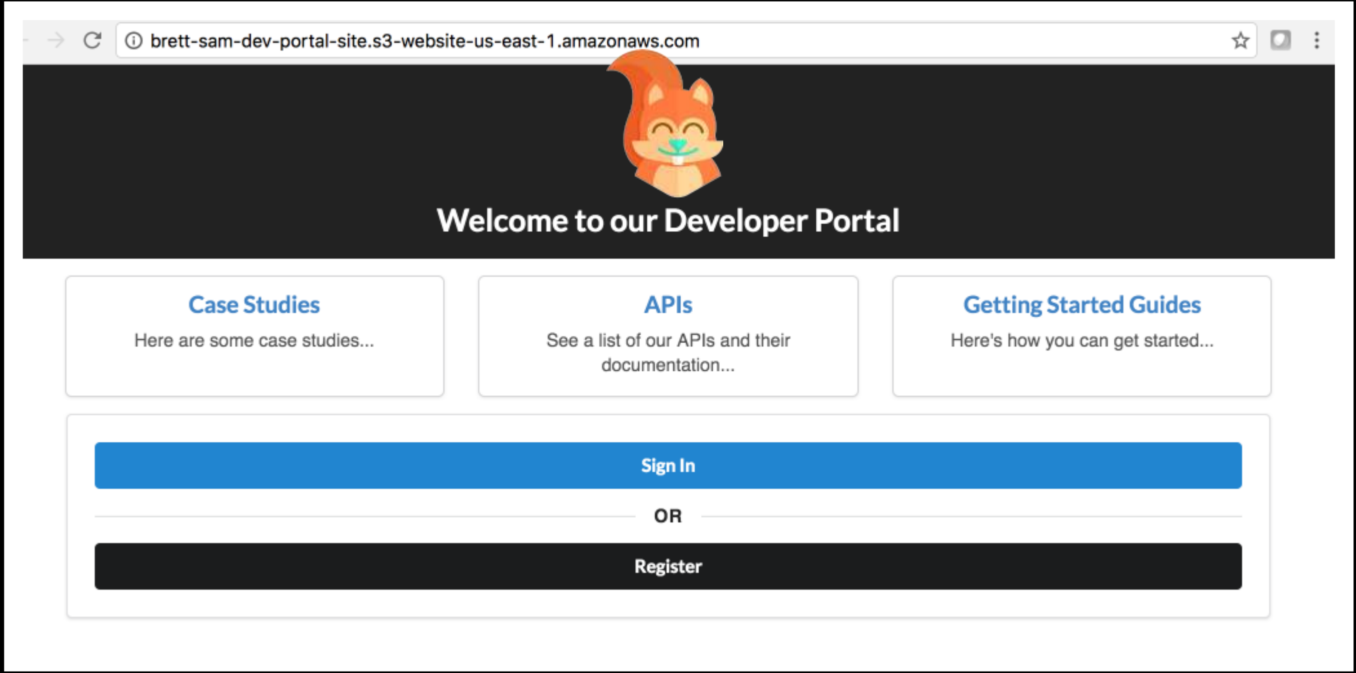
Monetize your microservices!

API Gateway Developer Portal
Open source reference implementation

New

SAM-based implementation available on GitHub

- Help developers consume your APIs
- Vend API Keys
- AWS Marketplace integration
- Supports Cognito authN


brett-sam-dev-portal-site.s3-website-us-east-1.amazonaws.com

Welcome to our Developer Portal

Case Studies
Here are some case studies...

APIs
See a list of our APIs and their documentation...

Getting Started Guides
Here's how you can get started...

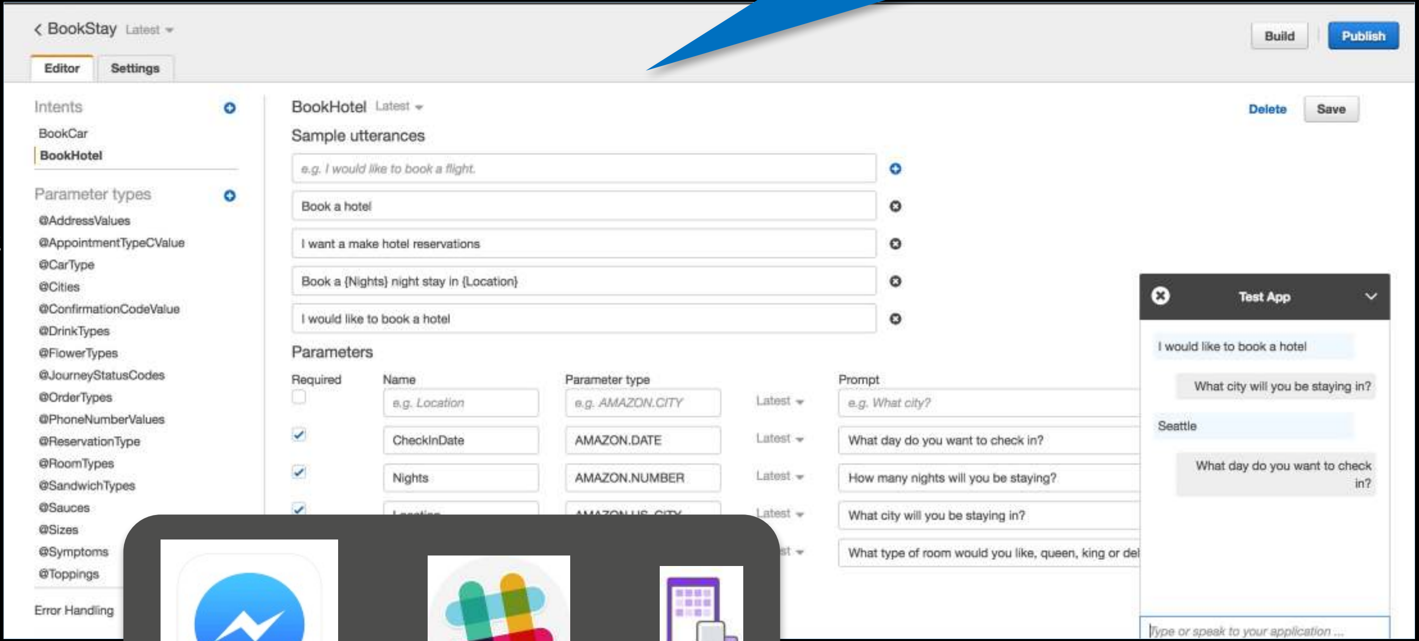
New places you can use Lambda functions

- Lambda Bots
- Amazon Kinesis Firehose
- On-prem storage
- Devices
- Edge/CDN

Lambda Bots and Amazon Lex Preview

- Text and speech
- Lambda functions run business logic
- Facebook, AWS Mobile Hub
- Slack and Twilio integration coming soon

I'd like to book a hotel



< BookStay Latest

Editor | Settings

Build | Publish

Intents
- BookCar
- BookHotel

Parameter types
- @AddressValues
- @AppointmentTypeCValue
- @CarType
- @Cities
- @ConfirmationCodeValue
- @DrinkTypes
- @FlowerTypes
- @JourneyStatusCodes
- @OrderTypes
- @PhoneNumberValues
- @ReservationType
- @RoomTypes
- @SandwichTypes
- @Sauces
- @Sizes
- @Symptoms
- @Toppings

Error Handling

BookHotel Latest

Delete | Save

Sample utterances
- e.g. I would like to book a flight.
- Book a hotel
- I want a make hotel reservations
- Book a {Nights} night stay in {Location}
- I would like to book a hotel

Parameters

Required	Name	Parameter type		Prompt
☐	e.g. Location	e.g. AMAZON.CITY	Latest	e.g. What city?
✓	CheckInDate	AMAZON.DATE	Latest	What day do you want to check in?
✓	Nights	AMAZON.NUMBER	Latest	How many nights will you be staying?
✓	[illegible]	[illegible]	Latest	What city will you be staying in?
			[illegible]	What type of room would you like, queen, king or del

Test App

I would like to book a hotel

What city will you be staying in?

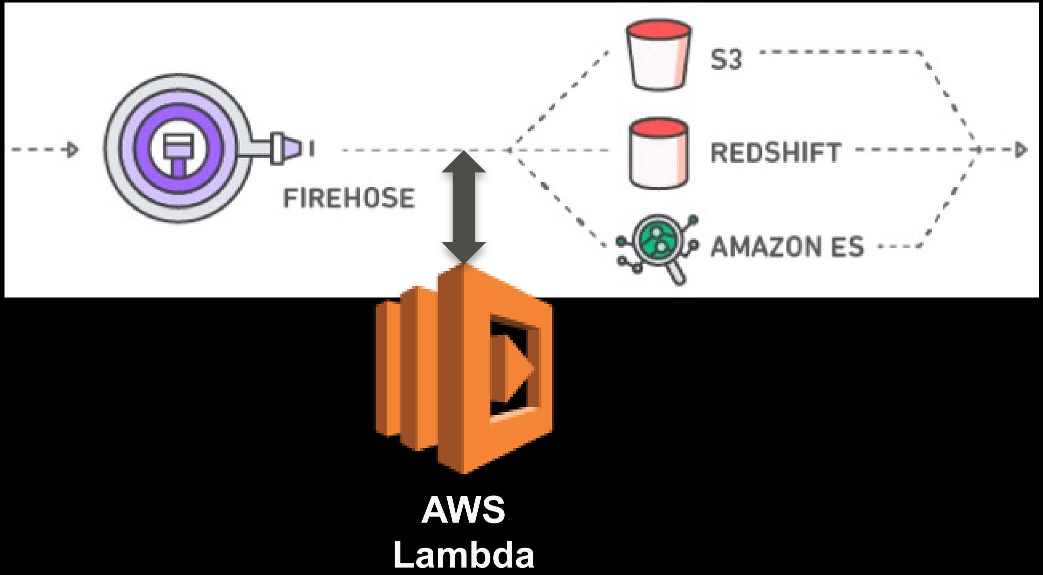
Seattle

What day do you want to check in?

Type or speak to your application ...

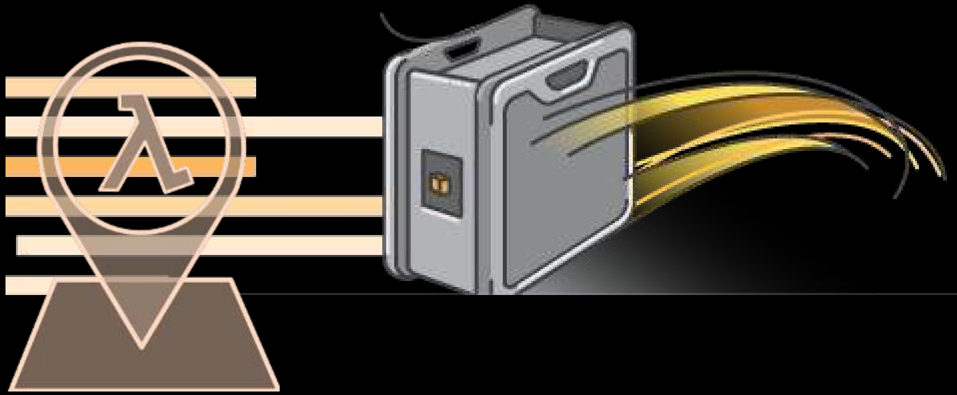
Amazon Kinesis Firehose integration Coming Soon

- Simple, real-time data streaming
- Transform, audit, or aggregate records in flight with Lambda
- Flexible buffering
- Lambda and Firehose both scale automatically



AWS Snowball Edge New

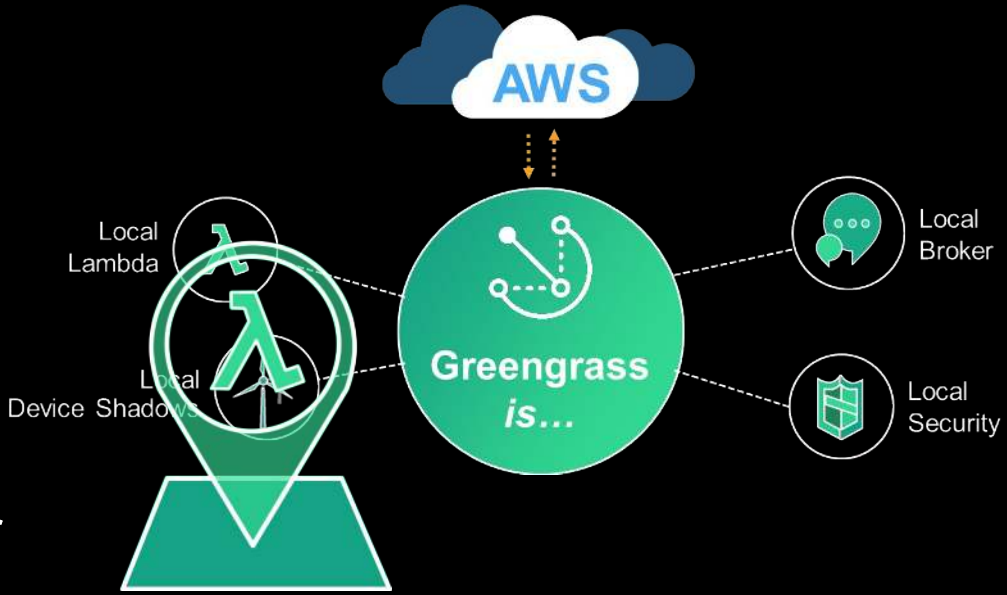
- Fast, simple, secure data transfer from on-prem to/from AWS Cloud
- 100 TB capacity
- Local S3 storage APIs
- **Local Lambda functions**
- Transcode multimedia content, compress in-real time, custom auditing



AWS Greengrass Preview

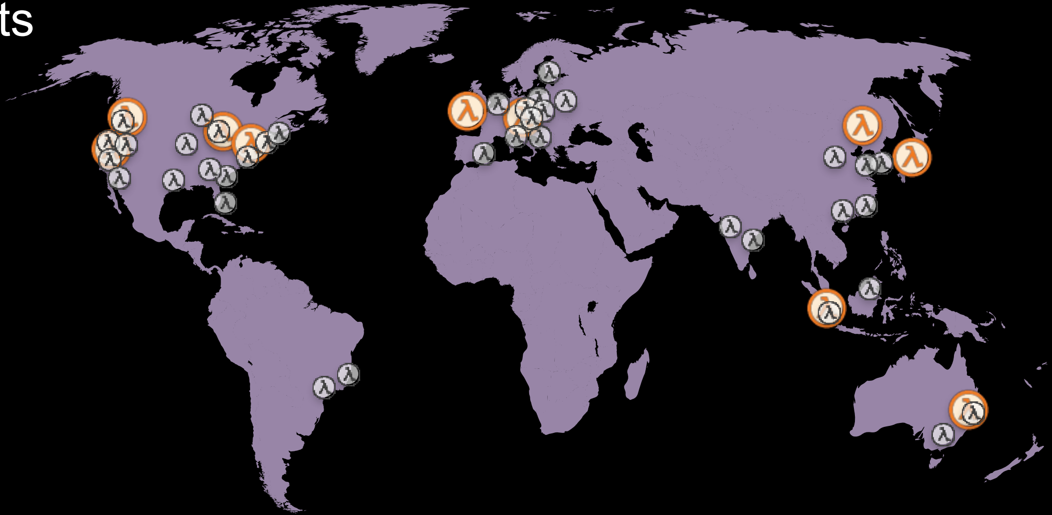
- Greengrass extends AWS processing onto devices
- Low-latency, near-real time

- **Lambda functions run right on the device**
- Cloud storage and compute via AWS IoT service
- BYOH – 1GHz, 128MB, x86 or ARM, Linux



Lambda@Edge Preview

- Low-latency request/response customization
- Supports viewer and origin events
- Preview limitations:
  - Node.js only
  - **50** ms max
  - Headers only
- Pricing: $0.60/M requests and $0.00000625125 per 128MB-s
  - 4K requests free/month



Sign up to join the preview!

Developer ecosystem — commercial

Code Libraries	Integrations	Deployment	Monitoring	APN Skills
ALGORITHMIA	zapier	CloudBees	DATADOG	amazon web services Partner Network
twilio		CODESHIP	loggly	
		MITOC GROUP	splunk>	
			sumologic	

Developer ecosystem — open source

APEX
serverless architecture

MEET SAM

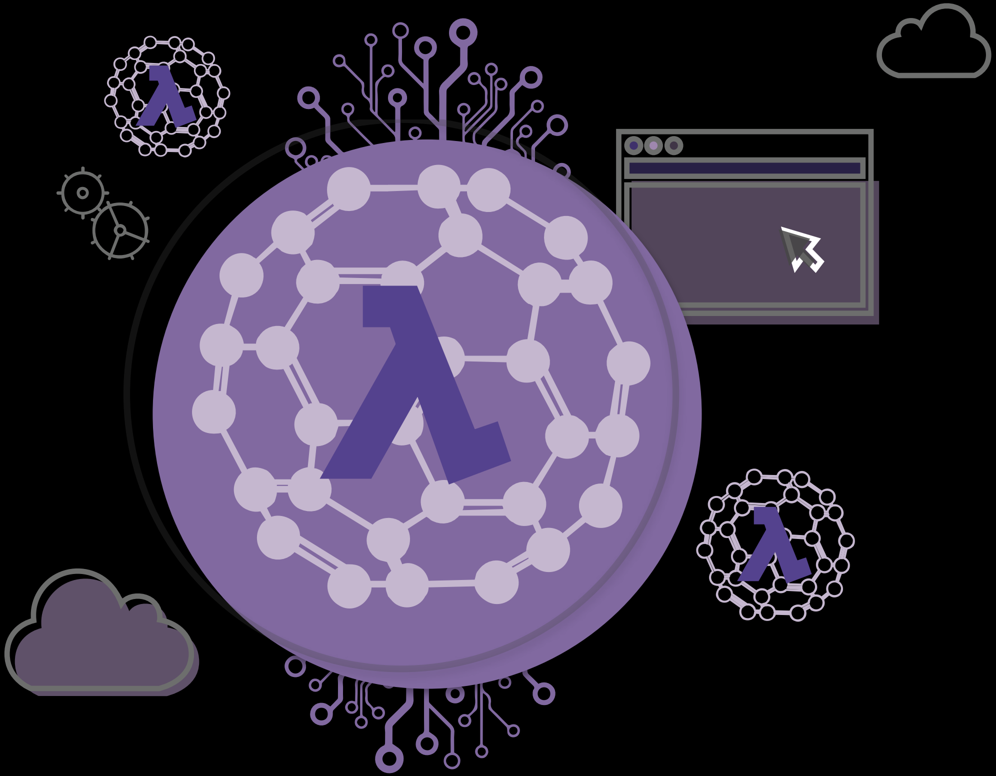
Chalice
Framework

Claudia.js

Gordon

SERVERLESS

Enjoy your serverless journey!



AWS re:Invent

Thank you!

julsimon@amazon.fr
@julsimon

amazon web services