

Lambdaを使ってJawsDays をHackしてみよう!

2016/3/12
JAWS DAYS 2016
Toshihiro Furuno

- WiFi SSID->jawsdays_workshop
- WiFi パスワード->workshop02
- 資料-> [https://github.com/toshihirocks/](https://github.com/toshihirocks/jaws2016-lambda-hands-on)

jaws2016-lambda-hands-on

前提

- ・ 初心者向けの講習なので難易度は高くありません。時間が余った方は改造するなど自由に遊んでみてください。
- ・ Lambdaのコード実装サンプルはNode.jsで行っております。Java,Pythonでのサンプルはない点はご了承ください。

やること、やらないこと

やること

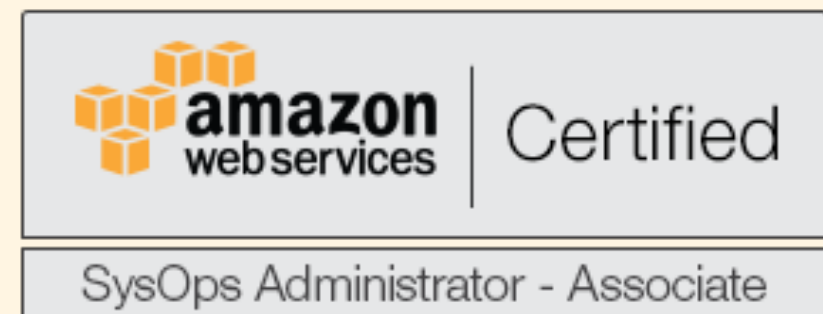
- ・ ハンズオン形式でLambdaで何かを作って遊んでみる(遊びなので業務向けではない)

やらないこと

- ・ Lambdaの詳細な説明
- ・ TwitterAPIの詳細

自己紹介

- 古野俊広(@toshihirocks)
- NTTソフトウェア株式会社
- AWS業務は1年ぐらい
- 好き
 - Lambda
 - Podcast
 - ワールドトリガー



本日のハンズオン

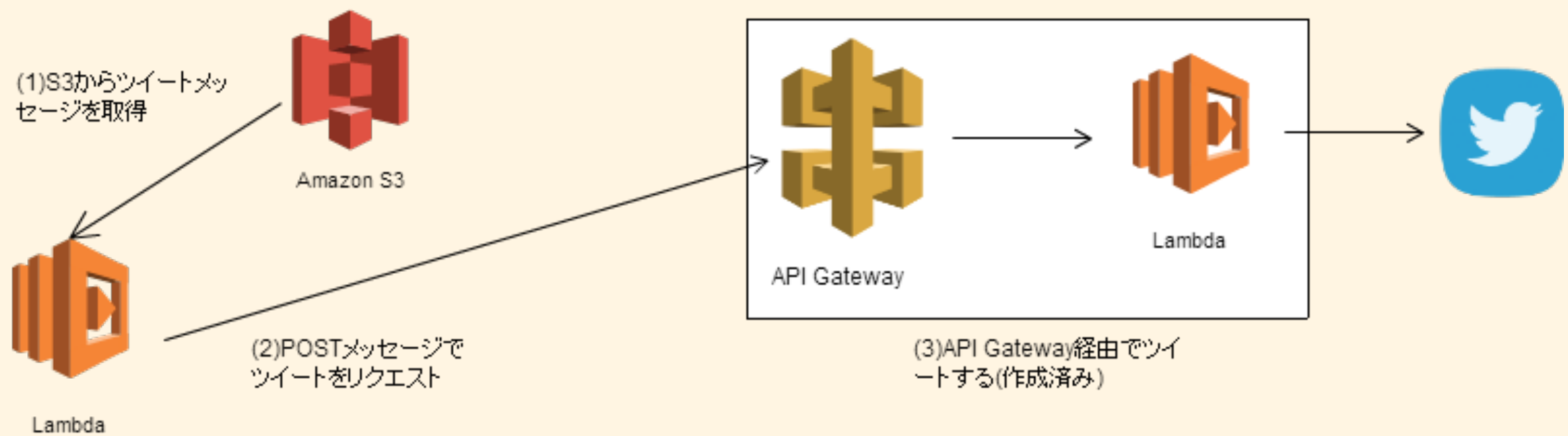


Lambdaを使っ
て #jawsdays にツ
イートしよう!

Lambdaを使ってJawsDaysをHackすっぞ

- ・ ツイートさせたい内容のテキストをS3に配置
- ・ IAMロールを付与したLambdaでS3からテキストファイルを取得し、ランダムにピックアップ
- ・ 予め作成してあるAPIGatewayに向けてPOSTリクエストを実施
- ・ APIGateway経由で@jaws2016_lambdaからハッシュタグ #jawsdays を付けてツイートされる(作成済み)

概要図



知識編

S3?

- ・オンラインストレージ
- ・高い堅牢性(99.999999999999%)
- ・安価
- ・容量無制限(1ファイル5TBまで)



今回はツイート文言を置く場所として利用

Lambda?

- ・ イベントドリブンや
cronのように定期タイミングでユーザーが書
いたコードを実行できる
- ・ コードはNode.js,Java,Python
- ・ 最近VPCも対応
- ・ 詳細は公式参照



IAMロール?

- ・ ユーザーやLambda、EC2などのリソースに対して役割(Role)を指定できる。例えば「S3のRead権限」など
- ・ ロールはLambda作成時に付与することで設定されたAPIのkickができる

今回の例ではLambdaからS3のバケットの情報を取得したいのでLambda向けの上記に該当するIAMロールを作成する

作業の流れ





- ・ S3バケットにツイート文言の配置
- ・ Lambda用IAMロールの作成
- ・ Lambda Functionの作成
- ・ Lambda Functionの定期実行の設定

準備編(S3)







サービス一覧からS3の選択

Amazon Web Services

Compute

-  **EC2**
Virtual Servers in the Cloud
-  **EC2 Container Service**
Run and Manage Docker Containers
-  **Elastic Beanstalk**
Run and Manage Web Apps
-  **Lambda**
Run Code in Response to Events

Storage & Content Delivery

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System** **PREVIEW**
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Import/Export Snowball**
Large Scale Data Transport
-  **Storage Gateway**
Hybrid Storage Integration








Database

 **RDS**

Developer Tools

-  **CodeCommit**
Store Code in Private Git Repositories
-  **CodeDeploy**
Automate Code Deployments
-  **CodePipeline**
Release Software using Continuous Delivery

Management Tools

-  **CloudWatch**
Monitor Resources and Applications
-  **CloudFormation**
Create and Manage Resources with Templates
-  **CloudTrail**
Track User Activity and API Usage
-  **Config**
Track Resource Inventory and Changes
-  **OpsWorks**
Automate Operations with Chef
-  **Service Catalog**
Create and Use Standardized Products
-  **Trusted Advisor**
Optimize Performance and Security

Security & Identity






Internet of Things

-  **AWS IoT**
Connect Devices to the Cloud

Game Development

-  **GameLift**
Deploy and Scale Session-based Multiplayer Games

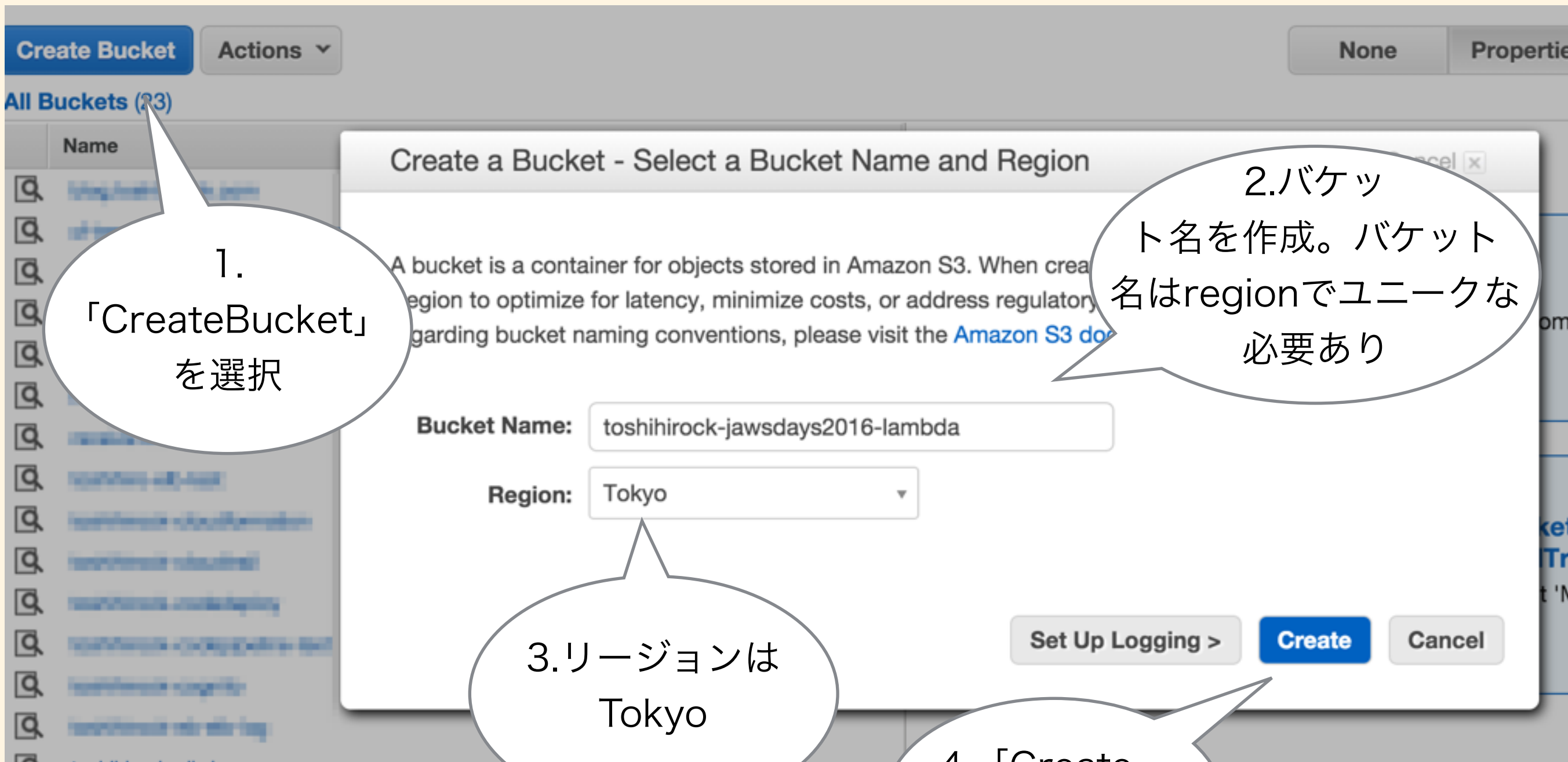
Mobile Services

-  **Mobile Hub**
Build, Test, and Monitor Mobile Apps
-  **Cognito**
User Identity and App Data Synchronization
-  **Device Farm**
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
-  **Mobile Analytics**
Collect, View and Export App Analytics
-  **SNS**
Push Notification Service

Application Services

-  **API Gateway**
Build, Deploy and Manage APIs
-  **AppStream**

バケットの作成



1.
「CreateBucket」
を選択

2.バケッ
ト名を作成。バケット
名はregionでユニークな
必要あり

3.リージョンは
Tokyo

4.「Create」
を選択

S3へ配置するテキストファイルの作成

- ・ 各自のPCの好きなエディタでツイートさせたい文言集を作成してください。文字コードはUTF-8で保存

- ・ 任意のデリミタ区切りで作成してください。

以下例はデリミタは改行(\r\n)

- ・ 作成後、「tweetList.txt」でローカルに保存

```
1   おそ松
2   カラ松
3   チョロ松
4   一松
5   十四松
6   トド松
```

確認

文字コード確認

```
$nkf -g tweetList.txt
```

改行コード確認.LF = \$.CR = ^Mで表示

```
$cat -e tweetList.txt
```

S3へテキストファイルのアップロード

先ほど作成したバケットを選択後、以下の操作でアップロード

The screenshot shows the 'Upload - Select Files and Folders' dialog box in the AWS S3 console. The dialog is titled 'Upload - Select Files and Folders' and has a 'Cancel' button in the top right corner. The 'Upload to:' field shows 'All Buckets / toshihiro-rock-jawsdays2016-lambda'. Below this, there is a text area with instructions: 'To upload files (up to 5 TB each) to Amazon S3, click **Add Files**. You can also drag and drop files and folders to the area below. To remove files already selected, click the **X** to the far right of the file name.' Below the text area is a large dashed box with the text 'Drag and drop files and folders to upload here.' Below this box is a list of files, currently showing 'tweetList.txt (20 bytes)' with an 'X' icon to its right. Below the file list are two buttons: '+ Add Files' and '- Remove Selected Files'. Below these buttons is a summary line: '1 Total upload size: 20 bytes'. At the bottom of the dialog are three buttons: 'Set Details >', 'Start Upload', and 'Cancel'. Three callouts are overlaid on the image: 1. A callout pointing to the 'Upload' button in the top left of the console, with the text '1. 「Upload」を選択'. 2. A callout pointing to the dashed drop area, with the text '2. テキストをドラッグアンドドロップでアップロード'. 3. A callout pointing to the 'Start Upload' button, with the text '3. 「Start Upload」を選択'.

1. 「Upload」を選択

2. テキストをドラッグアンドドロップでアップロード

3. 「Start Upload」を選択

確認

Upload

Create Folder

Actions ▾


🔍 Search by prefix

None

Properties

Transfers


All Buckets / toshihiro-rock-lambda-test

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	 tweetList.txt	Standard	60 bytes	Fri Mar 11 07:...

Transfers

☐ Automatically clear finished transfers

✓ Done

⬆ Upload:  Uploading tweetList.txt to toshihiro-rock-lambda-test

準備編(IAMロール)

サービス一覧からIAMの選択

Amazon Web Services

Category	Service	Description
Compute	EC2	Virtual Servers in the Cloud
	EC2 Container Service	Run and Manage Docker Containers
	Elastic Beanstalk	Run and Manage Web Apps
	Lambda	Run Code in Response to Events
Storage & Content Delivery	S3	Scalable Storage in the Cloud
	CloudFront	Global Content Delivery Network
	Elastic File System	Fully Managed File System for EC2 (PREVIEW)
	Glacier	Archive Storage in the Cloud
	Import/Export Snowball	Large Scale Data Transport
	Storage Gateway	Hybrid Storage Integration
Database	RDS	Managed Relational Database Service
	DynamoDB	Managed NoSQL Database
	ElastiCache	In-Memory Cache
Developer Tools	CodeCommit	Store Code in Private Git Repositories
	CodeDeploy	Automate Code Deployments
	CodePipeline	Release Software using Continuous Delivery
Management Tools	CloudWatch	Monitor Resources and Applications
	CloudFormation	Create and Manage Resources with Templates
	CloudTrail	Track User Activity and API Usage
Security & Identity	Identity & Access Management	Manage User Access and Encryption Keys
	Inspector	Analyze Application Security (PREVIEW)
	Trusted Advisor	Optimize AWS Resource Usage and Security
Internet of Things	AWS IoT	Connect Devices to the Cloud
	GameLift	Deploy and Scale Session-based Multiplayer Games
	Mobile Hub	Build, Test, and Monitor Mobile Apps
Game Development	Cognito	User Identity and App Data Synchronization
	Device Farm	Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
	Mobile Analytics	Collect, View and Export App Analytics
Mobile Services	SNS	Push Notification Service
	API Gateway	Build, Deploy and Manage APIs
	AppStream	Low Latency Application Streaming
Application Services	CloudSearch	Managed Search Service
	Elastic Transcoder	Easy-to-Use Scalable Media Transcoding
	SES	Simple Email Service

ロールの新規作成

1. 「Roles」を選択

2. 「Create New Role」を選択

Showing 36 results

<input type="checkbox"/>	Name ↕	Creation Time ↕
<input type="checkbox"/>	ec2Role	2015-07-11 07:41 UTC+0900
<input type="checkbox"/>		2015-07-21 15:52 UTC+0900
<input type="checkbox"/>		2015-08-04 22:02 UTC+0900
<input type="checkbox"/>	service-role	2015-09-18 20:56 UTC+0900
<input type="checkbox"/>	aws-opsworks-ec2-role	2015-09-13 20:30 UTC+0900
<input type="checkbox"/>	aws-opsworks-service-role	2015-09-13 20:30 UTC+0900
<input type="checkbox"/>	CloudTrail CloudWatchLogs Role	2015-07-01 03:15 UTC+0900

ロール名の決定

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Step 5: Review

Set Role Name

Enter a role name. You cannot edit the role name after the role is created.

Role Name

Maximum 64 characters. Use alphanumeric and '+=,@-_' characters

1. 任意の
RoleNameを入力

2. 「Next
Step」を選択

Cancel

Next Step

ロールタイプの選択

Lambdaにアタッチしたいロールを作成するのでLambdaを選択します

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Step 5: Review

Select Role Type

☒ AWS Service Roles

› Amazon EC2

Allows EC2 instances to call AWS services on your behalf.

Select

› AWS Directory Service

Allows AWS Directory Service to manage access for existing directory users and groups to AWS services.

Select

› AWS Lambda

Allows Lambda Function to call AWS services on your behalf.

Select

› Amazon API Gateway

Allows API Gateway to call AWS resources on your behalf.

Select

› AWS Config

Allows AWS Config to call AWS services and collect resource configurations on your behalf.

Select

☐ Role for Cross-Account Access

☐ Role for Identity Provider Access

「AWS
Lambda」を選択

ポリシーの選択

既にAWS側で用意しているS3へのReadOnly
のポリシーを付与

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Step 5: Review

Attach Policy

Select one or more policies to attach. Each role can have up to 10 policies.

Filter: Policy Type Showing 2 results

		Policy Name ⇅	Attached Entities ⇅	Creation Time ⇅	Edited Time ⇅
<input checked="" type="checkbox"/>		AmazonS3ReadOnlyAcc...	4	2015-02-07 03:40 UTC+0...	2015-02-07 03:40 UT...
<input type="checkbox"/>		AmazonS3FullAccess	2	2015-02-07 03:40 UTC+0...	2015-02-07 03:40 UT...

1. 「s3」と入力

2. 「AmazonS3ReadOnlyAccess」を選択

3. 「Next Step」を選択

Cancel Previous Next Step

レビュー

Create Role

[Step 1: Set Role Name](#)

[Step 2: Select Role Type](#)

[Step 3: Establish Trust](#)

[Step 4: Attach Policy](#)

Step 5: Review

Review

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

Role Name	lambda_test	Edit Role Name
Role ARN	arn:aws:iam::123456789012:role/lambda_test	
Trusted Entities	The identity provider(s) lambda.amazonaws.com	
Policies	arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess	Change Policies

確認後、
「Create Role」
を選択

[Cancel](#)

[Previous](#)

Create Role

Lambdaを作るう!

(後で利用)コードの準備

- ・ 以下の内容をコピーして任意のテキストエディタに貼っておく
- ・ <https://github.com/toshihirocks/jaws2016-lambda-hands-on/blob/master/index.js>

(後で利用)コードの準備

APIGatewayの
認証キーを指定





```
1 console.log('Loading function');
2
3 // API Gatewayの設定
4 const HOST = 'wjz7li7gmj.execute-api-northeast-1.amazonaws.com';
5 const PATH = '/prod/statuses/update';
6 const X_API_KEY = '';
7
8 // S3の設定
9 const BUCKET = 'toshihiro-rock-jaws2016-lambda';
10 const KEY = 'tweetList.txt';
11 const TEXT_DELIMITER = '\r\n';
12
```







S3の設定は
適宜自分の設定に
変更

サービス一覧からLambdaの選択





Amazon Web Services

Compute

-  **EC2**
Virtual Servers in the Cloud
-  **EC2 Container Service**
Run and Manage Docker Containers
-  **Elastic Beanstalk**
Run and Manage Web Apps
-  **Lambda**
Run Code in Response to Events

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System** PREVIEW
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Import/Export Snowball**
Large Scale Data Transport
-  **Storage Gateway**
Hybrid Storage Integration







Database

-  **RDS**
Managed Relational Database Service
-  **DynamoDB**
Managed NoSQL Database
-  **ElastiCache**
In-Memory Cache
-  **Redshift**




Developer Tools

-  **CodeCommit**
Store Code in Private Git Repositories
-  **CodeDeploy**
Automate Code Deployments
-  **CodePipeline**
Release Software using Continuous Delivery

Management Tools

-  **CloudWatch**
Monitor Resources and Applications
-  **CloudFormation**
Create and Manage Resources with Templates
-  **CloudTrail**
Track User Activity and API Usage
-  **Config**
Track Resource Inventory and Changes
-  **OpsWorks**
Automate Operations with Chef
-  **Service Catalog**
Create and Use Standardized Products
-  **Trusted Advisor**
Optimize Performance and Security

Security & Identity

-  **Identity & Access Management**
Manage User Access and Encryption Keys
-  **Directory Service**
Host and Manage Active Directory
-  **Inspector** PREVIEW
Analyze Application Security






Internet of Things

-  **AWS IoT**
Connect Devices to the Cloud






Game Development

-  **GameLift**
Deploy and Scale Session-based Multiplayer Games

Mobile Services

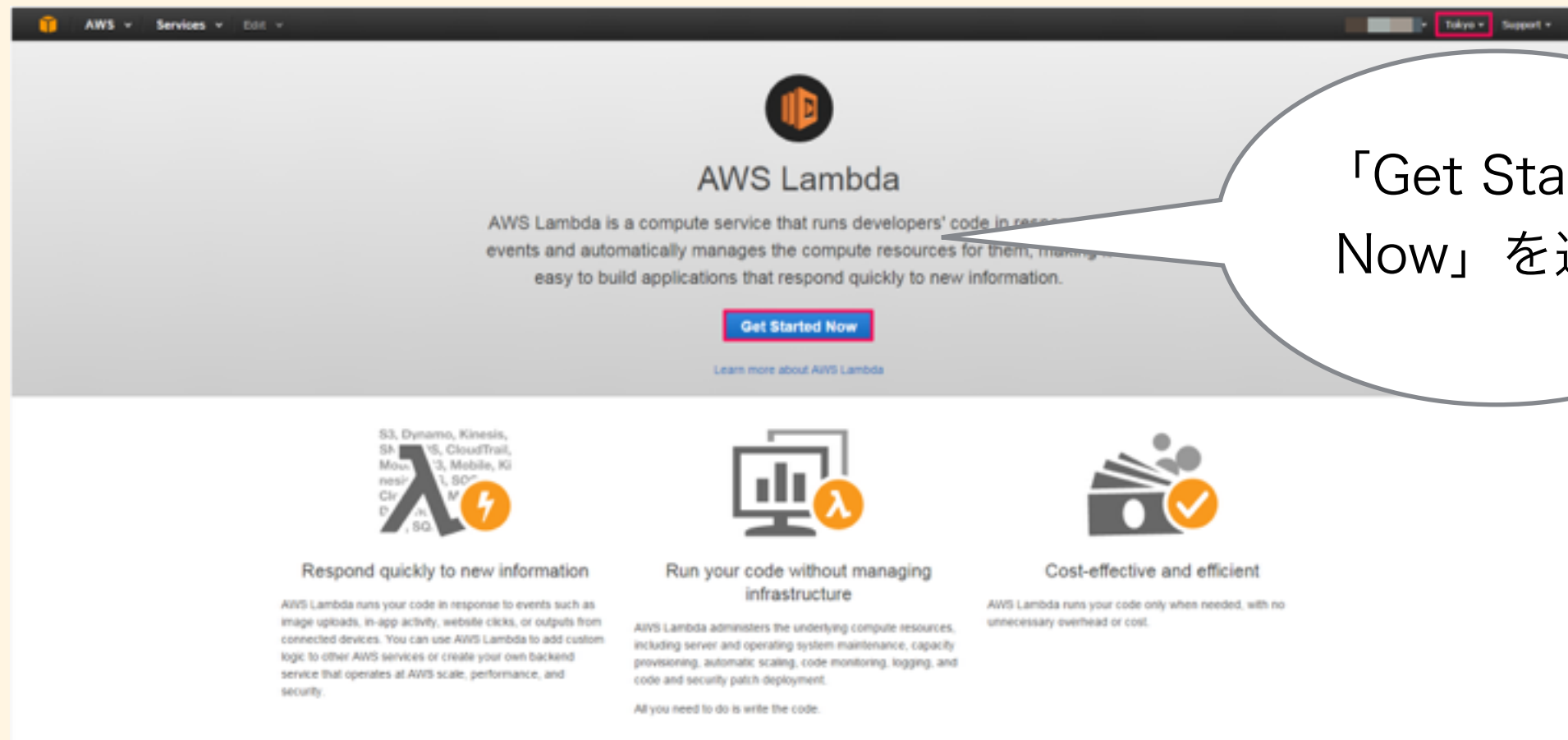
-  **Mobile Hub**
Build, Test, and Monitor Mobile Apps
-  **Cognito**
User Identity and App Data Synchronization
-  **Device Farm**
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
-  **Mobile Analytics**
Collect, View and Export App Analytics
-  **SNS**
Push Notification Service

Application Services

-  **API Gateway**
Build, Deploy and Manage APIs
-  **AppStream**
Low Latency Application Streaming
-  **CloudSearch**
Managed Search Service
-  **Elastic Transcoder**
Easy-to-Use Scalable Media Transcoding
-  **SES**

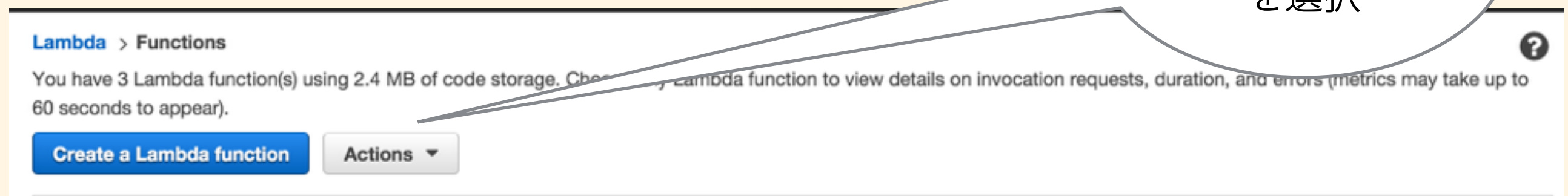
Functionの作成

※初めての方の画面



「Get Started Now」を選択

※既にFunctionがある人の画面



「Create a Lambda function」を選択

Blueprint

Filter

All languages

<< < Viewing 1-9 of 40 > >>

s3-get-object-python

An Amazon S3 trigger that retrieves metadata for the object that has been updated.

python2.7 · s3

config-rule-change-triggered

An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types.

nodejs · config

dynamodb-process-stream

An Amazon DynamoDB trigger that logs the updates made to a table.

nodejs · dynamodb

microservice-http-endpoint

A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.

nodejs · api-gateway

node-exec

Demonstrates running an external process using the Node.js child_process module.

nodejs

slack-echo-command-python

A function that handles a Slack slash command and echoes the details back to the user.

python2.7 · api-gateway

simple-mobile-backend

A simple mobile backend (read/write to DynamoDB).

nodejs · mobile

kinesis-process-record-python

An Amazon Kinesis stream processor that logs the data being published.

python2.7 · kinesis

splunk-kinesis

Demonstrates ingesting data from AWS Kinesis into Splunk Event Collector.

nodejs · splunk · kinesis

Cancel

Skip

利用しない
ので「Skip」を選
択

Functionの設定(1/2)

Lambda > New function

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name* tweetJawsDays2016

Description tweet jaws days 2016

Runtime* Node.js

Name及び
Descriptionは何
でもOK

Node.js

Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. [Learn more](#) about deploying Lambda functions.

Code entry type ☒ Edit code inline ☐ Upload a .ZIP file ☐ Upload a .ZIP from [Amazon S3](#)

```
1 console.log('Loading function');
2
3 exports.handler = function(event, context, callback) {
4     //console.log('Received event:', event, context, callback);
5     callback(null, 2);
6 }
```

Edit code
inline

先
ほどの内容を
コピペ

Functionの設定(2/2)

Lambda function handler and role

Handler*

index.handler



Index.handler

Role*

lambda_s3_exec_role



Ensure that popups are enabled to create a new role. [Learn more](#) about Lambda

先
ほど作成したS3へ
アクセスできるRole
名

Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. (selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda

Memory (MB)*

128



Timeout*

0

min

3

sec

You can optionally configure Lambda to access resources, such as databases, within your VPC. [Learn more](#) about VPCs within Lambda. **Please ensure your role has appropriate permissions to configure VPC. Select "Basic" to add these permissions.**

No VPC

VPC

No VPC



* These fields are required.

[Cancel](#)

[Previous](#)

[Next](#)

最後に
「Next」

レビュー

Lambda > New function

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

Review

Please review your Lambda function details. You can go back to edit changes for each section. When you are ready, click **Create function** to complete the setup process.

Lambda function

Edit

Name	tweetJawsDays2016
Description	tweet jaws days 2016
Runtime	NodeJS
Handler	index.handler
Role	lambda_s3_exec_role
Memory (MB)	128
Timeout	3

設定が問題なければ「Create function」

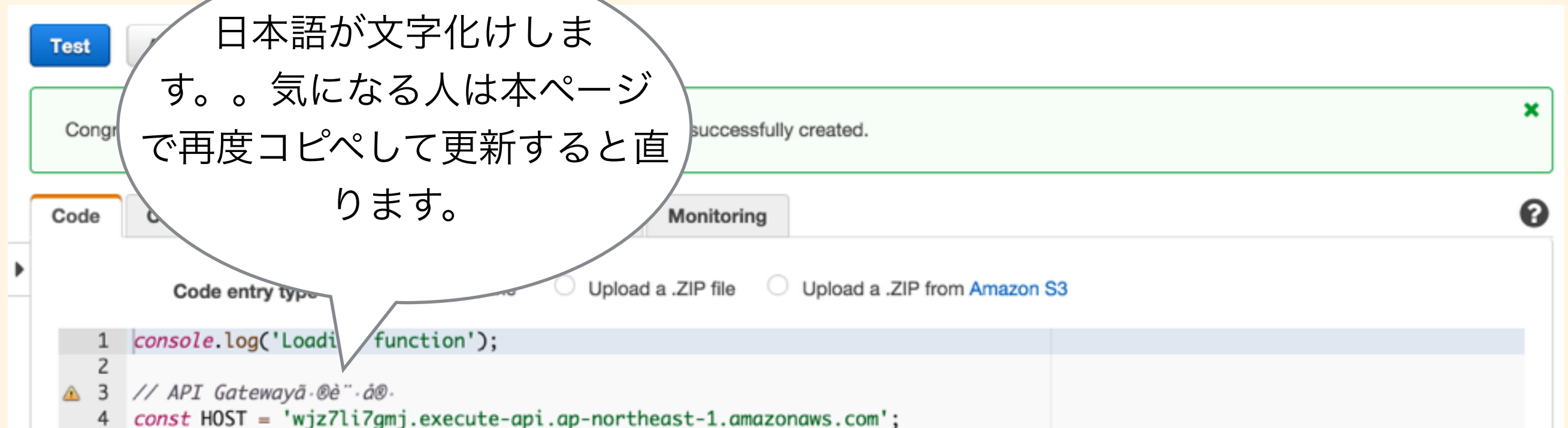
Cancel

Previous

Create function

作成

日本語が文字化けします。。気になる人は本ページで再度コピーして更新すると直ります。



テストしてみる

Lambda > **Functions** > **tweetJawsDays2016**

Test

Actions ▼

「Test」を選択

テストしてみる

Input test event

It looks like you have not configured a test event for this function yet. Use the editor below to configure a test event for your function with (please remember that this will actually execute the code!). You can always edit or delete the test event in the Actions list. Note that changes to the event will only be saved locally.

Sample event template

Hello World

```
1 {  
2   "key3": "value3",  
3   "key2": "value2",  
4   "key1": "value1"  
5 }
```

テンプレートは何でもOK(ツイート情報はS3から取得するので)

「Save and test」を選択

Cancel

Save

Save and test

テストしてみる

成功すれば画面下部
にツイートした文字列が表示
される

失敗すると赤いエラー表示
になる

✓ Execution result: succeeded ([logs](#))

The area below shows the result returned by your function execution using the context methods. [Learn more](#) about returning results from your function.

"かきくけこ"

❗ Execution result: failed ([logs](#))

The area below shows the result returned by your function execution using the context methods. [Learn more](#) about returning results from your function.

```
{
  "errorMessage": "Error. status code = 403. Error message = undefined",
  "errorType": "Error",
  "stackTrace": [
    "IncomingMessage.<anonymous> (/var/task/index.js:55:20)",
    "IncomingMessage.emit (events.js:117:20)",
    "_stream_readable.js:944:16",
    "process._tickDomainCallback (node.js:492:13)"
  ]
}
```

テストしてみる

Twitterでも確認



jaws2016_lambda @jaws2016_lambda · 2分
test #jawsdays



エラーが発生している場合

- ・ 画面下部(Log output)にログ情報が表示されるので確認
- ・ 必要に応じてconsole.logなどを仕込んでデバッグしてみる

Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

```
START RequestId: 34f7719e-e711-11e5-80de-c387c9822253 Version: $LATEST
2016-03-10T22:41:20.270Z      34f7719e-e711-11e5-80de-c387c9822253      S3 data = おそ松
カラ松
チョロ松
一松
十四松
トド松

2016-03-10T22:41:21.119Z      34f7719e-e711-11e5-80de-c387c9822253      response code = 200
2016-03-10T22:41:21.120Z      34f7719e-e711-11e5-80de-c387c9822253      {"created_at":"Thu Mar 10
22:41:20 +0000 2016","id":708060197726453800,"id_str":"708060197726453763","text":"一松 #jawsdays",
```

**botらしく
定期実行させてみる**

定期実行の設定

「Event sources」 タブを選択

The screenshot shows the AWS Lambda console interface for configuring event sources. At the top, there are buttons for 'Test' and 'Actions'. Below these are five tabs: 'Code', 'Configuration', 'Event sources', 'API endpoints', and 'Monitoring'. The 'Event sources' tab is selected and highlighted with an orange border. Below the tabs, a message states: 'You do not have any event sources for this function.' Underneath this message is a blue link with a plus icon that says '+ Add event source'. Two callout boxes provide instructions: one points to the 'Event sources' tab with the text '「Event sources」 タブを選択', and another points to the '+ Add event source' link with the text '「Add event source」を選択'.

Test Actions ▾

Code Configuration **Event sources** API endpoints Monitoring

You do not have any event sources for this function.

[+ Add event source](#)

「Add event source」を選択

定期実行の設定

Add event source

Configure your Lambda function to respond to events from the event sources listed below. You can also configure your Lambda function directly using the AWS [mobile SDK](#) for [Android](#) and [iOS](#).

Event source type CloudWatch Events - Schedule ⓘ

Rule name Tweet-15minutes-interval ⓘ

Rule description ⓘ

Schedule expression rate(15 minutes) ⓘ

Lambda will add the necessary permissions for **CloudWatch Events** to invoke your function. For [more](#) about the Lambda permissions model.

Enable event source ☒ Enable now ☐ Enable later ⓘ

Submit

「CloudWatch Events - Schedule」を選択

名称を適宜変更

任意の時間間隔を設定
(最小5分)

Submitして決定

確認

しばし待って確認

Twitter上でも確認



jaws2016_lambda @jaws2016_lambda · 3 時間
かきくけこ #jawsdays



jaws2016_lambda @jaws2016_lambda · 3 時間
さしすせそ #jawsdays



jaws2016_lambda @jaws2016_lambda · 3 時間
あいうえお #jawsdays



時間が余った人は

時間が余った人は

- ・ 好きに色々やってみてください！（定期実行間隔を変える、文言のパターンを増やすなど）
- ・ Lambdaから直接TwitterのAPIを叩くなどをしてみたい場合は以前書いたこちらを参考にやってみてください

<http://qiita.com/toshihiroock/items/d28505442526e0ae7793>

Thanks!