

# クラウド道場ハンズオン

初級 Lambda 編

2018.2.3

## 目次

構成図.....	1
ログイン .....	1
S3 の設定 .....	3
バケットの作成.....	3
Lambda の設定 .....	5
実行 .....	9
ログ .....	11

## 構成図



主に S3、Lambda にて構成されています。

## ログイン



AWS の HP にアクセスし、右上の「コンソールへログイン」をクリックします。  
( URL : <https://aws.amazon.com/jp/> )



アカウント:

ユーザー名:

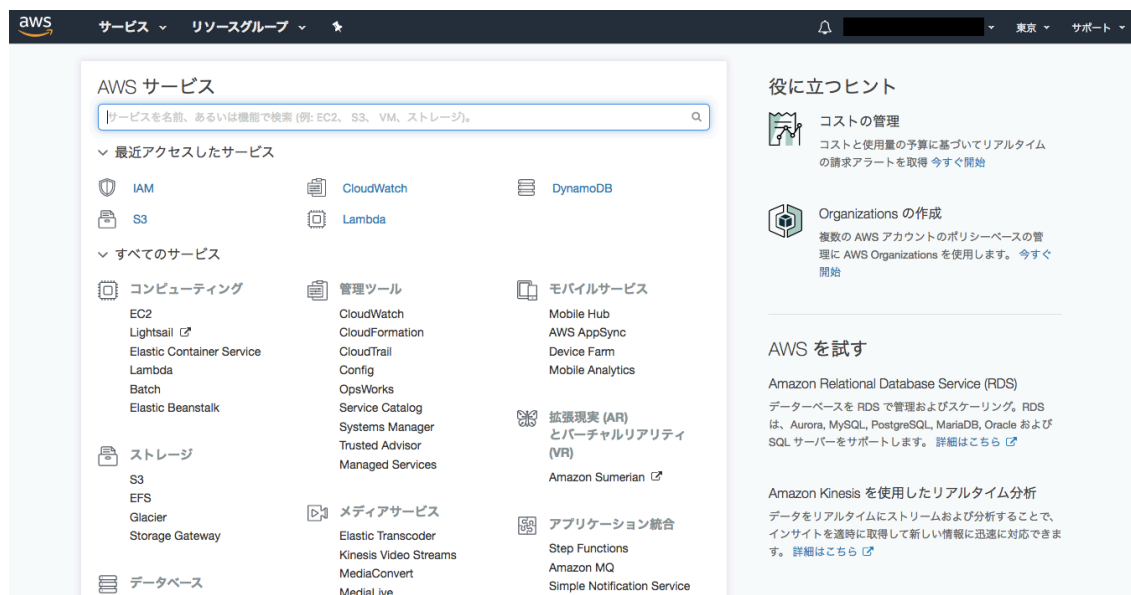
パスワード:

サインイン

[ルートアカウント認証情報を使用してサインイン](#)

アカウント情報を入力し、「サインイン」をクリックします。

## S3 の設定



ログインが完了すると、画像のような画面になります。

右上の囲い部分が「東京」になっていない場合は、クリックし、「アジアパシフィック（東京）」を選択しておきます。

「ストレージ」の「S3」をクリックします。

## バケットの作成



「S3」をクリックすると、画像のような画面になります。

「バケットを作成する」をクリックし、バケットを作成します。

バケットは、感覚的にはフォルダのようなものであり、ここでは振り分けするためにアップロードする画像や振り分け後の画像を保管する場所を作成します。



上記のようなウィンドウが出現するので、バケット名、リージョンを設定し、「作成」をクリックします。

バケット名はすべての S3 ユーザーの中で一意である必要があります。

ここでは、「20180203-yamadataro」など自分の名前などを入れるといいでしょう。

リージョンは「アジアパシフィック (東京)」を指定します。

Amazon S3 新機能のご紹介

バケット検索

バケットを作成する 削除する 空にする

5 バケット 0 パブリック 3 リージョン

バケット名	アクセス	リージョン	作成日
	非公開	米国東部 (オハイオ)	10月 25, 2017 7:08:00 午後 GMT+0900
lambda-test-20180201	非公開	アジアパシフィック (東京)	2月 1, 2018 3:17:46 午後 GMT+0900
	非公開	アジアパシフィック (東京)	12月 19, 2017 9:18:44 午後 GMT+0900
	非公開	アジアパシフィック (東京)	10月 26, 2017 11:49:39 午後 GMT+0900
	非公開	米国東部 (バージニア北部)	1月 16, 2018 5:42:40 午後 GMT+0900

\* オブジェクトは、オブジェクト ACL により、依然としてパブリックアクセス可能である場合があります。

完了すると作成したバケットが一覧に追加されます。  
その後、画面左上の「サービス」をクリックします。

## Lambda の設定

サービスを名前、あるいは機能で検索 (例: EC2、S3、VM、ストレージ)。

グループ A-Z

コンピューティング

EC2  
Lightsail  
Elastic Container Service  
Lambda  
Batch  
Elastic Beanstalk

開発者用ツール

CodeStar  
CodeCommit  
CodeBuild  
CodeDeploy  
CodePipeline  
Cloud9  
X-Ray

機械学習

Amazon SageMaker  
Amazon Comprehend  
AWS DeepLens  
Amazon Lex  
Machine Learning  
Amazon Polly  
Rekognition  
Amazon Transcribe  
Amazon Translate

拡張現実 (AR) とバーチャルリアリティ (VR)

Amazon Sumerian

アプリケーション統合

Step Functions  
Amazon MQ  
Simple Notification Service  
Simple Queue Service  
SWF

ストレージ

S3  
EFS  
Glacier  
Storage Gateway

管理ツール

CloudWatch  
AWS Auto Scaling  
CloudFormation  
CloudTrail

分析

Athena  
EMR

カスタマーエンゲージメント

Amazon Connect

サービスの一覧が表示されるので、「コンピューティング」から「Lambda」をクリックします。

aws サービス リソースグループ

東京 サポート

AWS Lambda

Lambda > 関数

関数 (12)

アクション 関数の作成

タグや属性によるフィルター、またはキーワードによる検索

関数名	説明	ランタイム	コードサイズ	最終更新日時
		Python 3.6	909 bytes	先月
		Python 3.6	1.2 kB	2 か月前
		Python 3.6	3.0 kB	先月
		Python 3.6	3.0 kB	3 か月前
		Python 3.6	2.3 kB	先月
		Python 3.6	3.0 kB	2 か月前

画像のような画面になるので、右上の「関数の作成」をクリックします。

Lambda > 関数 > 関数の作成

### 関数の作成

**一から作成**  
シンプルな「Hello World」の例で開始します。

**設計図**  
Lambda 関数の開始点として、事前設定されたテンプレートを選択します。

**設計図 情報** エクスポート

Q フィルターを追加

keyword: s3- ⓧ

**s3-get-object-python**  
An Amazon S3 trigger that retrieves metadata for the object that has been updated.  
python2.7 · s3

**s3-get-object**  
An Amazon S3 trigger that retrieves metadata for the object that has been updated.  
nodejs6.10 · s3

**s3-get-object-python3**  
An Amazon S3 trigger that retrieves metadata for the object that has been updated.  
python3.6 · s3

キャンセル 設定

画像のような画面になるので、「設計図」をチェックします。

チェックすると、設計図を選択するよう促されるので、検索欄に「s3-」と入力し、絞り込みを行います。

リストから「s3-get-object-python3」をチェックします。

似た名前のものがあるので注意しましょう。

問題なければ右下の「設定」をクリックします。

Lambda > 関数 > 関数の作成 > 設計図 s3-get-object-python3 の使用

### 基本的な情報 情報

**名前\***  
lambda-test

**ロール\***  
関数のアクセス許可を定義します。新しいロールは、作成後の数分間は使用できないことがあります。Lambda 実行ロールの詳細については、[こちら](#)を参照してください。  
テンプレートから新しいロールを作成 ▼

**ロール名\***  
新しいロールの名前を入力します。  
lambda-test

**ポリシーテンプレート**  
1 つ以上のポリシーテンプレートを選択します。関数が作成される前にロールが生成されます。各ポリシーテンプレートがロールに追加するアクセス権限については、[こちら](#)を参照してください。  
▼

**S3 オブジェクトの読み取り専用アクセス権限** ✕



画像のような画面になるので、各項目の設定を行います。

「名前」・・・任意（ここでは「lambda-test」としました）

「ロール」・・・「テンプレートから新しいロールを作成」

「ロール名」・・・任意（ここでは「lambda-test」としました）

「ポリシーテンプレート」・・・空欄のまま

をそれぞれ入力および選択します。

**s3** 削除

**バケット**  
イベントソースとして機能する S3 バケットを選択してください。バケットは関数と同じリージョンに存在する必要があります。

lambda-test-20180201 ▼

**イベントタイプ**  
Lambda 関数のトリガーとなるイベントを選択します。オプションで、イベントのプレフィックスやサフィックスを設定できます。しかし、それぞれのバケットの個々のイベントには、同一のオブジェクトキーを照合するプレフィックスやサフィックスを含んだ複数の設定はできません。

PUT ▼

**プレフィックス**  
キーが同じ文字から始まるオブジェクトのみに通知を制限するため、省略可能なプレフィックスを入力してください。

例: images/

**サフィックス**  
キーが同じ文字で終わるオブジェクトのみに通知を制限するため、省略可能なサフィックスを入力してください。

例: jpg

Lambda は、このトリガーから Lambda 関数を呼び出すのに必要な Amazon S3 のアクセス許可を追加します。Lambda アクセス許可モデルの詳細については [こちら](#) を参照してください。

**トリガーの有効化**  
今すぐトリガーを有効化するか、テスト用に無効化した状態でトリガーを作成します (推奨)。

☒

下にスクロールするとさらに設定項目が現れます。

「バケット」・・・前項で作成したバケット

「イベントタイプ」・・・「PUT」

「プレフィックス」・・・空欄のまま

「サフィックス」・・・空欄のまま

「トリガーの有効化」・・・チェック

をそれぞれ入力および選択します。

### Lambda 関数のコード

コードは、選択された設計図によって事前に設定されます。関数の作成後にコードを設定できます。

ランタイム  
Python 3.6

```

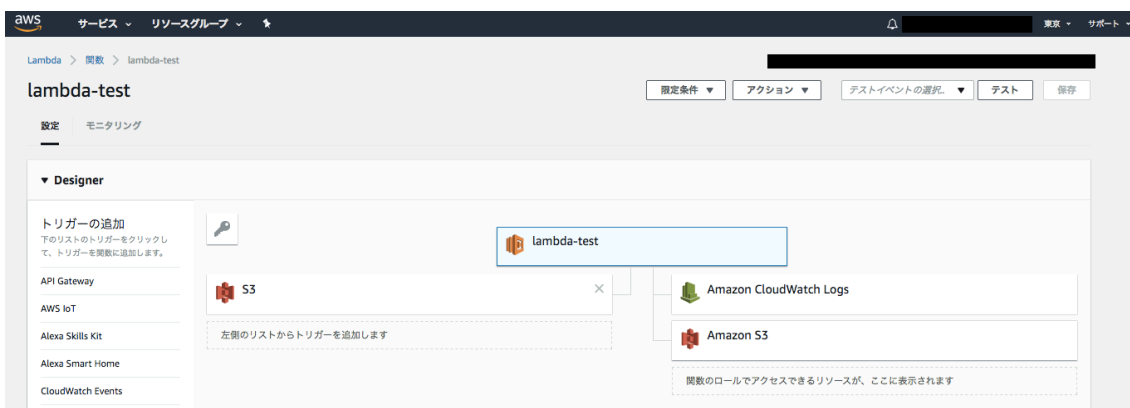
1
2 import json
3 import urllib.parse
4 import boto3
5
6 print('Loading function')
7
8 s3 = boto3.client('s3')
9
10
11 def lambda_handler(event, context):
12     #print("Received event: " + json.dumps(event, indent=2))
13
14     # Get the object from the event and show its content type
15     bucket = event['Records'][0]['s3']['bucket']['name']
16     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
17     try:
18         response = s3.get_object(Bucket=bucket, Key=key)
19         print("CONTENT TYPE: " + response['ContentType'])
20         return response['ContentType']
21     except Exception as e:
22         print(e)
23         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the s
24         raise e
25

```

\* これらのフィールドは必須です。

キャンセル 戻る 関数の作成

下までスクロールすると右下に「関数の作成」とあるので、クリックします。



完了すると画像のような画面になります。

左上の「サービス」をクリックし、再度 S3 にアクセスします。

## 実行

The screenshot shows the Amazon S3 console interface. At the top, there's a navigation bar with the AWS logo, service menu, resource group menu, and a search bar. Below the navigation bar, a message says "Amazon S3 へようこそ。新しいバケットを作成するか、既存のバケットを選択してプロパティを表示、設定します。" (Welcome to Amazon S3. Create a new bucket or select an existing bucket to view and configure its properties.) and a "ドキュメント" (Documentation) link.

The main content area shows the "Amazon S3" header with links for "新しいコンソールのご紹介" (Introducing the new console) and "ご存知でしたか?" (Did you know?). Below this is a search bar labeled "バケット検索" (Bucket search).

There are three buttons: "+ バケットを作成する" (Create bucket), "バケットを削除する" (Delete bucket), and "バケットを空にする" (Empty bucket). To the right, it shows "5 バケット 0 パブリック" (5 buckets, 0 public) and "3 リージョン" (3 regions).

バケット名 ↑	アクセス ↑	リージョン ↑	作成日 ↑
[redacted]	非公開 *	米国東部 (オハイオ)	10月 25, 2017 7:08:00 午後 GMT+0900
lambda-test-20180201	非公開 *	アジアパシフィック (東京)	2月 1, 2018 3:17:46 午後 GMT+0900
[redacted]	非公開 *	アジアパシフィック (東京)	12月 19, 2017 9:18:44 午後 GMT+0900
[redacted]	非公開 *	アジアパシフィック (東京)	10月 26, 2017 11:49:39 午後 GMT+0900
[redacted]	非公開 *	米国東部 (バージニア北部)	1月 16, 2018 5:42:40 午後 GMT+0900

\* オブジェクトは、オブジェクト ACL により、依然としてパブリックアクセス可能である場合があります。詳細

S3 にアクセスしたら、作成したバケットをクリックします。

The screenshot shows the details page for the bucket "lambda-test-20180201". The breadcrumb trail is "Amazon S3 > lambda-test-20180201". There are four tabs: "概要" (Overview), "プロパティ" (Properties), "アクセス権限" (Access permissions), and "管理" (Management). The "概要" tab is selected.

At the top, there are buttons: "アップロード" (Upload), "+ フォルダの作成" (Create folder), and "さらに ↓" (More). To the right, it says "アジアパシフィック (東京)" (Asia Pacific (Tokyo)).

The main content area has a light blue background with the text: "このバケットは空です。新しいオブジェクトをアップロードして開始します。" (This bucket is empty. Upload a new object to get started.).

Below the text are three icons with labels:

- Icon of a folder and document: "オブジェクトのアップロード" (Upload object)
- Icon of two people: "オブジェクトのプロパティの設定" (Configure object properties)
- Icon of a database cylinder: "オブジェクトへのアクセス許可の設定" (Configure access permissions for object)

まだ何もデータがないので画像のような画面になります。

左上の「アップロード」をクリックし、ファイルをアップロードしてみます。

アップロード

1 ファイルの選択

2 アクセス許可を設定する

3 プロパティを設定する

4 確認

1 ファイル   サイズ: 1.8 MB   ターゲットパス: lambda-test-20180201

+ さらにファイルを追加する

20170330-084508.pdf

- 1.8 MB

アップロード

次へ

PDF ファイルをアップロードしてみます。  
 ファイルをドラッグ&ドロップし、左下の「アップロード」をクリックします。

Amazon S3 > lambda-test-20180201

概要

プロパティ

アクセス権限

管理

Q

プレフィックスを入力し、Enter キーで検索します。ESC を押してクリアします。

アップロード

+ フォルダの作成

さらに ▾

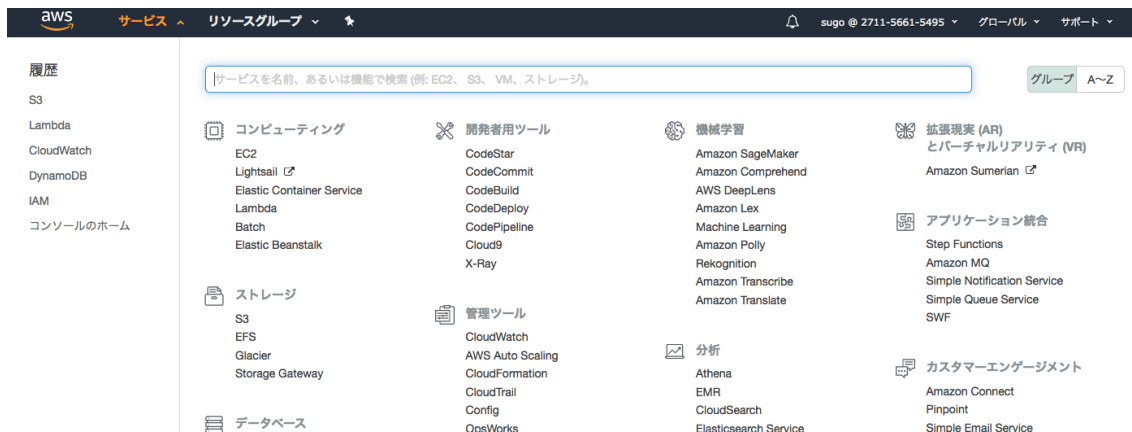
アジアパシフィック (東京)

表示中 1 ~ 1

<input type="checkbox"/>	名前	最終更新日時	サイズ	ストレージクラス
<input type="checkbox"/>	20170330-084508.pdf	2月 2, 2018 2:00:29 午前 GMT+0900	1.8 MB	スタンダード

表示中 1 ~ 1

完了すると画像のような画面になります。



実行した結果を確認したいので、左上の「サービス」をクリックします。  
「管理ツール」の「CloudWatch」をクリックします。

## ログ

画像のような画面になります。  
左のリストから「ログ」をクリックします。

CloudWatch > ロググループ

メトリクスフィルタの作成

アクション

フィルタ:

ロググループ名のプレフィックス

ロググループ	次の期間経過後にイベントを失効	メトリクスフィルタ	サブスクリプション
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/> /aws/lambda/lambda-test	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし
<input type="radio"/>	失効しない	0 フィルタ	なし

「/aws/lambda/」から始まるロググループを探します。  
その中から、「 / aws/ lambda/ [今回作成した Lambda 関数名] 」となっているものをクリックします。(今回 Lambda 関数名は「lambda-test」としましたので、「/aws/lambda/lambda-test」をクリックしました。)

CloudWatch > ロググループ > /aws/lambda/lambda-test のストリーム

イベントの検索		ログストリームの作成	ログストリームの削除
フィルタ: ログストリーム名のプレフィックス x			
<input type="checkbox"/> ログストリーム	直前のイベント時刻		
<input type="checkbox"/> 2018/02/01/[\$LATEST]2c95e64d1a0c4d7895c534db21fc82bd	2018-02-02 02:03 UTC+9		

クリックすると画像のような画面になります。  
S3 にファイルを一度アップロードしたので、そのログが記録されています。  
「直前のイベント時刻」を確認すると、S3 にファイルをアップロードした時間であることがわかります。  
S3 にファイルをいくつもアップロードした場合は、このログのリストがどんどん増えていきます。  
それでは、この「ログストリーム」をクリックします。

CloudWatch > ロググループ > /aws/lambda/lambda-test > 2018/02/01/[\$LATEST]2c95e64d1a0c4d7895c534db21fc82bd

イベントのフィルター		すべて 30秒
時間 (UTC +00:00)	メッセージ	
2018-02-01	いまのところ古いイベントはありません。再試行してください。	
▶ 17:03:14	START RequestId: ca4e5431-0771-11e8-975e-93c2f532e281 Version: \$LATEST	
▶ 17:03:14	CONTENT TYPE: application/pdf	
▶ 17:03:14	END RequestId: ca4e5431-0771-11e8-975e-93c2f532e281	
▶ 17:03:14	REPORT RequestId: ca4e5431-0771-11e8-975e-93c2f532e281 Duration: 644.51 ms Billed Duration: 700 ms Memory Size: 128 MB Max Memory Used: 42 MB	
	いまのところ新しいイベントはありません。再試行してください。	

クリックすると画像のような画面になります。

ログがいくつかキロ記されていることが確認できます。

中央あたりに「**CONTENT TYPE:** ~~~」というログがあるので見てみましょう。

S3にアップロードされたファイルがなんのファイルなのかが記録されています。

今回、PDF ファイルをアップロードし、ログには「**CONTENT TYPE:** application/pdf」と記録されており、ファイルの識別ができています。

画像ファイル（example.jpg など）をアップロードした場合には「**CONTENT TYPE:** image/jpeg」といったログが記録されます。