LambdaをつかってJawsDays
をHackしてみよう!

2016/3/12
JAWS DAYS 2016
Toshihiro Furuno

資料及びソースコードは以下にあります

https://github.com/toshihirock/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)

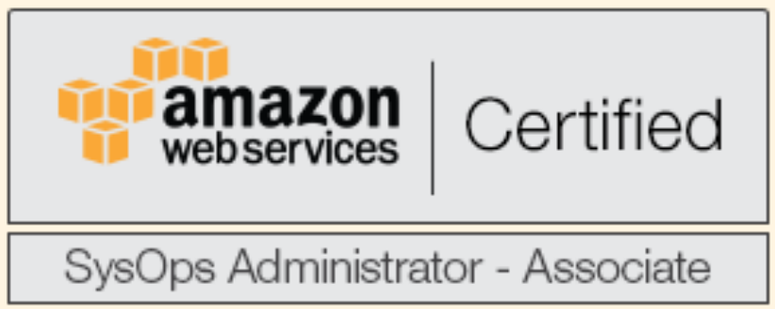
やらないこと

・Lambdaの詳細な説明

・TwitterAPIの詳細

自己紹介

- 古野俊広(@toshihirock)
- 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をつかってJawsDaysをHackすっぞ

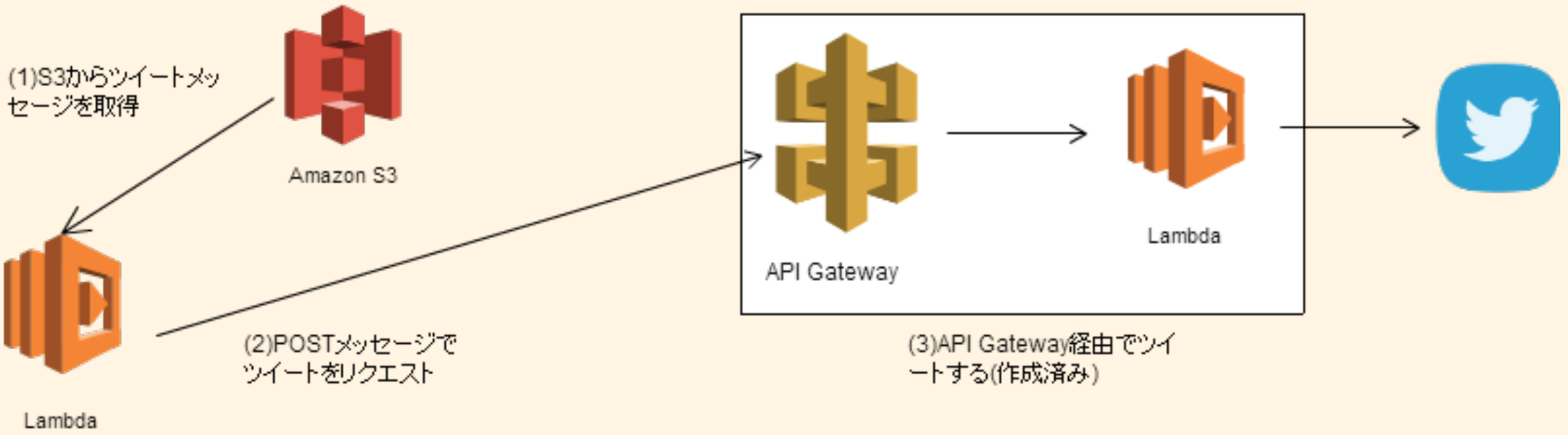
・ツイートさせたい内容のテキストをS3に配置

・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


(1)S3からツイートメッセージを取得
Amazon S3
Lambda
(2)POSTメッセージでツイートをリクエスト
API Gateway
Lambda
(3)API Gateway経由でツイートする(作成済み)

知識編

S3?

・オンラインストレージ

・高い堅牢性(99.999999999%)

・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

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

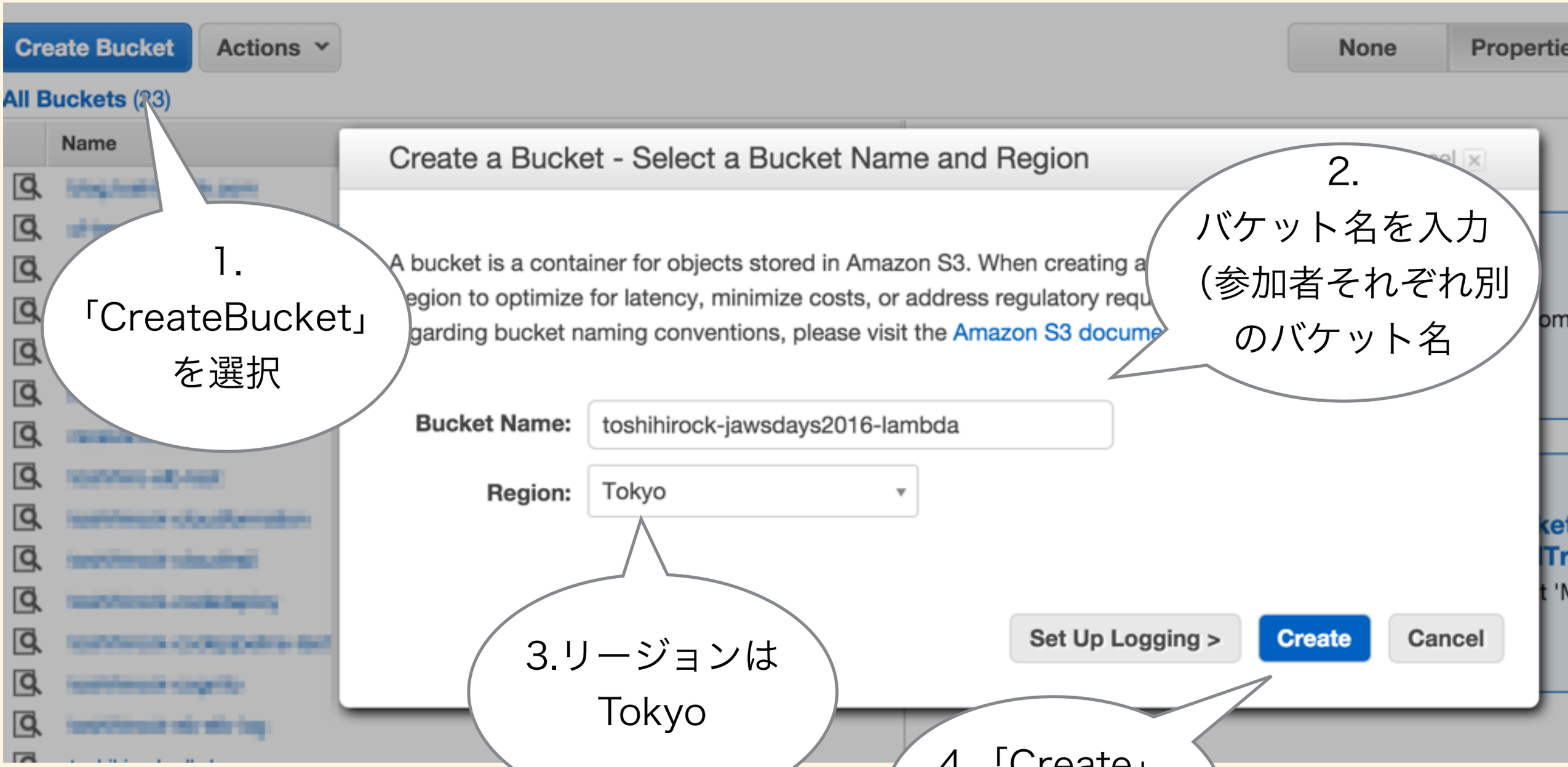
SNS
Push Notification Service

Application Services

API Gateway
Build, Deploy and Manage APIs

AppStream

バケットの作成

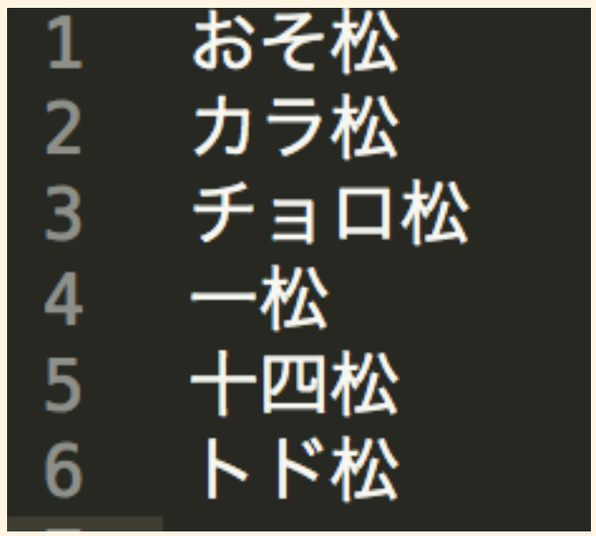

Create Bucket
Actions
None
Properties
All Buckets (23)
Name
Create a Bucket - Select a Bucket Name and Region
A bucket is a container for objects stored in Amazon S3. When creating a
egion to optimize for latency, minimize costs, or address regulatory requ
garding bucket naming conventions, please visit the Amazon S3 docume
Bucket Name:
toshihirock-jawsdays2016-lambda
Region:
Tokyo
Set Up Logging >
Create
Cancel
1.
「CreateBucket」
を選択
2.
バケット名を入力
（参加者それぞれ別
のバケット名
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)

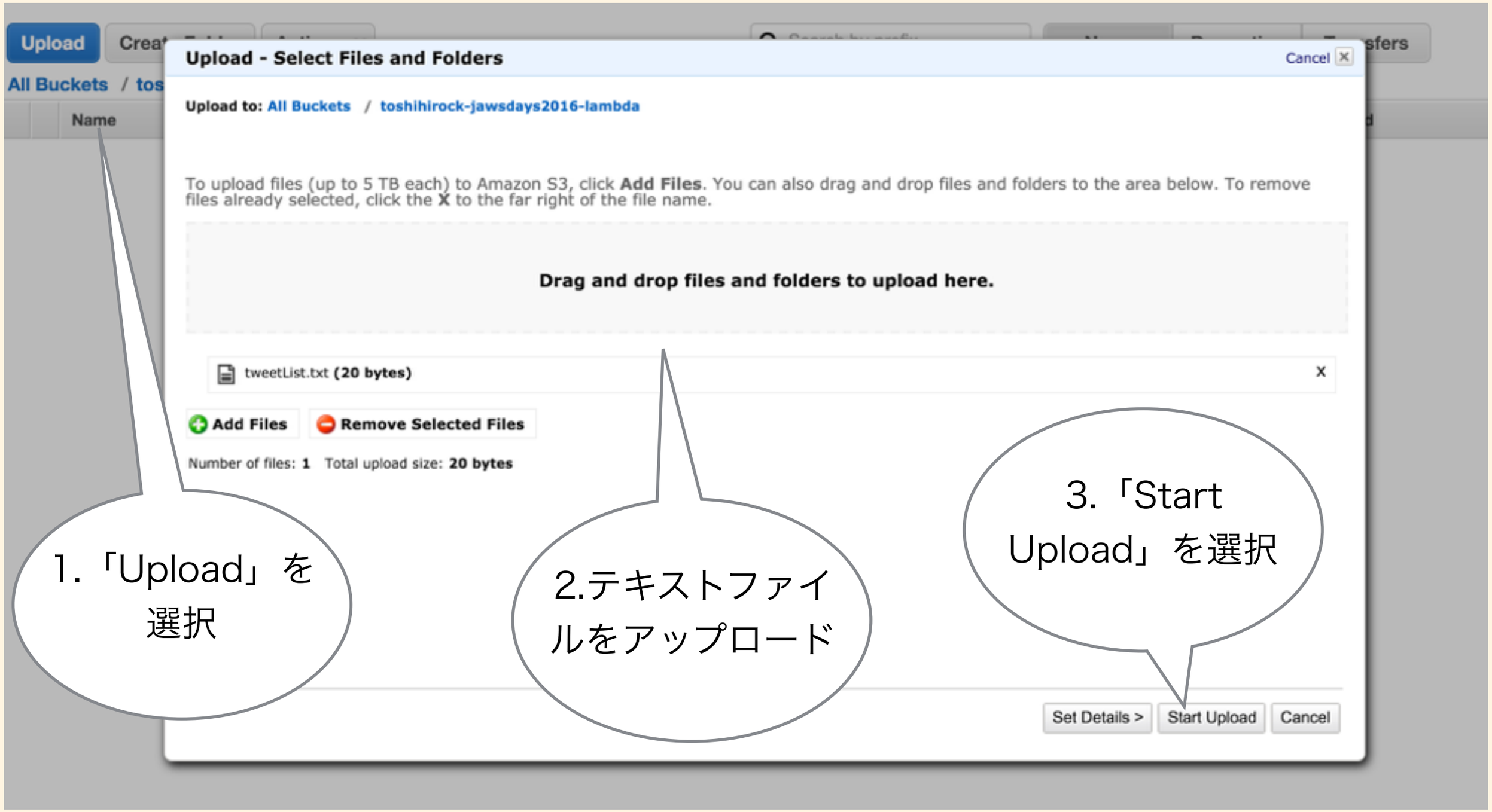
・作成後、「tweetList.txt」でローカルに保存



確認

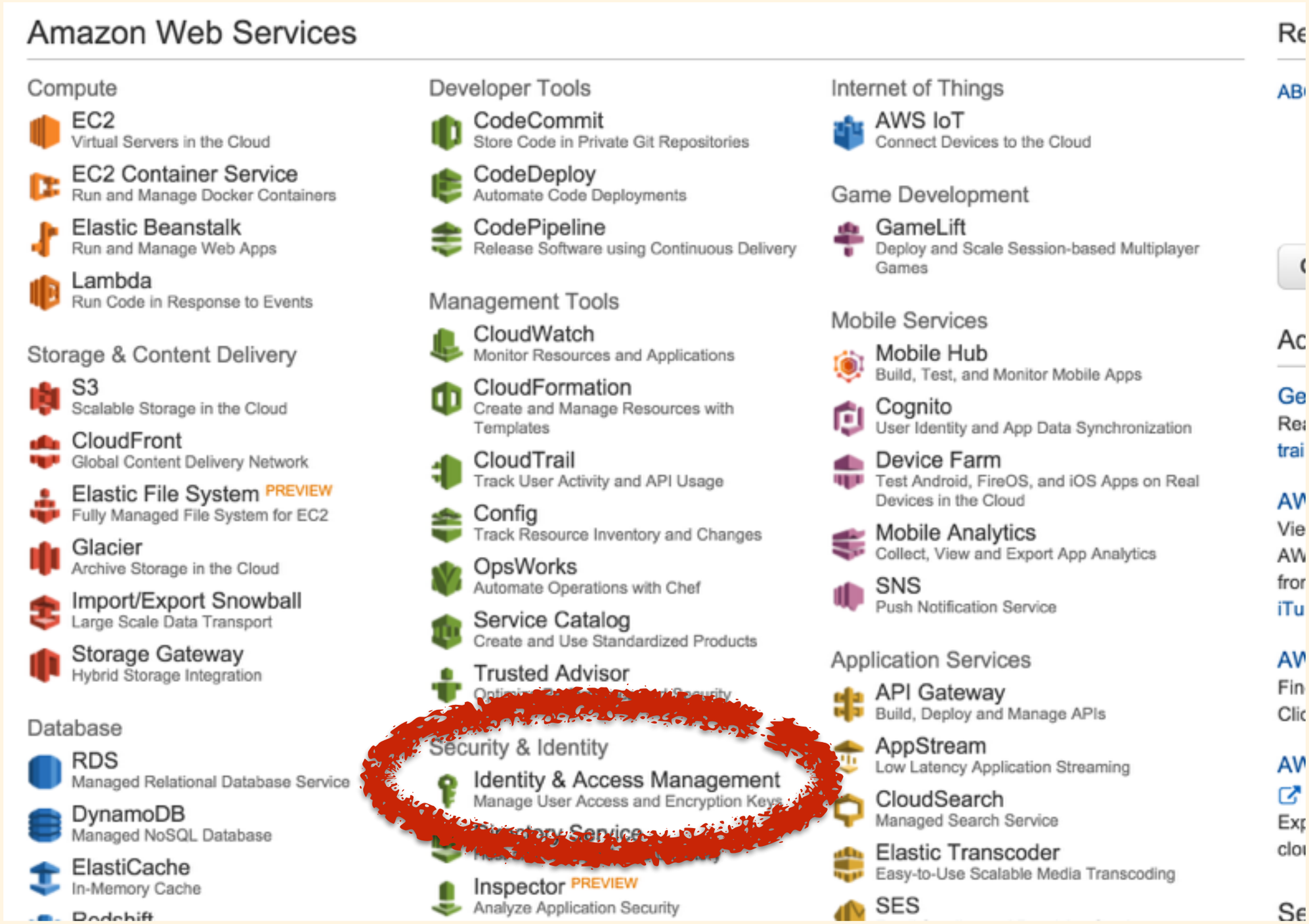
```
# 文字コード確認

$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



準備編(IAMロール)

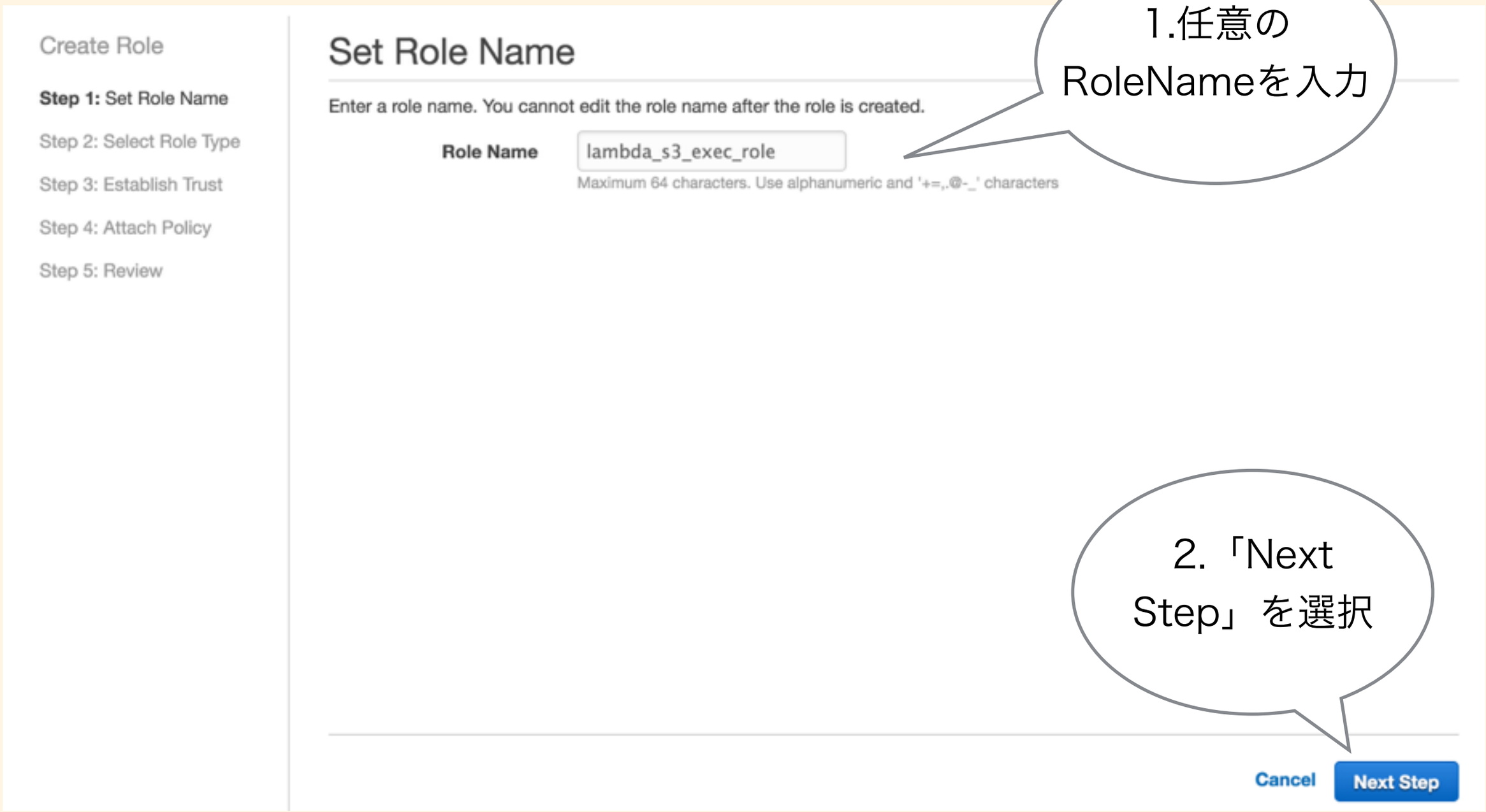
サービス一覧からIAMの選択



ロールの新規作成


Dashboard
Search IAM
Details
Groups
Users
Roles
Policie
Identity viders
Account ngs
Cr
Create New Role
Role Actions
Showing 36 results
Name
Creation Time
ecRole
2015-07-11 07:41 UTC+0900
2015-07-21 15:52 UTC+0900
2015-08-04 22:02 UTC+0900
ice-role
2015-09-18 20:56 UTC+0900
aws-opsworks-ec2-role
2015-09-13 20:30 UTC+0900
aws-opsworks-service-role
2015-09-13 20:30 UTC+0900
CloudTrail_CloudWatchLogs_Role
2015-07-01 03:15 UTC+0900
2. 「Create New Role」を選択
1. 「Roles」を選択

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
lambda_s3_exec_role
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

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.

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

AWS Lambda
Allows Lambda Function to call AWS services on your behalf.

Amazon API Gateway
Allows API Gateway to call AWS resources on your behalf.

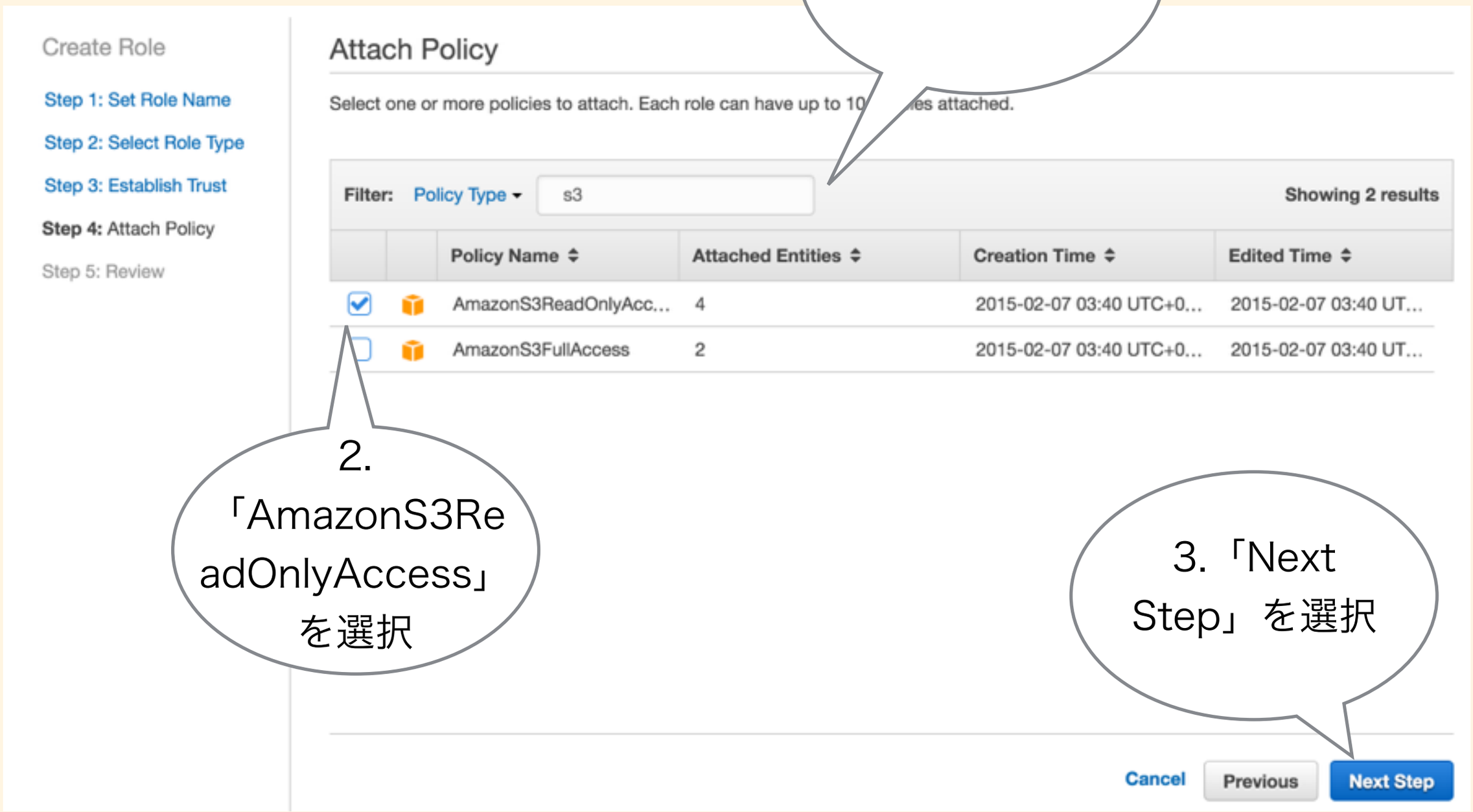
AWS Config
Allows AWS Config to call AWS services and collect resource configurations on your behalf.

「AWS Lambda」を選択

Role for Cross-Account Access

Role for Identity Provider Access

ポリシーの選択


1.「s3」と入力
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 es attached.
Filter:
Policy Type
s3
Showing 2 results
Policy Name
Attached Entities
Creation Time
Edited Time
AmazonS3ReadOnlyAcc...
4
2015-02-07 03:40 UTC+0...
2015-02-07 03:40 UT...
AmazonS3FullAccess
2
2015-02-07 03:40 UTC+0...
2015-02-07 03:40 UT...
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	lamba_test	Edit Role Name
Role ARN	[redacted]	
Trusted Entities	The identity provider(s) lambda.amazonaws.com	
Policies	arn:aws:iam::aws:policy/AmazonS3ReadOnly Access	Change Policies

確認後、
「Create Role」
を選択

Cancel　Previous　Create Role

Lambdaを作ろう!

(後で利用)コードの準備

・以下の内容をコピペして任意のテキストエディタに貼っておく

・https://github.com/toshihirock/jaws2016-lambda-hands-on/blob/master/index.js

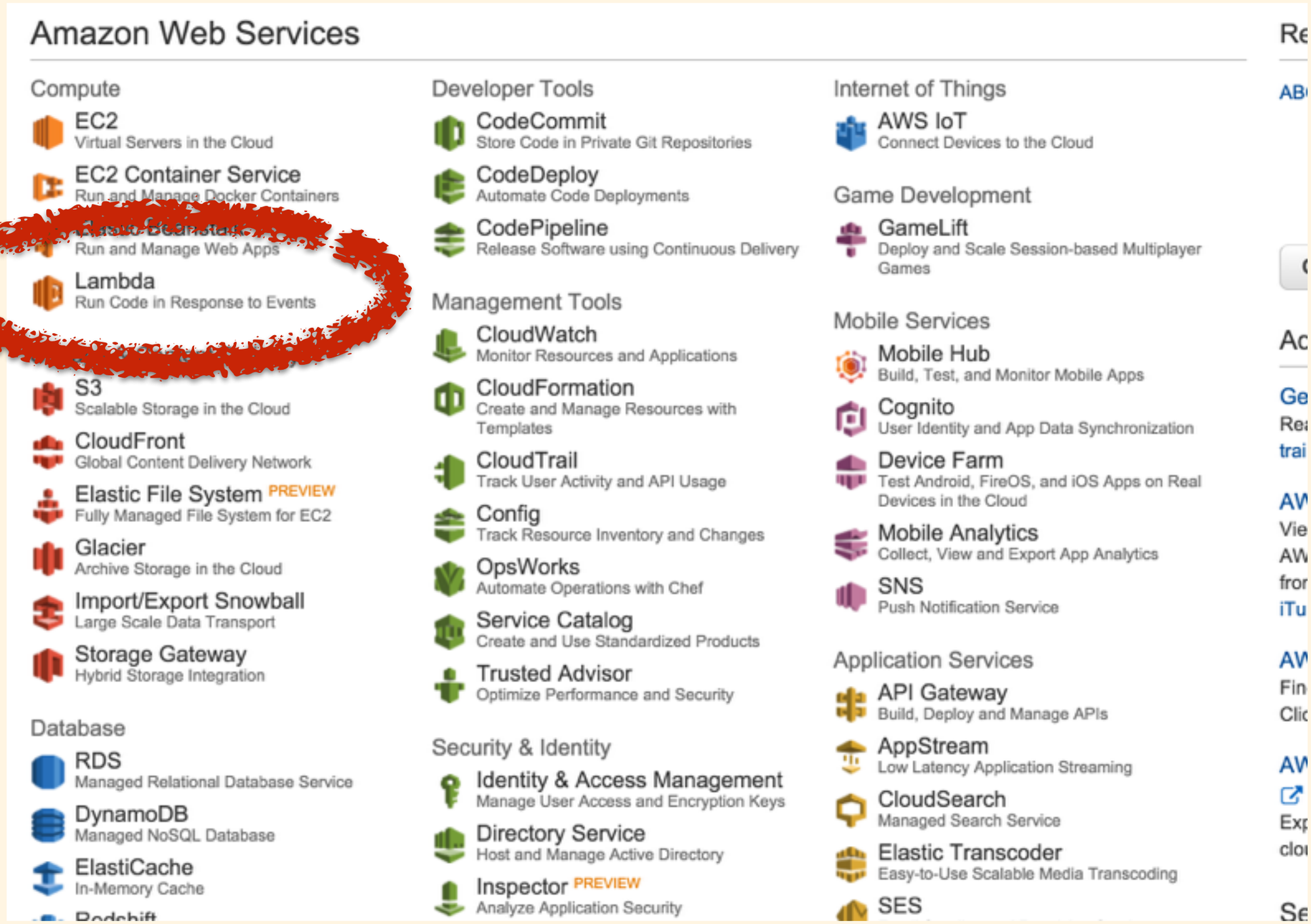
(後で利用)コードの準備

```
1  console.log('Loading function');
2
3  // API Gatewayの設定
4  const HOST = 'wjz7li7gmj.execute-a   .ap-northeast-1.amazonaws.com';
5  const PATH = '/prod/statuses/upd   te';
6  const X_API_KEY = '';
7
8  // S3の設定
9  const BUCKET = 'toshihirock-jaws2016lambda';
10 const KEY = 'tweetList.txt';
11 const TEXT_DELIMITER = '\r\n';
12
```

APIGatewayの
認証キーを指定

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



Functionの作成

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-

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

splun

Demor
from AW
Event Collector.

nodejs · splunk · kinesis

利用しない
ので「Skip」を選択

Cancel Skip

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 nee
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, cont
4      //console.log('Received event:',        null, 2));
```

Edit code inline

先ほどの内容をコピペ

Functionの設定(2/2)

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

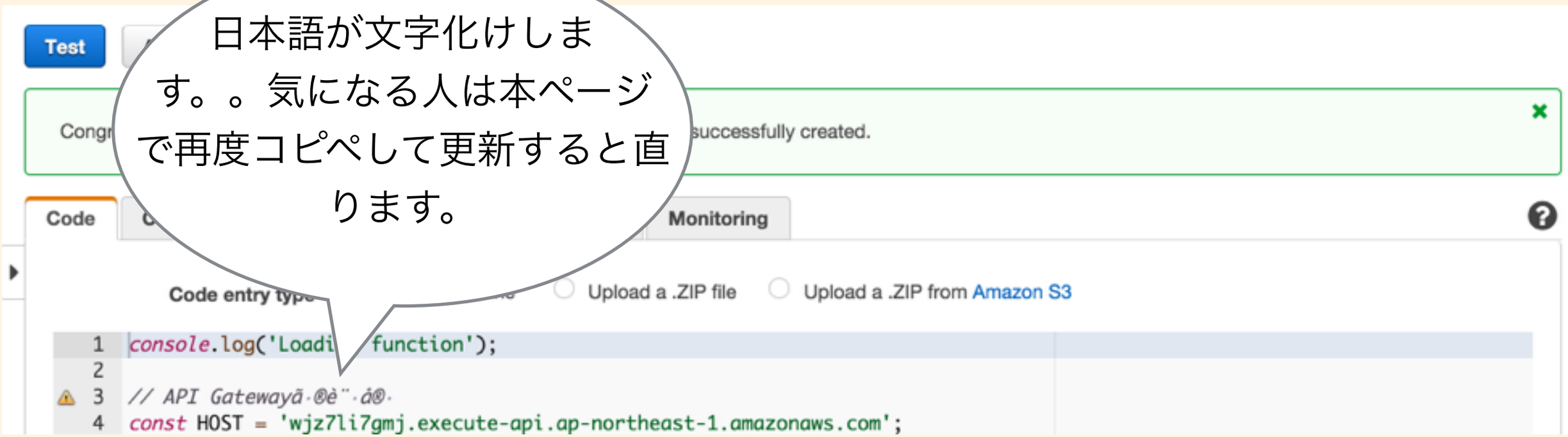
Edit

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「Crete function」

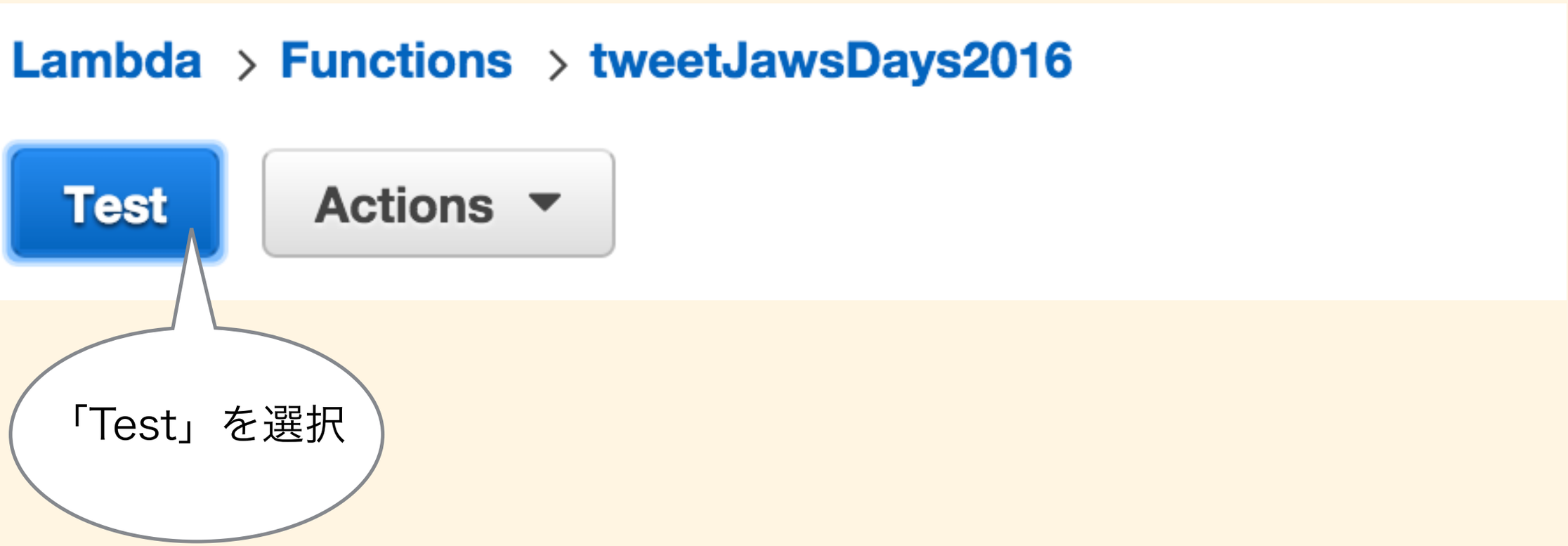
Cancel Previous **Create function**

作成

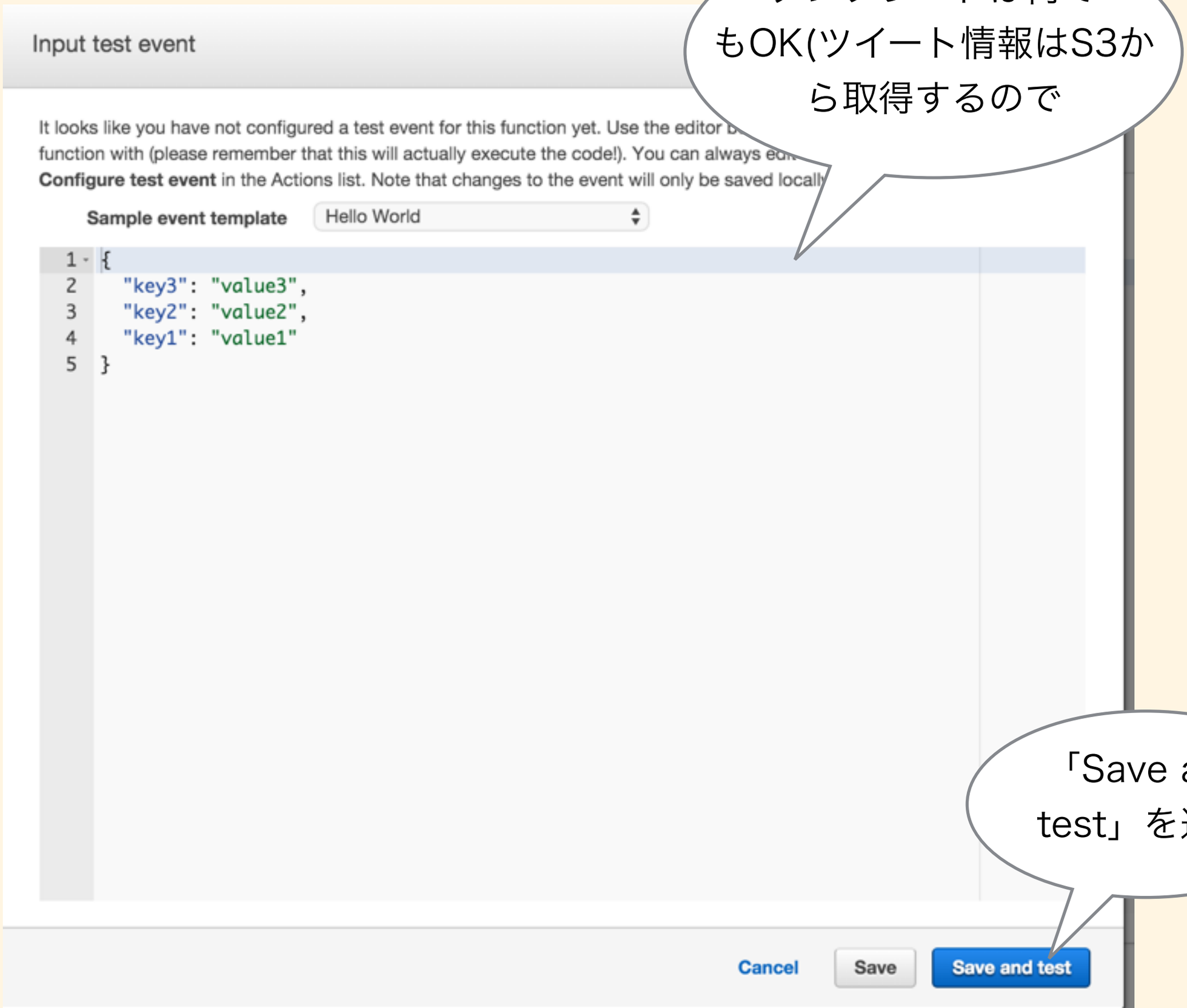


日本語が文字化けします。。気になる人は本ページで再度コピペして更新すると直ります。

テストしてみる


Lambda > Functions > tweetJawsDays2016
Test
Actions
「Test」を選択

テストしてみる

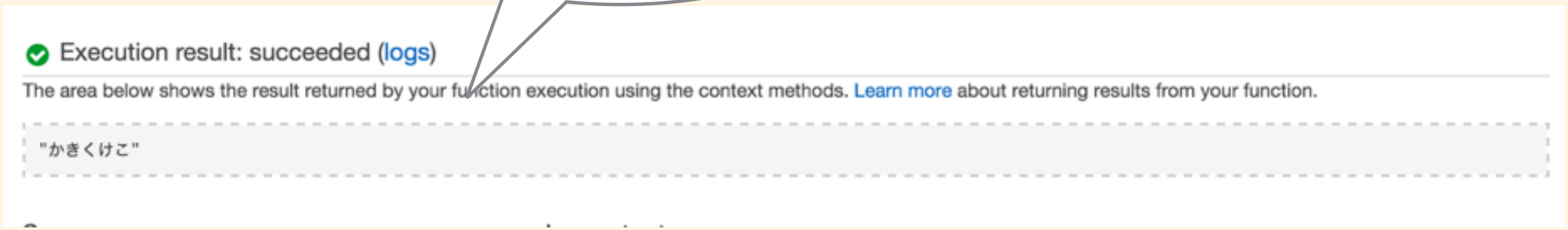


テンプレートは何でもOK(ツイート情報はS3から取得するので

「Save and test」を選択

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



テストしてみる

Twitterでも確認

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

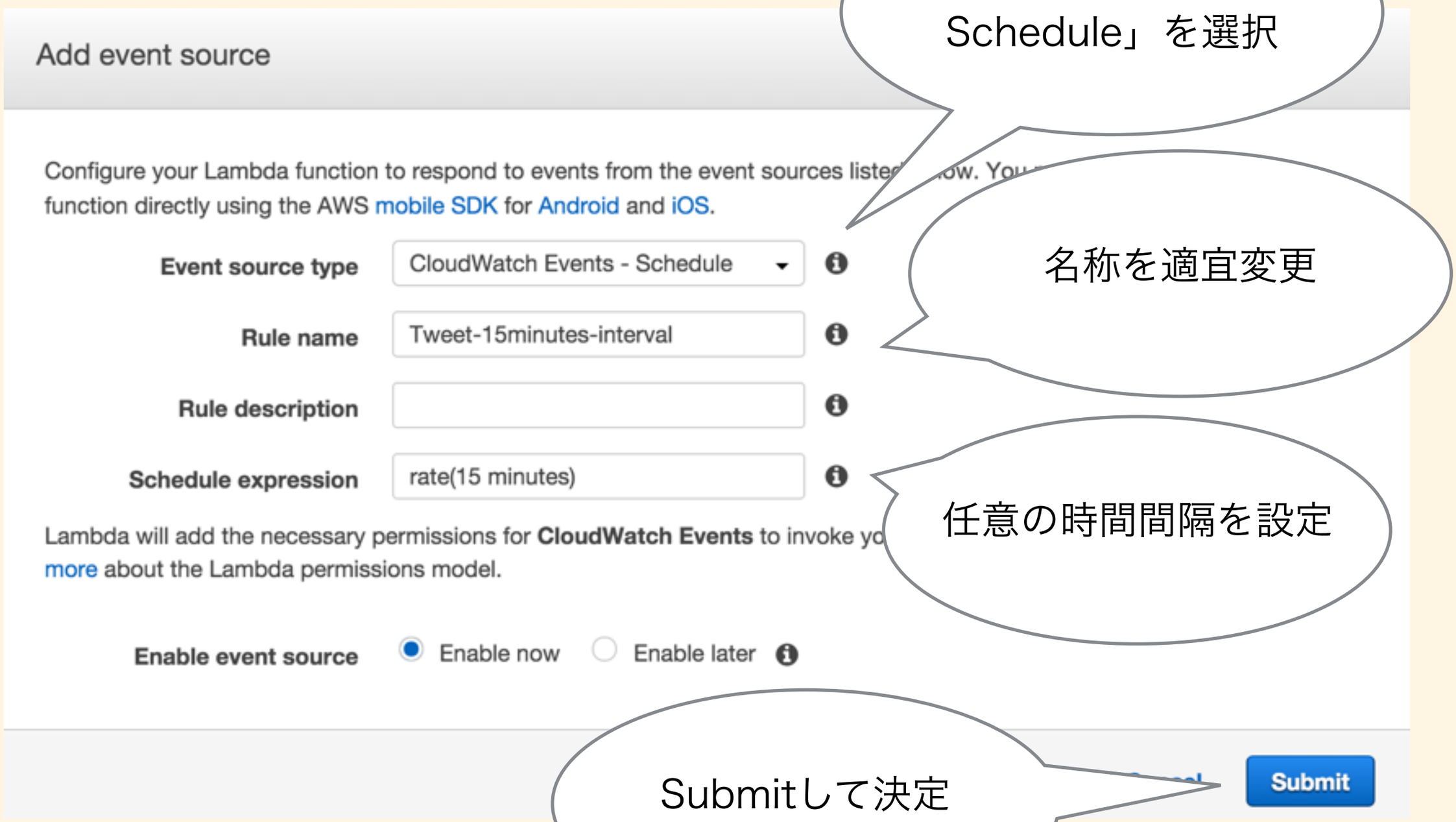
botらしく

定期実行させてみる

定期実行の設定


「Event sources」タブを選択
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 liste ow. You
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 yo
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
Submitして決定

確認

Twitter上でも確認

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/toshihirock/items/d28505442526e0ae7793

Thanks!