

2018年10月4日

株式会社インプレスR&D

<https://nextpublishing.jp/>

AWS でのサーバーレスサービス構築のノウハウ集！
『Amazon Web Services サーバーレスレシピ』発行
技術書典シリーズ・10月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレスR&Dは、『Amazon Web Services サーバーレスレシピ』(著者:矢田 裕基、太田 佳敬、佐々木 美穂、森岡 周平)を発行いたします。

『Amazon Web Servicesサーバーレスレシピ』
<https://nextpublishing.jp/isbn/9784844398448>



著者: 矢田 裕基, 太田 佳敬, 佐々木 美穂, 森岡 周平
小売希望価格: 電子書籍版 1400 円(税別) / 印刷書籍版 1600 円(税別)
電子書籍版フォーマット: EPUB3 / Kindle Format8
印刷書籍版仕様: B5 判 / カラー / 本文 82 ページ
ISBN: 978-4-8443-9844-8
発行: インプレス R&D

<< 発行主旨・内容紹介 >>

【AWS でサーバーレスサービスを設計・構築してみる！】

本書は、Amazon Web Service (AWS) を使ってサーバーレスな各種サービスを構築するための設計事例集です。各章では EC サイト、IoT、動画サービスなど様々なサービスを設計するためのノウハウを掲載しています。

〈本書の対象読者〉

AWS でサーバーレスな構成がしたい人

AWS でのサーバーレス設計のノウハウを広く知りたい人

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

EC サイトの構築や動画サービスなど、AWS でのサーバーレスサービスの構築ノウハウを掲載

第2章 Elasticsearchを利用したECサイトの構築 (太田 佳敬)

この章ではDynamoDBやElasticsearchを利用したサーバーレスのシステムを考えます。具体的には登録した商品を見たり、検索機能があったECサイトにします。具体的な機能としては次のようになります。

- ・商品が登録できる
- ・商品の詳細が見られる
- ・商品を検索できる

実際に運用するECサイトとしては次のような機能も必要ですが、今回は画面の都合上省略します。

- ・商品の一覧表示
- ・ユーザー管理
- ・決済サービス

機能は全てJSONを送るAPIで構成します。ウェブページはSPAとして作り、ユーザーの操作に応じてリクエストを送り、JSONで受け取った結果を元に表示を書き換えます。

2.1 全体構成

今回は全てAWS Serverless Application Model (以降、SAM) を使って作ります。図2.1の構成になりますが、前述のとおり薄く色を塗った部分のみを今回は解説します。

図2.1: 全体構成

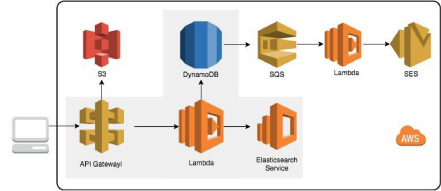


図2.1を簡単に説明すると、まずユーザーのアクセスをAmazon API Gatewayが受け、AWS Lambdaを起動させます。AWS Lambdaはアクセスに応じてDynamoDBから商品データを取得してレスポンスを送ったり、Amazon Elasticsearch Service (以降、ES) に関わらせてデータを検索したりします。これらについてSAMを利用して作っていきます。

開発に関してはaws cliコマンドと、SAMを手元のマシンで実行出来るaws-sam-cliのlocal機能を使って進めていきます。awsコマンドとsamコマンドが利用できるように公式サイトに沿ってインストール、および設定をお願いします。

なお、DynamoDBとElasticsearch ServiceはSAMに対応していません。DynamoDBはローカルで動かせるDynamoDB localというものがあるため、これを利用して開発をします。ESをローカルで動かせるソフトウェアはありませんが、通常のElasticsearchで代用できますのでそれを利用します。

DynamoDBを使う

今回、データベースはDynamoDBを利用します。AWS Lambdaを利用する場合、ほぼ間違いなくDynamoDBを使うこととなります。

今回の構成ではAWS Lambdaがユーザーからのアクセスに対して自動でスケールアップしていきます。Lambdaは無限に並列実行が可能のため(実際の動作では数千で上限値になりますが)、データベースへ非常に多数のクライアントが同時に接続します。DynamoDBは大量の接続に対しても適切なパフォーマンスが出るような設計になっており、Lambdaを使う場合は間違いなくこれを利用した方が良いでしょう。

もしその他のデータベースを扱いたいのであれば、データを取得する部分を適切に書き換えてください。

Elasticsearch Serviceを使う

まず、商品検索を行うのでどのように検索するかを考える必要があります。これは今回の話のメインでもある、Elasticsearchを利用します。

AWSでは、ElasticsearchをホスティングしているElasticsearch Serviceを提供しています。そのため、データを登録してクエリを投げただけで検索が出来るようになっています。ただし、ユーザー辞書が使えない、プラグインが使えない、といったいくつかの制限事項があります。

2.2 データ設計

DynamoDBもElasticsearchもスキーマレスのため何でもデータに入れますが、実際にコードから使う際にはデータ構造を意識するために、事前に想定しておきます。

AWS IoT の構築方法も紹介

するデータなどのメッセージの送受信をコントロールしており、主にWebsocketとMQTTと呼ばれる軽量なプロトコルが使用されています。またHTTP REST interfaceも送信するときに使用することができます。

Rules Engine

Message Brokerから送られてきた時に、どのようなアクションを行うかを設定することができます。アクションとは、AWSの他の機能へのアクションであり、例えばDynamoDBデータベースにデータを書き込んだり、Lambda関数を呼び出すことが可能です。また送られてきたデータがJSON形式の場合、データに対してSQLのようなクエリを書くことができ、データを抽出することで、特定のデータが来た時のみアクションを起こす、といったことが可能になります。例えばこのようなデータが送られてきたとします。

リスト3.1: CLI(Command Line Interface)

```
{
  "e": [
    { "n": "temperature", "u": "Cel", "t": 1234, "v": 22.5 },
    { "n": "light", "u": "lm", "t": 1235, "v": 135 },
    { "n": "acidity", "u": "ph", "t": 1235, "v": 7 }
  ]
}
```

ここで次のクエリを設定します。

リスト3.2: CLI

```
SELECT
(SELECT v FROM e WHERE n = 'temperature') as temperature
FROM
'my/topic'
```

ここではまずeというkeyの中の、nというKeyの値がtemperatureであるデータを取り出します。そしてそのデータのvのKeyの値を取り出し、temperatureというKeyの値としてデータを再編します。この結果として出力されるのが次のデータになります。

リスト3.3: データ

```
{"temperature":22.5}
```

この結果がデータとして次のアクションに渡されます。またこのようなクエリを設定し、指定されたKeyがなかった場合はアクションは行われません。

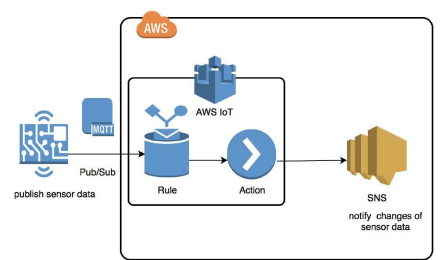
Security and Identity

デバイスがAWS IoTとやり取りする時には、証明書を使用したセキュアな通信をしなければなりません。Rules Engine・Message Brokerなどが他のAWS IoTの機能と連携する際にも証明書などの認証情報が必要になります。

3.3 基本構成

今回は部屋に気温センサーを置き、気温センサーが20℃を超えた場合に通知を行う想定で構成し、AWS IoTについて説明します。※今回はAWS IoTの仕組みについてがメインなのでraspberrypiの設定・気温の取得方法に関しては触れません。通知が可能になるという部分だけコマンドラインで確認します。

図3.2: 構成図



3.4 手順

この構成をもとにAWS IoTを使って機能をつくっていきましょう。

デバイスの登録

まずデバイスの登録を行います。

1. AWS IoT consoleを開き、「新規作成」をクリックします。

<<目次>>

第1章 AWS SAMとTypeScriptで作るアプリケーション開発(森岡 周平)

- 1.1 はじめに
- 1.2 AWS SAMとは
- 1.3 SAM CLIとは
- 1.4 TypeScriptとSAM Localを使用したアプリケーション開発

第2章 Elasticsearchを利用したECサイトの構築(太田 佳敬)

- 2.1 全体構成
- 2.2 データ設計
- 2.3 API設計
- 2.4 トップページを作る
- 2.5 商品を登録する
- 2.6 商品詳細を取得する
- 2.7 Elasticsearchを使う
- 2.8 Elasticsearchのインデックスを作る
- 2.9 Elasticsearchにデータを登録する
- 2.10 Elasticsearchからデータを検索する
- 2.11 AWSにデプロイする
- 2.12 ウェブページからアクセスする
- 2.13 まとめ

第3章 AWS IoT(佐々木 美穂)

- 3.1 AWS IoTとは?
- 3.2 AWS IoT Component
- 3.3 基本構成
- 3.4 手順
- 3.5 ポリシーの設定とは?
- 3.6 MQTTトピック・トピックフィルター
- 3.7 AWS IoT メッセージブローカー
- 3.8 sdkを利用した実行方法
- 3.9 まとめ

第4章 AWS Media Servicesで構築するサーバーレスな動画サイト(矢田 裕基)

- 4.1 AWS Media Servicesの登場
- 4.2 さっそく使ってみる
- 4.3 終わりに

第5章 AWS Media Servicesによるサーバーレスアーキテクチャーの歩き方(矢田 裕基)

- 5.1 AWSでサーバーレスアプリを作るにあたって
- 5.2 基本的なサービス
- 5.3 ユースケースから考える

<<著者紹介>>

矢田 裕基(やた ひろき)

AWS Lambdaに触れてから、如何にして安くAPI構築をするかを考え始めてサーバーレスという考え方に手を染めた。でも、現在はAndroidアプリを作っているエンジニア。最近、入門系の記事ばかり書いている人になりつつある。とある界限では「低温料理したお肉を提供する人」として認知されている。

太田 佳敬(おおた よしあき)

メインの仕事はマイクロサービスの Web アプリを開発する仕事だが、スマホのゲームから機械学習までとりあえずなんでもやる人。一部のマイクロサービスをサーバーレスにすることでコストを安くスケールもしやすくなるのではと思いい、サーバーレスに手を出し始める。

@ota42y <https://ota42y.com>

佐々木 美穂(ささき みほ)

大学院でIoTの研究をしつつインターンで Android アプリを作っている人。猫とことりんをこよなく愛しています。

@sasamihoo

森岡 周平(もりおか しゅうへい)

会社では主に Web アプリケーションを Rails で開発しています。最近では IDaaS の導入なども進めており、IDaaS のアイデンティティプロビジョニングをサーバーレスできないか考え諸々模索中。

<<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イブックスストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D (本社：東京都千代田区、代表取締役社長：井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らが、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社：東京都千代田区、代表取締役：唐島夏生、証券コード：東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp