報道各位

2018 年 12 月 26 日
株式会社インプレスR&D
https://nextpublishing.jp/

ユースケース集を通じて AWS Lambda を使いこなす！

『Go と SAM で学ぶ AWS Lambda』発行

技術書典シリーズ、12 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、『GoとSAMで学ぶAWS Lambda』(著者:杉田 寿憲)を発行いたします。

『GoとSAMで学ぶAWS Lambda』
https://nextpublishing.jp/isbn/9784844398813

著者:杉田 寿憲
小売希望価格:電子書籍版 1600 円(税別)／印刷書籍版 1800 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 88 ページ
ISBN:978-4-8443-9881-3
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

【ユースケース集を通じて AWS Lambda を学ぼう！】

本書は Go 言語での実装を通して、入門から中級程度の AWS Lambda の扱い方や、サーバーレスアーキテクチャの構成に不可欠な PaaS(API Gateway、S3、Dynamo DB など)やツール(SAM、direnv)の扱い方を学ぶ、AWS Lambda のユースケース集です。

(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。)

ユースケースを取り上げながら、Lambda の取り扱い方を学習

第2章　S3イベントの活用

2章からは、1章で紹介したツールを活用して開発したプロジェクトを見ながら、具体的にソースコードの意味やツールの活用方法を紹介します。実際に手を動かしながらコードを追ってみてください。

本章では、**「S3バケットにZipファイルがアップロードされたことをトリガーにして、LambdaでZipファイルを解凍し、別のS3バケットにアップロードする」**というユースケースを取り上げます。

そのために、次の項目を見ていきます。

・S3の概要とLambdaとの関係
・S3とLambdaを含むプロジェクトのSAMのテンプレート記述方法
・goenvの設定方法
・direnvの設定方法
・depの利用方法
・Lambdaのメインロジックのテスト方法
・Lambdaのハンドラーの類型
・SAMを利用したプロジェクトのデプロイ方法
・SAMを利用したプロジェクトの削除方法

2.1　S3

Amazon S3（Simple Cloud Storage Service）[1]はAWSが提供するオブジェクトストレージサービスです。「バケット」というリソースにオブジェクトとしてデータを保存します。

S3バケットは、S3のAPIに応じて次のイベントを発行します。

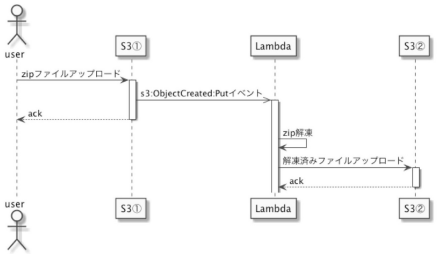
・s3:ObjectCreated:*
・s3:ObjectCreated:Put
・s3:ObjectCreated:Post
・s3:ObjectCreated:Copy
・s3:ObjectCreated:CompleteMultipartUpload
・s3:ObjectRemoved:*
・s3:ObjectRemoved:Delete
・s3:ObjectRemoved:DeleteMarkerCreated
・s3:ReducedRedundancyLostObject

このユースケースで利用する`s3:ObjectCreated:Put`以外のイベントをトリガーに、Lambdaを起動することもできます。さらに、イベントの通知先としてLambda以外にもSQSとSNSがサポート[2]されています。

イベントソースとなるサービスとLambdaは同一リージョンにデプロイされている必要があります。

2.2　シーケンス

図2.1: シーケンス図



1.https://aws.amazon.com/jp/s3/

2.https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/NotificationHowTo.html

S3 バケットにファイルがアップロードされたら Slack への投稿する、という具体的な例をもとに解説

第3章　SNSとSQSによるファンアウト

3.1　概要

本章では、ファンアウトパターンを取り上げます。具体的には、S3バケットにファイルがアップロードされたことをトリガーにしてSNSでSQSとLambdaに通知し、さらにSQSはLambdaに通知して処理を実行し、Lambdaはまた別の処理を並列に実行するというものです。

ファンアウトパターン[1]はクラウドデザインパターンのひとつです。AWSのクラウドデザインパターンの解説ページでは次のように説明されています。

> 処理を呼び出すプロセスから直接各処理を呼び出すのではなく、間にノーティフィケーション（通知）コンポーネントとキューイングコンポーネントを入れることで、非同期かつ並列に処理が可能になる。処理を呼び出すプロセスは、通知先への通知の後に、処理を続行できる。また、通知先の処理については知っている必要がないため、処理を増やしたい場合も通知コンポーネントへの通知先登録を増せば対応できる。また、キューごとに処理を割り当てて動作させれば、並列に処理を実行できる。

本章では通知コンポーネントとしてSNSを、キューイングコンポーネントとしてSQSを採用しています。Lambdaでは、アップロードしたファイルの拡張子とファイル名をそれぞれログに出力するという簡易な処理を行っていますが、実際にはサムネイルの生成と画像解析とタグ付けなど、責務の異なる処理を並行処理するのに適したデザインパターンです。

Eメールなど、Slack以外の通知手段を増やしたいときには、他のコンポーネントに影響なくSNSのサブスクリプションを追加することで実現できます。

本章では次の項目を説明します。

・SQSの概要とLambdaとの関係
・SNSの概要とLambdaとの関係
・S3、SQS、SNS、Lambdaを含むプロジェクトのSAMのテンプレート記述方法
・クラウドデザインパターンのひとつであるファンアウトパターンの実装
・Slackの Incoming Webookの利用方法
・イベントデータ・イベント発行をJSONファイルを使って擬似的にテストする方法
・CloudFormationのトラブルシューティング方法

3.2　SQS

SQS（Amazon Simple Queue Service）[2]はAWSが提供するメッセージキューイングサービスです。コンポーネント同士を分離し、スケールアウトをサポートします。

Lambdaは、2018年6月にSQSをイベントソースとして利用できるようになりました[3]。それまではサブスクライバー（SQSメッセージの受信個）がメッセージをポーリングし、処理済みのメッセージを明示的に削除する必要がありました。しかし、Lambdaのサポートにより SQSのエンキューをトリガーにLambdaを起動し、Lambdaが正常に終了すればメッセージは自動的に削除してくれるようになりました。

ただし、トリガーとして対応しているキューは標準キューのみでFIFOキューは従来どおりのポーリングが必要です。FIFOキューは標準キューの改良版で、順序と1回のみの配信（最低1回ではない）が保証されていますが、スループットは標準キューに劣ります。ユースケースによって使い分けてください[4]。

また、明示的にポーリングしないとはいえ、内部的にはLambdaがポーリング（ロングポーリング）するため、APIコールに対する課金[5]がなくなるわけではないので注意しましょう。

Lambdaと連携して使用するに時には、特にビジビリティタイムアウト（可視性タイムアウト）[6]の概念を理解し、ビジビリティタイムアウトの時間とLambdaの実行時間を適切に設定する必要があります。SQSにエンキューされたメッセージは、サブスクライバーが受信するとインフライト状態になります。インフライト状態のメッセージは他のサブスクライバーから見えません。この見えない状態の時間制限が、ビジビリティタイムアウトです。ビジビリティタイムアウトの時間が過ぎたとき、メッセージを受信したサブスクライバーによりメッセージが削除されていない（サブスクライバー内での処理が失敗した）ときは、再び他のサブスクライバーから見えるようになります。

Lambdaにおいては5分を上限とするタイムアウトを設定することができます。もし、このタイムアウトをビジビリティタイムアウトよりも長くしてしまうと、LambdaでSQSからのメッセージに応じた処理をしている最中にメッセージが可視状態に戻るので、メッセージが二重に処理される可能性があります。そのため、「ビジビリティタイムアウト＞Lambdaのタイムアウト」と設定してください。マネジメントコンソールでは誤った設定ができないようになっており、samコマンドでもエラーになるようにはなっていますが、template.ymlには記述できるので念頭に置いておいてください。

3.3　SNS

SNS（Amazon Simple Notification Service）[7]はAWSが提供するPub/Subメッセージングサービスです。メッセージはプッシュ型で、トピックをサブスクライブできるのはLambdaをはじめ次のとおりです。

・Lambda
・SQS

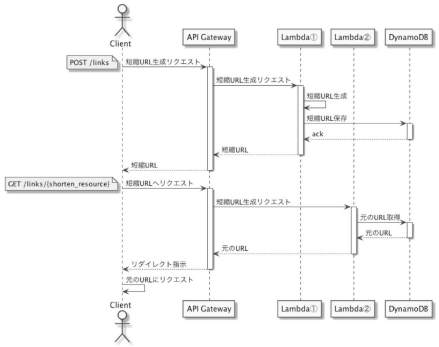
1.http://aws.clouddesignpattern.org/index.php/CDPFanout%E3%83%91%E3%82%BF%E3%83%BC%E3%83%B3
2.https://aws.amazon.com/jp/sqs/

3.https://aws.amazon.com/jp/blogs/aws/aws-lambda-adds-amazon-simple-queue-service-to-supported-event-sources/
4.https://docs.aws.amazon.com/ja_jp/AWSSimpleQueueService/latest/SQSDeveloperGuide/FIFO-queues.html
5.https://aws.amazon.com/jp/sqs/pricing/
6.https://docs.aws.amazon.com/ja_jp/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html
7.https://aws.amazon.com/jp/sns/

API Gateway、DynamoDB、Lambda を使って短縮 URL を実際に作りながら解説

4.4 シーケンス

図 4.1: シーケンス図



4.5 フォルダー構成

図 4.2: フォルダー構成

```
url-shortener-lambda-go
├── handlers
│   ├── db
│   │   ├── db.go
│   │   └── testhelper.go
│   ├── redirect
│   │   ├── main.go
│   │   └── main_test.go
│   └── shorten
│       ├── main.go
│       └── main_test.go
├── .gitignore
├── .go-version
├── Gopkg.lock
├── Gopkg.toml
├── Makefile
└── template.yml
```

ソースコード全体は GitHub のリポジトリー[11]で公開しています。

4.6 ソースコード

本書では go1.11.2 を利用するため、プロジェクトのルートで次のコマンドを実行して.go-version を作成してください。

```
$ goenv local 1.11.2
```

プロジェクトのルートフォルダーは Go の規約に則って$GOPATH/src/github.com/toshi0607/url-shortener-lambda-go のように作成してください。

.go ファイルが紙面の都合上インデントが半角スペース×2 になっていますが、goimports などのフォーマッタでハードタブに変更してください。

まず短縮 URL を生成する Lambda①を実装します。API Gateway へのリクエストがマッピングされた event から短縮対象の URL を取り出し、短縮 URL を生成して DynamoDB に保存しています。

11 https://github.com/toshi0607/url-shortener-lambda-go

＜＜目次＞＞

＜＜著者紹介＞＞

杉田 寿憲(すぎた としのり)
株式会社メルペイのバックエンドエンジニア。Go と GCP のマイクロサービスと闘争中。一緒に開発してくれる仲間を探しています。

＜＜販売ストア＞＞

電子書籍:
Amazon Kindle ストア、楽天 kobo イーブックストア、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。また自らも、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イア&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