AWS Lambda実践ガイド

第1章 Lambda で実現するサーバーレスシステム

なぜ Lambda を使うのか？

- 保守・運用の手間がない
  - マネージドサービスの特徴
- 高負荷に耐えられる
  - 必要に応じてスケーリングされる
- 低コスト
  - Lambda の料金は実行時間に対して決まる

EC2 だと...

- 運用・保守上の問題
  - 脆弱性対応、環境構築
- 高負荷への耐性
  - 負荷耐性は基本的にEC2の構成によるため、突然の高負荷に耐えられない可能性
- (比較的)高コスト
  - 例えば、夜の10時にだけ実行させたい処理があったとしたら？

Lambda はサーバレスアーキテクチャ

実行環境がAWSによって実行ごとに用意される。
そのため、ユーザが実行環境としてサーバを持つ必要がない ← サーバレス
ユーザは実行したいプログラムを「関数」として登録する
利用可能な言語：C#, **Go**, Java, Node.js, Python, **Ruby** (本の情報は少し古い)

開発しやすいLambda

- 実行したい機能を「関数」として登録するだけでよい
- 一つ一つの「関数」は小さく作るのでテストが容易

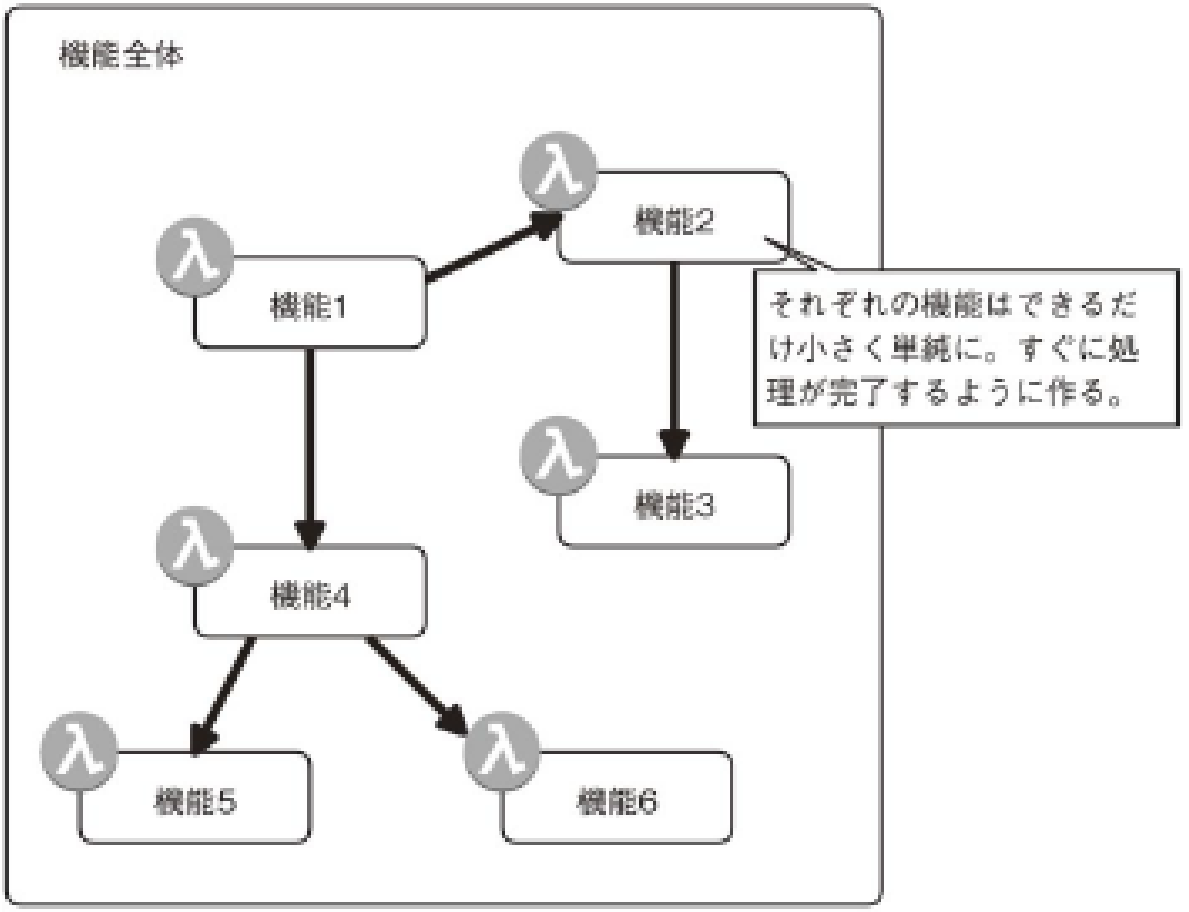


図 1-4　たくさんの Lambda 関数を組み合わせてシステム全体を構築する

Lambda の制限

- ステートレス
  - 実行環境が都度破棄されるため、状態を保持しておくことはできない
- 最大稼働時間
  - 最大でも5分でタイムアウトになる

つまり、Lambda は必要に応じて、ちょっとした処理をするときに便利だが、継続的な処理には向かない
継続的な処理はEC2の方が適している

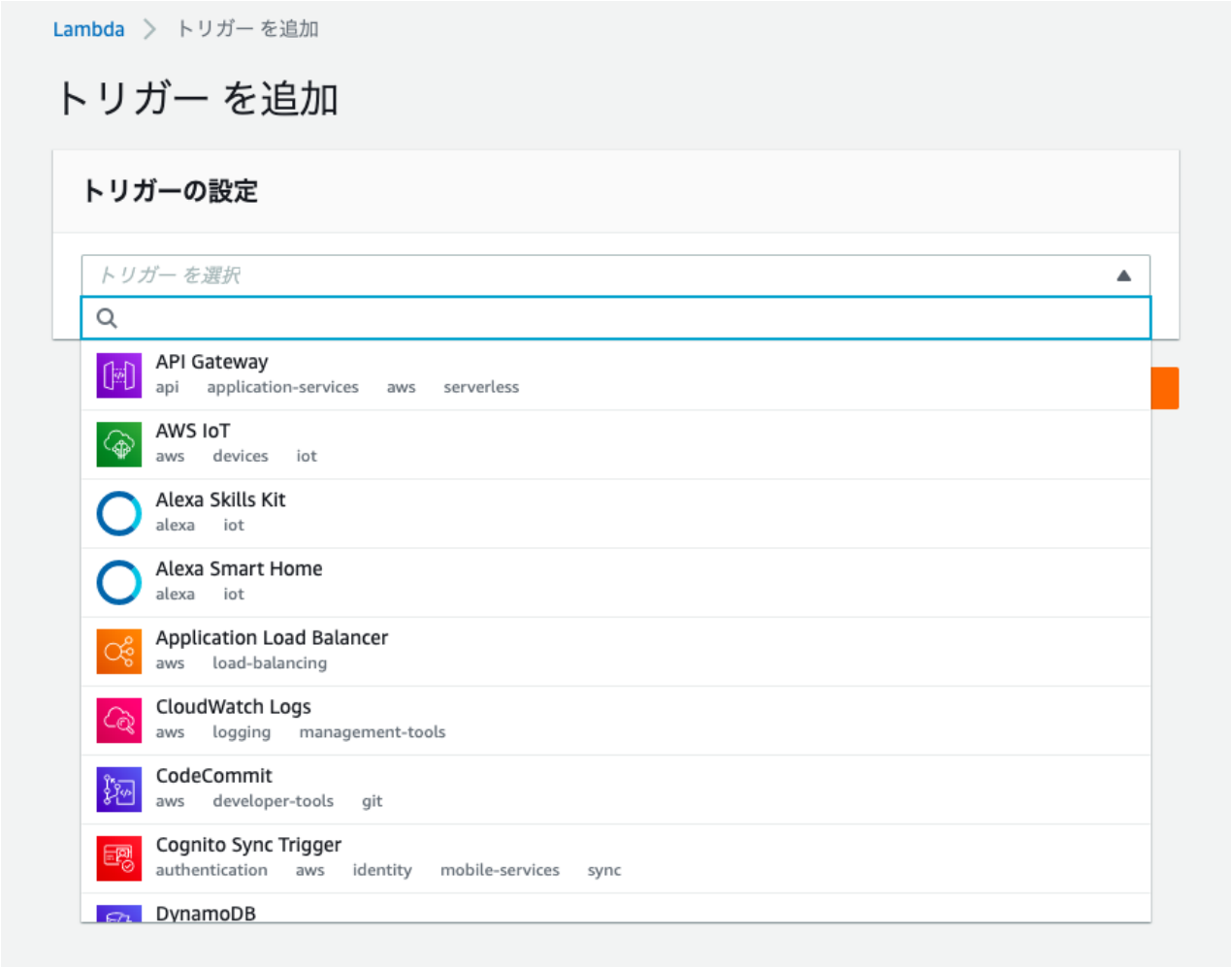
Lambda の使い方

一定の時間帯に実行したい処理に使うこともできるが、、、
他サービスとの連携で使われることが多い

→ イベントドリブンの糊付けプログラミング

イベントソース

いろいろなサービスがトリガーとして選択できる →



イベントソース

イベントドリブンとはイベントソースと呼ばれるもののある出来事をトリガーとして処理をすること

API Gateway をトリガーとしてDynamoDBと連携する例

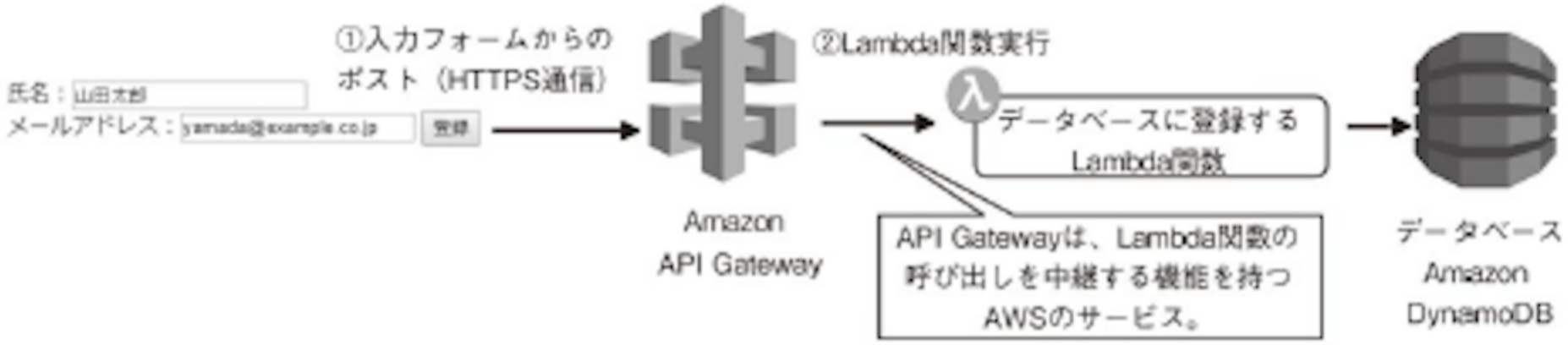


図 1-7　API Gateway と Lambda 関数を組み合わせて REST 形式の Web API を作る

まとめ

- EC2での処理をLambdaに置き換えることで低コスト化などメリットがある
- Lambdaにはいくつかの制限があることに注意
- Lambdaはイベントドリブンで他のサービスを結びつける

ところで...

本で取り上げられていた API Gateway → Lambda → DynamoDB のプログラムを以前組んだのでサンプルとして置いてあります(index.js)
POST で来たデータをごにょごにょしてDynamoDBに突っ込むというプログラムになってます