

AWS Lambda を使った APIサーバー公開ハンズオン

JAWS-UG × Shingen.py コラボ回 vol.1

自己紹介



Issei Hamada

Sony Biz Networks Corporation

- JAWS-UG 山梨運営メンバー
- 出身地: 神奈川県
- インフラエンジニアとしてキャリアをスタートし、今は生成 AI にどっぷりです
- **Japan AWS Ambassador (2023~)**
- 好きな AWS サービスは、**Amazon Q Developer** です



JAWS-UG
AWS User Group - Japan
YAMANASHI

Copyright 2025 JAWS-UG Yamanashi

後半パートの目標

前半で作成した自分のコードを**ビルド**し、
AWSへ**デプロイ**する

後半パートの目標

■いわゆる「アプリケーション開発」の流れ

① コードを書いて、



●コーディング

② 動かすのに必要な部品を全て
まとめて、



●ビルド

③ 公開する



●デプロイ

後半パートの目標

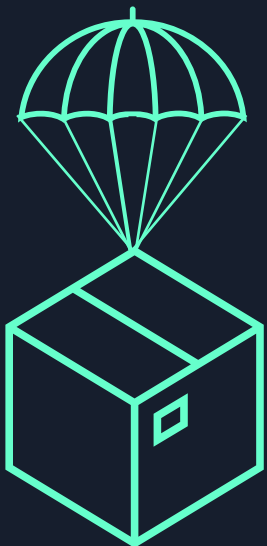
ビルド



- アプリケーションを作るとき、誰かが作ってくれた便利ツール「ライブラリ」を使う事が非常に多い
 - Ex: 予測分析アプリを作る為に、重回帰分析の計算が必要
 - 「scikit-learn」というライブラリに式が入っている
 - アプリを動かす為には、この「ライブラリ」を揃える必要がある
- プログラミング言語によっては、人間が書いたコードを機械語に翻訳(コンパイル)する必要がある
- これらの作業を、合わせてビルドと呼ぶ

後半パートの目標

デプロイ



- ビルドしたアプリを動かすには、コンピュータが必要
- 不特定多数の人間に触ってもらうには、このコンピュータを公開する必要がある
 - ・全てを公開すると危険なので、一部の領域のみを公開する
- アプリケーションを正しく配置する作業を**デプロイ**と呼ぶ

後半パートの目標

■アプリケーション開発のここが大変!!



開発・ビルド・公開時の環境を揃えなければならない

アプリを動かす為のサーバを用意しなければならない



全世界から見えるよう、サーバを公開しなければならない

後半パートの目標

■アプリケーション開発のここが大変!!

aws



開発・ビルド・公開時の環境を揃えなければならない



アプリを動かす為のサーバを用意しなければならない



全世界から見えるよう、サーバを公開しなければならない

ここで活躍するのが……

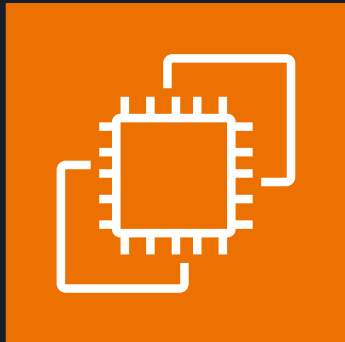
AWS とは?



- 世界で最も包括的で、幅広く採用されているクラウドコンピューティングサービス
- 世界中の、地理的に分散されたデータセンター上に構築されている
 - 「地域(リージョン)」は、必ず2つ以上のデータセンターで構成されている
 - 障害が起きづらい
- あらゆる場面に対応出来る、様々なサービスが用意されている
 - 2025年時点で250個らしい

AWS とは?

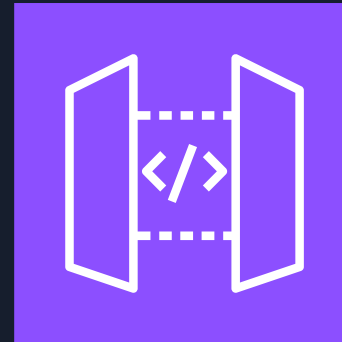
■今回のハンズオンでも、複数のサービスを組み合わせて API を実装する



Amazon Elastic Compute
Cloud (Amazon EC2)



AWS CloudFormation



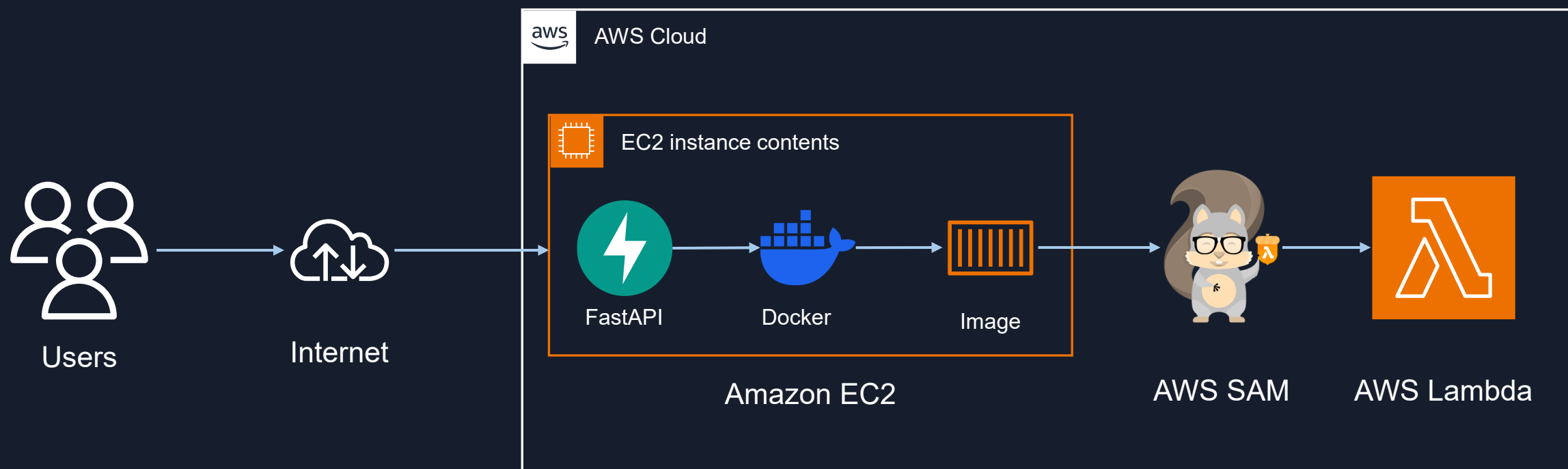
Amazon API Gateway



AWS Lambda

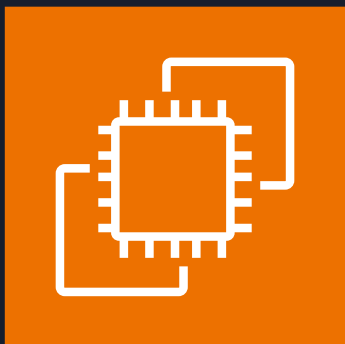
環境説明

- AWS 上に構築した Code Server から、API を Lambda としてデプロイする



環境説明

Amazon EC2 とは



- AWS が提供する、仮想マシンサービス
 - ・ いわゆるサーバ
 - ・ 恐らく AWS で S3 と並んで有名
- OS 等、必要なソフトウェアがセットアップされた状態で立ち上がる為、すぐに利用を始められる
- 今回は開発用サーバとして利用

環境説明

AWS Lambda とは



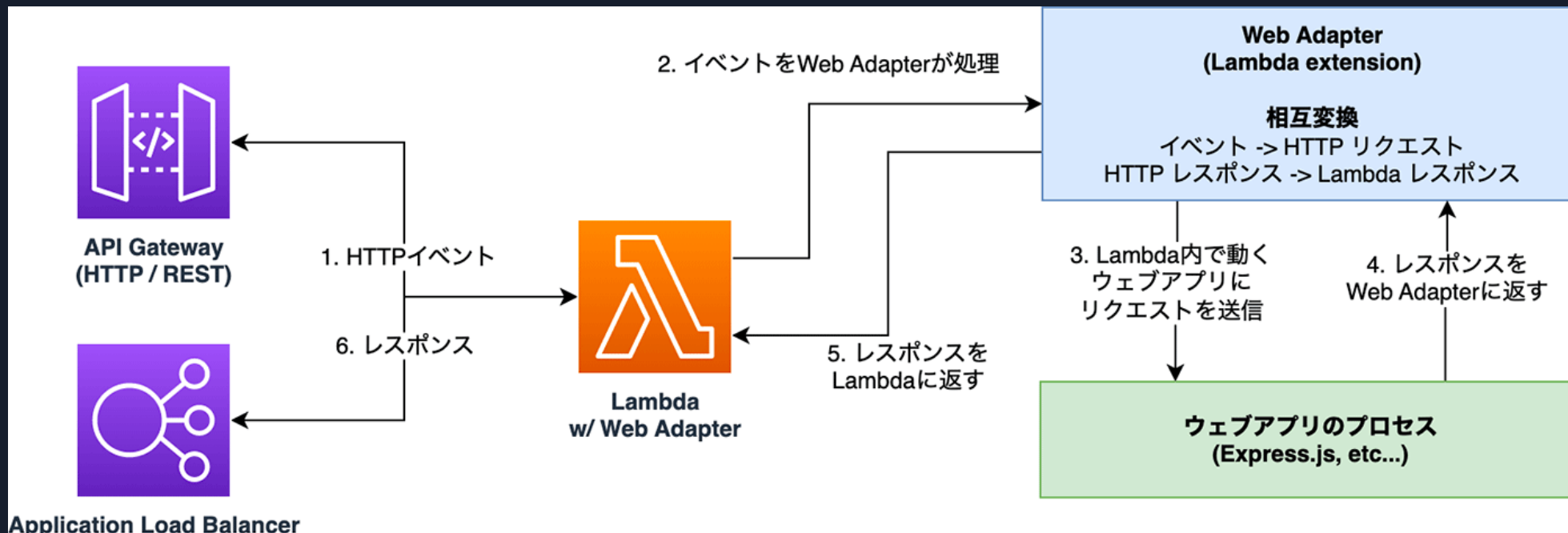
- サーバ管理無しにコードを実行出来る、サーバレスコンピューティングサービス
 - ・ 正確には、コードを実行するその時だけコンピュータが立ち上がり、実行が終わると止まる
- Lambda を使う事で開発者はインフラの管理をせず、コード開発に集中できるようになる
- 今回は、先ほど開発した FastAPI のコードを Lambda としてデプロイして実行する

環境説明

上級者向け

■今回は、Lambda Web Adapter という拡張機能を利用

- Lambda に引き渡されるイベントを HTTP 形式に変換してアプリに伝える機能



環境説明

上級者向け

■Lambda Web Adapter の使い方

- アプリと拡張機能を含めたコンテナイメージを作成して、Lambda をデプロイする

■デプロイ手順

1. Dockerfile を作成
2. コンテナイメージをビルド
3. イメージを元に Lambda をデプロイ

※ 今回は、濱田にて Dockerfile 作成済み

環境説明

AWS SAM とは



- AWS にサーバレスリソースをデプロイする事に特化した、CloudFormation の拡張書式
 - CloudFormation = インフラストラクチャをコードで作成出来るようにするサービス
- CloudFormation と比較して少ないコード量でサーバレスリソースを定義出来る他、ビルド等便利な機能を備えている
- 今回は、Lambda をデプロイする為に利用

ハンズオン

■手順

- 1人に1台の開発サーバを用意しています
 - スプレッドシートを確認して、空いているサーバを利用して下さい
 - <https://docs.google.com/spreadsheets/d/1MLHjcC36LWxqiYA0aFjIdvEbtr-Vk3z0ie9EeCtETZI/edit?usp=sharing>
- 開発サーバに入ったら、以下の手順書を git clone (DL) して作業を始めて下さい
 - <https://github.com/issei-hamada/hmddev-lambda-web-adapter-workshop.git>



JAWS-UG

AWS User Group - Japan

YAMANASHI