具体的なユースケースから学ぶ、サーバーレスアプリケーションの活用方法

野村 侑志

アマゾン ウェブ サービス ジャパン合同会社
ソリューションアーキテクト

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

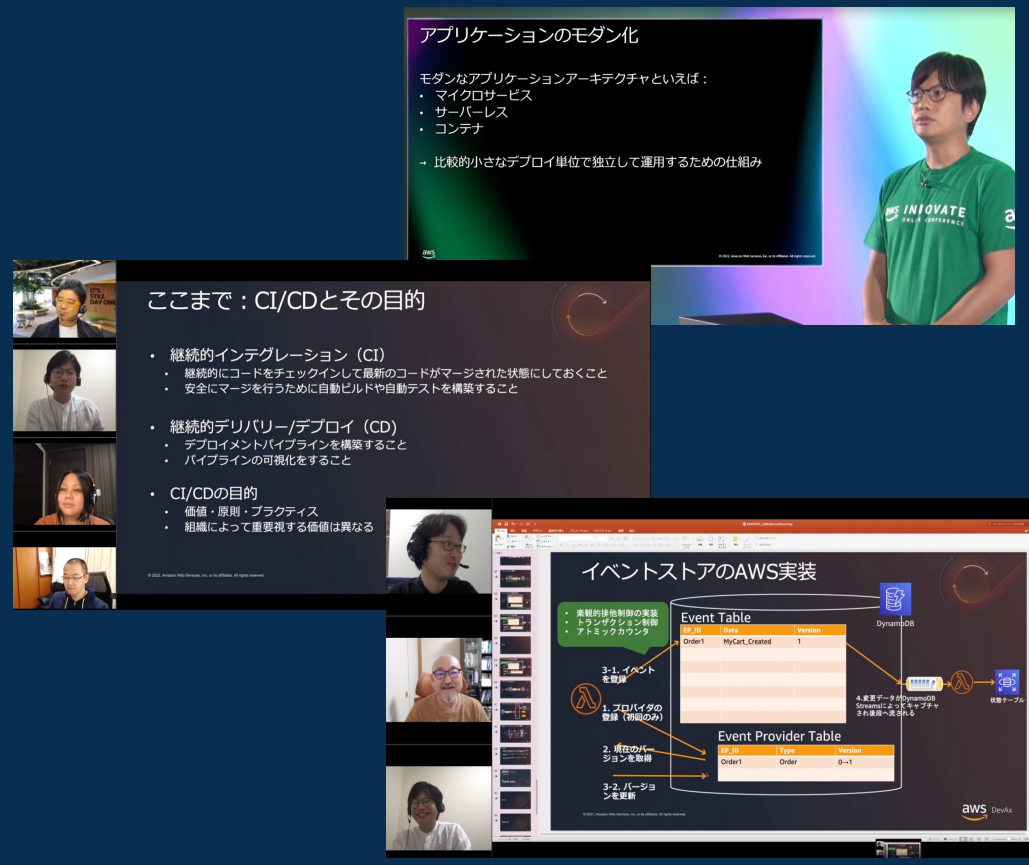
自己紹介

名前：野村　侑志　twitter：@ugnomura

略歴：AWSソリューションアーキテクト
← 生保会社 ← 米国 ← 複合機開発
← 音楽大学（打楽器）

好きなAWSサービス：Amazon SQS

興味：モダンアプリケーション, API, コンテナ
圏論, Haskell, 競プロ など

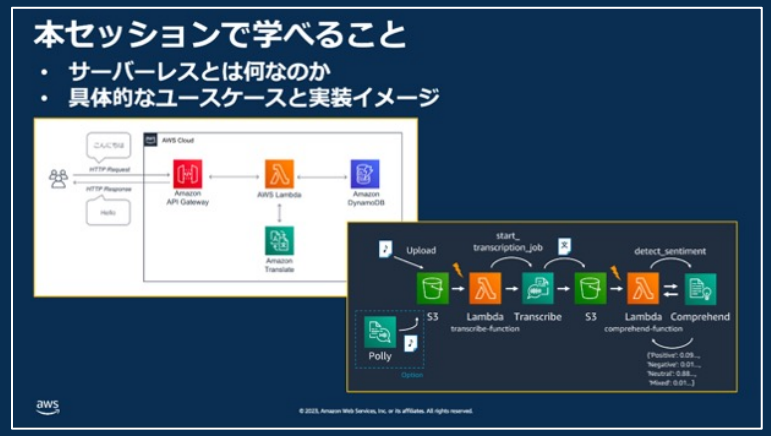


モダンアプリケーション開発系の登壇などの
活動を活発に行なっています！

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

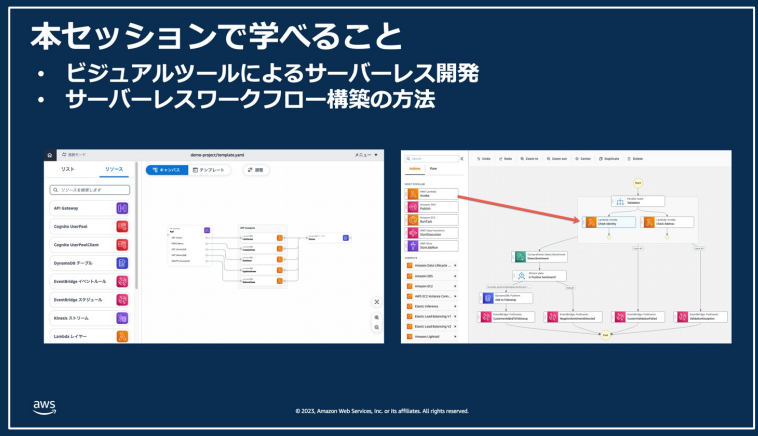
関連セッションについて

- 3つの関連セッションで構成しています。
- 本セッションでは、サーバーレスの基本的な考え方や関連サービスの基礎、組み合わせ例についてご紹介します。AWS Lambda / Amazon API Gateway / Amazon DynamoDB について簡単にご紹介する内容が含まれます。



具体的なユースケースから学ぶ、サーバーレスアプリケーションの活用方法

サーバーレスアプリケーションを効率よく開発しよう！
AWS SAMとローカル開発



ブラウザで開発する
サーバーレスアプリケーション

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

本セッションの対象

- サーバーレスという単語を聞いたことがあって
  これから触っていきたいと思っている方

- AWSをインフラとしてだけではなく
  モダンな形で使っていきたいと思っている方

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

アジェンダ

サーバーレスとは
サーバーレスのよくある使い方
サーバーレスの始め方

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスとは

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスとは?

サーバーがない?

サーバーの存在を意識しない

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスとは?

サーバーがない?

もう一歩踏み込んで頂くと...

サーバーの存在を意識しない

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

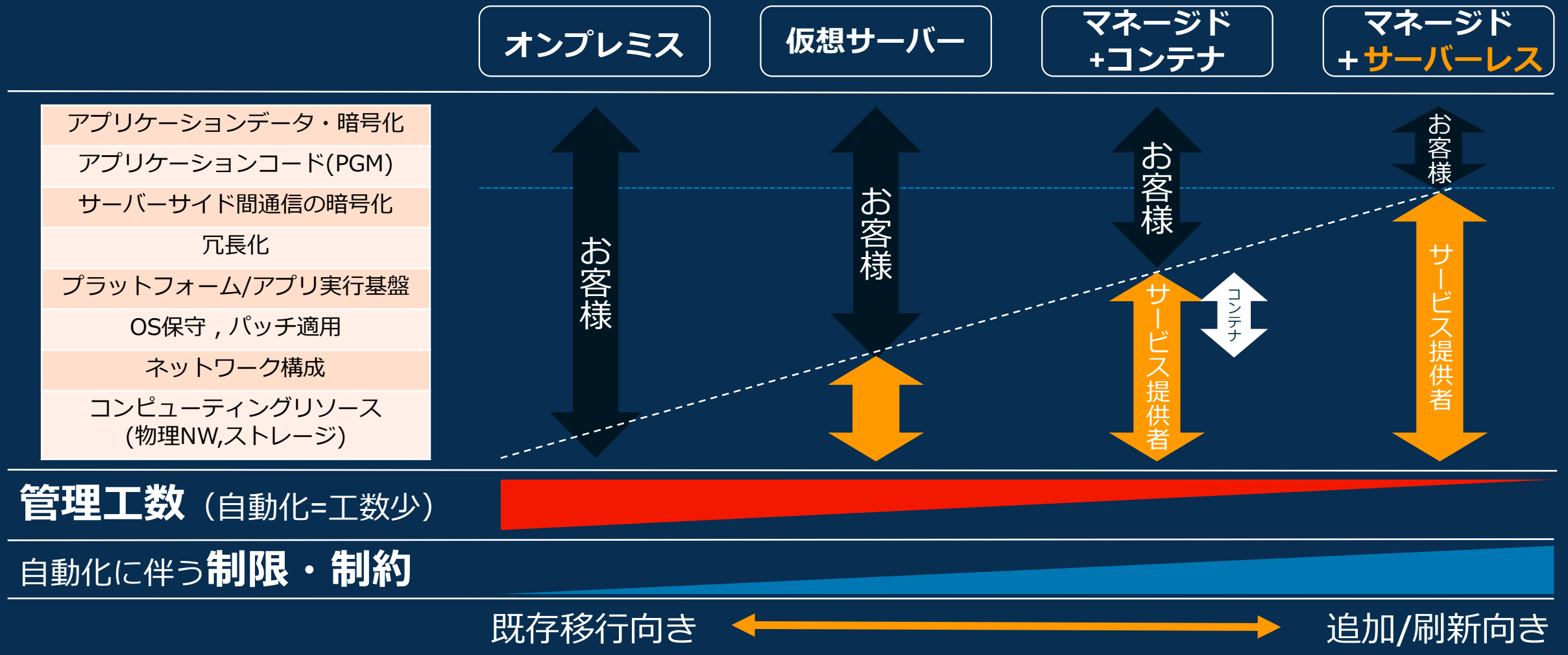
サーバーレスとは?

サーバーがない?

ユーザーコントロール可能な
サーバーを前提としない

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスとは?

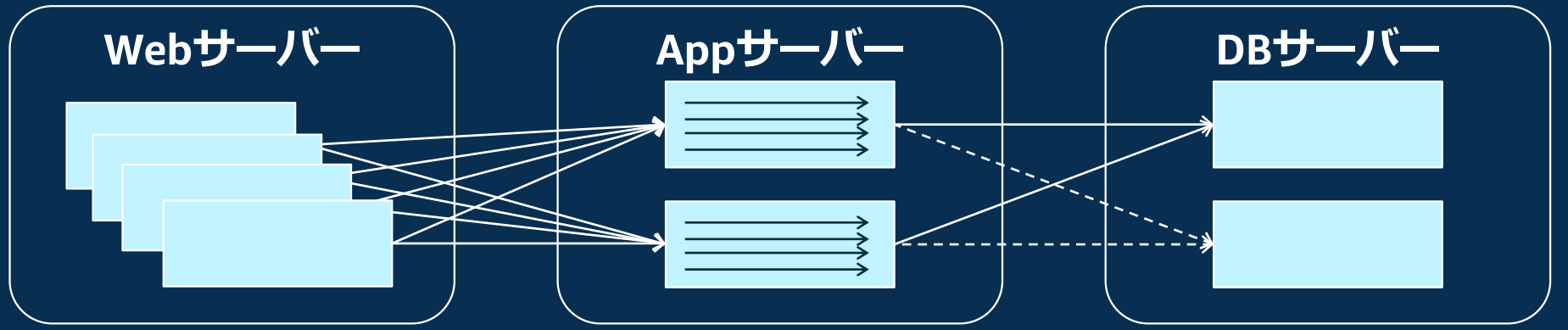


© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスとは? - 従来型のシステム設計

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスアーキテクチャーの例（従来型との対比）



【必要な作業】

- ✓ 規模の見積もり
- ✓ 可用性の設計
- ✓ 安定的かつ継続的なインフラ運用と保守

【得られるメリット】

- ✓ スケーラビリティ
- ✓ 設計済みの可用性確保
- ✓ インフラ運用と保守作業からの解放

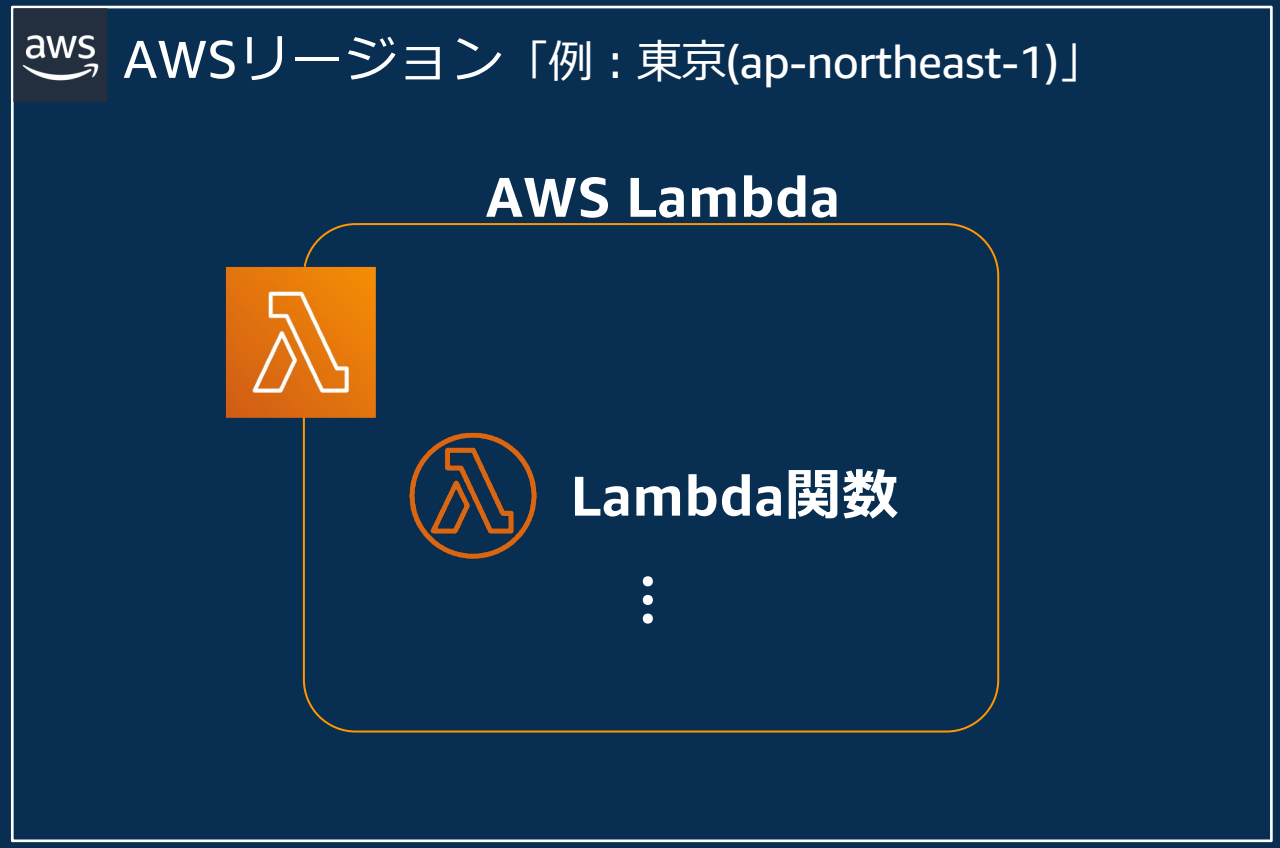
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda とは?

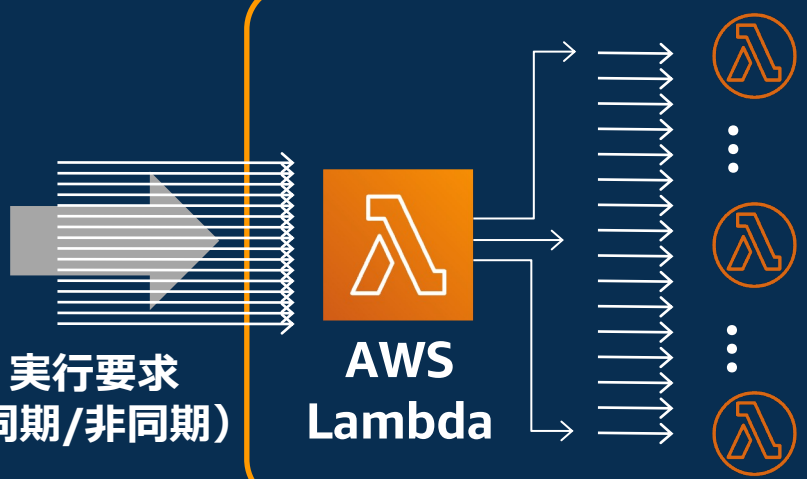
お客様が開発した **関数** を**サーバーレス**に実行できるサービス
→ コスト効率と管理性に優れたイベントドリブンでのプログラム実行基盤



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda とは? - イベント処理の担い手

Lambda は 「関数」を「開発・デプロイ・実行」する単位として扱い、
様々なイベントをトリガーに実行する



Lambda関数

【代表的な特徴（仕様）】

- 実行時間は最長で15分（デフォルト3秒）
- リソース量はメモリサイズのみを指定
- ライフサイクル直接制御不可
  （挙動はLambdaサービスが制御）

主な特徴・メリット

可用性/拡張性 設計不要

仮想サーバー（OS）管理不要

使用量に応じた課金

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS のサービスモデル

事業に集中できるサービスモデルを徹底追及することが AWS のミッション

オンプレミス	独自構築 on EC2	マネージドサービス	サーバーレス アーキテクチャ
アプリケーション作成	アプリケーション作成	アプリケーション作成	アプリケーション作成
スケールアウト設計	スケールアウト設計	スケールアウト設計	スケールアウト設計
定形運用設計	定形運用設計	定形運用設計	定形運用設計
ミドルウェアパッチ	ミドルウェアパッチ	ミドルウェアパッチ	ミドルウェアパッチ
ミドルウェア導入	ミドルウェア導入	ミドルウェア導入	ミドルウェア導入
OSパッチ	OSパッチ	OSパッチ	OSパッチ
OS導入	OS導入	OS導入	OS導入
HWメンテナンス	HWメンテナンス	HWメンテナンス	HWメンテナンス
ラッキング	ラッキング	ラッキング	ラッキング
電源・ネットワーク	電源・ネットワーク	電源・ネットワーク	電源・ネットワーク

開発者が担当　AWSが担当

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスの
よくある使い方

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスのよくある使い方

- Web APIの実行環境として
- データ加工処理の一環として

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWSサービスの本質はビルディングブロック

サービスを組み合わせることで、
素早くアプリケーション構築が可能

サーバーレスのよくある使い方

- Web APIの実行環境として
- データ加工処理の一環として

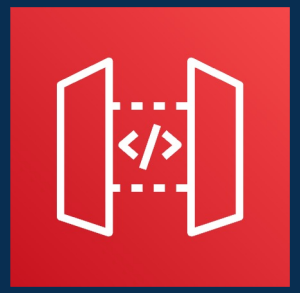
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Web API on サーバーレス

前捌き

プログラム実行

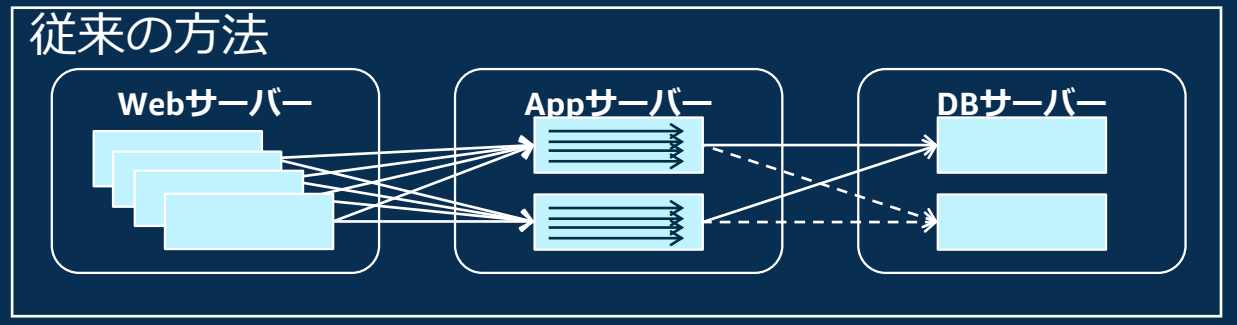
データ保持



Amazon API Gateway

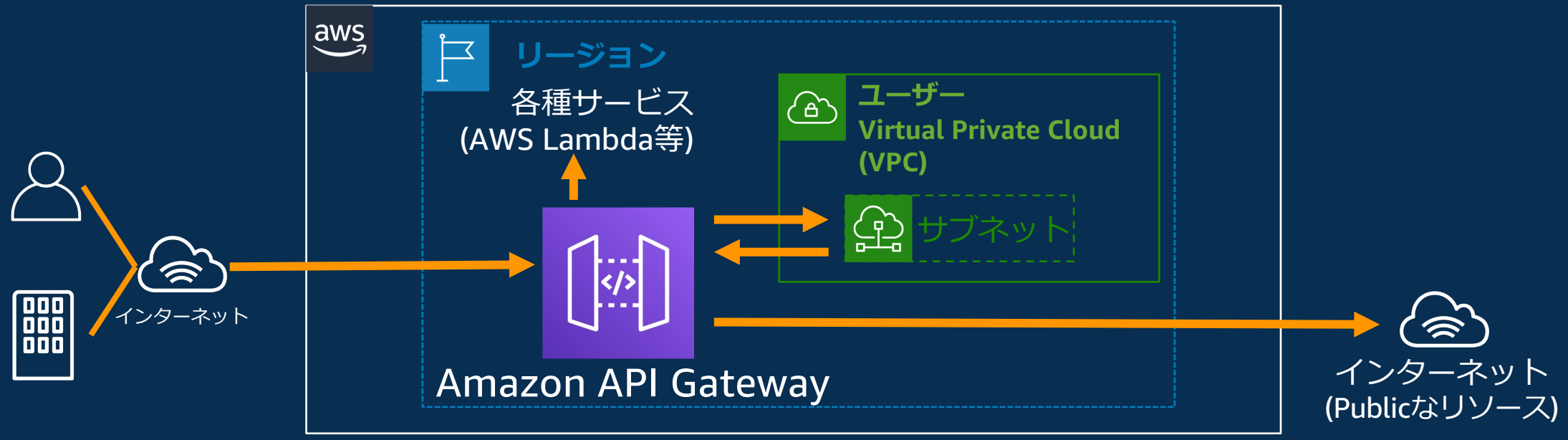
AWS Lambda

Amazon DynamoDB



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon API Gateway



主な特徴・メリット

- 可用性/拡張性 設計不要
- 仮想サーバー(OS)管理不要
- 使用量に応じた課金

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DynamoDB

どんな規模にも対応する高速で柔軟な キーバリュー データベース

スケールに応じたパフォーマンス

規模に関係なく、一貫した数ミリ秒台の応答時間を実現
事実上無制限のスループットでアプリケーションを構築可能

サーバーレス

サーバーのプロビジョニング、ソフトウェアのパッチ適用、アップグレードは不要
自動スケールアップ/ダウン
データの継続的なバックアップ

高いセキュリティ

デフォルトですべてのデータを暗号化
堅牢なセキュリティのためAWS Identity and Access Managementと完全に統合

グローバルデータベース

複数のAWSリージョンにまたがりテーブルを簡単に複製することで、ローカルデータに素早くアクセスできるグローバルアプリケーションを構築

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Web API on サーバーレス（再掲）

前捌き

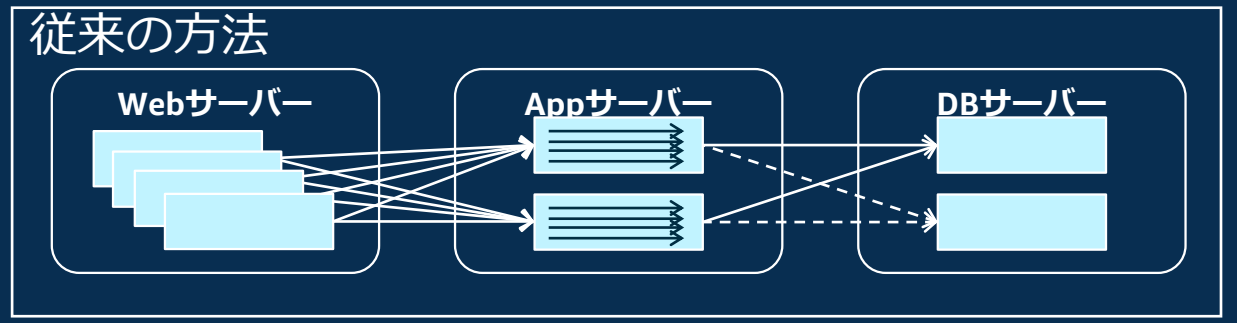
プログラム実行

データ保持

Amazon API Gateway

AWS Lambda

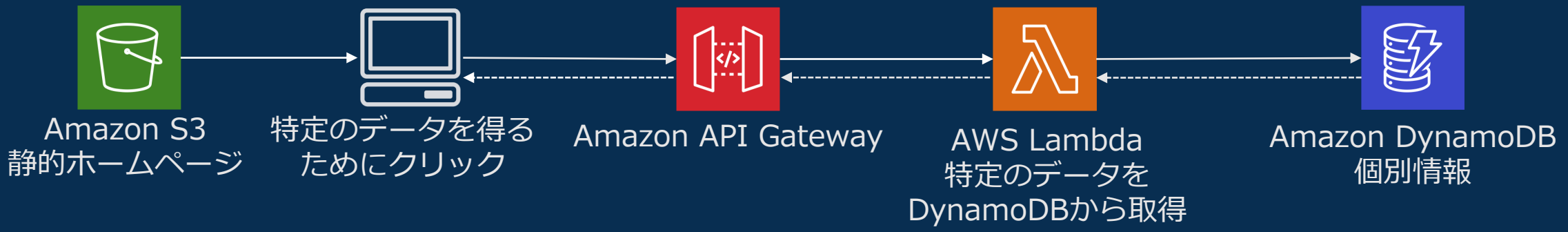
Amazon DynamoDB



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

ウェブアプリケーションの一部として

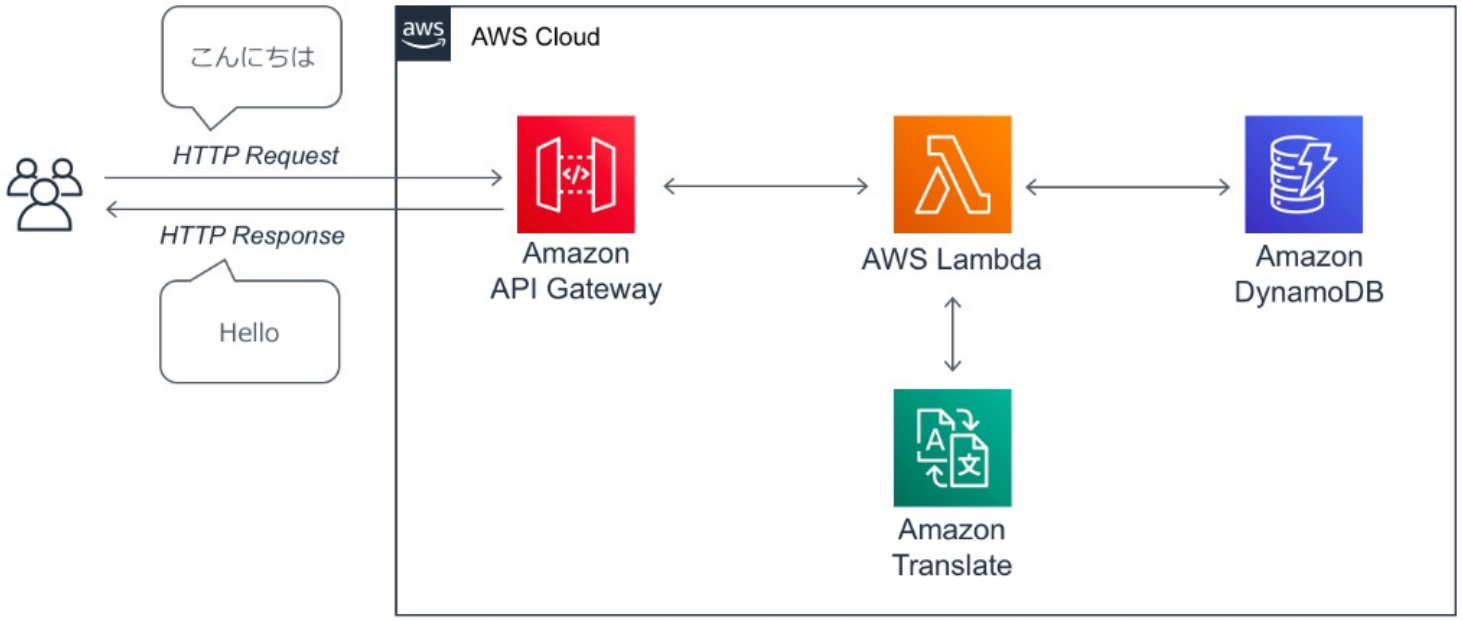
サーバーレスのウェブアプリケーションやバックエンドを構築し、
ウェブ、モバイルリクエストを処理し、永続化レイヤーでデータを保存



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hands-on for Beginners Serverless #1

AWS Hands-on for Beginners
～オリジナルAPIの作成を通してサーバーレスの基本を学ぼう～



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスのよくある使い方

- Web APIの実行環境として
- データ加工処理の一環として

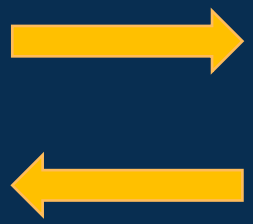
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

データ加工処理 on サーバーレス

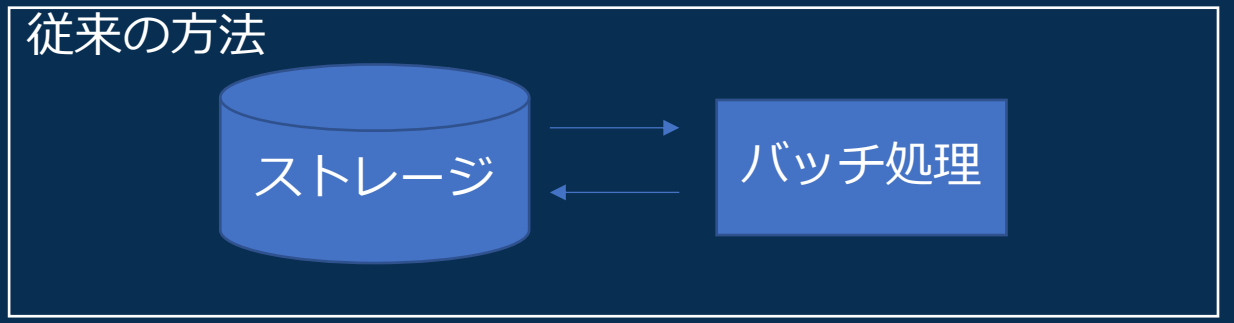
処理前後のデータを保持

データの加工

Amazon S3



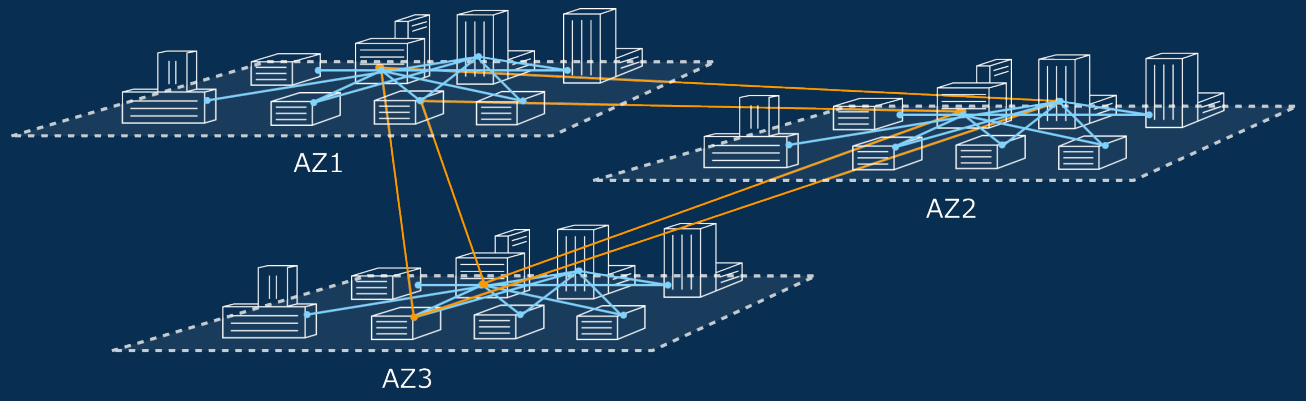
AWS Lambda

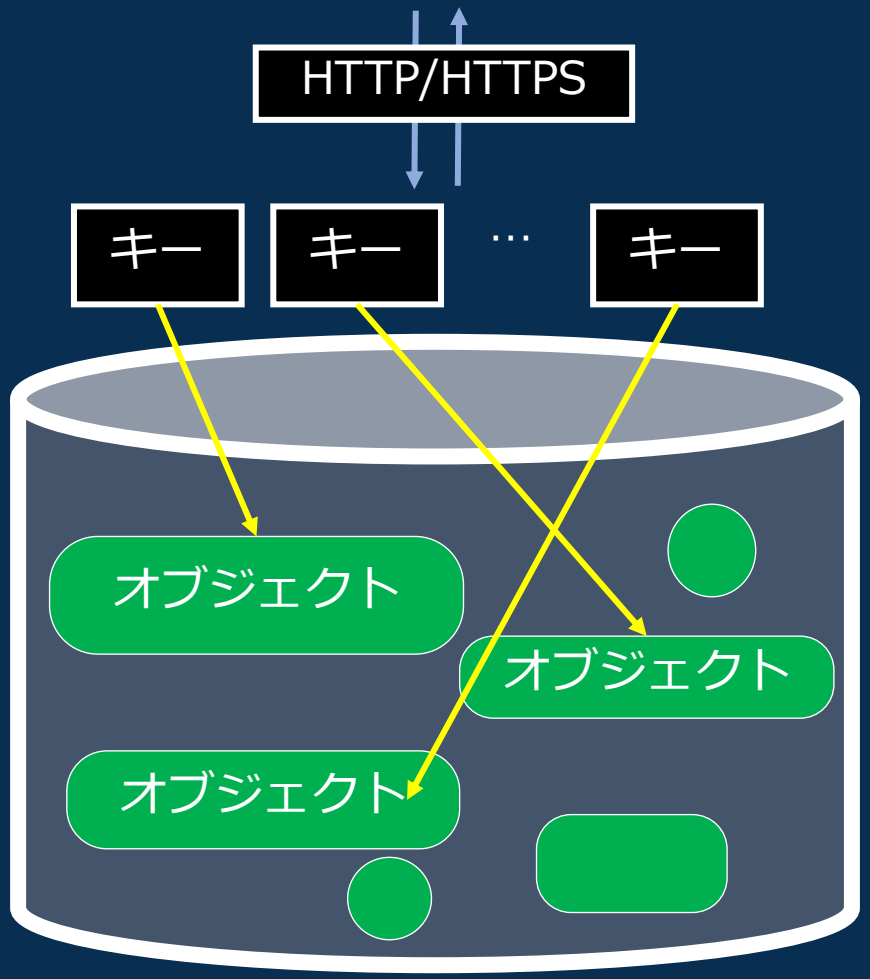


© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3

- AWSが提供するオブジェクトストレージ
- 自動的にオブジェクトを冗長化
- 低コストに大容量のデータを保存

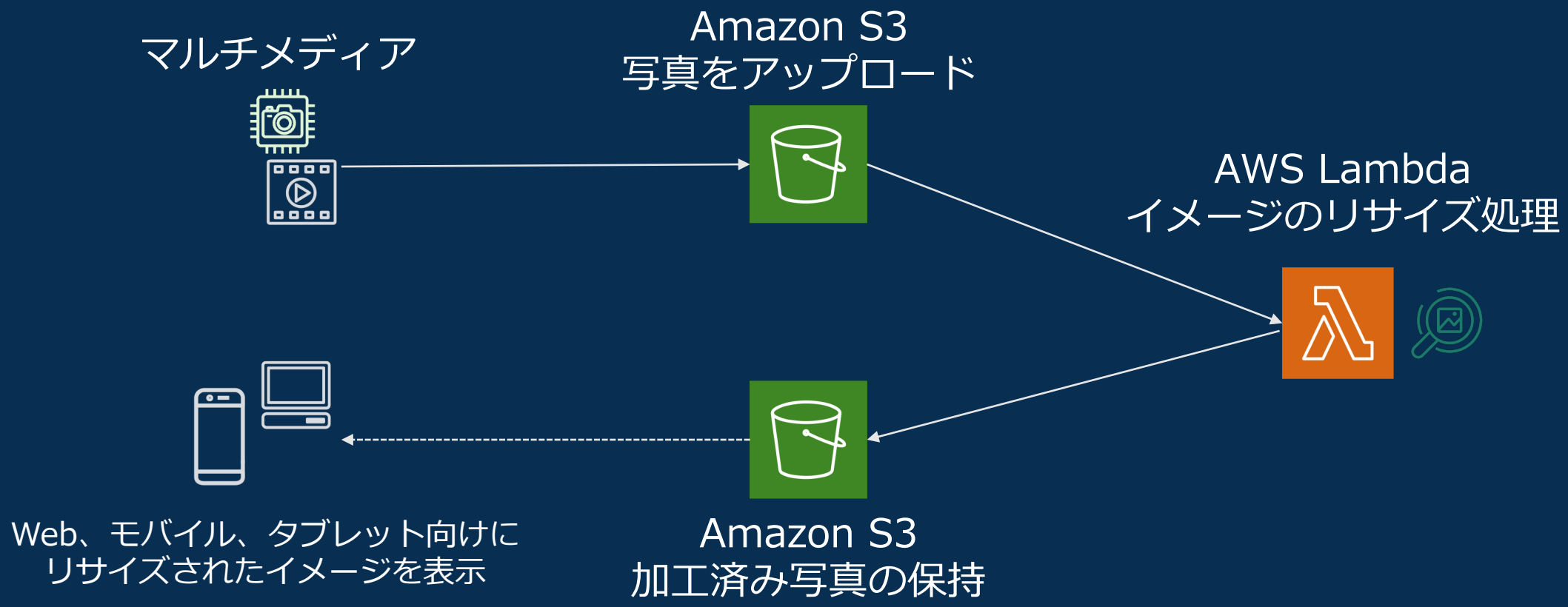




© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

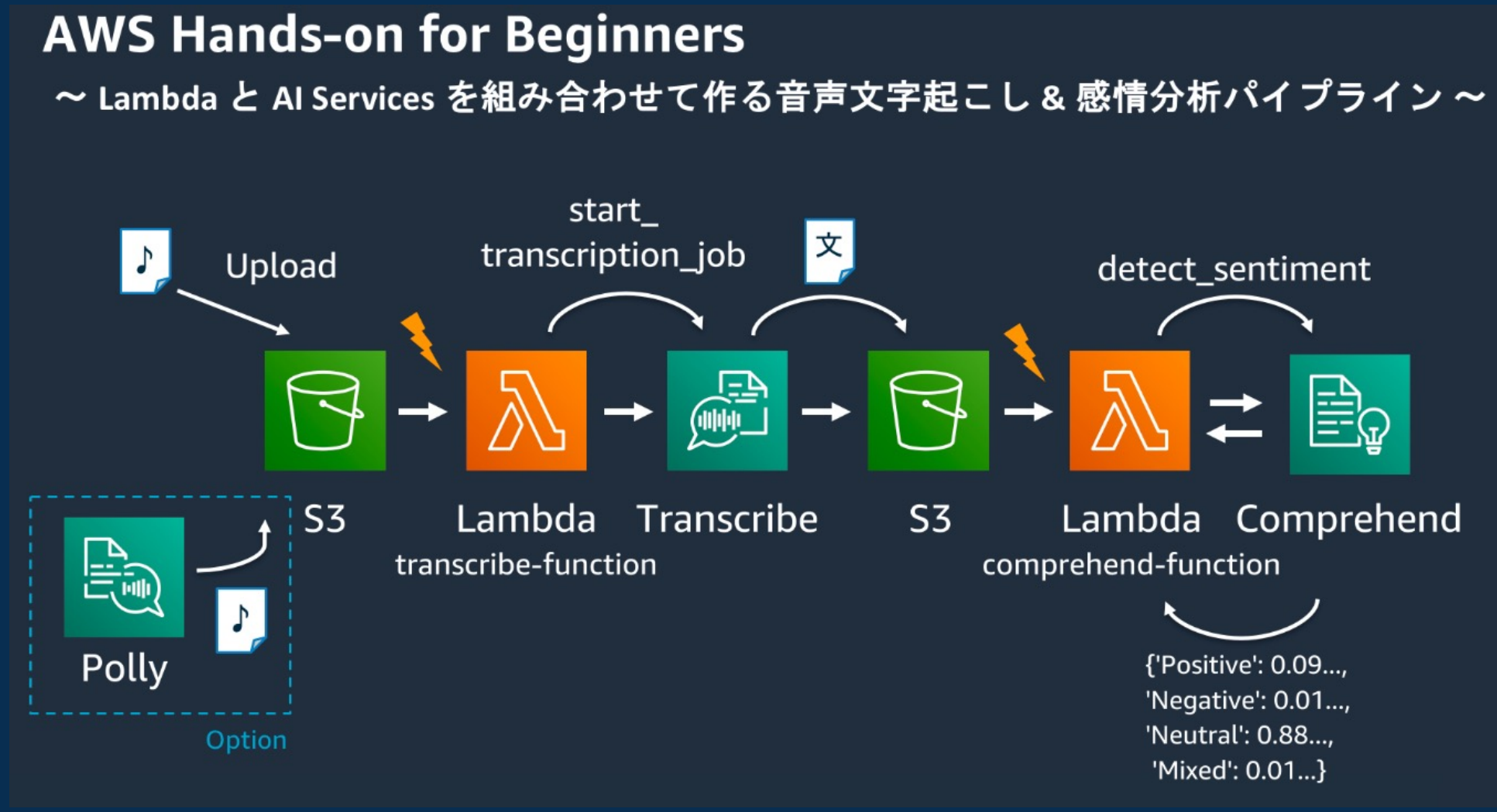
リアルタイムなデータ加工処理

AWSにアップロードされたデータをリアルタイムに処理。
例: 画像サムネイルの作成



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hands-on for Beginners Serverless #3



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスのよくある使い方

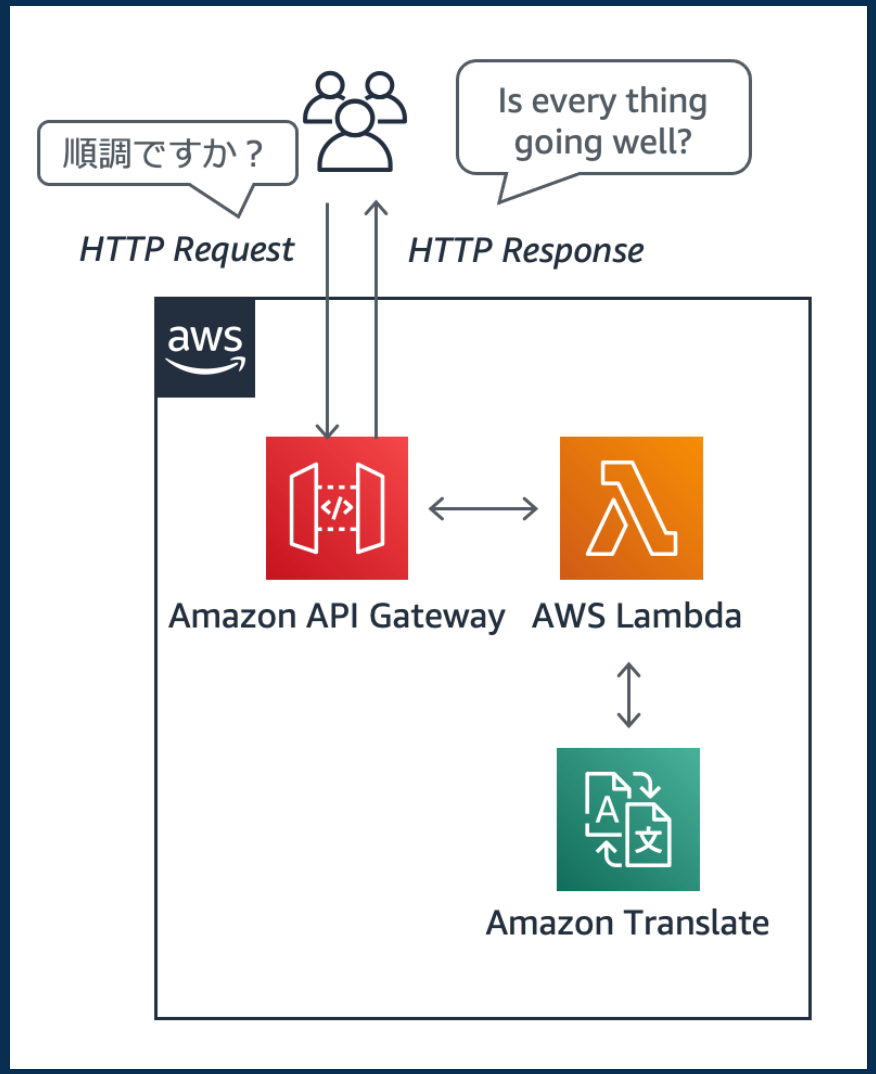
- Web APIの実行環境として
- データ加工処理の一環として

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレスの始め方

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

ウェブAPI デモ
(Hand-on for Beginners Serverless #1 より)



Amazon API Gateway

- Web APIの入口

AWS Lambda

- プログラムの実行

Amazon Translate

- プログラムの中で呼び出される翻訳サービス

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

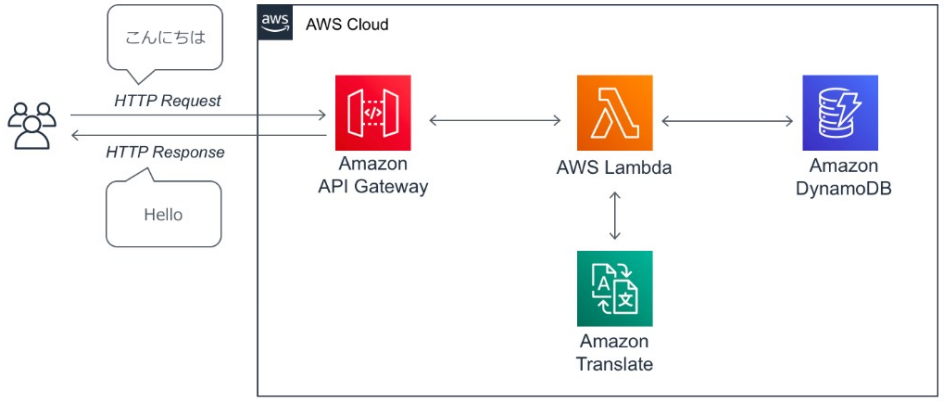
まとめ

- サーバーレスとは
  - ユーザーコントロール可能なサーバーを前提としないアーキテクチャ構成
- サーバーレスのよくある使い方
  - ウェブアプリケーションの構成要素
  - データ加工処理の一部として
- サーバーレスの始め方
  - ハンズオンをやってみよう！意外と簡単！

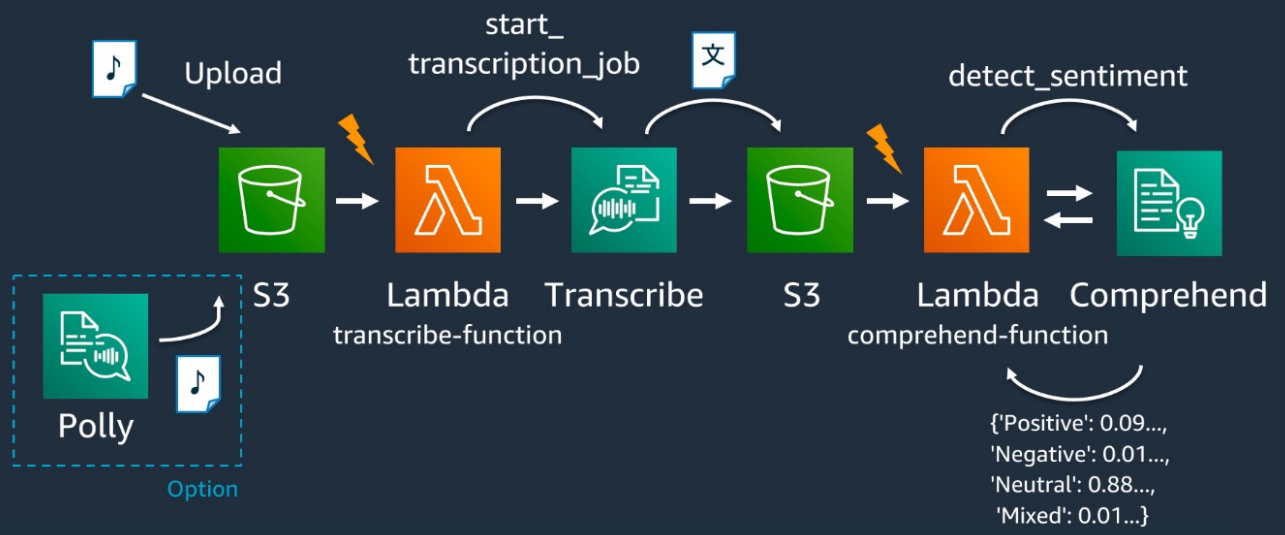
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

まずはハンズオンをやってみよう！

AWS Hands-on for Beginners
～オリジナルAPIの作成を通してサーバーレスの基本を学ぼう～



AWS Hands-on for Beginners
～ Lambda と AI Services を組み合わせて作る音声文字起こし & 感情分析パイプライン ～



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

サーバーレス自己学習ガイド

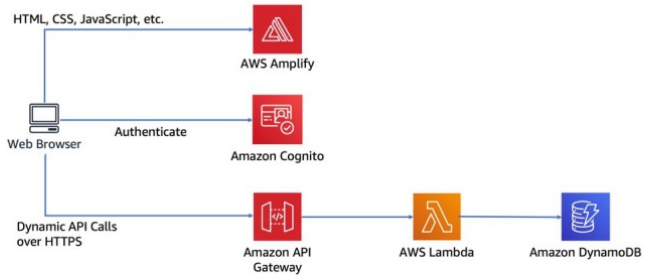
https://aws.amazon.com/jp/serverless/patterns/redirect-serverless-steps

- ハンズオンから始まる6つのステップ
  - Hands-on for Beginners も包含
- ご自分のペースで主要ポイントを学べます
- 開発作業の諸処で役立つサーバーレス技術情報サイトもご紹介

サーバーレスの始め方（1/2）

1 最初のトライ: サーバーの準備も実行環境構築も不要、いきなりアプリ開発を体験!

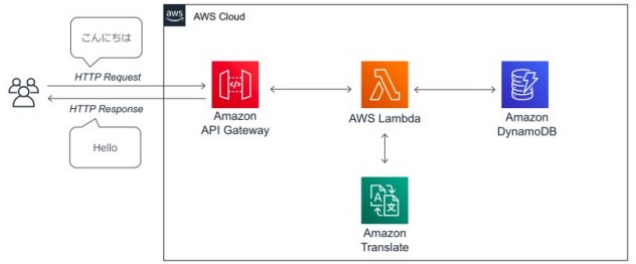
最初のサーバーレスWebアプリ：手順に沿えば、多くのサーバーレスサービスに触れながら、Webアプリが作れます。



go.aws/2Sh6I5M

「動的 Web / モバイルバックエンド」パターン

5-10分 x 11本のハンズオンで、サーバーレスな機能 API を作りながら、少しずつサービス自体の理解を深めていけます。



go.aws/2UhYdKw

「機能API」パターン

処理フローにチャレンジ

2 処理フローや例外処理の理解

新しいビジュアルエディタでアプリケーションの処理フローを視覚的にデザインできます。

エラー処理を含む処理の流れをフローとして定義すれば、アプリケーション全体の可読性を高め、保守を容易にします。

Thank you!

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.