

AWS Builders Online Series

T 3 - 2

はじめてのサーバーレス - AWS でのサーバーレス アプリケーション開発

池谷 友基

アマゾン ウェブ サービス ジャパン 合同会社
技術支援本部 テクニカル アカウント マネージャ



自己紹介

池谷 友基 (いけがや ともき)

アマゾン ウェブ サービス ジャパン
技術支援本部
テクニカルアカウントマネージャー



エンタープライズサポートプランにご加入頂いている
お客様向けに AWS の技術支援を行っています

好きなAWSサービス : AWS Support



対象者、本セッションで学べること

サーバーレスをこれから使っていきたいと考えている方に向けて
サーバーレスの基本的な考え方や関連サービスの基礎、良くある使い方
についてご紹介します。

- ・ サーバーレスという言葉を聞いたことがあるけれど
何から始めれば良いのかわからない方
- ・ インフラとしての AWS から
モダンな AWS の使い方へシフトしたいと考えている方



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、良くある使い方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。

本セッションで学べること

- ・本セッションで学べること
- ・なぜモダナイゼーションが必要か
- ・モダナイゼーションを進める上でのコンテナ活用のメリット
- ・AWS コンテナサービスを活用することで解決できる課題

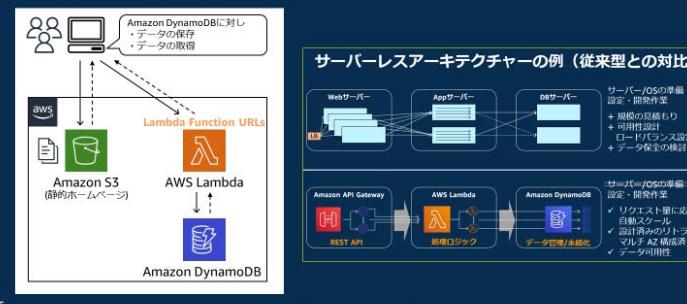


aws

T3-1: はじめてのコンテナワークフロー - AWS でのコンテナ活用の第一歩

本セッションで学べること

- ・サーバーレスとは何か
- ・サーバーレスな Web アプリケーション



aws

T3-2: はじめてのサーバーレス - AWS でのサーバーレスアプリケーション開発

本セッションで学べること

- ・オブザーバビリティとは何か、なぜ必要か
- ・AWS でオブザーバビリティを実現する方法



aws



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

T3-3: AWS ではじめるオブザーバビリティ - システムのどこで・何が・なぜ起こってるのかを理解する

aws

アジェンダ

サーバーレスとは

サーバーレスの良くある使い方

サーバーレスの始め方



サーバーレスとは



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

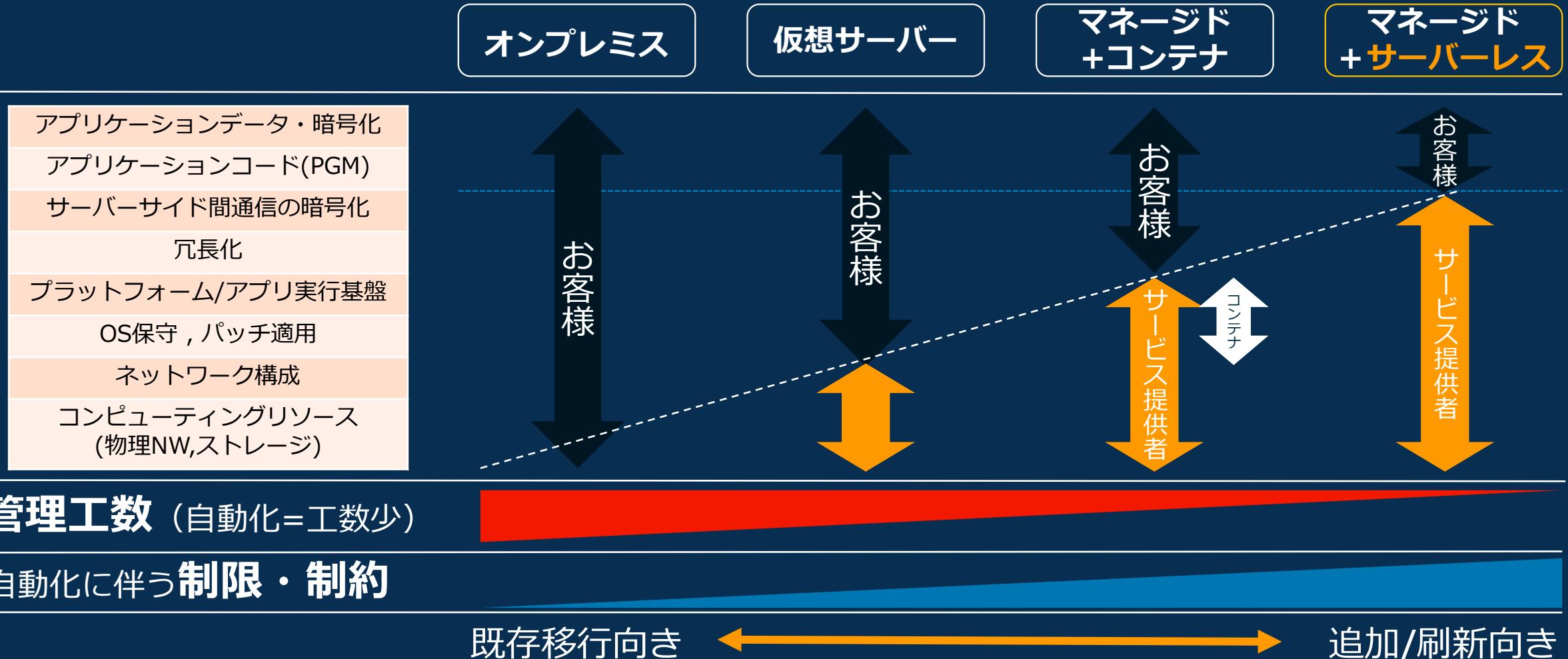


サーバーレスとは

サーバーがない

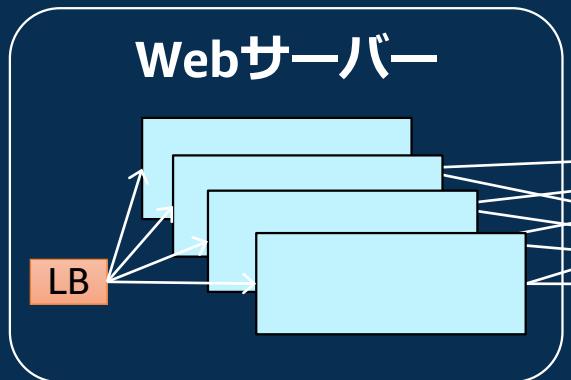
ユーザー権限をロール可能な
サーバーを前提としない

サーバーレスとは



サーバーレスとは - 従来型のシステム設計

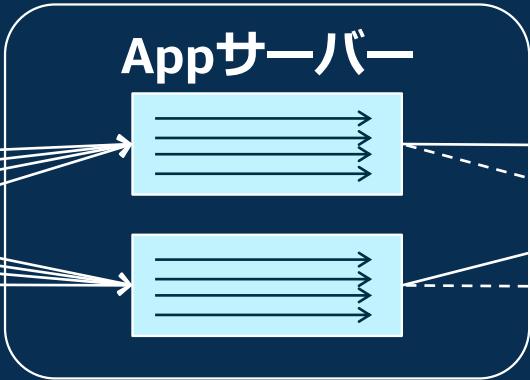
プレゼンテーション層



Web
サーバー

Apache, Nginx
等

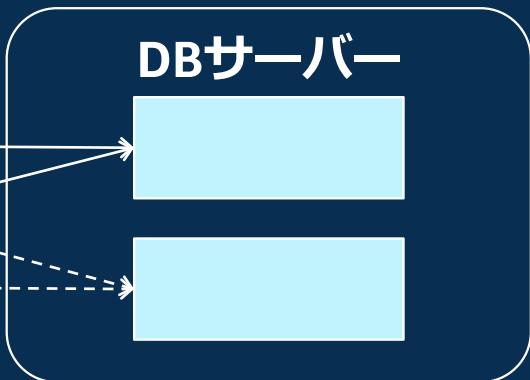
アプリケーション層



アプリケーション
サーバー

Tomcat,
WebLogic 等

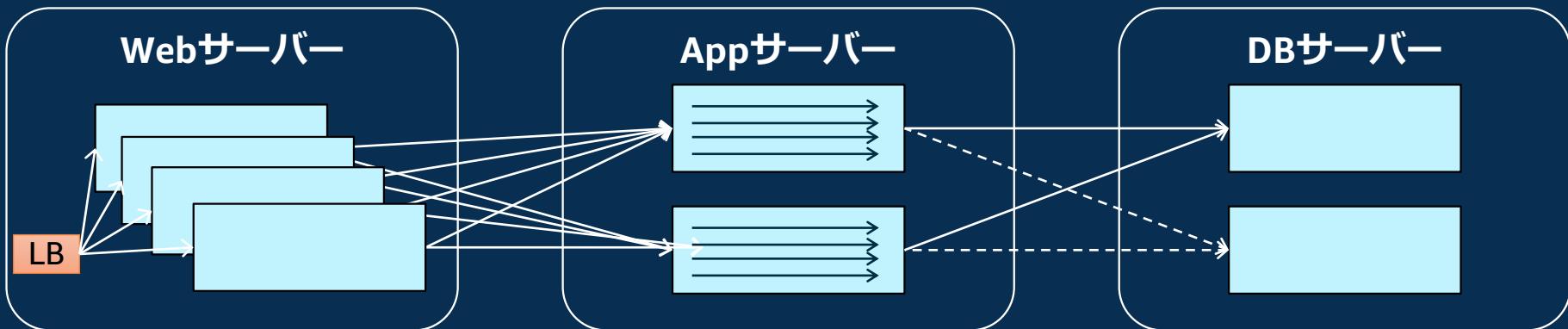
永続層



データベース
サーバー

Oracle DB,
MySQL 等

サーバーレスアーキテクチャーの例（従来型との対比）



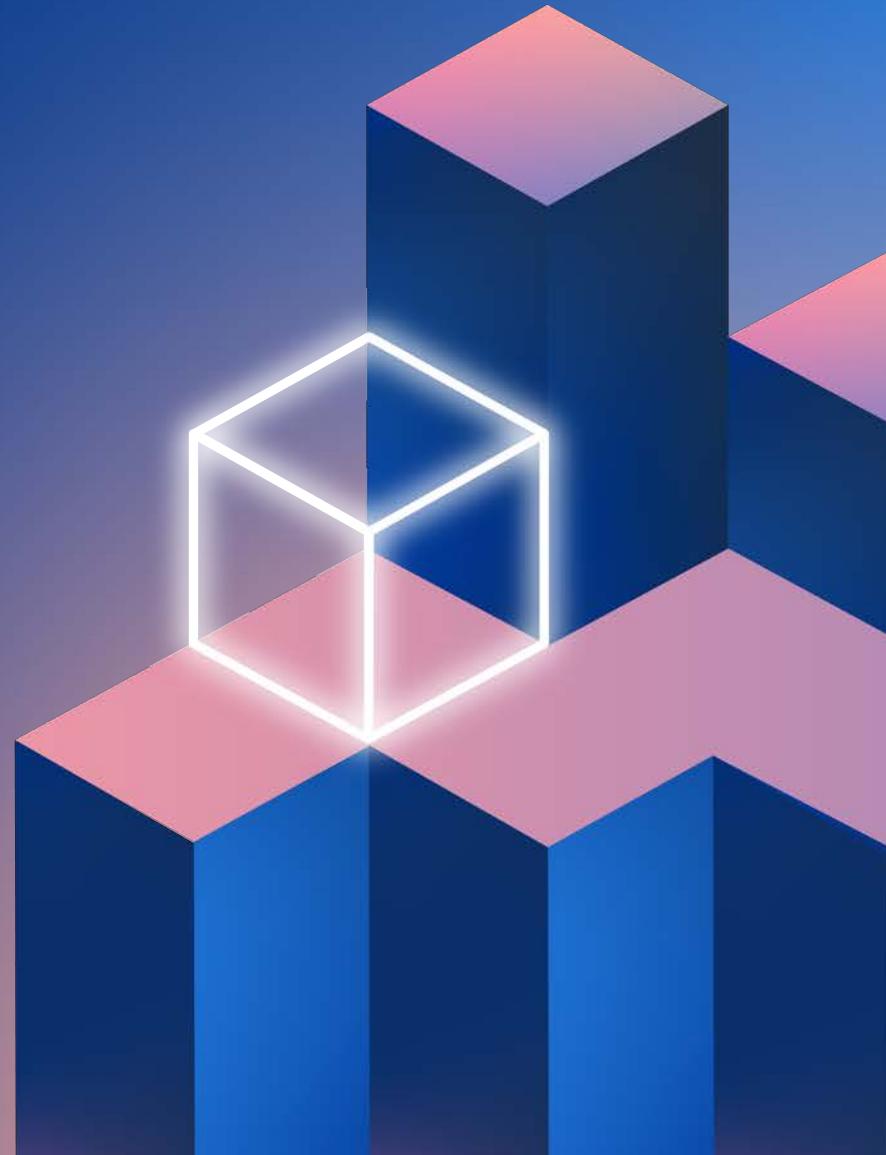
サーバー/OSの準備・構築
設定・開発作業
+ 規模の見積もり
+ 可用性設計
ロードバランス設定等
+ データ保全の検討



サーバー/OSの準備・構築
設定・開発作業
✓ リクエスト量に応じて
自動スケール
✓ 設計済みのリトライ
マルチ AZ 構成済
✓ データ可用性



AWS Lambda



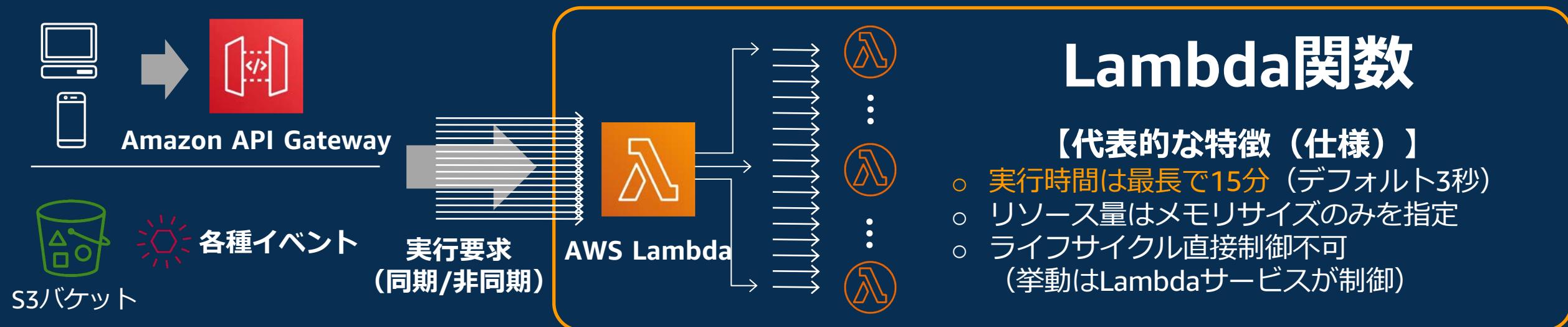
AWS Lambda とは

お客様が開発した **関数(プログラム)** 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



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



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

サーバーレスの良くある使い方





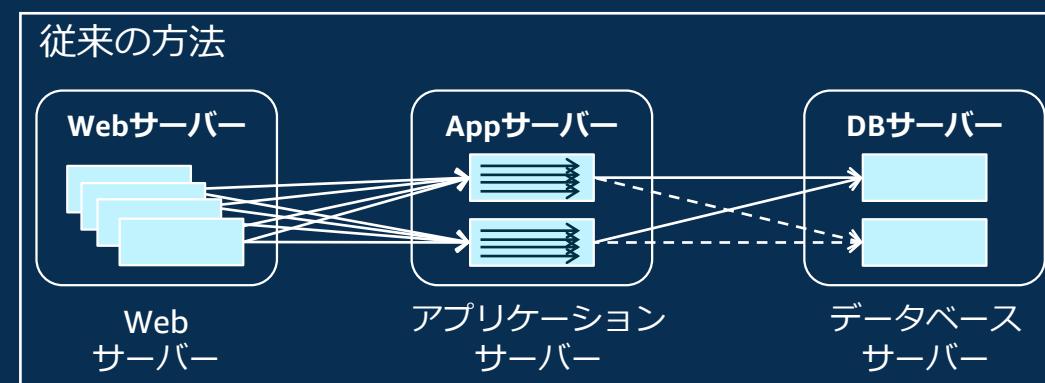
AWSサービスの本質はビルディングブロック
サービスを組み合わせることで、
素早くアプリケーション構築が可能



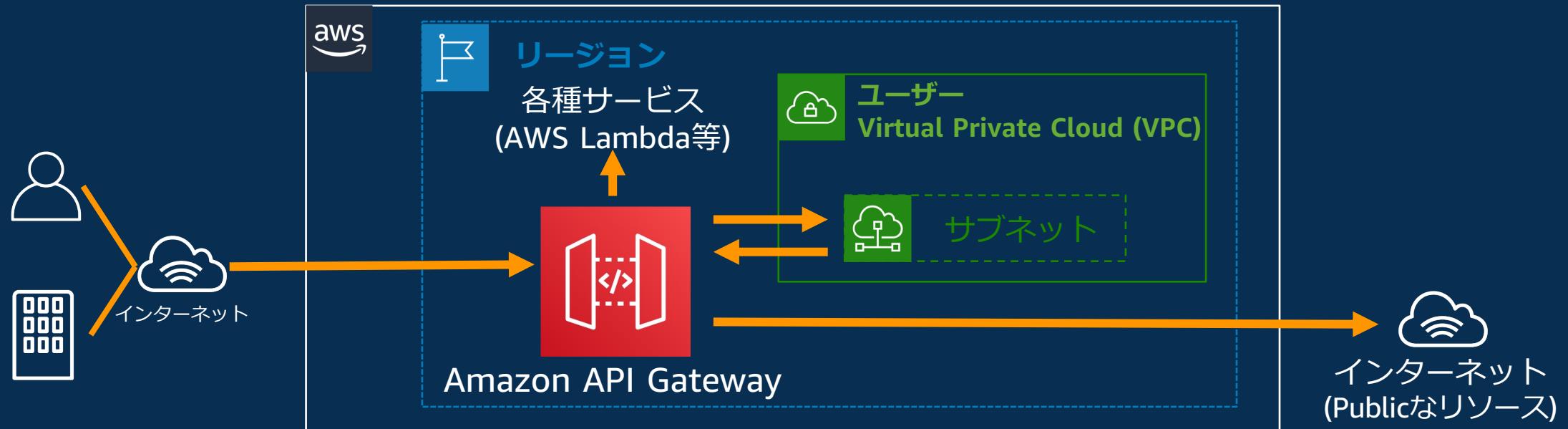
サーバーレスのよくある使い方

1. Web アプリケーションの実行環境として
2. データ加工処理の一環として

Web アプリケーション の実行環境として



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

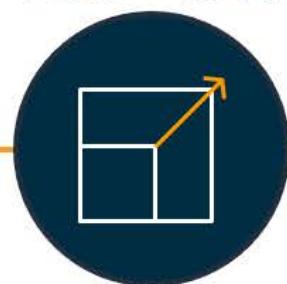
使用量に応じた課金

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



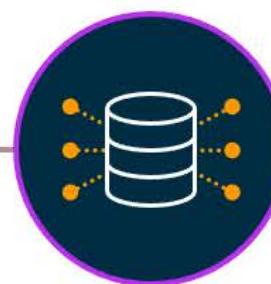
規模に関係なく、一貫した
数ミリ秒台の応答時間実現
事実上無制限のスループットで
アプリケーションを構築可能

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



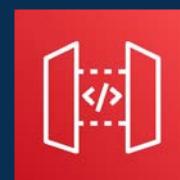
複数のAWSリージョンにまたがりテーブルを
簡単に複製することで、ローカルデータに
素早くアクセスできるグローバルアプリケー
ションを構築



サーバーレスのよくある使い方

1. Web アプリケーションの実行環境として

前捌き プログラム実行 データ保持



Amazon API Gateway



AWS Lambda



Amazon DynamoDB

2. データ加工処理の一環として

データ加工処理の一環として

処理前後のデータを保持

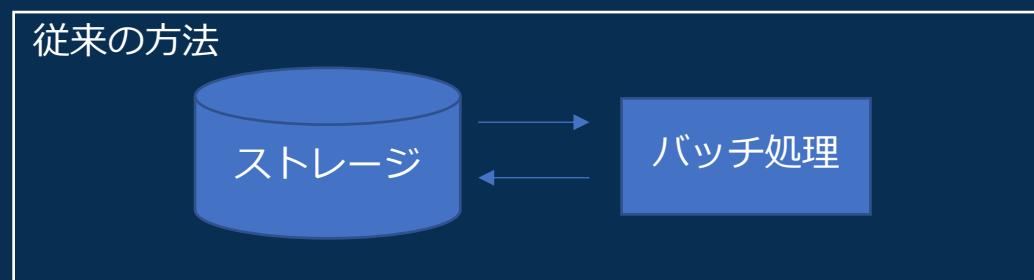


Amazon S3

データの加工



AWS Lambda

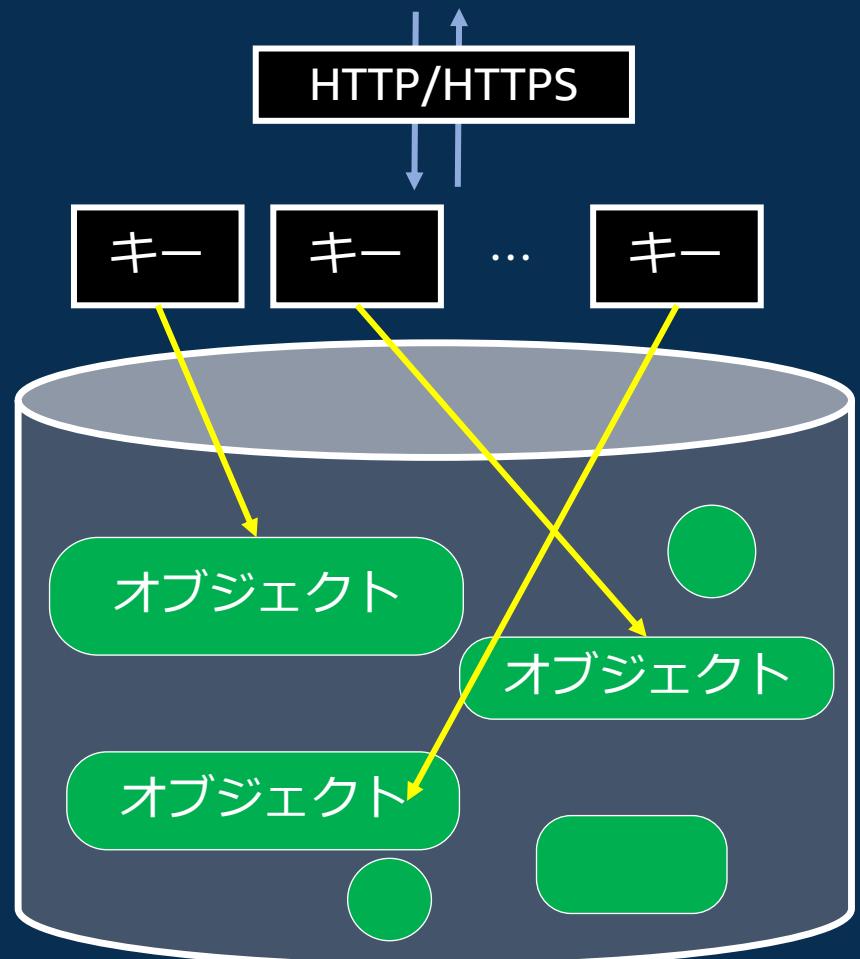
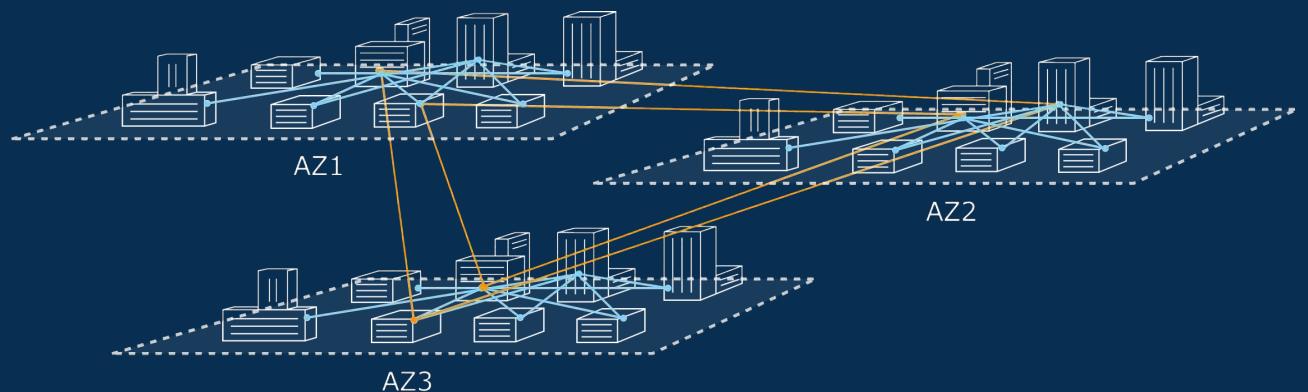


Amazon S3

AWSが提供するオブジェクトストレージ

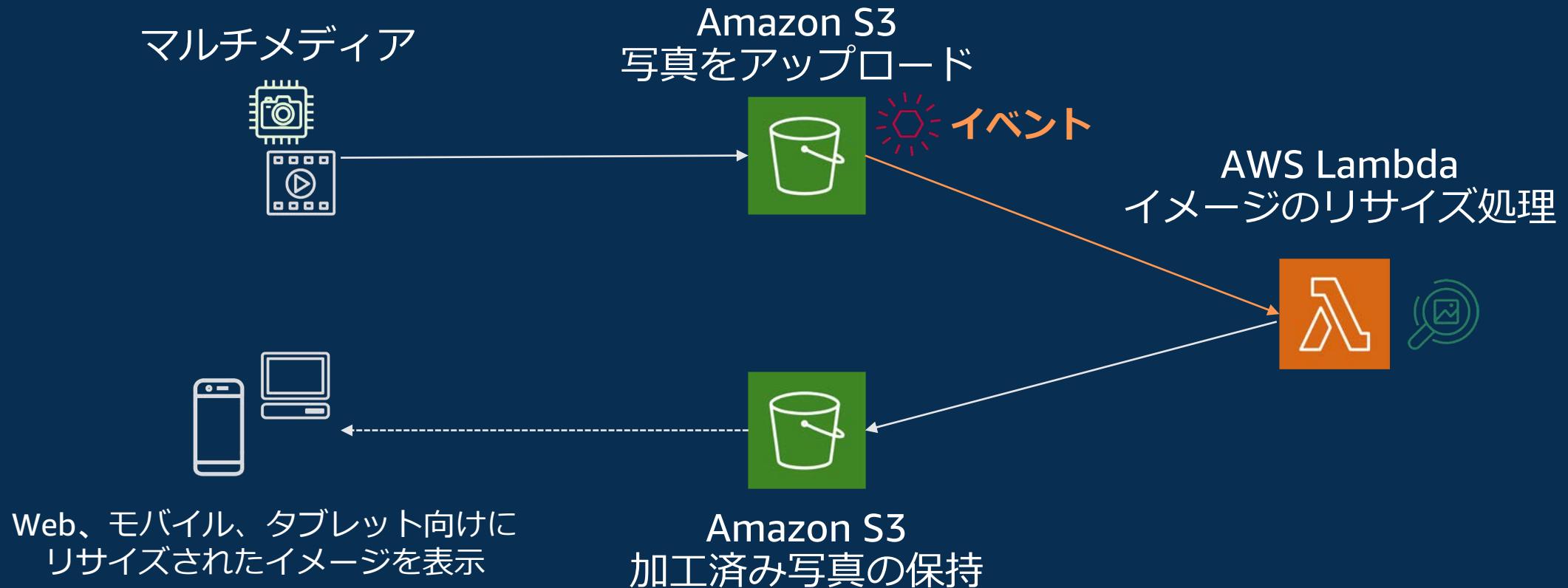
自動的にオブジェクトを冗長化

低成本に大容量のデータを保存



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: サムネイル画像の作成



サーバーレスのよくある使い方

1. Web アプリケーションの実行環境として



2. データ加工処理の一環として

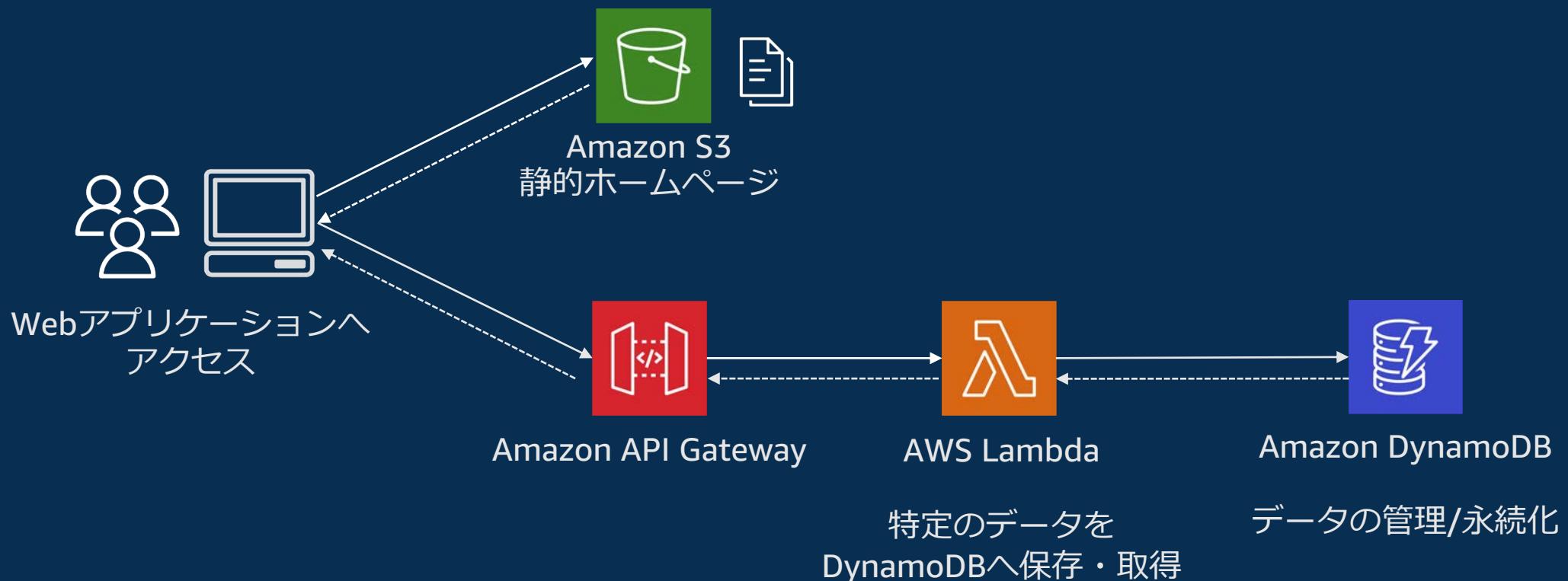


サーバーレスの始め方



サーバーレスなWebアプリケーション

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

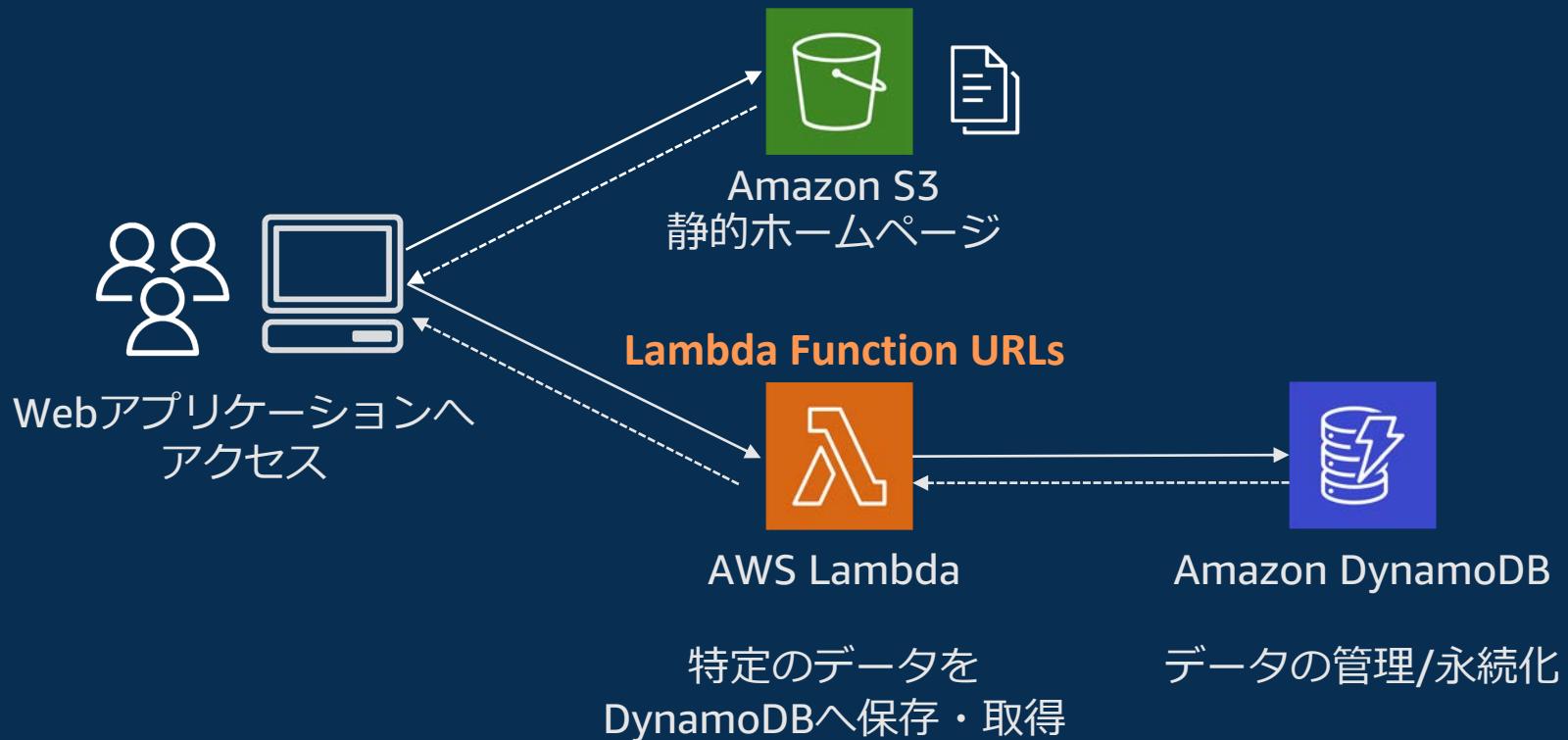


AWS Lambda Function URLs

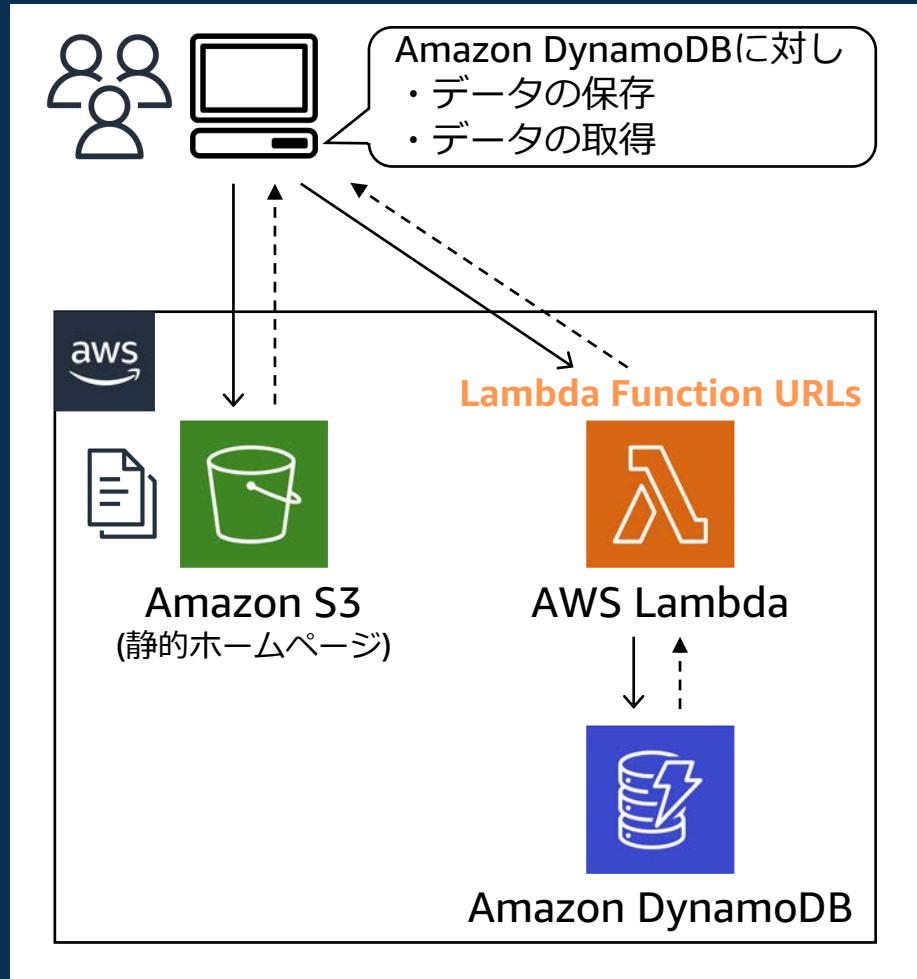
HTTP リクエストで AWS Lambda 関数を同期的に呼び出すシンプルな方法

Function URLs は任意の Lambda 関数に HTTPS エンドポイントを追加

- IAM 認証をサポート



デモ：サーバーレスなWebアプリケーション



Amazon S3

- 静的ホームページ (Web API の入口)

AWS Lambda

- プログラムの実行
- Function URLs (HTTPS エンドポイント)

Amazon DynamoDB

- データの管理/永続化

demo0.p... 65% コンソールのホーム | us-east-1 us-east-1.console.aws.amazon.com/console/home?region=us-east-1 aws サービス 検索 [Alt+S] バージニア北部 WSParticipantRole/Participant @ 7897-8078-8176 + i

コンソールのホーム 情報

デフォルトレイアウトにリセット + ウィジェットを追加

最近アクセスしたサービス 情報

Lambda, DynamoDB, S3, Security Hub, Simple Notification Service, Amazon EventBridge

すべてのサービスを表示

AWS へようこそ :

AWS の開始方法 AWS を最大限に活用するために基礎を学び、有益な情報を見つけましょう。

トレーニングと認定 AWS のエキスパートから学び、スキルと知識を深めましょう。

AWS の最新情報 AWS Health に移動

AWS Health 情報

未解決の問題 0 過去 7 日間

スケジュールされている変更 0 近日および過去 7 日間

その他の通知 0 過去 7 日間

コストと使用状況 情報

今月のコスト **<\$0.01** 詳細

予測される月末のコスト

ソリューションを構築 情報

シンプルなウィザードと自動化されたワークフローで構築を開始できます。

仮想マシンを起動

デモ内容のご説明

1. Lambda 関数の作成
2. Lambda/DynamoDBの設定
3. Lambda Function URLs の設定
4. S3の設定紹介/Webアプリのテスト

Amazon DynamoDBに対し
・データの保存
・データの取得

Amazon S3 (静的ホームページ)

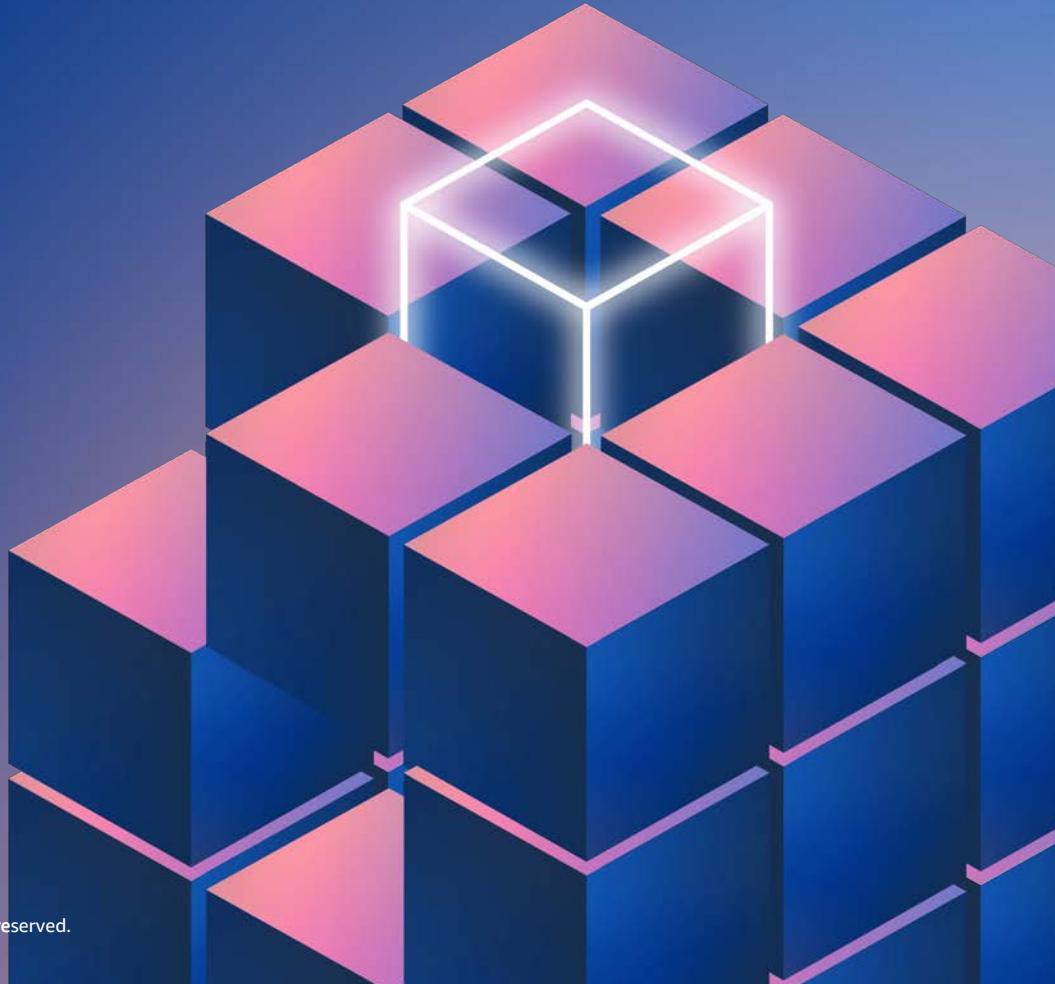
AWS Lambda

Amazon DynamoDB

Lambda Function URLs

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
- ・サーバーレスの良くある使い方
 - ・Web アプリケーションの実行環境として
 - ・データ加工処理の一部として
- ・サーバーレスの始め方
 - ・サーバーレスな Web アプリケーション



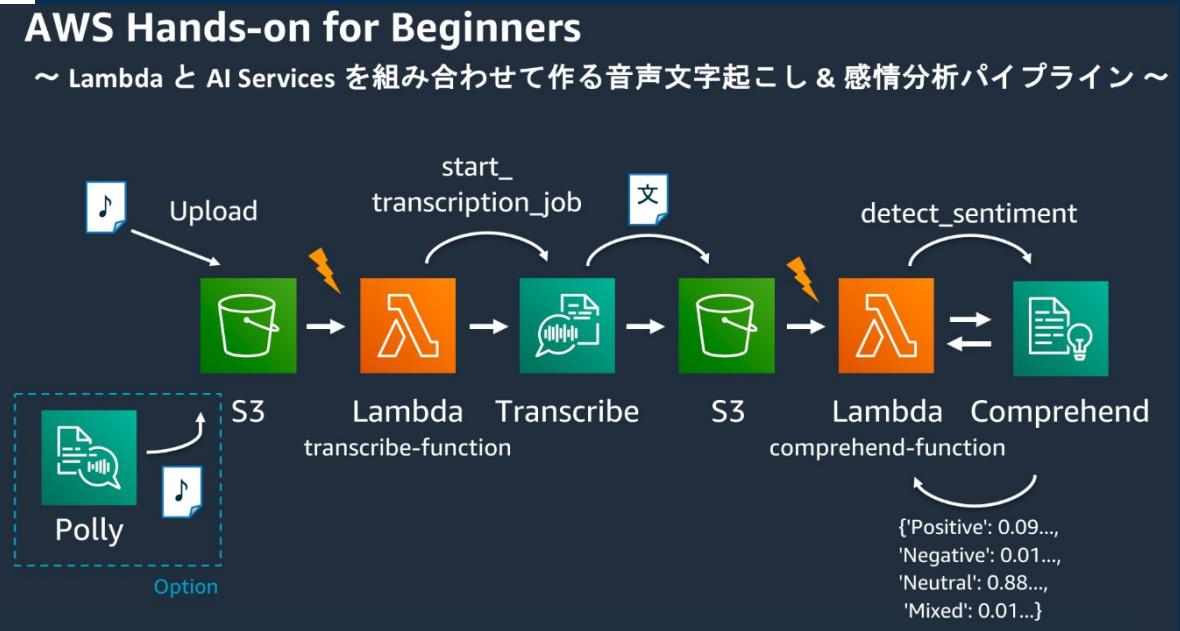
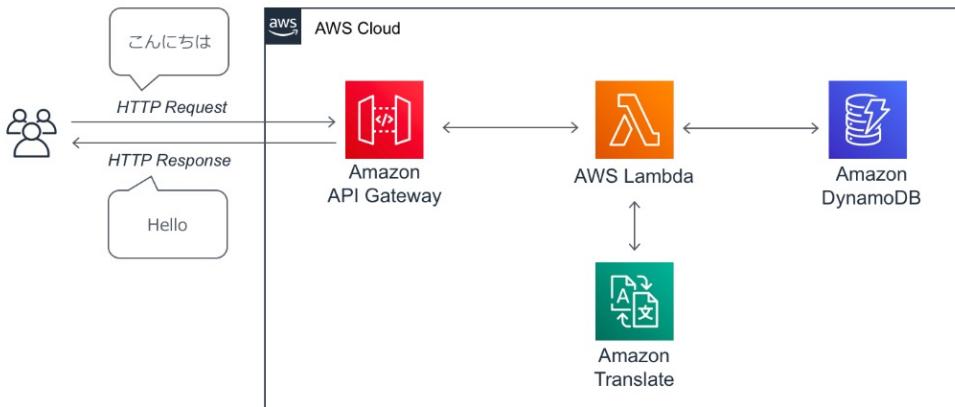
今後に向けて

ハンズオンをやってみよう！

<https://aws.amazon.com/jp/events/aws-event-resource/hands-on/>



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～



参考：形で考えるサーバーレス設計

<https://aws.amazon.com/jp/serverless/patterns/serverless-pattern/>



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
も含みます

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



aws サーバーレスの始め方 (1/2) 2023.03 Edition

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。

Dynamic API Calls over HTTPS

go.aws/2Sh6l5M

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きます。

AWS Cloud

Amazon API Gateway

AWS Lambda

Amazon DynamoDB

Amazon Translate

「動的 Web / モバイルバックエンド」パターン

「機能API」パターン

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。

Start

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!



AWS TRAINING & CERTIFICATION

600+ ある AWS Skill Builder の無料デジタルコースで学ぼう

30 以上の AWS ソリューションの中から、自分にもっとも関係のあるクラウドスキルとサービスにフォーカスし、自習用のデジタル学習プランとランプアップガイドで学ぶことができます。

自分に合ったスキルアップ方法で学ぼう

[EXPLORE.SKILLBUILDER.AWS »](#)



あなたのクラウドスキルを AWS 認定で証明しよう

業界で認められた資格を取得して、スキルアップの一歩を踏み出しましょう。AWS Certified の取得方法と、準備に役立つ AWS のリソースをご覗ください。

[受験準備のためのリソースにアクセスしよう »](#)



AWS Builders Online Series に ご参加いただきありがとうございます

楽しんでいただけましたか? ぜひアンケートにご協力ください。
本日のイベントに関するご意見/ご感想や今後のイベントについての
ご希望や改善のご提案などがございましたら、ぜひお聞かせください。

-  aws-apj-marketing@amazon.com
-  twitter.com/awscloud_jp
-  facebook.com/600986860012140
-  <https://www.youtube.com/user/AmazonWebServicesJP>
-  <https://www.linkedin.com/showcase/aws-careers/>
-  twitch.tv/aws

