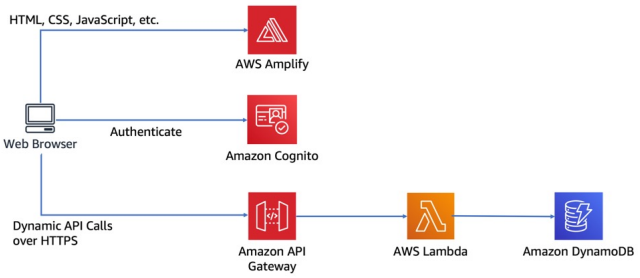
サーバーレスの始め方（1/2）

2023.09 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!

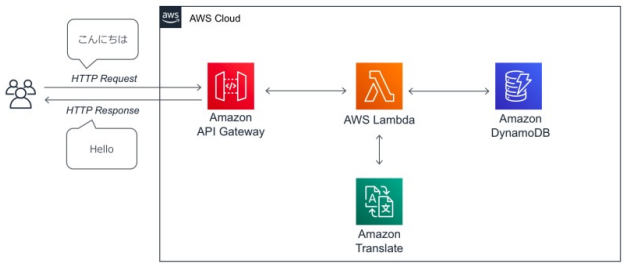
最初のサーバーレスWebアプリ：手順に沿えば、多くのサーバーレスサービスに触れながら、Webアプリが作れます。



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。



bit.ly/3PcEve3

「機能API」パターン

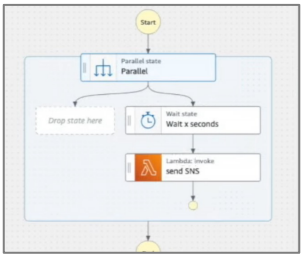
シンプルなサーバーレスアプリの
次は処理フローにチャレンジ

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。

【ハンズオン】
- ローコードで作るワークフロー

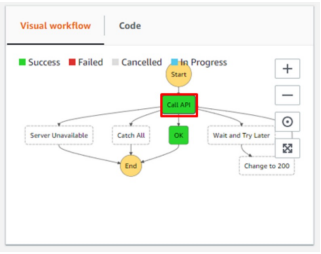
bit.ly/3paQP2r



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。

【ハンズオン】
- サーバーレスアプリでエラーに対処する

go.aws/3J0PSE7



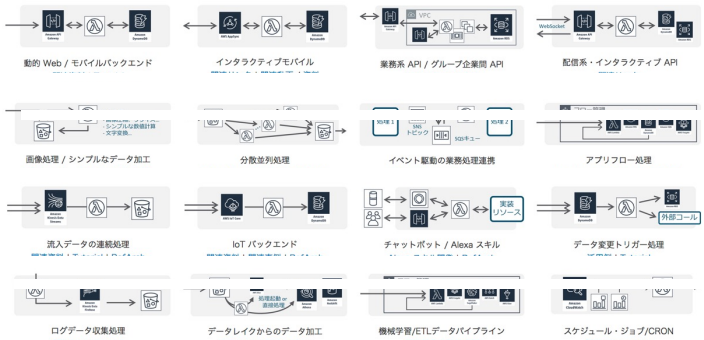
利用シーン/目的に応じた
サーバーレスの基本形を知る!

3 やりたいこと駆動で基本構造を選択

サーバーレスは組み合わせの形で識別できます。よく使われる基本的なユースケースパターンとして何があり、どんなサービスの組み合わせなのかを知っておけば、実際のプロジェクト時に対象サービスに学習範囲を限定でき、それでいながらアプリ設計スピード向上、品質向上が期待できます。

サーバーレスパターン

go.aws/37WpOF2

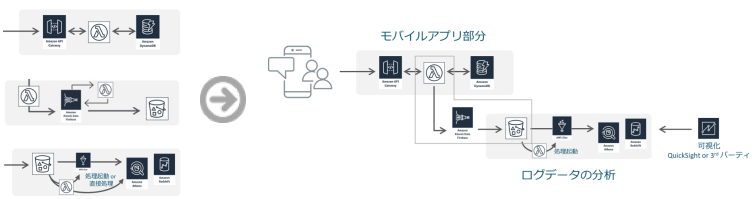


基本ケースの組み合わせという考え方

実績ある 16の基本ケースをさらに組み合わせてアプリケーション全体を設計することもできます。

パターン組み合わせ設計

go.aws/2sNwK8f



サーバーレスの始め方（2/2）

2023.09 Edition

4 開発工程を円滑にするフレームワーク

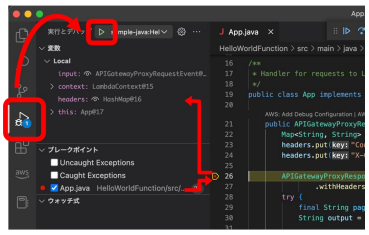
【ハンズオン】初めての SAM　bit.ly/3o6Fm2w

サーバーレスアプリの開発・保守に便利なフレームワーク Serverless Application Model（SAM）。クラウド上の開発環境 AWS Cloud9 を使って、SAM をイチから体験できます。

5 サーバーレスアプリ開発をローカル PC で

ローカル開発環境の整備　bit.ly/3SEBy76

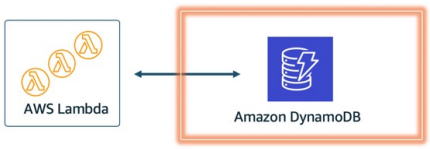
使い慣れた自身の開発環境をサーバーレス対応へ。ローカル PC 環境でサーバーレス開発やデバッグをできるようにしましょう。



6 データソースの選択

スケールする NoSQL 型

大量のデータ入力、ID/キーによるデータ取得が中心の API 型処理を水平方向にスケールアウト。加えて、マネージドで可用性設計や DB ver. up 作業を軽減。

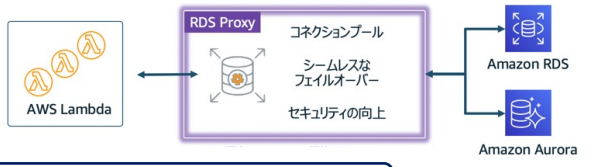


考慮点
- アプリから見た I/F
  - SQL? API?
- データ設計
  - RDB型? Key-Value?
- DBの可用性設計/スケーラビリティ

DynamoDB 設計ベストプラクティス　go.aws/2Gb2CGR

使い慣れたリレーショナルDB

アンチパターンと言われた『サーバーレス x RDB』もいまや普通の選択肢。馴染みある RDB のデータ設計とサーバーレスアプリは共存します。



RDBと共に使う → RDS Proxy　go.aws/2RFqasQ

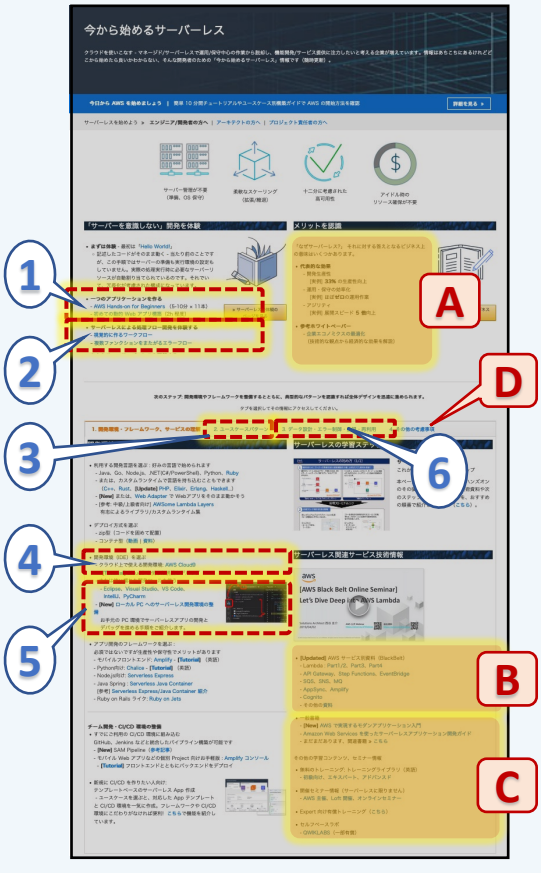
中級者向け

追加のワンステップ!

サーバーレスアプリケーションの設計をすすめるにあたって考慮しておくべき性質**「べき等性」**について理解しましょう

builders.flash
go.aws/32jquGK

これから続く開発工程で必要な情報はココから!



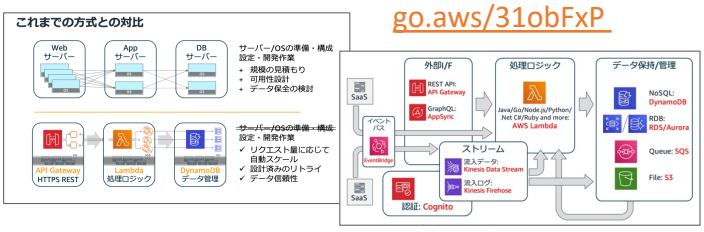
「今から始めるサーバーレス」サイトの歩き方

go.aws/2RRIOPq

本資料で紹介する各ステップの内容は、本サイトの右図の場所に配置されています。以下のような追加情報（A-D）もあります。

A サーバーレスの価値、おさらい、全体観

go.aws/31obFxP



B サービス別資料へのリンク（BlackBelt 資料）

C 一般書籍 トレーニング情報

D Tips・チューニング その他のチェック項目