



ブラウザで開発する サーバーレスアプリケーション

下川 賢介

アマゾン ウェブ サービス ジャパン合同会社
サーバーレススペシャリスト ソリューションアーキテクト

自己紹介

下川 賢介 / Kensuke Shimokawa

アマゾンウェブサービスジャパン合同会社

コンピュータ事業本部

サーバーレス スペシャリスト ソリューションアーキテクト

サーバーレスの活用支援を担当



[_kensh](#)



https://speakerdeck.com/_kensh



https://qiita.com/_kensh

本セッションで取り上げるAWSサービス・機能

本セッションで取り上げる AWS のサービス・機能は以下の通りです。詳細は、AWS クラウドサービス活用資料集 または AWS ドキュメント よりご確認ください。

- AWS Step Functions
- Amazon EventBridge
- AWS SAM
- Amazon API Gateway
- AWS Lambda
- AWS Application Composer

AWS クラウドサービス活用資料集

アマゾン ウェブ サービスの公式イベントのアーカイブおよびオンデマンドコンテンツの動画や資料がご利用いただけます。

[AWS イベントお申込 »](#)

[AWS 初心者向け »](#)

[サービス別資料 »](#)

[ハンズオン資料 »](#)

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



本セッションの対象になる方

- サーバーレスアプリケーションの簡単な始め方を探っておられる方
- ワークフローの手軽な構築に興味がある方
- チームコラボレーションを活発にしてイテレーションを高速化したい方

サーバーレスの設計方針
のチーム共有って
どうするの？



ワークフロー
開発を
効率化したい

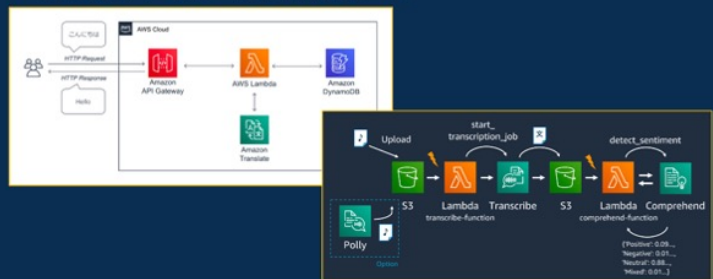


関連セッションについて

- 本セッションは、3つの関連セッションの一つとなっています。
- 本セッションでは、AWS Lambda/Amazon API Gateway/AWS SAM/Amazon DynamoDB などのサービスの説明は行いません。サービスの基礎について知りたい場合、他の関連セッションをご覧ください。

本セッションで学べること

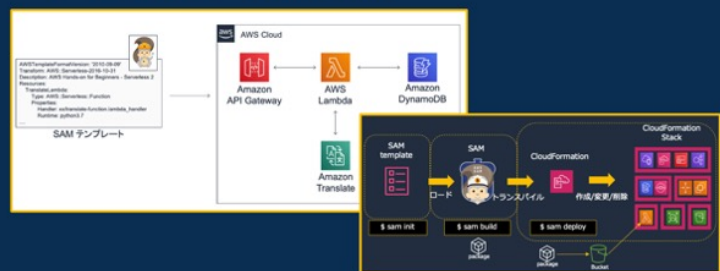
- サーバーレスとは何なのか
- 具体的なユースケースと実装イメージ



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

本セッションで学べること

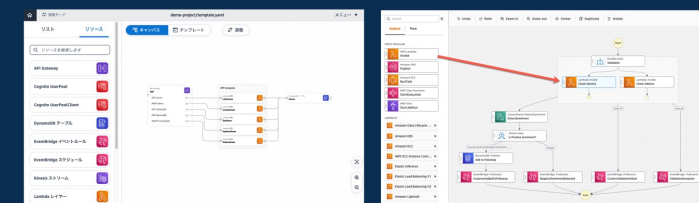
- AWS SAMによるサーバーレス開発の流れ
- サーバーレスのローカル開発の方法



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

本セッションで学べること

- ビジュアルツールによるサーバーレス開発
- サーバーレスワークフロー構築の方法

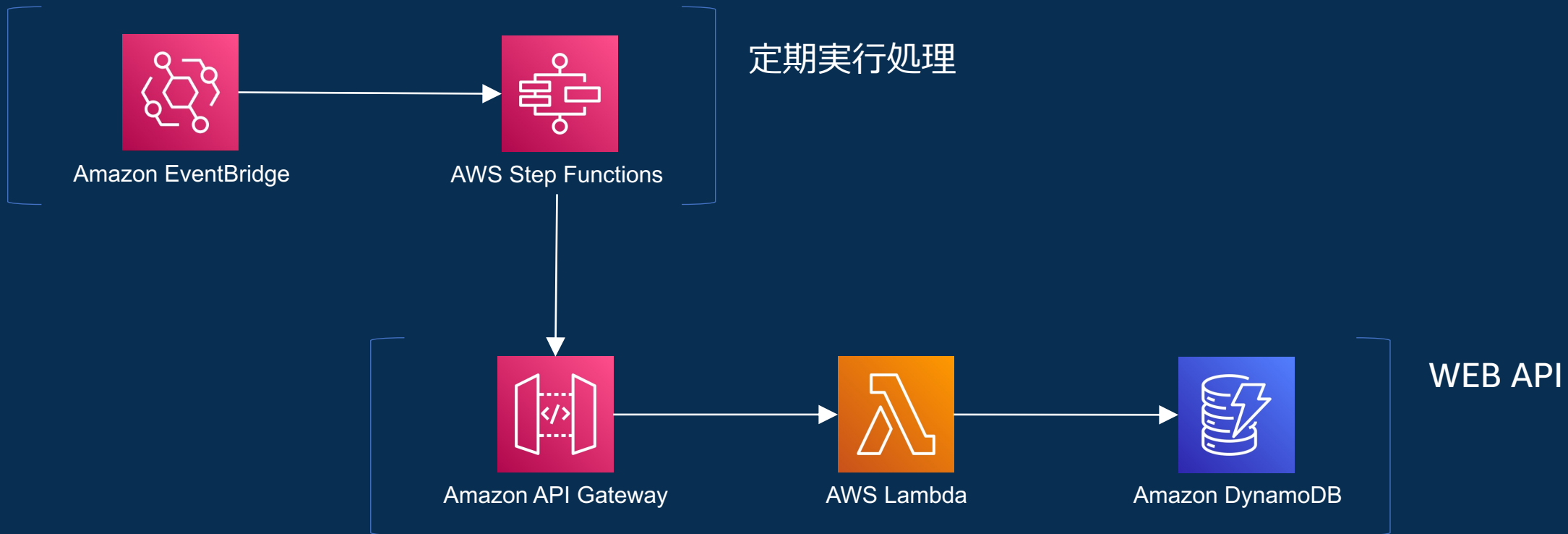


ブラウザで開発する
サーバーレスアプリケーション

アジェンダ

- サーバーレスアプリケーションを視覚的に設計する
 - Demo
- サーバーレスワークフローを視覚的に実装する
 - Demo
- まとめ

今回作成するアーキテクチャ



サーバーレスアプリケーションを 視覚的に設計する

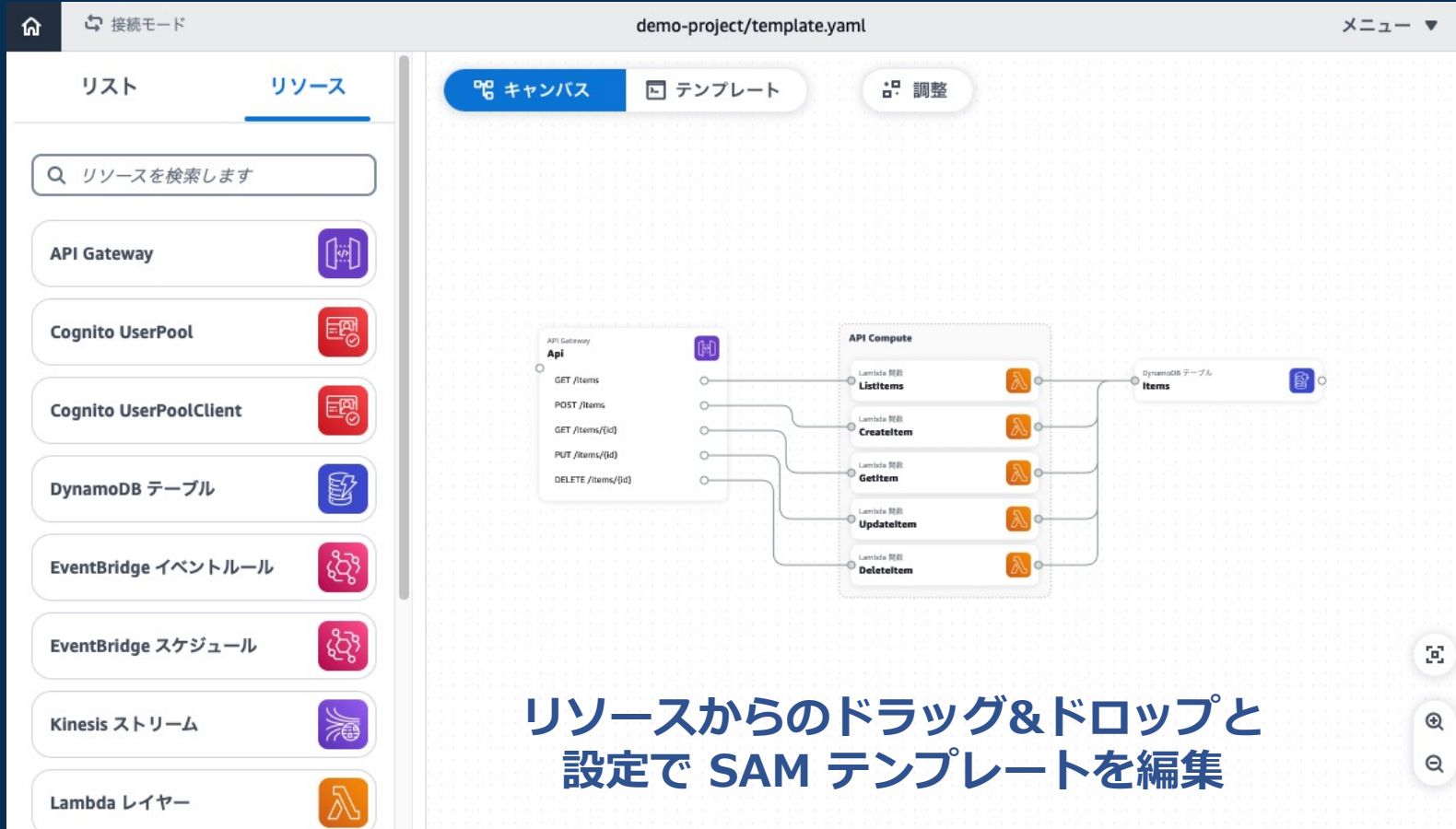


初学者におけるサーバーレス開発

- 複数の AWS サービスの構成パターンの理解が必要
- サービスをどのように連携させるのか、良いデフォルト設定の理解が必要
- デプロイできるように、Infrastructure as Code (IaC) への理解が必要
- アーキテクチャをどのように構成したかをチームで共有、可視化したい

AWS Application Composer

AWS 管理コンソール UI で IaC コード(AWS SAM テンプレートファイル) を作成するサービス



The screenshot displays the AWS Application Composer interface. On the left, a sidebar lists various AWS resources such as API Gateway, Cognito UserPool, and Lambda. The main area shows a visual representation of a SAM template named 'demo-project/template.yaml'. It features a 'キャンバス' (Canvas) and 'テンプレート' (Template) tabs. The diagram illustrates an 'API Gateway' resource connected to an 'API Compute' resource, which is further connected to a 'DynamoDB テーブル' (DynamoDB Table). The API Compute resource is composed of several Lambda functions: ListItems, CreateItem, GetItem, UpdateItem, and DeleteItem. A search bar at the top left of the sidebar contains the text 'リソースを検索します' (Search for resources).

リソースからのドラッグ&ドロップと
設定で SAM テンプレートを編集

AWS SAM テンプレートを編集
(既存ファイルの読み込みも可能)

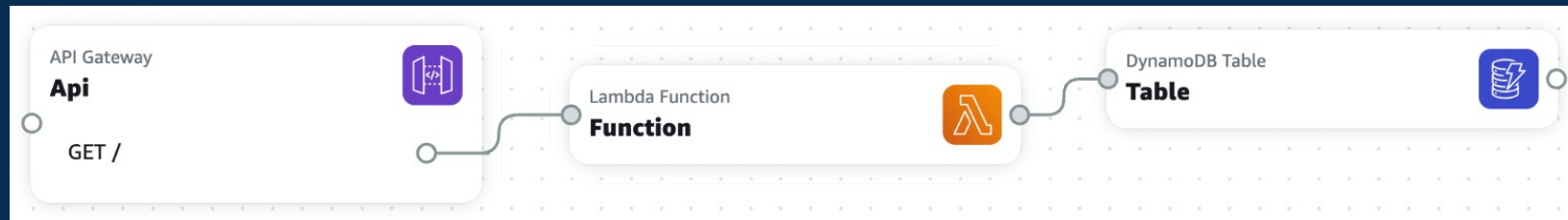
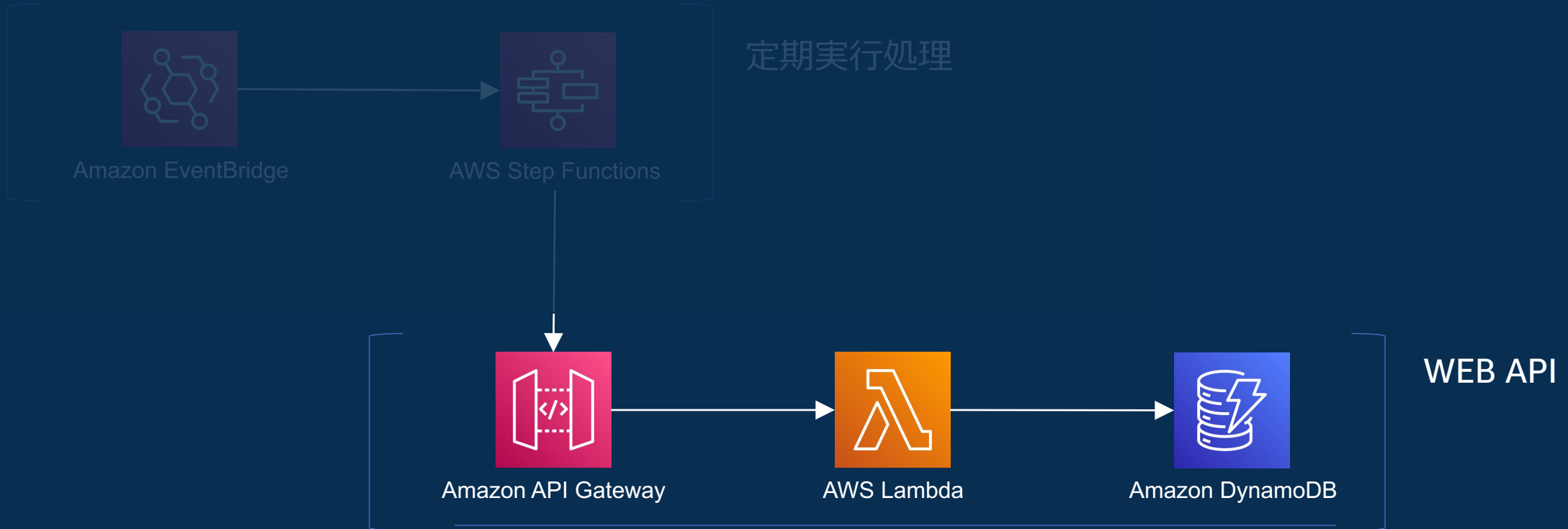
サーバーレスサービスに対応

ローカルファイルシステムに接続することで、ローカルのファイルを編集可能

- 任意の統合開発環境(IDE)
SAM CLI と統合された
開発フローを実現

(*) AWS SAM = AWS Serverless Application Model

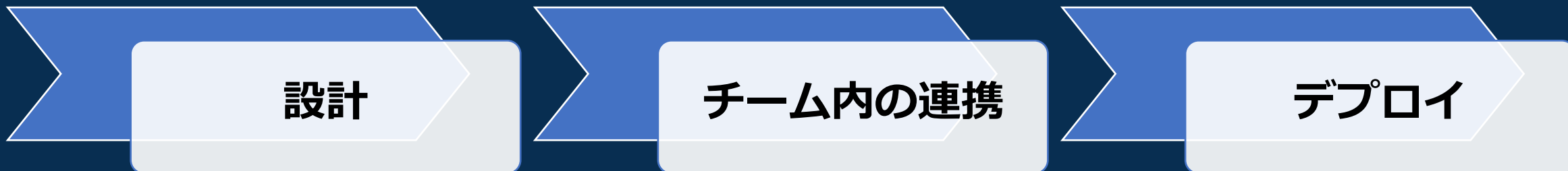
今回作成するアーキテクチャ (再掲)



Demo



サーバーレスの視覚的な設計のまとめ



視覚的にアーキテクチャを構成

- ドラッグ&ドロップ
- 直感的な構成
- 共通のタスクを簡素化
 - IAM ポリシー
 - デフォルト設定

チームコラボレーションとイテレーションの加速

- 何百行もの設定に目を通すことなく、チームコラボレーションを促進
- IaC とアーキテクチャのビジュアル表現を常に同期

IaC によるデプロイ

- 組み込みのベストプラクティスが考慮されたテンプレートによるデプロイ

サーバーレスワークフローを 視覚的に実装する

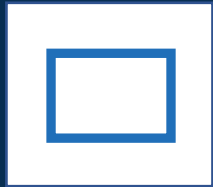


初学者におけるワークフロー実装

- 複数の AWS サービスの組み合わせによるエラーパターンの複雑化
- プログラミングに不慣れな担当者による実装が困難
- 少しずつ動かしながらの段階的な構築をしたい
- ワークフローをどのように構成したかをチームで共有、可視化したい

AWS Step Functions

AWS Step Functions は、ワークフローとしてサービスを繋ぎ合わせ アプリケーションを構築するためのサーバーレスなオーケストレーションサービス



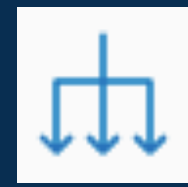
Task

処理の実行



Choice

処理の分岐



Parallel

並列実行



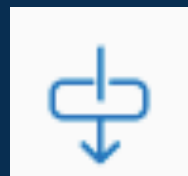
Map

配列要素の反復処理



Wait

指定された時間待機



Pass

入力を次の出力にパススルー



Success

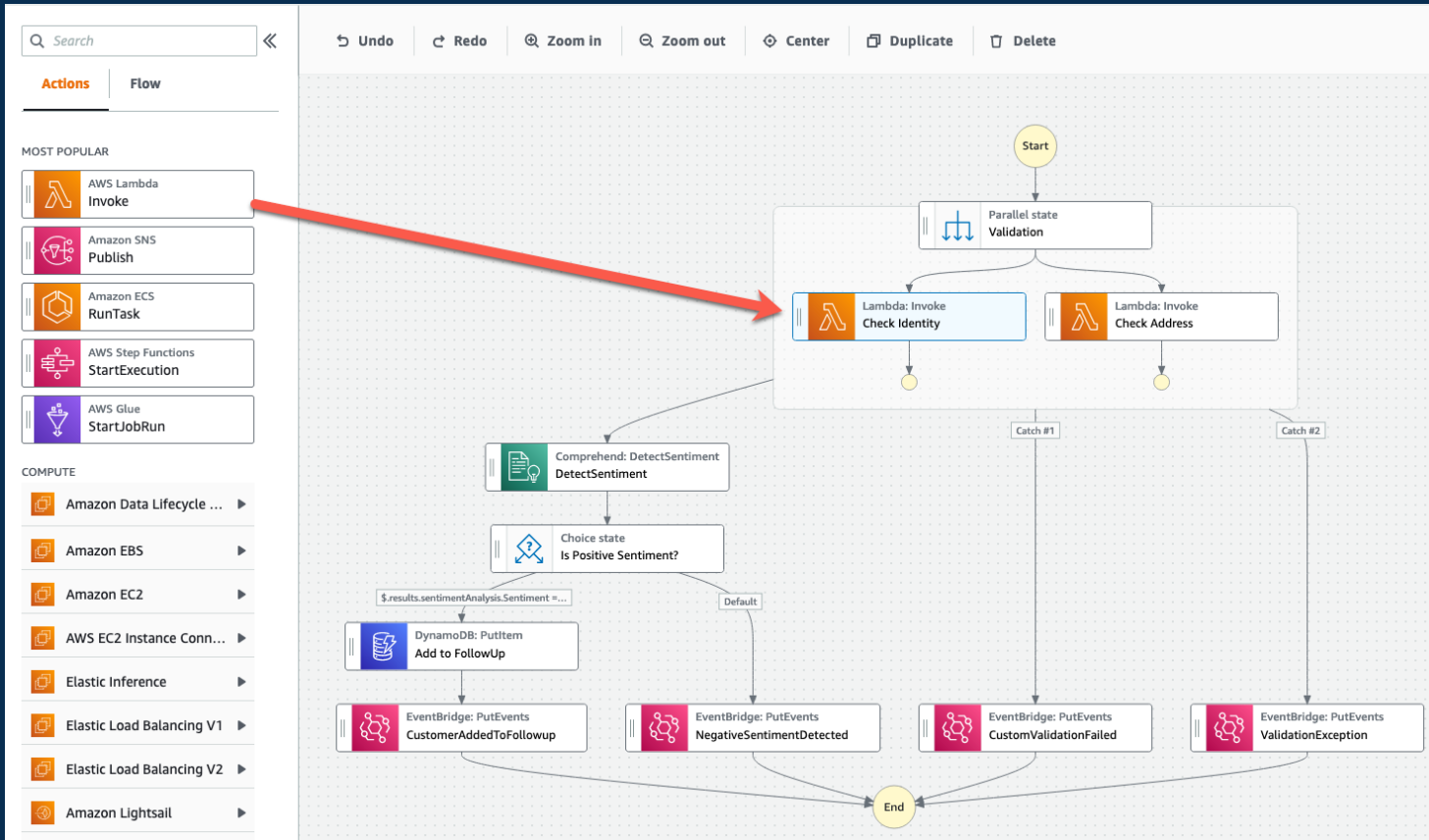
成功で実行を終了



Fail

失敗で実行を終了

AWS Step Functions Workflow Studio



サーバーレスワークフローの構築

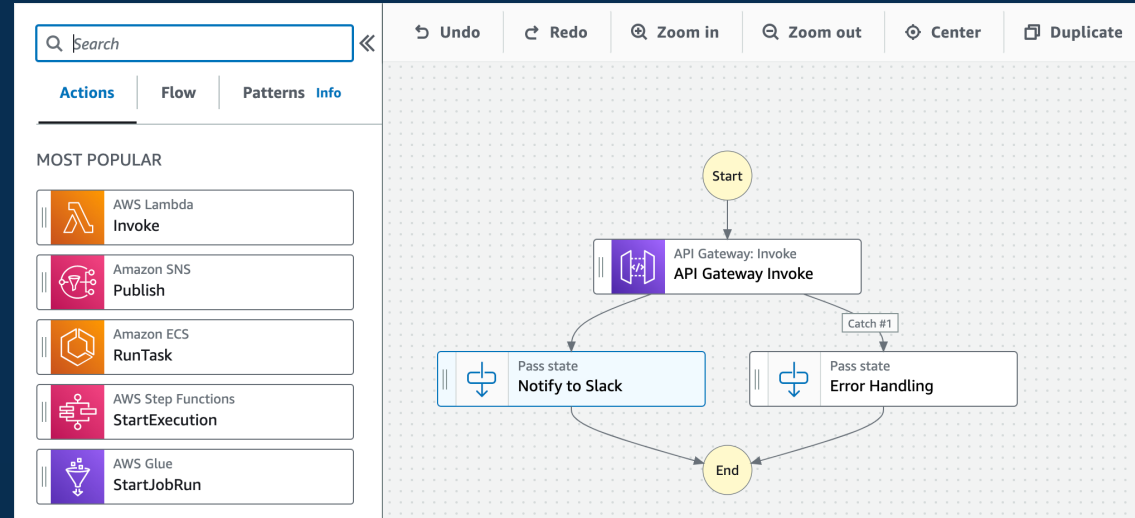
分散型アプリケーションの構築

ビジネス プロセスの可視化

オーケストレーションを最小限のコードで
実装可能

ビジュアルでドラッグ&ドロップによる
ワークフロー構築

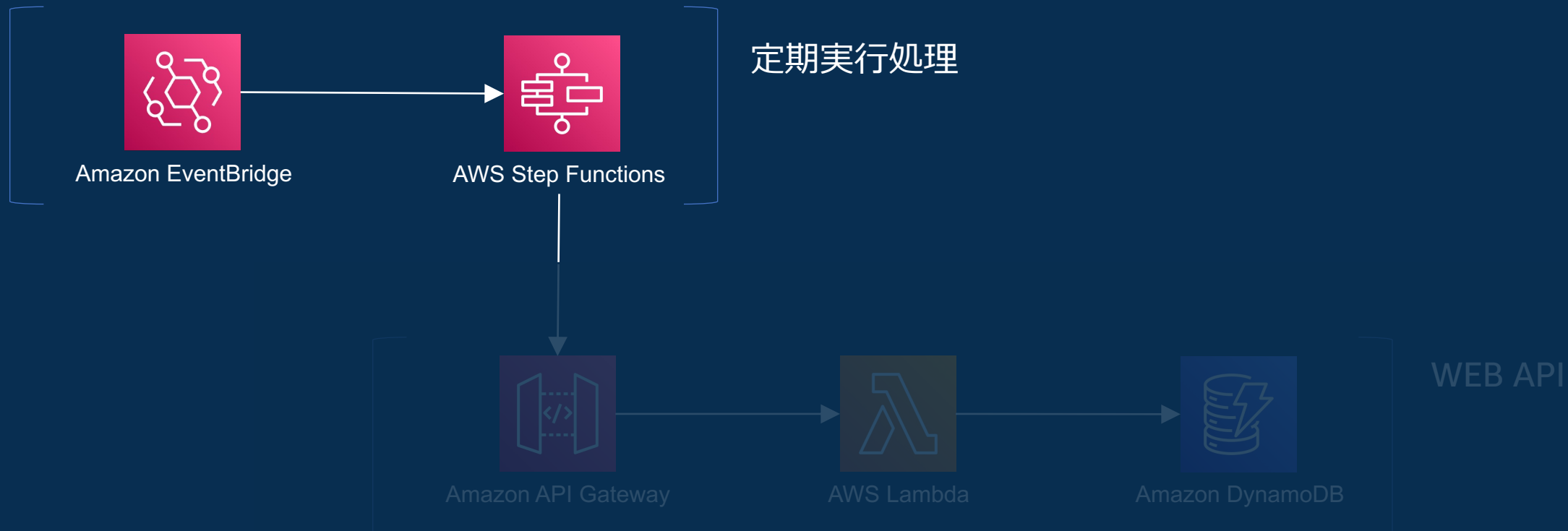
今回作成するアーキテクチャ (再掲)



Demo



今回作成するアーキテクチャ (再掲)



Amazon EventBridge Scheduler



AWS マネージドなサーバーレススケジューラーで、定期的なタスクの実行、管理に利用可能

270 以上のサービス、6000 以上の API アクションを呼び出すことができる

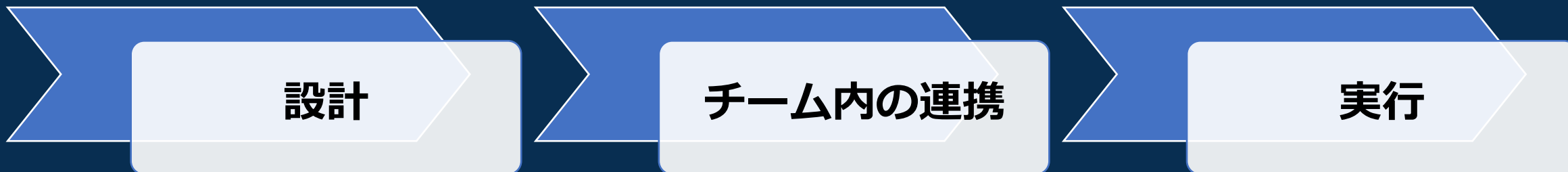
定期的な実行に加えて、一回限りの実行や、UTC 以外のタイムゾーン、サマータイムをサポート

1か月あたり 14,000,000 回までは無料

Demo



ワークフローの視覚的な実装のまとめ



視覚的にワークフローを構成

- ドラッグ&ドロップでの構成
- プログラミング部分を最小化
- 直感的な構成

チームコラボレーション

- ワークフローデザインの可視化

視覚的な実行と イテレーションの加速

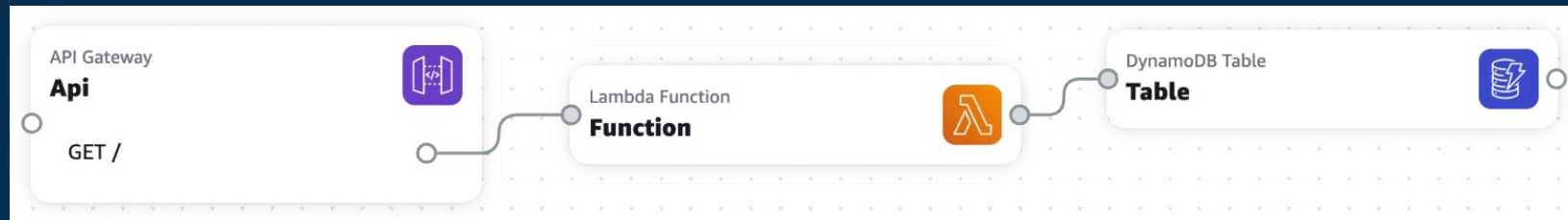
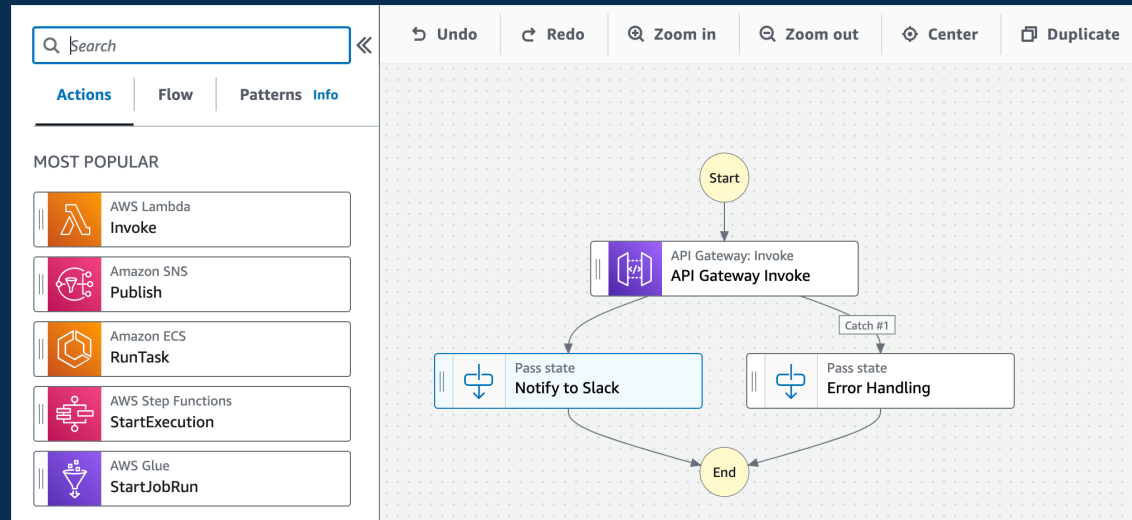
- ワークフロー実行結果の可視化
- 入出力情報の可視化
- 容易な再実行

まとめ



今回作成したアーキテクチャ

AWS Step Functions Workflow Studio



まとめ

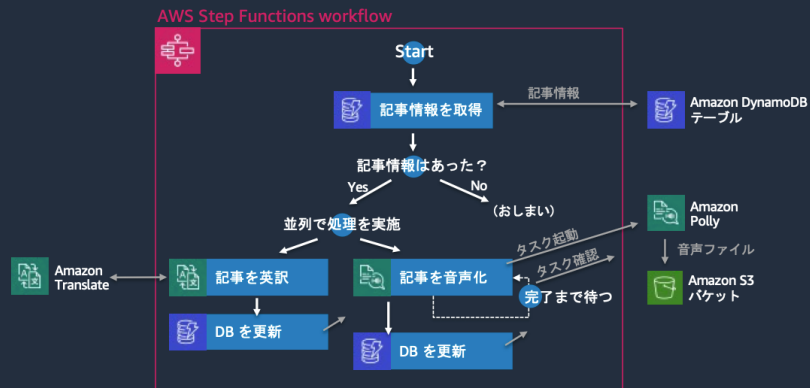
- これから サーバーレス を始めてみようとするときに Application Composer は直感的に IaC (AWS SAM) テンプレートを作成できる
- ワークフローを作る際に、Step Functions Workflow Studio を利用すると、ドラッグ&ドロップで、ワークフローの作成と編集ができる
- 可視化サービスが充実してきているのでチーム連携のコラボレーション用途としても利用しやすくなった

今後に向けて

本セッションでご紹介した内容と類似したものを
具体的な画面とデモを見ながら進めることができるハンズオンです

AWS Hands-on for Beginners - AWS Step Functions 入門

～ビジュアルツールを使ってローコードにワークフローを作成する～



- AWS Step Functions の使い方を 1.5時間でハンズオン形式で学ぶコンテンツ
- 日本語の記事（テキスト）を英訳化 & 音声化するサーバーレスなワークフローを作成
- AWS に関する前提知識やプログラミングのご経験がなくてもご実施いただけます



サーバーレス パターン

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

代表的な実装ブロックを形（パーツ）として理解

組み合わせによってアプリケーション全体のたたき台をイメージ

このような考え方により応用性と開発スピードを向上



代表的な適用シーン/ユースケースと実装形

<p>動的 Web / モバイルバックエンド 関連資料 Tutorial Tutorial (中級編) テンプレートから始める → こちら 見積もり例</p>	<p>リアルタイムモバイルアプリケーション AWS マンガ ワークショップ Solution リンク 見積もり例</p>	<p>業務系 API / グループ企業間 API Tutorials Private API 記事 関連事例 OpenAPI の利用</p>	<p>Push 配信系・インタラクティブ API 関連リンク AppRepository サンプル 解説動画 (英語)</p>
<p>画像処理 / シンプルなデータ加工 Tutorial 関連事例 Solution リンク テンプレートから始める → こちら 見積もり例</p>	<p>分散並列処理 (like MapReduce) 関連事例1 関連事例2 [New] Distributed Map による CSVファイル処理 (Tutorial Solution)</p>	<p>イベント駆動の業務処理連携 Tutorial イベント駆動アーキテクチャ FIFO デモ (順序保証、重複排除) テンプレートから始める → こちら</p>	<p>アプリケーションフロー処理 Tutorial (Workflow / エラー処理) 短時間・高速フロー処理向け Express ローコードの処理フロー Tutorial</p>
<p>流入データの連続処理 関連資料 RefArch Tutorial1 Tutorial2</p>	<p>IoT バックエンド 関連資料 関連事例 RefArch 関連 Solution1 Solution2 見積もり例</p>	<p>チャットボット / Alexa スキル Alexa スキル開発 ワークショップ Solution Solution2</p>	<p>データ変更トリガー処理 活用例 Tutorial Streaming におけるエラー処理</p>
<p>ログデータ収集処理 関連事例 Solution1 SIEM Solution データ変換ブループリント 見積もり例</p>	<p>データレイク周りのデータ加工 Solution リンク DB Loader or より包括的なソリューション</p>	<p>機械学習/ETLデータパイプライン 関連記事1 関連記事2 関連動画 Managed Airflow</p>	<p>スケジュール・ジョブ / SaaS イベント 関連 Doc 関連 Tutorial [New] EventBridge Scheduler テンプレートから始める → こちら</p>



サーバーレス自己学習ガイド



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

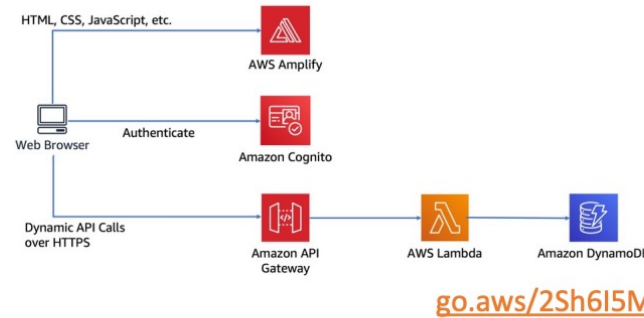
- ハンズオンから始まる6つのステップ
 - Hands-on for Beginnersも包含
- ご自分のペースで主要ポイントを学べます
- 開発作業の諸処で役立つサーバーレス技術情報サイトもご紹介



サーバーレスの始め方 (1/2)

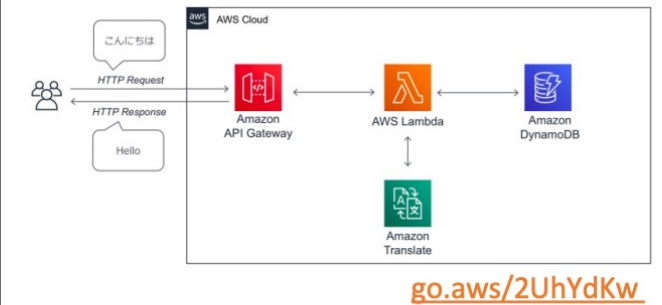
1 最初のトライ: サーバーの準備も実行環境構築も不要、いきなりアプリ開発を体験!

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。



「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きます。

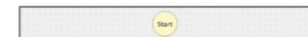


「機能API」パターン

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!

KENSUKE SHIMOKAWA



[_kensh](#)



https://speakerdeck.com/_kensh



https://qiita.com/_kensh