



[AWS Hands-on for Beginners - Serverless #3] AWS Lambda と AWS 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

アマゾン ウェブ サービス ジャパン株式会社
Solutions Architect
金澤 圭 / Kei Kanazawa
(収録日: 2020/2/6)



自己紹介

□ 名前

金澤 圭 Kei Kanazawa



@ketancho



□ ロール

テクニカルソリューションアーキテクト

□ 経歴

- ・ お客様先で新規事業や新規プロダクト開発をするエンジニア
- ・ AWS をはじめとした技術を教えることが好きです

□ 好きなAWSサービス



AWS Lambda

AWS Hands-on for Beginners とは



実際に手を動かしながら AWS の各サービスを学んでいただきます



初めてそのサービスをご利用される方がメインターゲットです



お好きな時間、お好きな場所でご受講いただけるオンデマンド形式です



テーマごとに合計1~2時間の内容 & 細かい動画に分けて公開
スキマ時間の学習や、興味のある部分だけの聴講も可能

内容についての注意点

- 資料では2020年2月6日収録時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。資料作成には十分注意しておりますが、資料とAWS公式ウェブサイトとで記載内容に相違があった場合、AWS公式ウェブサイトの記載を優先させていただきます。
- マネージメントコンソールについても、収録時点のものとなります。差異がある場合がございますので、ご注意ください。
- ハンズオンでは AWS の各種サービスの利用、リソースの作成を行います。無料枠を超えるハンズオンもございますが、その場合はご利用料金が発生することをあらかじめご認識ください。
- 学習後のリソースの削除についても、お客様の責任でご実施いただくようお願いいたします。

本シリーズのゴール

- 音声の文字起こし、および、その感情分析を行うパイプラインをサーバーレスアーキテクチャで構築する
- S3 トリガーで AWS Lambda を非同期に呼び出す方法を理解する
- AWS の各種 AI Services の特徴と使い方を理解する

本シリーズの前提条件・知識

- AWS アカウントをお持ちであること
 - ハンズオンの作業が同一AWSアカウントの他のリソースに影響が出る場合があります
 - ハンズオン用にAWSアカウントを取得していただくことをオススメします
 - AdministratorAccess ポリシーのついた IAM ユーザーで作業を進めています
- サーバーレス#1のハンズオン(※1)をご視聴いただいていると、より理解を深めやすいと思います

(※1) AWS Hands-on for Beginners - Serverless #1

https://pages.awscloud.com/event_JAPAN_Hands-on-for-Beginners-Serverless-2019_LP.html

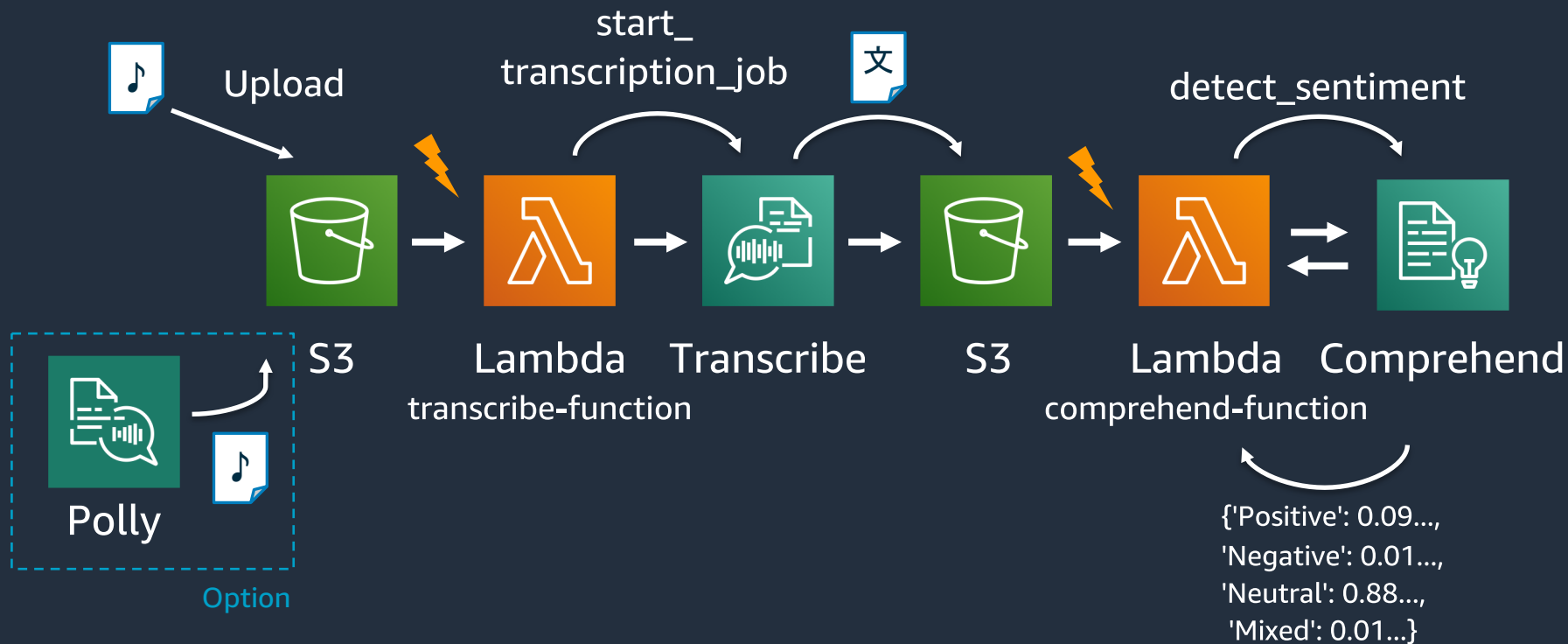
本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

今回のハンズオンで構築する構成



AWS Lambda の特徴

- サーバーの **プロビジョニング/管理なしでプログラムを実行** できるサービス
- コードの実行やスケーリングに必要なことは、Lambda 側で実施するので、開発者の方は **コードを書くことにより集中できる**
- リクエストベースの料金体系



実行回数

(無料枠あり)

+



実行時間

(単価は確保したメモリ量による)

(無料枠あり)

AWS における Compute サービス

復習



Amazon EC2



Container Services



AWS Lambda

自由度

高い 😊



低い 😞

管理の手間

多い 😞

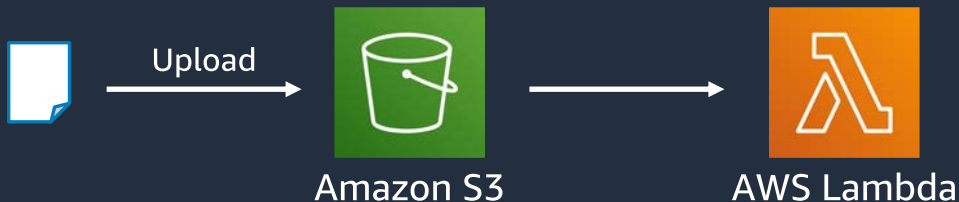


少ない 😊

AWS Lambda のイベントソースと呼び出しタイプ

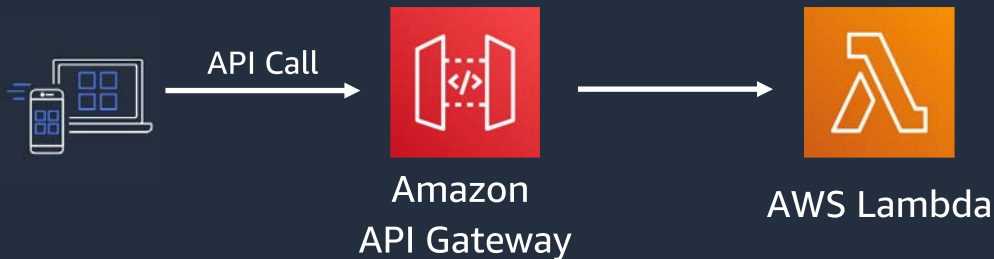
復習

- 非同期呼び出しの例



Lambda へのリクエストが受け付けられたかどうかのみを返却

- 同期呼び出しの例



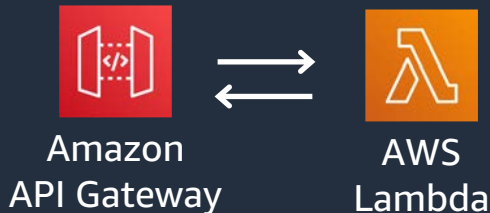
Lambda の実行完了時にレスポンスが返却

本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

AWS Lambda と他の AWS サービスとの連携パターン

① Lambda 関数を同期的に呼び出す



② Lambda 関数を非同期的に呼び出す



③ Lambda がイベントを読み取る (ポーリングする)

◆ストリームベース



◆ストリームベースではない



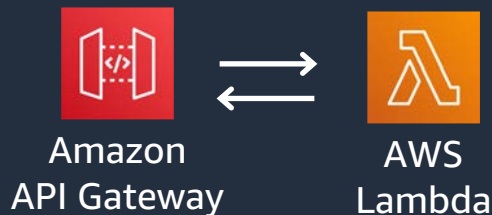
AWS Lambda と他の AWS サービスとの連携パターン

- **ポーリングベース**: Lambda がポーリングして、データがあれば関数を実行
 - └ **ストリームベース**
 - └ 呼び出しタイプ: 同期 ← “③ Lambda がイベントを読み取る”
 - └ ~~呼び出しタイプ: 非同期~~
 - └ **ストリームベースではない**
 - └ 呼び出しタイプ: 同期 ← “③ Lambda がイベントを読み取る”
 - └ ~~呼び出しタイプ: 非同期~~
- **ポーリングベースではない**: Lambda はイベントソースから呼ばれる
 - └ 呼び出しタイプ: 同期 ← “① Lambda 関数を同期的に呼び出す”
 - └ 呼び出しタイプ: 非同期 ← “② Lambda 関数を非同期的に呼び出す”

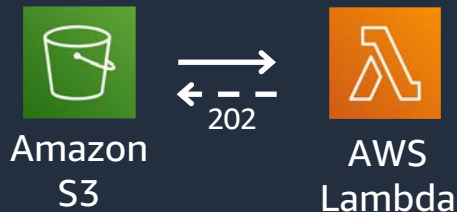
AWS Lambda と他の AWS サービスとの連携パターン

再掲

① Lambda 関数を同期的に呼び出す



② Lambda 関数を非同期的に呼び出す

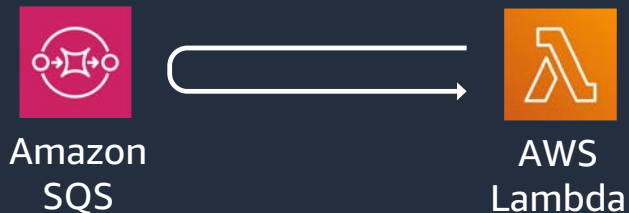


③ Lambda がイベントを読み取る (ポーリングする)

◆ストリームベース

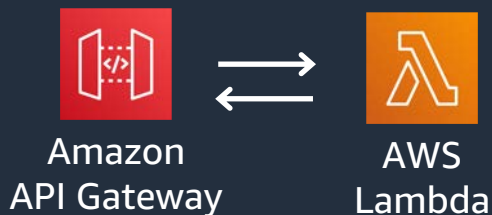


◆ストリームベースではない



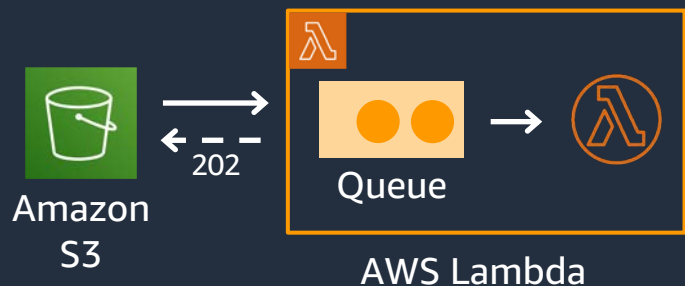
(参考) AWS Lambda 開発者ガイド - 他のサービスで AWS Lambda を使用する
https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/lambda-services.html

① Lambda 関数を同期的に呼び出す



- イベントソースが Lambda 関数を呼び出し、その**実行が終わるのを待つ**形
- イベントソースに **Lambda の実行結果が返される**
- エラー発生時の動きは、イベントソースとなった AWS サービスの設定によって決まる
- 利用シーン
 - Lambda 側の処理を待つ必要があり、イベントソースに実行結果を返す必要がある場合

② Lambda 関数を非同期的に呼び出す



- イベントソースが Lambda を呼び出すと、イベント情報は Lambda 内部のキューに格納されその後、Lambda ファンクションが実行される
- イベントソースに **Lambda の実行結果は返されない**
- **エラー時の動き**
 - 実行時エラー: リトライ回数を Lambda で設定 (N=0~2)
 - 同時実行数不足: リトライ期間を設定 (60秒~6時間)
- 利用シーン
 - Lambda 側の処理を待つ必要がない場合
 - Lambda 側の処理に時間がかかる場合

(参考) AWS Lambda – 非同期呼び出し

https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/invoke-async.html

③ Lambda がイベントを読み取る

◆ ストリームベース



- DynamoDB Streams と Kinesis Data Streams がこの形
- Lambda がイベントソースに対して、新しいレコードがないかをポーリング
- 失敗レコードの有効期限切れ or 処理が成功するまでそのシャードからの読み込みが行われない

◆ ストリームベースではない

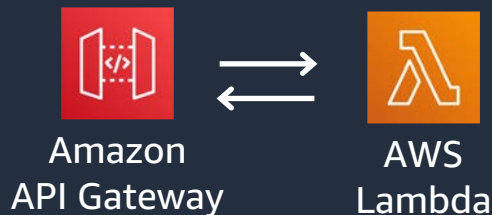


- Amazon SQS がこの形
- Lambda がイベントソースに対して、新しいレコード（メッセージ）がないかをポーリング
- 失敗した場合、そのメッセージはキューに戻り、Visibility Timeout 後に再度処理が行われる
処理のブロックは行われない

AWS Lambda と他の AWS サービスとの連携パターン

再掲

① Lambda 関数を同期的に呼び出す



② Lambda 関数を非同期的に呼び出す



③ Lambda がイベントを読み取る (ポーリングする)

◆ストリームベース



◆ストリームベースではない

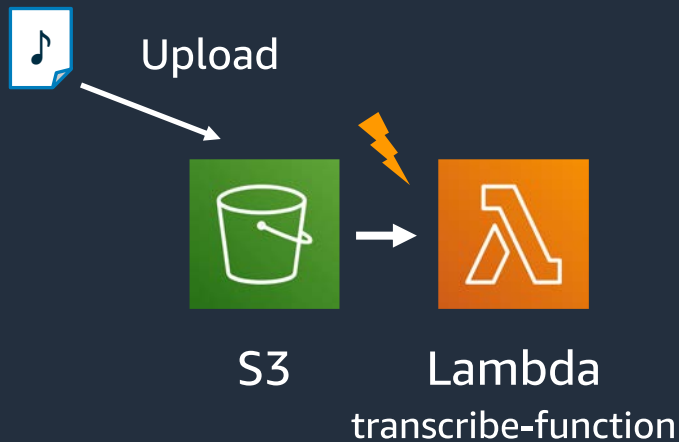


(参考) AWS Lambda 開発者ガイド - 他のサービスで AWS Lambda を使用する
https://docs.aws.amazon.com/ja_ip/lambda/latest/dg/lambda-services.html

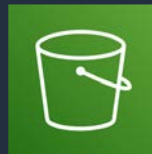
本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

今回のハンズオンで構築する構成



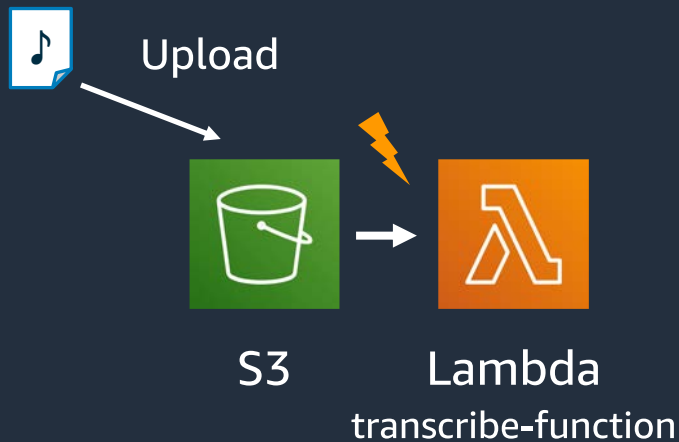
Amazon Simple Storage Service (Amazon S3)



- 高い耐久性
 - 99.999999999% (イレブンナイン)
 - 標準で少なくとも3つのアベイラビリティゾーン (AZ) にまたがってデータが格納される
- 容量無制限、安価なストレージ
- 静的 Web サイトホスティング機能 など
- 様々な AWS サービスと連携

(参考) AWS Black Belt Seminar: Amazon S3/Glacier
<https://aws.amazon.com/jp/blogs/news/webinar-bb-amazon-s3-glacier-2019/>

今回のハンズオンで構築する構成

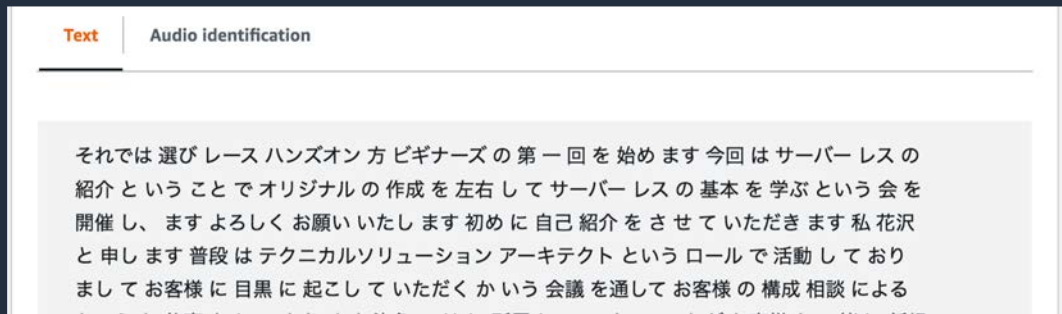


本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ



- 音声をテキストに変換する **文字起こし** サービス
 - 2019/11 に日本語対応
 - 保存された音声ファイルに加え、リアルタイム変換も可能（現在、リアルタイム変換は日本語未対応）

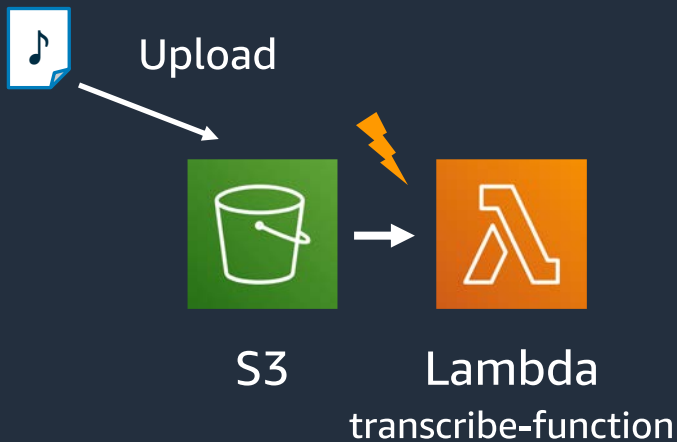


- ユースケースの例
 - 記者会見やインタビューなどの録音データを文字起こし
 - 動画コンテンツの字幕作成
 - コールセンター業務の通話の文字起こし

(参考) AWS Black Belt Seminar: Amazon AI Service
<https://aws.amazon.com/jp/blogs/news/webinar-bb-ai-service-2019/>

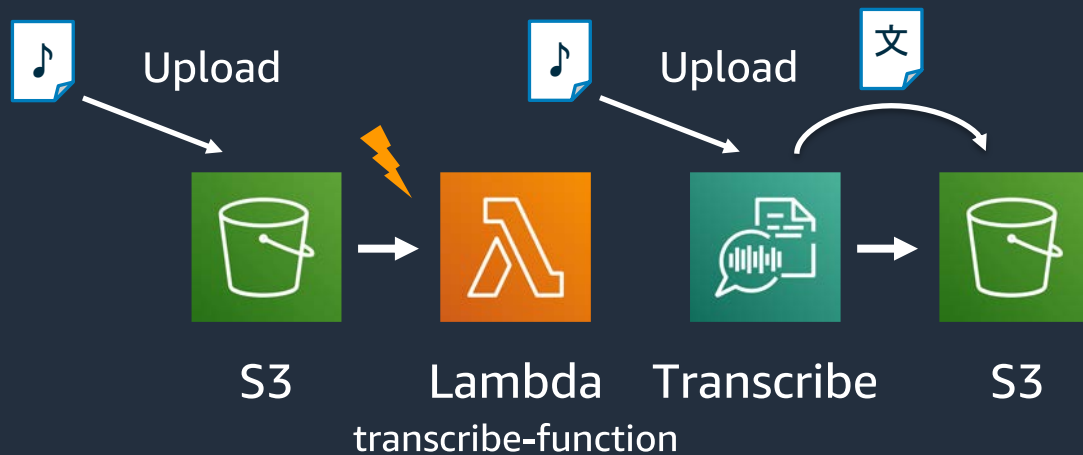
今回のハンズオンで構築する構成

前回まで



今回のハンズオンで構築する構成

今回

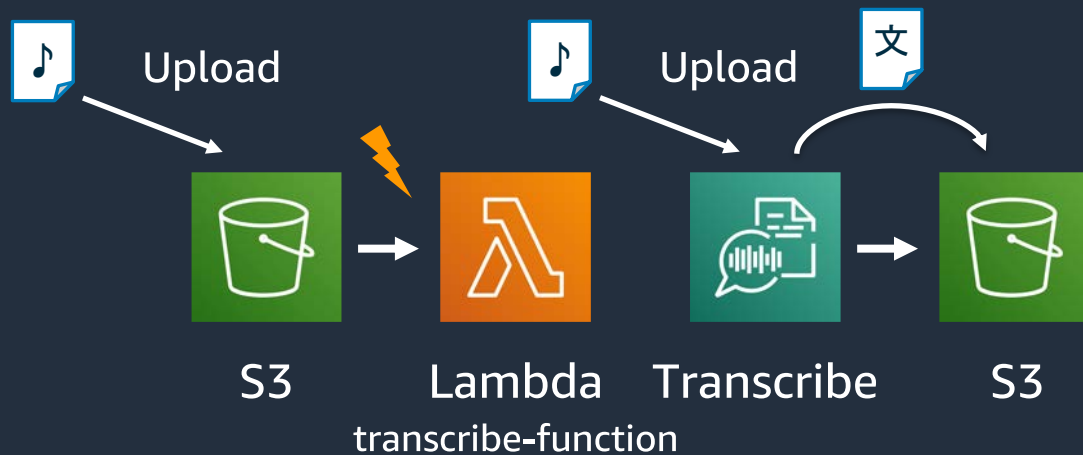


本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

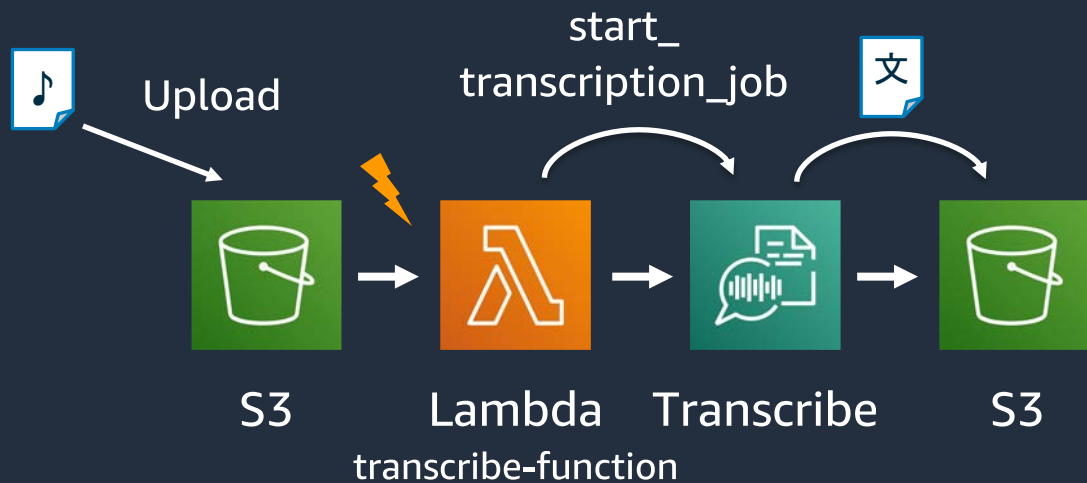
今回のハンズオンで構築する構成

前回まで



今回のハンズオンで構築する構成

今回



本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを **Comprehend** で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

Amazon Comprehend



- 機械学習を使用した **自然言語処理** サービス
 - テキストの中の有用な情報を発見・分析
 - キーフレーズやエンティティ（場所、日付）の取得、感情分析（ポジネガ分析）
 - 2019/11 に日本語対応（東京リージョンでは未対応）
- ユースケースの例
 - 社内ドキュメントのキーフレーズやエンティティを検索エンジンでインデックス
 - レビューやアンケート結果、ソーシャルメディアからの Feedback を感情分析

（参考）AWS Black Belt Seminar: Amazon AI Service
<https://aws.amazon.com/jp/blogs/news/webinar-bb-ai-service-2019/>

Insights Info

Entities | Key phrases | Language | Sentiment | Syntax

Analyzed text

アマゾン ウェブ サービス (AWS) とは? Amazon 社内のビジネス課題を解決するために生まれた IT インフラストラクチャ のノウハウをもとに、2006 年、アマゾン ウェブ サービス (AWS) はウェブサービスという形態で、企業を対象に IT インフラストラクチャ サービスの提供を開始しました。クラウドコンピューティングの主なメリットの1つは、必要な時に、必要なだけ、低価格で IT リソースを提供することができることです。クラウドを使用することで、企業はもはや数週間・数カ月も前から、サーバーや他の IT インフラストラクチャ を計画・調達する必要がなくなります。クラウドなら、即座に何百・何千ものサーバーを数分で起動し、迅速に結果を出すことができます。

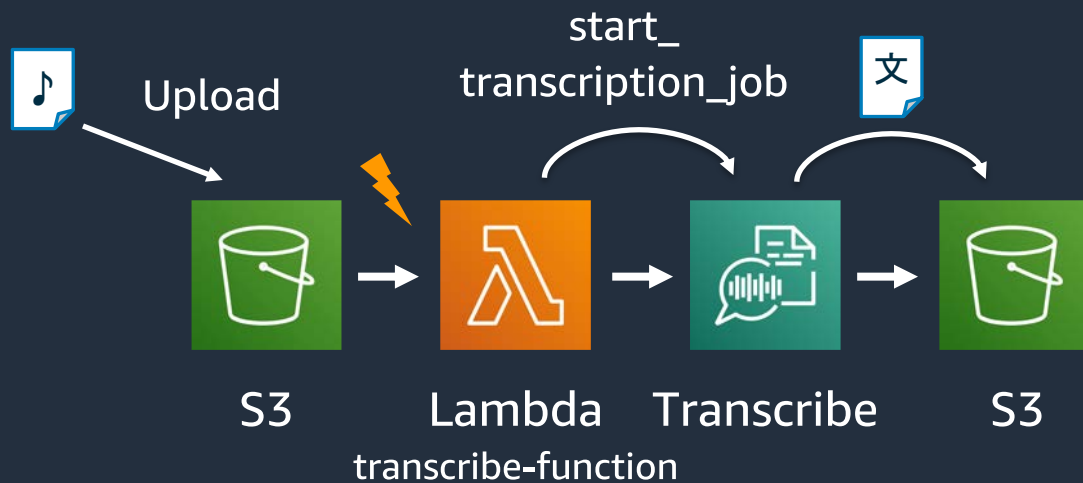
▼ Results

< 1 2 >

Entity	Category	Confidence
アマゾン ウェブ サービス	Organization	0.90
AWS	Organization	0.94
Amazon 社内	Organization	0.73
IT インフラストラクチャ	Other	0.83
2006 年	Date	0.99+
アマゾン ウェブ サービス	Title	0.74

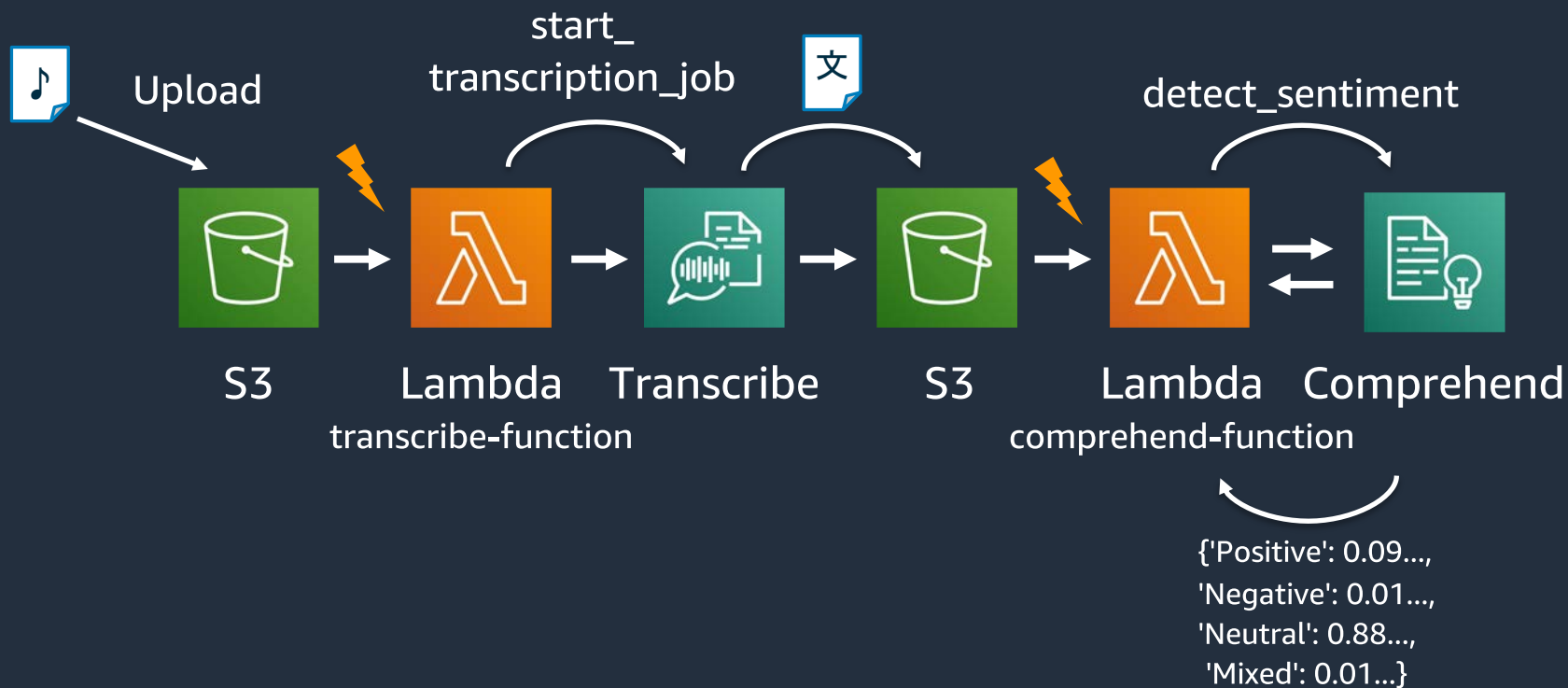
今回のハンズオンで構築する構成

前回まで



今回のハンズオンで構築する構成

今回



本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

Amazon Polly



- **テキストを音声に変換** するサービス
 - 英語、日本語をはじめ複数の言語をサポート
 - SSML (Speech Synthesis Markup Language) や Lexicon の設定を行えます
- ユースケースの例
 - カスタマーセンターの自動音声応答
 - 記事（テキスト）を音声に変換する

(参考) AWS Black Belt Seminar: Amazon AI Service
<https://aws.amazon.com/jp/blogs/news/webinar-bb-ai-service-2019/>

プレーンテキスト

SSML

?

```
< speak>
  こんにちは、金澤です。
  < amazon:effect name="whispered">囁いたり、</amazon:effect>
  < prosody volume="x-loud">声の大きさを変えたりできますよ！</prosody>
</ speak>
```

149 文字を使用

言語とリージョン

日本語

音声

☐ Mizuki, 女性

☒ Takumi, 男性

▶ 音声を聴く

📄 ダウンロード MP3

サンプルレート: 22050Hz

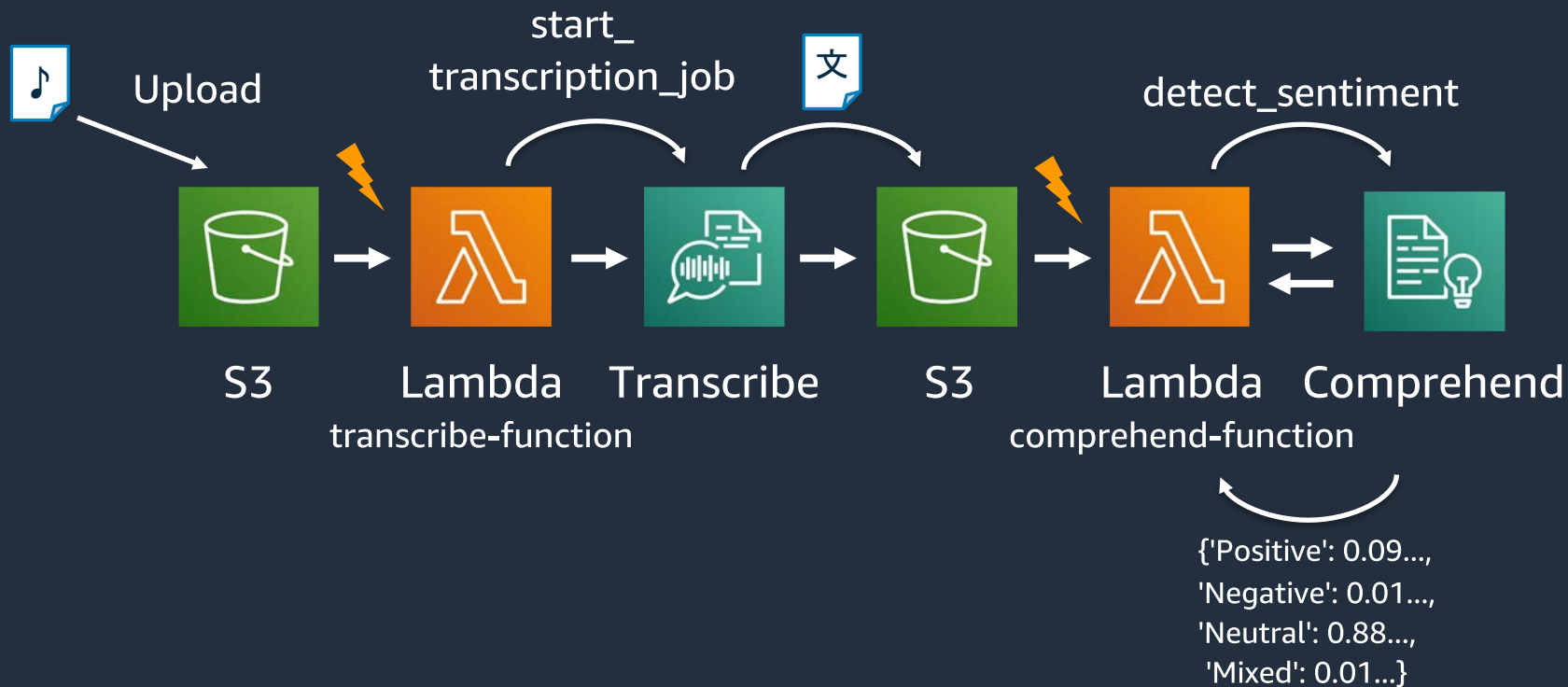
ファイル形式を変更

S3 に向けて合成

S3 タスク設定の変更

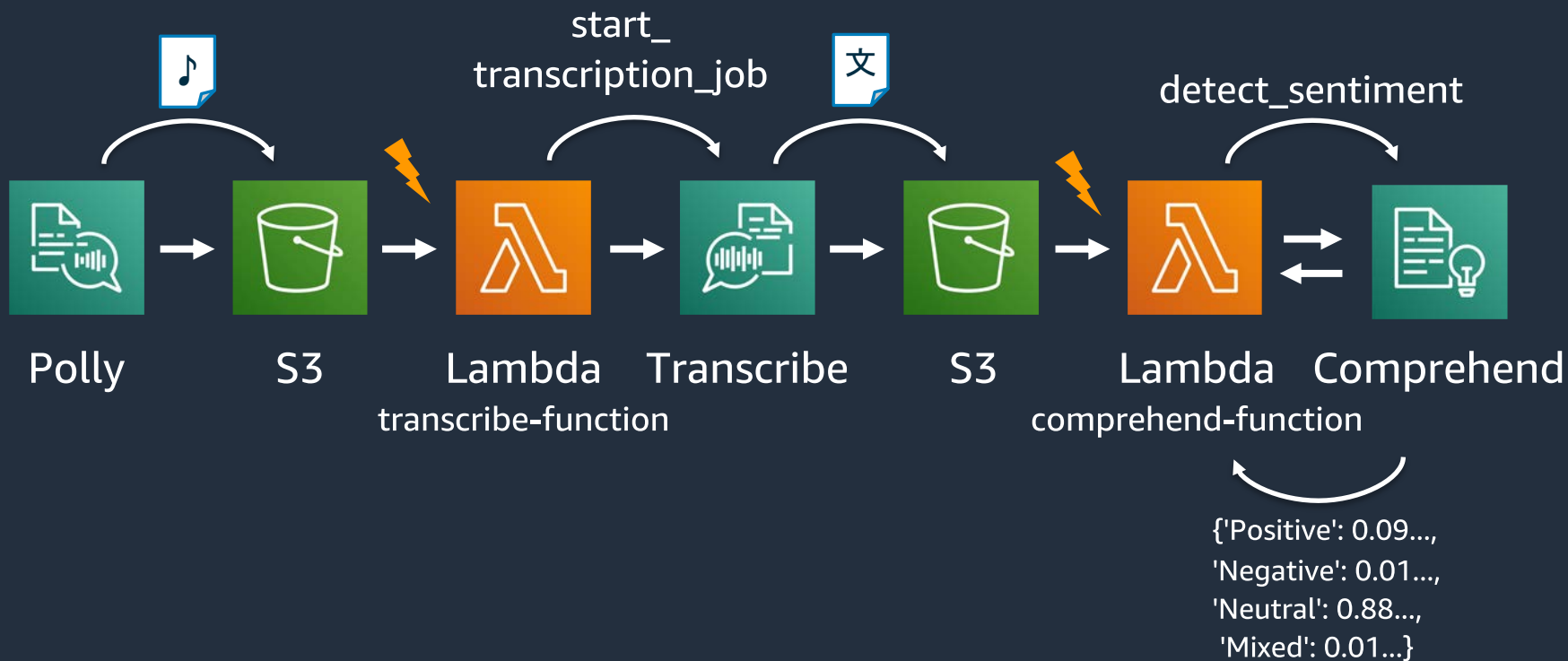
今回のハンズオンで構築する構成

前回まで



今回のハンズオンで構築する構成

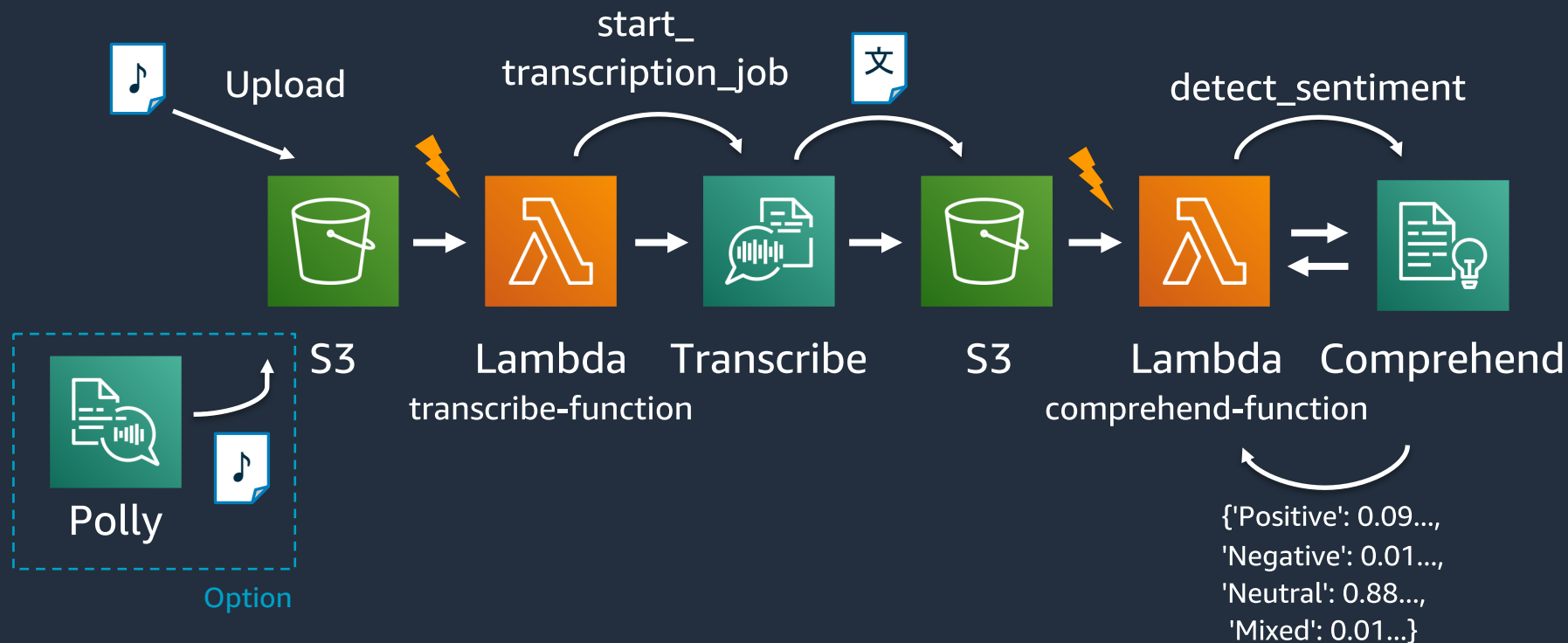
今回



本シリーズのアジェンダ

- 1) 今回のハンズオンで構築する構成の紹介と AWS Lambda の復習
- 2) AWS Lambda と他の AWS サービスとの連携パターン
- 3) S3 トリガーで Lambda ファンクションを起動する
- 4) Amazon Transcribe を使って文字起こしを試してみる
- 5) S3 への音声ファイルアップロードをトリガに Lambda を起動し Transcribe するパイプラインを作る
- 6) パイプラインで文字起こしたテキストを Comprehend で感情分析する
- 7) [Option] Amazon Polly で作成した音声ファイルをパイプラインに投入する
- 8) クリーンアップ 、まとめ + Next Challenge のススメ

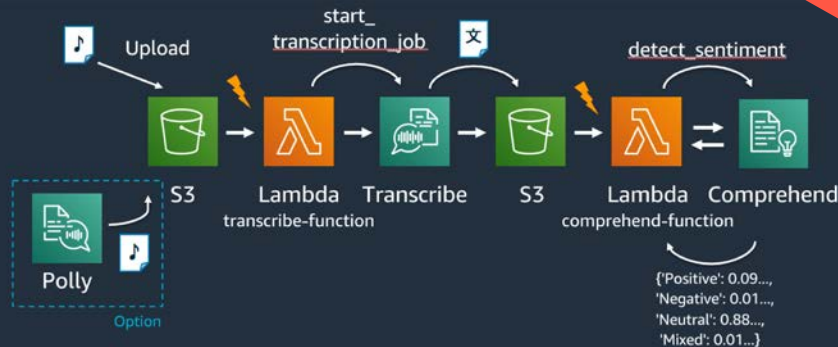
今回のハンズオンで構築した構成



本シリーズのまとめ + Next Challenge のススメ

アンケートも
よろしくお願いします！

- 他サービスからの **AWS Lambda** 呼び出しパターン をご紹介
- “非同期呼び出し” の例として
S3 + Lambda パターンを実装
- 各種 AWS AI Services と連携し、
文字起こし + 感情分析パイプラインを構築



• Next Challenge 案

- 今回の構成をテンプレートから自動構築するには？ → [サーバーレス #2](#) を Check !!
- Amazon Translate と組み合わせてみると何ができる？ → [サーバーレス #1](#) を Check !!
- ELB + EC2 + RDS 構成を作り、AWS AI Services を組み合わせる？ → [Scalable Web 編](#) を Check !!

アンケートも
よろしくお願いします！

