デベロッパーガイド

Amazon Translate

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Translate: デベロッパーガイド

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスはAmazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Amazon Translate とは 1
ユースケース 1
初めてお使いになる方 2
Amazon Translate の料金 2
Amazon Translate API リファレンス 2
サポートされている言語 3
サポートされている言語 3
Amazon Translate 機能でサポートされている言語 7
仕組み 8
入力コンテンツでサポートされている形式 8
翻訳のカスタマイズ 8
言語の自動検出 9
例外処理 9
設定 10
にサインアップする AWS アカウント 10
管理アクセスを持つユーザーを作成する 11
のセットアップ AWS CLI 12
プログラマチックアクセス権を付与する 12
AWS SDKs の使用 14
入門 16
使用開始 コンソール 16
開始方法 (AWS CLI) 18
コマンドラインを使用したテキストの翻訳 18
次のステップ 19
開始方法 (SDK) 19
SDK for Java の使用 19
AWS SDK for Python の使用 20
その他の SDK の例 20
翻訳の処理モード 21
リアルタイム翻訳 21
リアルタイム翻訳 (コンソール) 21
リアルタイム翻訳 (API) 26
非同期バッチ処理 29
利用可能なリージョン 29

前提条件 ... 30
ジョブの実行 ... 35
モニタリングと分析 ... 42
結果の取得 ... 44
翻訳のカスタマイズ ... 47
翻訳禁止タグの使用 ... 47
コンソールで翻訳禁止タグを使用する ... 48
API で翻訳禁止タグを使用する ... 48
カスタム用語によるカスタマイズ ... 49
カスタム用語の作成 ... 50
カスタム用語の使用 ... 52
SDK for Python の使用例 ... 53
用語の暗号化 ... 55
ベストプラクティス ... 55
簡潔さの使用 ... 57
簡潔さ設定の使用 ... 57
サポートされている言語 ... 57
不敬表現のマスキング ... 58
不敬表現設定の使用 ... 59
サポートされていない言語 ... 59
形式の設定 ... 60
形式設定の使用 ... 60
サポートされている言語 ... 61
並列データによるカスタマイズ ... 62
利用可能なリージョン ... 63
Amazon Translate の並列データ入力ファイル ... 64
並列データの追加 ... 70
並列データの表示と管理 ... 73
コードの例 ... 78
基本 ... 78
アクション ... 79
シナリオ ... 97
Amazon Transcribe ストリーミングアプリケーションを構築する ... 97
Amazon Lex chatbot を構築する ... 98
Amazon SNS アプリケーションの構築 ... 99
顧客からのフィードバックを分析するアプリケーションの作成 ... 101

翻訳ジョブを開始する ... 107
Tagging ... 110
新しいリソースへのタグ付け ... 111
タグの表示、編集、削除 ... 112
セキュリティ ... 114
データ保護 ... 115
保管中の暗号化 ... 116
転送中の暗号化 ... 117
Identity and Access Management ... 117
対象者 ... 118
アイデンティティを使用した認証 ... 118
ポリシーを使用したアクセスの管理 ... 122
Amazon Translate で IAM が機能する仕組み ... 125
アイデンティティベースのポリシーの例 ... 132
AWS マネージドポリシー ... 138
トラブルシューティング ... 141
モニタリング ... 143
CloudWatch によるモニタリング ... 146
を使用した Amazon Translate API コールのログ記録 AWS CloudTrail ... 148
Amazon Translate の CloudWatch メトリクスとディメンション ... 150
EventBridge によるモニタリング ... 152
コンプライアンス検証 ... 155
耐障害性 ... 156
インフラストラクチャセキュリティ ... 156
VPC エンドポイントAWS PrivateLink ... 157
Amazon Translate VPC エンドポイントに関する考慮事項 ... 157
Amazon Translate 用のインターフェイス VPC エンドポイントの作成 ... 157
Amazon Translate 用の VPC エンドポイントポリシーの作成 ... 158
ガイドラインとクォータ ... 160
サポートしている AWS リージョン ... 160
コンプライアンス ... 160
スロットリング ... 160
ガイドライン ... 160
Service Quotas ... 161
ドキュメント履歴 ... 164
API リファレンス ... 179

AWS 用語集 180
..... clxxxi

Amazon Translate とは

Amazon Translate は高度な機械学習テクノロジーを使用して、高品質の翻訳をオンデマンドで提供するテキスト翻訳サービスです。Amazon Translate を使用すれば、非構造化テキストドキュメントの翻訳や、複数の言語で動作するアプリケーションの構築が可能です。Amazon Translate でサポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。

トピック

- ユースケース
- Amazon Translate を初めてお使いになる方へ
- Amazon Translate の料金
- Amazon Translate API リファレンス

ユースケース

Amazon Translate を使用して以下を行います。

Amazon Translate を組み込めば、アプリケーションで複数言語のユーザーエクスペリエンスを実現できます。

- ミーティングメモ、技術者によるレポート、ナレッジベースの記事、投稿など、社内で作成したコンテンツを翻訳できます。
- メール、ゲーム内チャット、カスタマーサービスなど、対人コミュニケーションを翻訳できるので、顧客と社員のニーズに合わせた言語で連絡を交わすことができます。

会社の受信データを処理および管理できます。

- ソーシャルメディア、ニュースフィードなどのテキストをさまざまな言語で分析できます。
- eDiscovery のケースなど、さまざまな言語で情報を検索できます。

Amazon Translate を他の AWS のサービスに組み込めば、言語に依存しない処理が可能になります。

- Amazon Comprehend を使用したソーシャルメディアストリームなどといった非構造化テキストから、ネームドエンティティ (固有表現)、センチメント、キーフレーズを抽出できます。
- Amazon Transcribe を使って字幕やライブキャプションをさまざまな言語で作成できます。
- Amazon Polly を使用して、翻訳されたコンテンツを読み上げることができます。

- Amazon S3 に保存されているドキュメントレポジトリを翻訳できます。
- データベース Amazon DynamoDB、Amazon Aurora、Amazon Redshift に保存されているテキストを翻訳できます。
- AWS Lambda または AWS Glue を使用してワークフローをシームレスに統合できます。

Amazon Translate を初めてお使いになる方へ

Amazon Translate を初めてお使いになる場合は、以下のトピックから開始することをお勧めします。

1. Amazon Translate の仕組み — Amazon Translate の概要です。
2. Amazon Translate の開始方法 – AWS アカウントを設定し、Amazon Translate の使用を開始する方法が説明されています。
3. SDK を使用した Amazon Translate のコード例 AWS SDKs – Amazon Translate APIs。

以下のリソースを使用して Amazon Translate サービスについて学ぶこともできます。

- AWS 機械学習 ブログには、Amazon Translate に関する役立つ記事が掲載されています。
- Amazon Translate ディープダイブビデオシリーズでは、Amazon Translate に関する紹介動画を提供しています。

Amazon Translate の料金

他の AWS 製品と同様に、Amazon Translate を使用するための契約や最低契約金はありません。Amazon Translate の使用料金の詳細については、「Amazon Translate 料金表」を参照してください。

Amazon Translate API リファレンス

「Amazon Translate API リファレンス」は独立したドキュメントになりました。詳細については、「Amazon Translate API リファレンス」を参照してください。

サポートされている言語と言語コード

Amazon Translate ではソース言語 (入力言語) をターゲット言語 (出力言語) に翻訳されます。ソース言語とターゲット言語の組み合わせは、言語ペアと呼ばれます。

> **Note**
>
> Amazon Translate では、ソース言語とターゲット言語に同じ言語を指定した翻訳に対して課金されることはありません。ソース言語を auto に設定した場合、自動検出の使用に対して課金される場合があります。詳細については、「言語の自動検出」を参照してください。

トピック

- サポートされている言語
- Amazon Translate 機能でサポートされている言語

サポートされている言語

Amazon Translate は、以下の表に示されている言語間のテキスト翻訳をサポートしています。言語コード列には ISO 639-1 の 2 桁の言語コードが使用されています。言語の国別バージョンについては、この表は RFC 5646 形式に従い、ダッシュの後に ISO 3166 によって定められた 2 桁の国コードを付加しています。例えば、メキシコのスペイン語の言語コードは `es-MX` です。

言語	言語コード
アフリカーンス語	af
アルバニア語	sq
アムハラ語	am
アラビア語	ar
アルメニア語	hy
アゼルバイジャン語	az

言語	言語コード
ベンガル語	bn
ボスニア語	bs
ブルガリア語	bg
カタロニア語	ca
簡体字中国語	zh
繁体字中国語	zh-TW
クロアチア語	hr
チェコ語	cs
デンマーク語	da
ダリ語	fa-AF
オランダ語	nl
英語	en
エストニア語	et
ファルシ語 (ペルシア語)	fa
フィリピン語、タガログ語	tl
フィンランド語	fi
フランス語	fr
フランス語 (カナダ)	fr-CA
グルジア語	ka
ドイツ語	de

言語	言語コード
ギリシャ語	el
グジャラート語	gu
ハイチクレオール語	ht
ハウサ語	ha
ヘブライ語	he
ヒンディー語	hi
ハンガリー語	hu
アイスランド語	is
インドネシア語	id
アイルランド語	ga
イタリア語	it
日本語	ja
カンナダ語	kn
カザフ語	kk
韓国語	ko
ラトビア語	lv
リトアニア語	lt
マケドニア語	mk
マレー語	ms
マラヤーラム語	ml

言語	言語コード
マルタ語	mt
マラーティー語	mr
モンゴル語	mn
ノルウェー語 (ブークモール)	no
パシュト語	ps
ポーランド語	pl
ポルトガル語 (ブラジル)	pt
ポルトガル語 (ポルトガル)	pt-PT
パンジャブ語	pa
ルーマニア語	ro
ロシア語	ru
セルビア語	sr
シンハラ語	si
スロバキア語	sk
スロベニア語	sl
ソマリ語	so
スペイン語	es
スペイン語 (メキシコ)	es-MX
スワヒリ語	sw
スウェーデン語	sv

言語	言語コード
タミル語	ta
テルグ語	te
タイ語	th
トルコ語	tr
ウクライナ語	uk
ウルドゥー語	ur
ウズベク語	uz
ベトナム語	vi
ウェールズ語	cy

Amazon Translate 機能でサポートされている言語

以下のセクションでは、Amazon Translate の機能でサポートされている言語について説明します。

- リアルタイムのドキュメント翻訳 — 英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。リアルタイム翻訳の詳細については、「リアルタイム翻訳」を参照してください。
- 詳細度 – この機能でサポートされている言語については、「」を参照してくださいAmazon Translate での簡潔さの使用。
- 不敬マスキング — この機能でサポートされている言語については「Amazon Translate における不適切な言葉やフレーズのマスキング」を参照してください。
- 形式 — この機能でサポートされている言語については「Amazon Translate の形式設定」を参照してください。

Amazon Translate の仕組み

Amazon Translate サービスを使用して、ソース言語 (入力コンテンツの言語) からターゲット言語 (翻訳出力用に選択した言語) にコンテンツを翻訳します。バッチジョブでは、ファイルを 1 つ以上のソース言語から 1 つ以上のターゲット言語に変換できます。サポートされている言語の詳細については、「」を参照してくださいサポートされている言語と言語コード。

トピック

- 入力コンテンツでサポートされている形式
- 翻訳のカスタマイズ
- 言語の自動検出
- 例外処理

入力コンテンツでサポートされている形式

Amazon Translate では、入力コンテンツに対して次の形式がサポートされています。

- リアルタイム翻訳の場合：
  - 入力テキスト – UTF-8 形式のプレーンテキスト。Amazon Translate は出力コンテンツを UTF-8 テキストとして提供します。
  - 1 つの入力ファイル – プレーンテキスト (.txt)、HTML (.html)、または Word (.docx) コンテンツを含むファイル。Amazon Translate は、出力コンテンツを入力ファイルと同じ形式でファイルとして提供します。
- バッチ翻訳ジョブの場合：
  - 入力ファイルのコレクション – Amazon S3 の場所にアップロードする 1 つ以上のファイル。サポートされているファイル形式には、プレーンテキスト (.txt)、HTML (.html)、Word (.docx)、Excel (.xlsx)、PowerPoint (.pptx)、XLIFF 1.2 (.xlf) などがあります。Amazon Translate は出力コンテンツをファイルとして提供します。各出力ファイルのファイル形式は、入力ファイル形式と一致します。

翻訳のカスタマイズ

次の機能を使用して、Amazon Translate で作成する翻訳をカスタマイズできます。

- Do-not-translateタグ – 開始タグと終了タグを使用して、翻訳しないコンテンツ (HTML コンテンツ) を指定します。
- Custom terminology (カスタム用語) — Amazon Translate でブランド名などの特定の用語を翻訳する方法を定義します。
- Brevity – ほとんどの翻訳の翻訳出力の長さを短くします (簡潔でない翻訳出力と比較して) 。Brevity は、リアルタイムテキスト翻訳でサポートされています。
- 冒涜的 – 翻訳出力の冒涜的な単語やフレーズをマスクします。
- 形式 – 翻訳出力の言語形式のレベルを設定します。
- 並列データ – 指定した翻訳サンプル例のスタイル、トーン、単語の選択を反映するように翻訳出力を適応させます。

詳細については、「翻訳のカスタマイズ」を参照してください。

言語の自動検出

Amazon Translate ではソーステキストで使用されている言語を自動で検出できます。言語の自動検出を使用するには、 `auto` をソース言語として指定します。Amazon Translate がユーザーに代わって Amazon Comprehend を呼び出し、ソーステキストで使用されている言語を判断します。言語の自動検出を選択することで、お客様は Amazon Comprehend のサービス条件および契約に同意するものとみなされます。Amazon Comprehend の料金の詳細については「Amazon Comprehend の料金」を参照してください。

例外処理

サポートされていないソース言語またはターゲット言語を指定した場合、Amazon Translate から次の例外が返されます。

- UnsupportedLanguagePairException – Amazon Translate によりすべての対応言語間の翻訳がサポートされます。この例外は、ソース言語またはターゲット言語のいずれかがサポートされていない場合に返されます。さらなる詳細については、Supported languages を参照してください。
- DetectedLanguageLowConfidenceException – 言語自動検出を使用していて Amazon Translate で正確なソース言語が検出されたか定かではない場合に、この例外が返されます。信頼性のレベルが低い場合でも、それが許容される範囲である場合は例外で返されたソース言語を使用できます。

セットアップ

Amazon Translate を初めて使用する場合は、事前に以下のタスクをすべて実行してください。

タスクのセットアップ

- にサインアップする AWS アカウント
- 管理アクセスを持つユーザーを作成する
- AWS Command Line Interface（AWS CLI) をインストールして設定する
- プログラマチックアクセス権を付与する
- AWS SDK でのこのサービスの使用

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一部では、電話またはテキストメッセージを受信し、電話のキーパッドに検証コードを入力します。

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、ルートユーザーアクセスが必要なタスクの実行にはルートユーザーのみを使用するようにしてください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 を保護し AWS IAM Identity Center、 を有効にして管理ユーザーを作成します。

を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、アカウント所有者AWS Management Consoleとして にサインインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドのルートユーザーとしてサインインするを参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、IAM ユーザーガイドの AWS アカウント 「ルートユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

   を ID ソース IAM アイデンティティセンターディレクトリ として使用する方法のチュートリアルについては、「 AWS IAM Identity Center ユーザーガイド」の「デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン 「 ユーザーガイド」の AWS 「 アクセスポータルにサインインする」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「グループの結合」を参照してください。

AWS Command Line Interface（AWS CLI) をインストールして設定する

を使用して AWS CLI 、Amazon Translate をインタラクティブに呼び出します。

をインストールして設定するには AWS CLI

1. をインストールします AWS CLI。手順については、『AWS Command Line Interface ユーザーガイド』の次のトピックを参照してください。

   の最新バージョンをインストールまたは更新する AWS Command Line Interface

2. AWS CLIを設定します。手順については、『AWS Command Line Interface ユーザーガイド』の次のトピックを参照してください。

   AWS Command Line Interfaceの設定

プログラマチックアクセス権を付与する

ユーザーが の AWS 外部で を操作する場合は、プログラムによるアクセスが必要です AWS Management Console。プログラムによるアクセスを許可する方法は、 がアクセスするユーザーのタイプによって異なります AWS。

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラマチックアクセス権を必要とするユーザー	目的	方法
ワークフォースアイデンティティ (IAM アイデンティティセンターで管理されているユーザー)	一時的な認証情報を使用して、 AWS CLI、 AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 • については AWS CLI、AWS Command Line Interface ユーザーガイドの「 を使用する AWS CLI ように AWS IAM Identity Centerを設定する」を参照してください。 • AWS SDKs、ツール、API については、 AWS APIs「 SDK およびツールリファレンスガイド」の「IAM アイデンティティセンター認証」を参照してください。AWS SDKs
IAM	一時的な認証情報を使用して、 AWS CLI、 AWS SDKs、または AWS APIs。	「IAM ユーザーガイド」の「 AWS リソースでの一時的な認証情報の使用」の手順に従います。
IAM	(非推奨) 長期認証情報を使用して、AWS CLI、 AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 • については AWS CLI、「 AWS Command Line Interface ユーザーガイド」の「IAM ユーザー認証情報を使用した認証」を参照してください。 • AWS SDKs「 SDK とツールリファレンスガイド」の

プログラマチックアクセス権を必要とするユーザー	目的	方法
		「長期認証情報を使用した認証」を参照してください。 AWS SDKs • API AWS APIs「IAM ユーザーのアクセスキーの管理」を参照してください。

AWS SDK でのこのサービスの使用

AWS Software Development Kit (SDKsは、多くの一般的なプログラミング言語で使用できます。各 SDK には、デベロッパーが好みの言語でアプリケーションを簡単に構築できるようにする API、コード例、およびドキュメントが提供されています。

SDK ドキュメント	コード例
AWS SDK for C++	AWS SDK for C++ コード例
AWS CLI	AWS CLI コード例
AWS SDK for Go	AWS SDK for Go コード例
AWS SDK for Java	AWS SDK for Java コード例
AWS SDK for JavaScript	AWS SDK for JavaScript コード例
AWS SDK for Kotlin	AWS SDK for Kotlin コード例
AWS SDK for .NET	AWS SDK for .NET コード例
AWS SDK for PHP	AWS SDK for PHP コード例
AWS Tools for PowerShell	Tools for PowerShell のコード例
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) コード例
AWS SDK for Ruby	AWS SDK for Ruby コード例

SDK ドキュメント	コード例
AWS SDK for Rust	AWS SDK for Rust コード例
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP コード例
AWS SDK for Swift	AWS SDK for Swift コード例

可用性の例

必要なものが見つからなかった場合。このページの下側にある [Provide feedback (フィードバックを送信)] リンクから、コードの例をリクエストしてください。

Amazon Translate の開始方法

Amazon Translate の使用を開始する最も簡単な方法は、コンソールを使って何らかのテキストを翻訳することです。コマンドラインから API オペレーションを試すこともできます。また、いずれかの AWS SDKs をインストールして Amazon Translate API オペレーションを使用することもできます。

トピック

- 使用開始 コンソール
- 開始方法 (AWS CLI)
- 開始方法 (SDK)

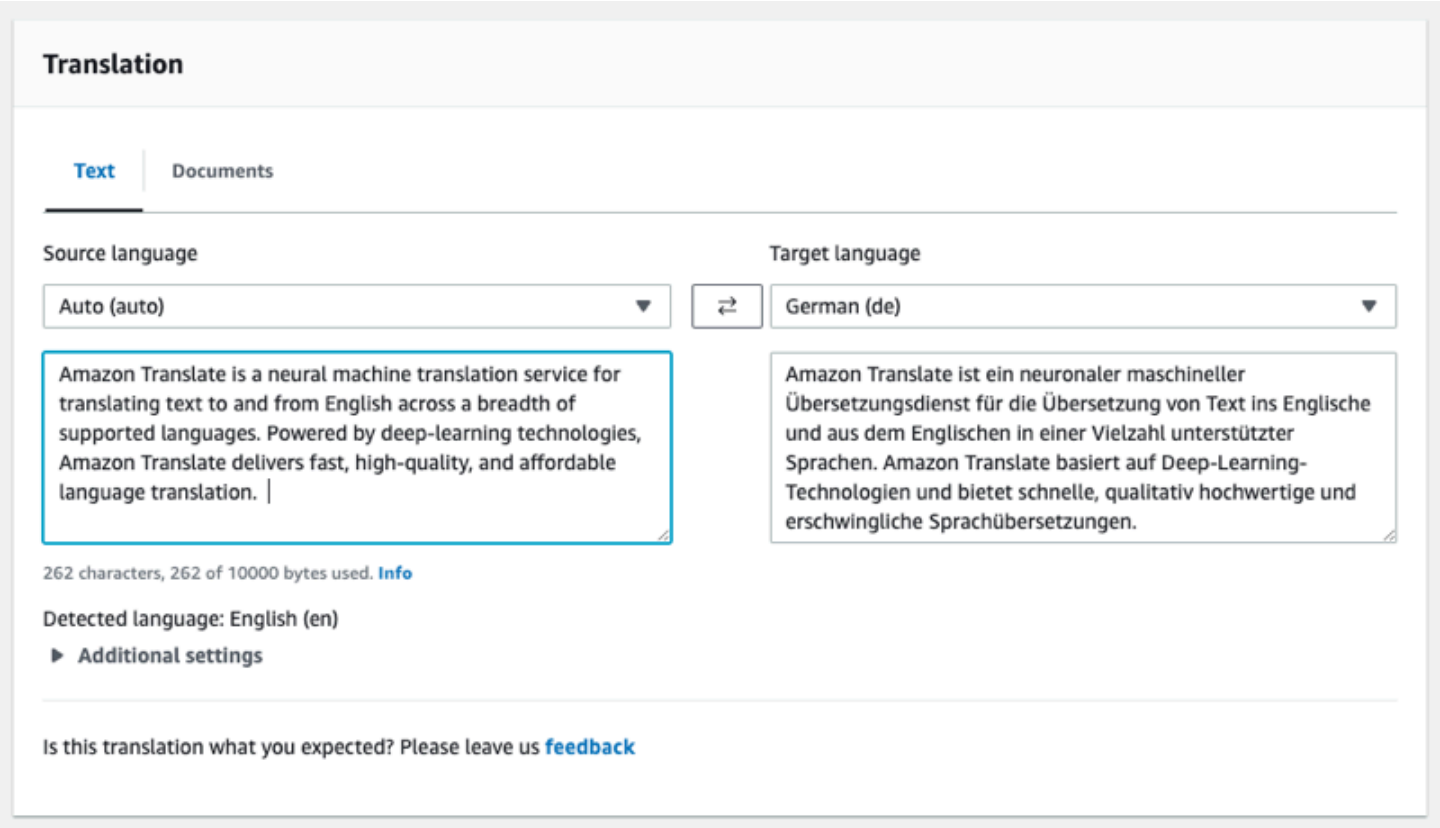
使用開始 コンソール

Amazon Translate の使用を開始する最も簡単な方法は、コンソールを使って何らかのテキストを翻訳することです。コンソールを使用して、最大 10,000 バイトのテキストを翻訳できます。「Amazon Translate の仕組み」の概念と用語を確認してから、先に進むことをお勧めします。

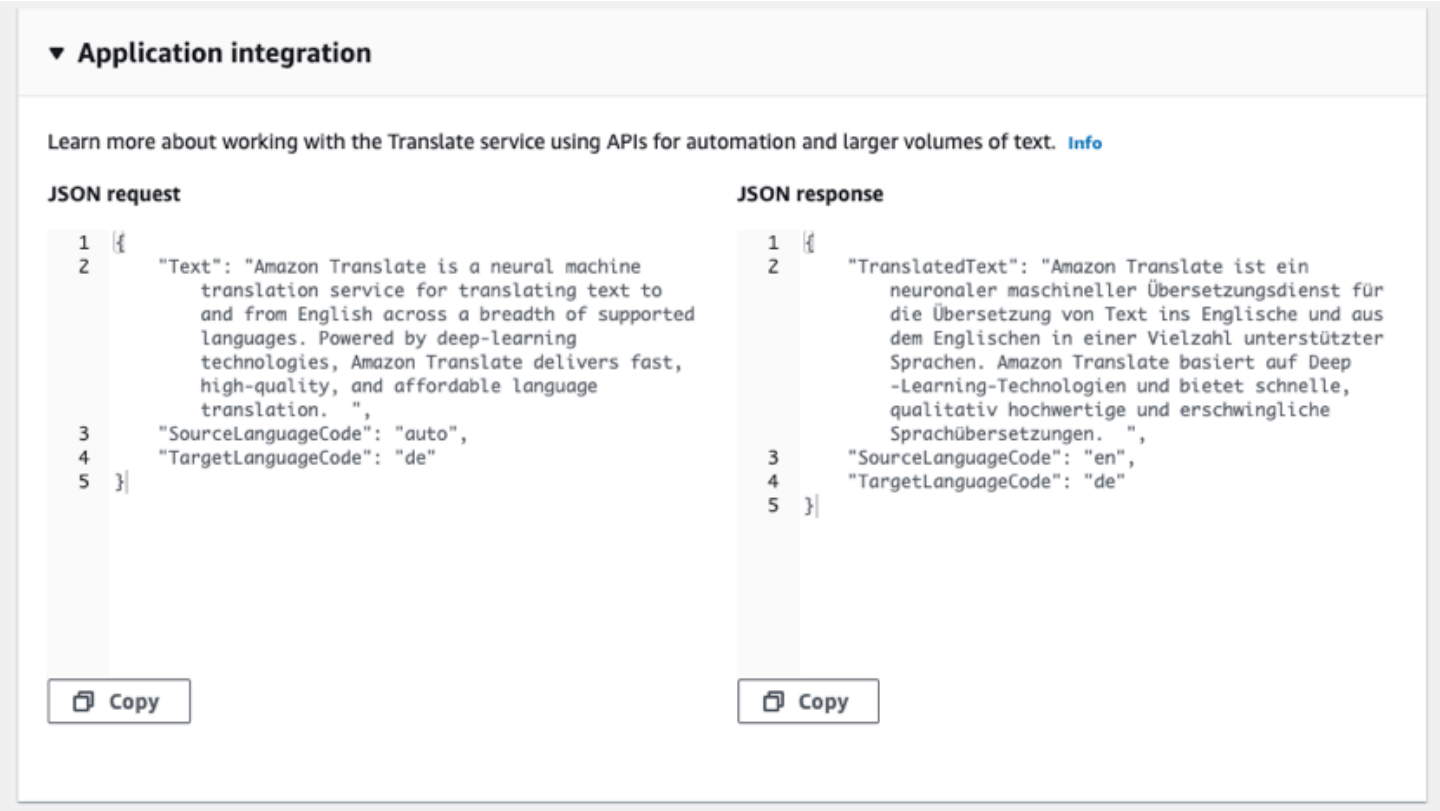
Amazon Translate コンソール を開きます。

Amazon Translate を初めて使用する場合は、[Launch real-time translation (リアルタイム翻訳を起動)] を選択します。

[リアルタイムでの翻訳] で、ターゲット言語を選択します。ソース言語は Amazon Translate によって自動検出されますが、ユーザーが選択することもできます。左側のテキストボックスに翻訳するテキストを入力します。翻訳されたテキストが右側のテキストボックスに表示されます。



[アプリケーション統合] セクションでは、TranslateText オペレーションの JSON 入力と出力を確認できます。



開始方法 (AWS CLI)

以下の演習では、AWS コマンドラインインターフェイス (AWS CLI) を使用してテキストを翻訳します。以下の演習を行うには、CLI に精通している必要があります。また、テキストエディタが必要です。詳細については、「AWS Command Line Interface（AWS CLI) をインストールして設定する」を参照してください。

コマンドラインから Amazon Translate を使用するには、Amazon Translate サービスをサポートするリージョンからコマンドを実行する必要があります。使用可能なエンドポイントとリージョンのリストについては、AWS 全般リファレンスの「Amazon Translate のリージョンとエンドポイント」を参照してください。

コマンドラインを使用したテキストの翻訳

次の例では、コマンドラインから TranslateText オペレーションを使用してテキストを翻訳する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。コマンドラインで、以下のように入力します。

```
aws translate translate-text \
```

```
            --region region \
            --source-language-code "en" \
            --target-language-code "es" \
            --text "hello, world"
```

レスポンスは以下の JSON です。

```
{
    "TargetLanguageCode": "es",
    "Text": "Hola, mundo",
    "SourceLanguageCode": "en"
}
```

次のステップ

Amazon Translate のその他の使用方法については、「SDK を使用した Amazon Translate のコード例 AWS SDKs」を参照してください。

開始方法 (SDK)

AWS にはSDKs が用意されています。SDK は、署名の計算、リクエストの再試行処理、エラー処理など、クライアントの API 接続の詳細の多くを管理します。詳細については、AWS SDK を参照してください。

次の例は、Java と Python を使用して Amazon Translate TranslateText オペレーションを使用する方法を示しています。SDK を使用して、Amazon Translate API オペレーションについて学び、独自のアプリケーション構成要素として使用します。

トピック

- を使用したテキストの翻訳 AWS SDK for Java
- を使用したテキストの翻訳 AWS SDK for Python (Boto)
- その他の SDK の例

を使用したテキストの翻訳 AWS SDK for Java

AWS は、Java で TranslateText オペレーションを使用する方法の GitHub の例を提供します。この例を実行するには、 が必要です AWS SDK for Java。SDK for Java をインストールする手順については、「AWS SDK for Java 2.x のセットアップ」を参照してください。

を使用したテキストの翻訳 AWS SDK for Python (Boto)

次の例は、Python で TranslateText オペレーションを使用する方法を示しています。この例を実行するには、 経由で Python SDK をインストールします AWS CLI。手順については、the section called “のセットアップ AWS CLI” を参照してください。

```
import boto3

translate = boto3.client(service_name='translate', region_name='region', use_ssl=True)

result = translate.translate_text(Text="Hello, World",
            SourceLanguageCode="en", TargetLanguageCode="de")
print('TranslatedText: ' + result.get('TranslatedText'))
print('SourceLanguageCode: ' + result.get('SourceLanguageCode'))
print('TargetLanguageCode: ' + result.get('TargetLanguageCode'))
```

サポートされている言語コードのリストについては、「サポートされている言語と言語コード」を参照してください

その他の SDK の例

.NET と SAP ABAP を使用する例SDK を使用した Amazon Translate のコード例 AWS SDKsについては、「」を参照してください。

翻訳の処理モード

ドキュメントを翻訳するときは、2 つの異なる翻訳処理モードを使用できます。リアルタイム翻訳または非同期バッチ処理です。使用するモードは、ターゲットドキュメントのサイズと種類に基づいており、翻訳ジョブの送信方法と結果の表示方法に影響します。

- リアルタイム翻訳 — 少量のテキスト (またはテキストファイル) を翻訳する同期リクエストを行うと、Amazon Translate は翻訳されたテキストを直ちに応答します。
- 非同期バッチ処理 – ドキュメントのコレクションを Amazon Simple Storage Service (Amazon S3) の場所に配置し、非同期処理ジョブを開始して翻訳します。Amazon Translate は、翻訳された出力ドキュメントを指定された Amazon S3 の場所に送信します。

リアルタイム翻訳

Amazon Translate では、ドキュメントとテキストの翻訳オペレーションがリアルタイムで行われ、翻訳がすぐに返されます。コンソールまたは API を使用してリアルタイムの翻訳を実行できます。

トピック

- コンソールを使用したリアルタイム翻訳
- API を使用したリアルタイム翻訳

コンソールを使用したリアルタイム翻訳

リアルタイム翻訳にコンソールを使用する場合、入力テキストを [ソース言語] テキストボックスに貼り付けるか、入力テキストをファイルとして入力します。必要に応じて、希望する形式レベル、冒涜的なマスキング、簡潔さ、カスタム用語などの機能を設定できます。

リアルタイム翻訳では言語の自動検出を使用できますが、料金が発生する場合があります。詳細については、「言語の自動検出」を参照してください。

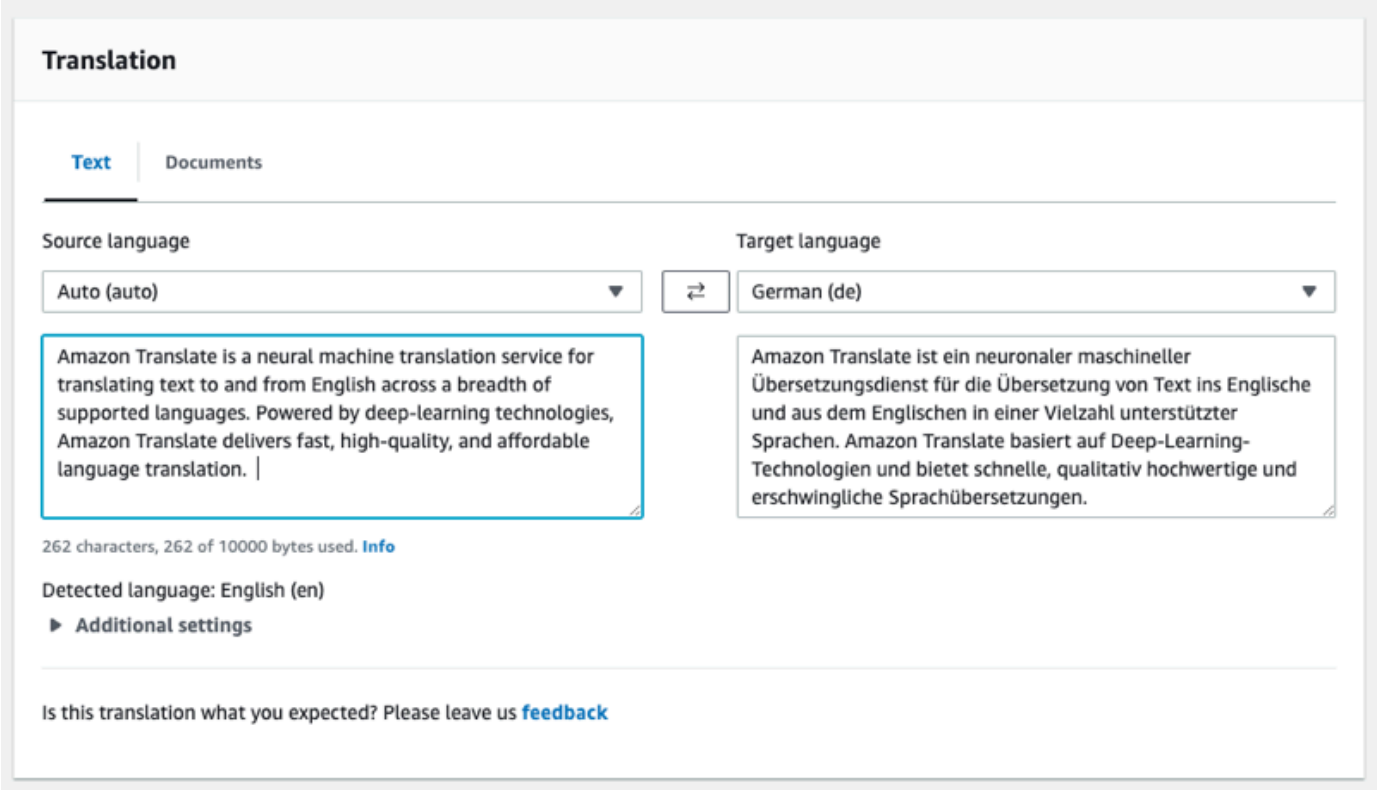
トピック

- テキストの翻訳
- ドキュメントの翻訳
- 同等の API リクエストとレスポンスデータを表示する
- 翻訳機能を使用する

テキストの翻訳

Amazon Translate コンソールを使用して、最大 10,000 バイトのテキストを翻訳できます。

1. Amazon Translate コンソール を開きます。
2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. [ソース言語] では、ソーステキストの言語を選択するか、自動検出の場合は値を [Auto (自動)] のままにします。
4. [ターゲット言語] で言語を選択します。
5. [ソース言語] テキストボックスにテキストを入力するか貼り付けます。コンソールの [ターゲット言語] テキストボックスに、翻訳されたテキストが表示されます。



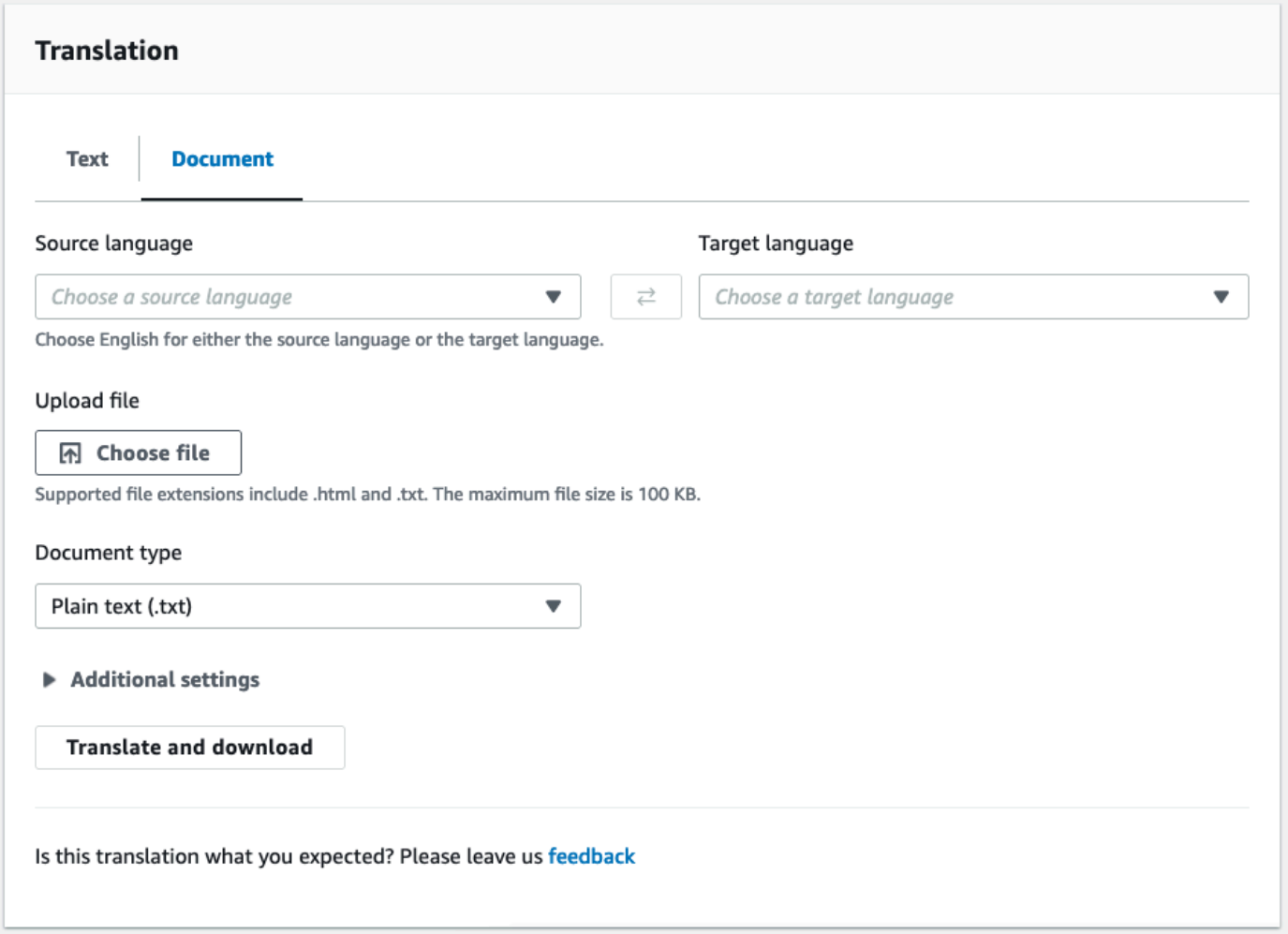
ドキュメントの翻訳

リアルタイムのドキュメント翻訳は、英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。

Amazon Translate コンソールを使用してドキュメントを翻訳するには、以下の手順を実行します。

1. Amazon Translate コンソール を開きます。

2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. 翻訳パネルで、[ドキュメント] タブを選択します。



4. [ソース言語] では、ソーステキストの言語を選択するか、自動検出の場合は [Auto (自動)] を選択します。
5. [ターゲット言語] で言語を選択します。ソース言語が英語でない場合、ターゲット言語に英語を選択する必要があります。
6. [ファイルをアップロード] で [ファイルの選択] を選択し、ソースファイルへのパスを入力します。ファイルの最大サイズは 100 KB です。
7. [ドキュメントタイプ] では、翻訳ソースファイルの形式を選択します。ドキュメント翻訳では、プレーンテキスト、HTML、または Word (.docx) の入力ファイルがサポートされています。
8. [翻訳] を選択します。

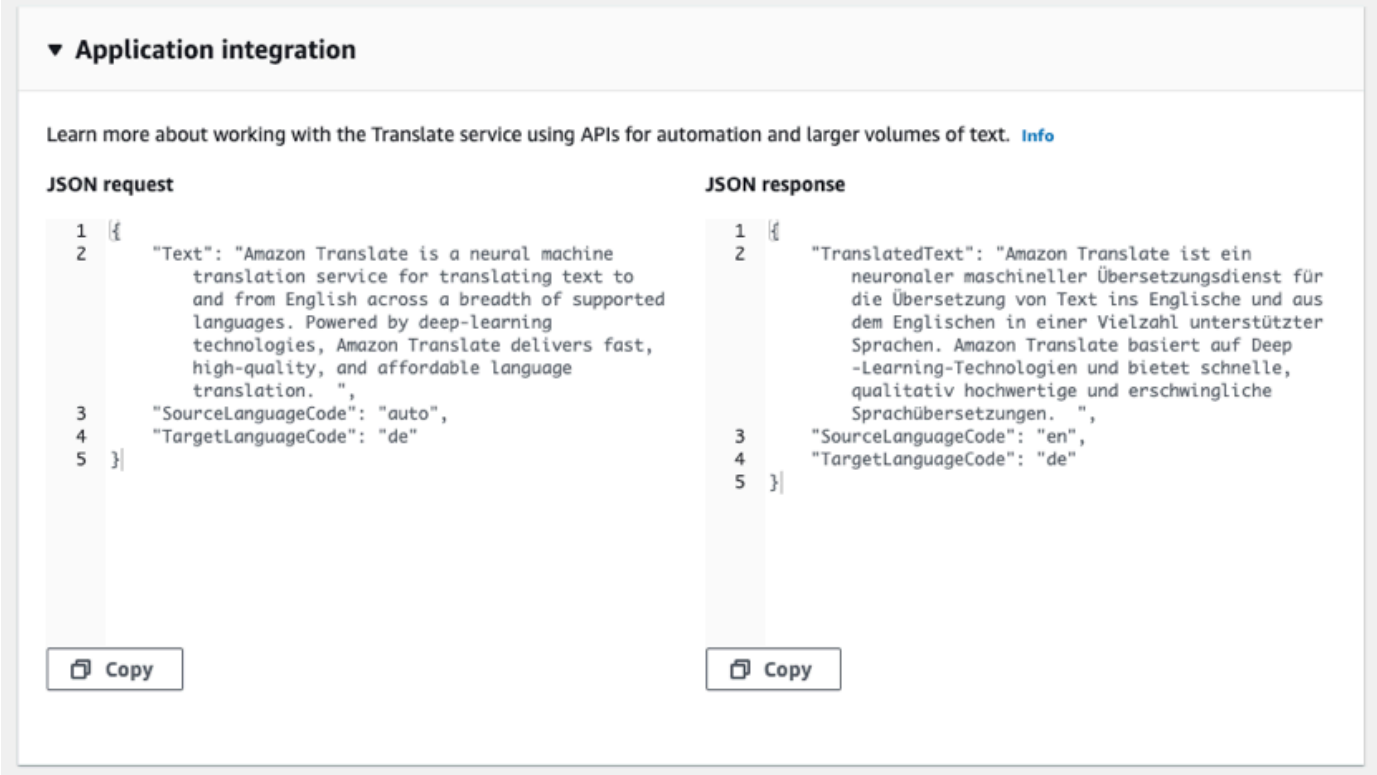
翻訳タスクが完了したら、[翻訳をダウンロード] を選択して、翻訳したドキュメントをローカルのハードドライブにダウンロードします。翻訳したドキュメントの形式 (テキスト、HTML、Word) は入力ドキュメントと一致します。

同等の API リクエストとレスポンスデータを表示する

コンソールを使用して入力テキストまたはドキュメントを翻訳すると、同等の API リクエストデータとレスポンスデータを JSON 形式で表示できます。

1. [翻訳] パネルの下にある [アプリケーション統合] パネルを展開します。

   コンソールには同等の翻訳リクエストデータが JSON 形式で表示されます。



2. [JSON リクエスト] をコピーして TranslateText または TranslateDocument API オペレーションに使用できます。
3. [JSON レスポンスパネル] の JSON 出力は API が生成する出力と一致します。

翻訳機能を使用する

Amazon Translate コンソールで翻訳機能を使用するには、以下の手順を実行します。

1. Amazon Translate コンソール を開きます。
2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. 前の手順で説明したように、ソース言語、ターゲット言語、および入力データ (テキストまたはドキュメント) を指定します。
4. [追加設定] では、次の設定を使用して翻訳ジョブの出力をカスタマイズすることができます。

   Custom terminology (カスタム用語)

   カスタム用語ファイルを選択します。ファイルの入力テキストにソース用語のエントリがある場合、Amazon Translate は用語ファイルの翻訳を使用します。

   詳細については、「カスタム用語による翻訳のカスタマイズ」を参照してください。

   鮮やかさ

   ほとんどの翻訳の翻訳出力の長さを短縮します (簡潔でない翻訳出力と比較して) 。Amazon Translate では、テキストの翻訳には簡潔さがサポートされていますが、ドキュメントの翻訳には対応していません。ソース言語とターゲット言語が簡潔にするためにサポートされていない言語ペアを形成している場合、Amazon Translate は簡潔な設定を無視します。

   サポートされている言語の詳細については、「」を参照してくださいAmazon Translate での簡潔さの使用。

   Profanity (不敬)

   翻訳出力内の不適切な単語やフレーズをマスクします。Amazon Translate は、サポートされているすべての言語で不敬表現のマスキングをサポートしているわけではありません。

   詳細については、「Amazon Translate における不適切な言葉やフレーズのマスキング」を参照してください。

   形式

   一部のターゲット言語では、[形式] をフォーマルまたはインフォーマルに設定できます。形式がターゲット言語をサポートしていない場合、Amazon Translate は形式設定を無視します。

   詳細については、「Amazon Translate の形式設定」を参照してください。
5. ドキュメント翻訳の場合、[翻訳] をクリックすると、選択した翻訳機能を使用してドキュメントが翻訳されます。

テキスト翻訳の場合、各翻訳機能を選択すると、コンソールは翻訳されたテキストにその機能を適用します。

API を使用したリアルタイム翻訳

Amazon Translate では、インタラクティブなアプリケーションをサポートするために、以下のリアルタイム翻訳オペレーションを提供しています。

- TranslateText — テキストブロックを翻訳します。
- TranslateDocument — ファイル (プレーンテキスト、HTML、または.docx) の内容を翻訳します。

これらの同期オペレーションは、翻訳結果をアプリケーションに直接返します。これらのオペレーションで言語の自動検出を使用すると、料金が発生する場合があります。詳細については、「言語の自動検出」を参照してください。

テキストの翻訳

TranslateText オペレーションを使用して、単一のテキストブロックを翻訳します。

コマンドラインを使用したテキストの翻訳

次の例では、コマンドラインから TranslateText オペレーションを使用する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

コマンドラインで、以下のコマンドを入力します。

```
aws translate translate-text \
            --region region \
            --source-language-code "en" \
            --target-language-code "es" \
            --text "hello, world"
```

コマンドは次のJSONで応答します。

```
{
    "TargetLanguageCode": "es",
    "TranslatedText": "Hola, mundo",
```

```
    "SourceLanguageCode": "en"
}
```

JSON ファイルを使用したテキストの翻訳

この例では、JSON ファイルを使用して長いテキストブロックを翻訳する方法を示します。ソース言語とターゲットの言語をコマンドラインで指定するか、JSON ファイルで指定します。

> **Note**
>
> JSON ファイルの例は、読みやすいようにフォーマットされています。`"Text"` フィールドの形式を変更して改行を削除します。
> 例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

JSON ファイルを使用してテキストを翻訳するには

1. 次のテキストを、`translate.json` という JSON ファイルにコピーします。

```
{
    "Text": "Amazon Translate translates documents between languages in
    real time. It uses advanced machine learning technologies
    to provide high-quality real-time translation. Use it to
    translate documents or to build applications that work in
    multiple languages.",
    "SourceLanguageCode": "en",
    "TargetLanguageCode": "fr"
}
```

2. で AWS CLI、次のコマンドを実行します。

```
aws translate translate-text \
            --region region \
            --cli-input-json file://translate.json > translated.json
```

このコマンドは、次の JSON テキストを含む JSON ファイルを出力します。

```
{
    "TargetLanguageCode": "fr",
    "TranslatedText": "Amazon Translate traduit les documents entre
```

```
    les langue en temps réel. Il utilise des technologies
    avancées d'apprentissage de la machine pour fournir
    une traduction en temps réel de haute qualité. Utilisez-le
    pour traduire des documents ou pour créer des applications
    qui fonctionnent en plusieurs langues.",
    "SourceLanguageCode": "en"
}
```

ドキュメントの翻訳

テキスト、HTML、または Word (.docx) ドキュメントを翻訳し、その翻訳の結果をアプリケーションに直接返すには、TranslateDocument を使用します。

リアルタイムのドキュメント翻訳は、英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。ソース言語コードを指定するか、自動検出を使用できます。

コマンドラインを使用したドキュメントの翻訳

次の例では、コマンドラインから TranslateDocument オペレーションを使用する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

コマンドラインで、以下のコマンドを入力します。

```
aws translate translate-document \
            --region region \
            --source-language-code "en" \
            --target-language-code "es" \
            --document-content fileb://source-lang.txt
            --document ContentType=text/plain
            --query "TranslatedDocument.Content"
            --output text | base64
            --decode > target-lang.txt
```

コマンドは次のJSONで応答します。

```
{
    "SourceLanguageCode": "en",
    "TargetLanguageCode": "es",
    "TranslatedDocument":{
```

```
        "Content": blob
    }
}
```

Amazon Translate による非同期バッチ処理

大きいドキュメントの集合 (サイズは 5 GB 以下) を翻訳するには、Amazon Translate 非同期バッチ処理オペレーションである StartTextTranslationJob を使用します。これは、ソーシャルメディアへの投稿やユーザーレビューなどの短いドキュメントの集合に最適です。また、瞬時の翻訳が不要な状況にも最適です。

非同期バッチ翻訳を実行するには通常、以下の手順を実行します。

1. Amazon S3 バケット内の入力フォルダに一連のドキュメントを保存します。
2. バッチ翻訳ジョブを開始します。
3. リクエストの一環として、Amazon S3 の入力フォルダとそのすべてのサブフォルダに対する読み取りアクセス権がある IAM ロールを Amazon Translate に付与します。このロールには、出力 Amazon S3 バケットに対する読み取り/書き込みアクセス権も必要です。
4. バッチ翻訳ジョブの進行状況をモニタリングします。
5. 指定した出力バケットからバッチ翻訳ジョブの結果を取得します。

利用可能なリージョン

バッチ翻訳は、次の AWS リージョンでサポートされています。

- 米国東部(オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)
- 米国西部 (オレゴン)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (ソウル)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- カナダ (中部)

- 欧州 (フランクフルト)
- 欧州 (アイルランド)
- 欧州 (ロンドン)
- 欧州 (パリ)
- 欧州 (ストックホルム)

トピック

- バッチ翻訳ジョブの前提条件
- バッチ翻訳ジョブの実行
- バッチ翻訳ジョブのモニタリングと分析
- バッチ翻訳結果の取得

バッチ翻訳ジョブの前提条件

Amazon Translate がバッチ翻訳ジョブを正常に実行するには、以下の前提条件を満たす必要があります。

- 入力および出力ドキュメントを含む Amazon S3 バケットは、呼び出す API エンドポイントと同じ AWS リージョン内に存在している必要があります。
- バッチ入力ドキュメントの集合は、サイズが 5 GB 以下とします。
- バッチ翻訳ジョブでは、最大 100 万個のドキュメントを送信できます。
- 各入力ドキュメントは 20 MB 以下で、100 万字未満とします。
- 入力ファイルは、Amazon S3 バケットのフォルダ内に存在している必要があります。バケットの最上位レベルに入力ファイルを追加すると、バッチ翻訳ジョブを実行するときに Amazon Translate によってエラーがスローされます。この要件は入力ファイルに適用されます。出力ファイルにフォルダは不要で、Amazon Translate を Amazon S3 バケットの最上位レベルに配置できます。
- 入力ファイルフォルダには、ネストされたフォルダを含めることができます。ネストされたどのフォルダにも details という名前が付けられていないことを確認してください。付いていると、バッチ翻訳ジョブを実行するときに Amazon Translate によってエラーがスローされます。

サポートされるファイル形式

Amazon Translate では、バッチ翻訳ジョブで次の種類のファイルがサポートされています。

- プレーンテキスト
- HTML
- Word ドキュメント (.docx)
- PowerPoint プレゼンテーションファイル (.pptx)
- Excel ワークブックファイル (.xlsx)
- XML Localization Interchange File Format (XLIFF) ファイル (.xlf) Amazon Translate では XLIFF バージョン 1.2 のみがサポートされています。

Amazon Translate ではファイルを UTF-8 エンコーディングする必要があります。

前提となるアクセス許可

バッチ翻訳ジョブを実行する前に、 AWS アカウントに IAM のサービスロールが必要です。このロールには、Amazon Translate に以下を付与するアクセス許可ポリシーがある必要があります。

- Amazon S3 の入力フォルダとそのすべてのサブフォルダへの読み取りアクセス権
- 出力バケットへの読み取り/書き込みアクセス権

また、ロールを引き受け、アクセス許可を取得することを Amazon Translate に許可する信頼ポリシーが含まれている必要があります。この信頼ポリシーは、`translate.amazonaws.com` サービスプリンシパルによる `sts:AssumeRole` アクションの実行を許可する必要があります。

Amazon Translate コンソールを使用してバッチ翻訳ジョブを作成する場合、Amazon Translate でこのロールを自動的に作成できるようにするオプションがあります。 AWS CLI または Amazon Translate API を使用してバッチ翻訳ジョブを実行する場合は、リクエストでロールの Amazon リソースネーム (ARN) を指定します。

詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

Example アクセス許可ポリシー

次のアクセス許可ポリシーの例では、Amazon S3 のサンプル入力フォルダへの読み取りアクセス権が付与されます。サンプル出力バケットへの読み取り/書き込みアクセス権が付与されます。

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::input-bucket-name/*",
        "arn:aws:s3:::output-bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": [
        "arn:aws:s3:::input-bucket-name",
        "arn:aws:s3:::output-bucket-name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::output-bucket-name/*"
    }
  ]
}
```

Example 信頼ポリシー

次の信頼ポリシーでは、ポリシーが属する IAM ロールが Amazon Translate に引き受けられるようにしています。

混乱した代理問題を軽減するために、信頼ポリシーを使用している AWS アカウントを確認することをお勧めします。次の例では、aws:SourceArn および aws:SourceAccount 条件キーを使用してソースアカウントを検証しています。バッチ翻訳ジョブを送信する AWS アカウントを入力します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "translate.amazonaws.com"
```

```
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:translate:*:111122223333:*"
        },
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

暗号化をカスタマイズするための前提となるアクセス許可

Amazon Translate で暗号化設定をカスタマイズできますが、最初に IAM のサービスロールにアクセス許可を追加する必要があります。

Amazon Translate では、バッチ翻訳ジョブを実行したときに生成される翻訳出力が暗号化されます。デフォルトでは、この暗号化は AWS マネージドキー を使用して実行されます。このタイプのキーは、 によって作成 AWS され、 アカウントの AWS Key Management Service （ AWS KMS) に保存されます。ただし、この KMS キーをユーザー自身が管理することはできません。これは、ユーザーに代わって AWSによってのみ管理と使用が行われます。

オプションで、 AWS アカウントで作成、所有、管理する KMS キーであるカスタマーマネージドキーを使用して出力を暗号化することもできます。

キーには、Amazon Translate によるキーの使用を可能にするキーポリシーが必要です。キーポリシーではこれを、Amazon Translate による Amazon S3 バケットへのアクセスを許可するサービスロールにアクセス許可を付与することで実行します。

キーポリシーは、次のポリシーステートメントの例に示すように、出力の暗号化に必要な AWS KMS オペレーションをサービスロールが実行できるようにします。

Example KMS キーポリシーステートメント

```
{
  "Effect": "Allow",
  "Principal":
  {
```

```
    "AWS":
    [
      "arn:aws:iam::111122223333:role/AmazonTranslateServiceRoleS3FullAccess"
    ]
  },
  "Action":
  [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

詳細については、AWS Key Management Service デベロッパーガイドの「AWS KMSでのキーポリシー」を参照してください。

別の AWS アカウントの AWS KMS キーを使用するアクセス許可

Amazon Translate を使用するアカウントとは異なる AWS アカウントにある KMS キーを使用する場合は、以下を行う必要があります。

1. IAM で Amazon Translate のサービスロールを更新する。
2. でキーポリシーを更新します AWS KMS。

サービスロールを更新するには、次の例に示すように、他の AWS アカウントの KMS キーを使用して必要な AWS KMS オペレーションを実行できるようにするポリシーをアタッチします。

Example 別のアカウントの KMS キーへのアクセス権を付与する IAM ポリシー

```
{
  "Effect": "Allow",
  "Action":
  [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:DescribeKey"
  ],
```

```
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/key-id"
}
```

KMS キーポリシーを更新するには、次のポリシーステートメントの例に示されているように、キーの使用が許可されているプリンシパルとしてサービスロールと管理者ユーザーを追加します。

Example IAM ロールによるキーの使用を許可するための KMS キーポリシーステートメント

```
{
  "Effect": "Allow",
  "Principal":
  {
    "AWS":
    [
      "arn:aws:iam::444455556666:role/AmazonTranslateServiceRoleS3FullAccess",
      "arn:aws:iam::444455556666:admin"
    ]
  },
  "Action":
  [
    "kms:Decrypt",
    "kms:CreateGrant",
    "kms:GenerateDataKey",
    "kms:RetireGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

詳細については、「 AWS Key Management Service デベロッパーガイド」の「他のアカウントのユーザーに AWS KMS キーの使用を許可する」を参照してください。

バッチ翻訳ジョブの実行

Amazon Translate コンソール、、または Amazon Amazon Translateを使用して AWS CLI、バッチ翻訳ジョブを実行できます。

> **Note**
>
> バッチ翻訳ジョブは長時間実行されるオペレーションであり、完了するまでにかなりの時間がかかる場合があります。たとえば、小さいデータセットのバッチ翻訳に数分かかり、巨大

なデータセットのバッチ翻訳には最大 2 日かかることがあります。完了時間はリソースの可用性によっても異なります。

Amazon Translate コンソール

Amazon Translate コンソールを使用して翻訳ジョブを実行するには、バッチ翻訳ページでジョブを作成します。

1. Amazon Translate コンソールを開きます。
2. 左側のナビゲーションメニューで、[Batch translation (バッチ翻訳)] を選択します。
3. [Translation jobs (翻訳ジョブ)] ページで、[Create job (ジョブの作成)] を選択します。コンソールに [Create translation job (翻訳ジョブの作成)] ページが表示されます。
4. [Job settings (ジョブ設定)] で、次の作業を実行します。
   a. [Name (名前)] で、バッチ翻訳ジョブのカスタム名を入力します。
   b. [ソース言語] で、ソースファイルの言語を選択します。ソースファイルの言語がわからない場合や、入力ドキュメントに異なるソース言語が含まれている場合は、`auto` を選択します。Amazon Translate が各ファイルのソース言語を自動検出します。
   c. [ターゲット言語] では、最大 10 個の言語を選択できます。Amazon Translate が各ソースファイルを各ターゲット言語に翻訳します。
5. [Input data (入力データ)] で、次の作業を行います。
   a. [Input S3 location (S3 での入力先)] で、Amazon S3 内にあって、翻訳ソースファイルが含まれている入力フォルダを指定します。Amazon S3 でフォルダに移動することでフォルダを提供するには、[Select folder (フォルダを選択)] を選択します。
   b. [File format (ファイル形式)] で、翻訳ソースファイルの形式を選択します。
6. [Output data (出力データ)] で、次の作業を行います。
   a. [Output S3 location (S3 での出力先)] で、Amazon S3 内にあって、Amazon Translate からの翻訳の出力先となる出力フォルダーを指定します。Amazon S3 でフォルダに移動することでフォルダを提供するには、[Select folder (フォルダを選択)] を選択します。
   b. オプションで、 () で管理するカスタマーマネージドキーを使用して出力を暗号化する場合は、暗号化設定をカスタマイズ (詳細) を選択しますAWS KMS。 AWS Key Management Service

デフォルトでは、Amazon Translate は AWSがユーザーに代わって作成、管理、使用する KMS キーを用いて翻訳出力を暗号化します。また、独自の KMS キーで出力を暗号化する場合はこのオプションを選択します。

現在の AWS アカウントから KMS キーを使用する場合は、「 AWS Key Management Service キーの選択」で選択します。また、別の AWS アカウントから KMS キーを使用する場合は、そのキーの Amazon リソースネーム (ARN) を入力します。

> **Note**
>
> 独自の KMS キーを使用できるようにするには、IAM で Amazon Translate のサービスロールにアクセス許可を追加する必要があります。別のアカウントの KMS キーを使用する場合は、 でキーポリシーも更新する必要があります AWS KMS。詳細については、「暗号化をカスタマイズするための前提となるアクセス許可」を参照してください。

7. [Customizations - optional (カスタマイズ - オプション)]では、次の設定を使用して翻訳ジョブの出力をカスタマイズすることができます。

   Profanity (不敬)

   翻訳出力内の不適切な単語やフレーズをマスクします。ジョブに複数のターゲット言語を指定する場合、すべてのターゲット言語が不敬表現のマスクキングをサポートしている必要があります。いずれかのターゲット言語が不敬表現のマスキングをサポートしていない場合、翻訳ジョブはどのターゲット言語でも不敬表現をマスクしません。

   詳細については、「Amazon Translate における不適切な言葉やフレーズのマスキング」を参照してください。

   鮮やかさ

   Amazon Translate は、バッチ翻訳ジョブの簡潔さをサポートしていません。

   詳細については、「Amazon Translate での簡潔さの使用」を参照してください。

   形式

   一部のターゲット言語では、[形式] をフォーマルまたはインフォーマルに設定できます。ジョブに複数のターゲット言語を指定した場合、Translate はサポートされていないターゲット言語の形式設定を無視します。

詳細については、「Amazon Translate の形式設定」を参照してください。

Custom terminology (カスタム用語)

各用語のソース用語例と翻訳例で構成されています。ジョブに複数のターゲット言語を指定した場合、Translate は、用語ファイルにソース用語のエントリがある要求されたターゲット言語ごとに、指定された用語を使用します。

詳細については、「カスタム用語による翻訳のカスタマイズ」を参照してください。

Parallel data (並列データ)

テキストのセグメントの翻訳方法を示した例で構成されています。ジョブに複数のターゲット言語を指定する場合、並列データファイルにはすべてのターゲット言語の翻訳が含まれている必要があります。

並列データをバッチ翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。

> **Note**
>
> Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「Amazon Translate 料金表」を参照してください。

詳細については、「並列データによる翻訳のカスタマイズ (Active Custom Translation)」を参照してください。

8. [Access permissions (アクセス許可)] で、Amazon Translate に対し、Amazon S3 において入力ファイルと出力ファイルに必要なアクセス許可を付与する IAM ロールを提供します。
   - アカウントにこの IAM ロールがすでにある場合は、[Use an existing IAM role (既存の IAM ロールを使用する)] を選択し、[IAM role (IAM ロール)] でそれを選択します。
   - アカウントにこの IAM ロールがない場合は、[Create an IAM role (IAM ロールを作成する)] を選択します。[IAM role (IAM ロール)] で、Input and output S3 buckets (S3 バケットの入力と出力)] を選択します。[Role name (ロール名)] にカスタムの名前を入力します。翻訳ジョブを作成すると、Amazon Translate によってロールが自動的に作成されます。IAM のロール名にはプレフィックスとして AmazonTranslateServiceRole- が付きます。

> **Note**
>
> 独自の KMS キーでの翻訳出力の暗号化を選択した場合は、[Create an IAM role (IAM ロールを作成する)] を選択できません。この場合、既存の IAM ロールを使用する必要があります。また、KMS キーには、ロールでキーを使用できるようにするキーポリシーが必要です。
> 詳細については「暗号化をカスタマイズするための前提となるアクセス許可」を参照してください。

9. [Create job (ジョブの作成)] を選択します。

   コンソールが [Translation jobs (翻訳ジョブ)] ページに戻ります。このページの上部にあるバナーにジョブ作成ステータスが表示されます。数分後、ジョブがテーブルに表示されます。

10. [Name (名前)] 列でジョブ名を選択すると、ジョブの詳細ページが開きます。

    翻訳ジョブの実行中は、[Status (ステータス)] フィールドに[In progress (進行中)] と表示されます。

11. ステータスが [Completed (完了)] になったら、[Output file location (出力ファイルの場所)] でリンクを選択して翻訳出力に進みます。コンソールが Amazon S3 の出力バケットに移動します。

12. 出力ファイルをダウンロードするには、それぞれのチェックボックスを選択して、[ダウンロード] を選択します。

AWS CLI

を使用して翻訳ジョブを実行するには AWS CLI、 `start-text-translation-job` コマンドを使用し、 `parallel-data-names`パラメータに並列データリソースの名前を指定します。

Example Start-text-translation-job コマンド

次の例では、Amazon S3 の入力バケットに保存されている Excel ファイルを送信して、翻訳ジョブを実行します。このジョブは、リクエストに含まれる並列データによってカスタマイズされます。

```
$ aws translate start-text-translation-job \
> --input-data-config ContentType=application/vnd.openxmlformats-officedocument.spreadsheetml.sheet,S3Uri=s3://amzn-s3-demo-bucket/input/ \
> --output-data-config S3Uri=s3://amzn-s3-demo-bucket/output/ \
> --data-access-role-arn arn:aws:iam::111122223333:role/my-iam-role \
> --source-language-code en \
```

```
> --target-language-codes es it \
> --job-name my-translation-job
```

コマンドが成功すると、Amazon Translate からジョブ ID と次のステータスでの応答があります。

```
{
    "JobId": "4446f95f20c88a4b347449d3671fbe3d",
    "JobStatus": "SUBMITTED"
}
```

翻訳ジョブの出力をカスタマイズする場合は、次のパラメータを使用できます。

`--settings`

翻訳出力を構成するための設定で、以下のオプションを含みます。

翻訳出力で簡潔性を有効にします。Amazon Translate は、バッチ翻訳ジョブの簡潔さをサポートしていません。詳細については、「Amazon Translate での簡潔さの使用」を参照してください。

不敬表現のマスキングを有効にして、不適切な単語やフレーズをマスクします。有効にするには、不敬表現のパラメータを `Profanity=MASK` に設定します。詳細については、「Amazon Translate における不適切な言葉やフレーズのマスキング」を参照してください。いずれかのターゲット言語が不敬表現のマスキングをサポートしていない場合、翻訳ジョブはどのターゲット言語でも不敬表現をマスクしません。

翻訳出力の形式レベルを設定します。`Formality` パラメータを `FORMAL` または `INFORMAL` に設定します。ジョブに複数のターゲット言語を指定した場合、Translate はサポートされていないターゲット言語の形式設定を無視します。詳細については、「Amazon Translate の形式設定」を参照してください。

`--terminology-names`

翻訳ジョブに追加するカスタム用語リソースの名前。このリソースでは、各用語のソース用語と翻訳例を挙げています。ジョブに複数のターゲット言語を指定した場合、Translate は、用語ファイルにソース用語のエントリがある要求されたターゲット言語ごとに、指定された用語を使用します。

このパラメータでは、カスタム用語リソースが 1 つだけ許容されます。

使用可能なカスタム用語リソースのリストについては、`list-terminologies` コマンドを使用してください。

詳細については、「カスタム用語による翻訳のカスタマイズ」を参照してください。

`--parallel-data-names`

翻訳ジョブに追加する並列データリソースの名前。このリソースには、テキストのセグメントの翻訳方法を示した例が含まれています。ジョブに複数のターゲット言語を指定する場合、並列データファイルにはすべてのターゲット言語の翻訳が含まれている必要があります。

並列データを翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。

このパラメータでは、並列データリソースが 1 つだけ許容されます。

> **Note**
>
> Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「Amazon Translate 料金表」を参照してください。

使用可能な並列データリソースのリストについては、`list-parallel-data` コマンドを使用してください。

詳細については、「並列データによる翻訳のカスタマイズ (Active Custom Translation)」を参照してください。

翻訳ジョブのステータスを確認するには、`describe-text-translation-job` コマンドを使用します。

Example Describe-text-translation-job コマンド

次の例では、ジョブ ID を指定してジョブのステータスを確認します。この ID は、ジョブが `start-text-translation-job` コマンドで開始されたときに Amazon Translate により提供されたものです。

```
$ aws translate describe-text-translation-job \
> --job-id 4446f95f20c88a4b347449d3671fbe3d
```

Amazon Translate からジョブプロパティでの応答があります。これにはそのジョブのステータスが含まれます。

```
{
    "TextTranslationJobProperties": {
        "JobId": "4446f95f20c88a4b347449d3671fbe3d",
        "JobName": "my-translation-job",
        "JobStatus": "COMPLETED",
        "JobDetails": {
            "TranslatedDocumentsCount": 0,
            "DocumentsWithErrorsCount": 0,
            "InputDocumentsCount": 1
        },
        "SourceLanguageCode": "en",
        "TargetLanguageCodes": [
            "es",
            "it"
        ],
        "SubmittedTime": 1598661012.468,
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/input/",
            "ContentType": "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/output/111122223333-
TranslateText-4446f95f20c88a4b347449d3671fbe3d/"
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/my-iam-role"
    }
}
```

`JobStatus` 値が `IN_PROGRESS` である場合は、数分が経過してから `describe-text-translation-job` を再実行し、ステータスが `COMPLETED` になるまで続けます。ジョブが完了すれば、`OutputDataConfig` の `S3Uri` フィールドにある場所に翻訳結果をダウンロードできます。

Amazon Translate API

Amazon Translate API を使用してバッチ翻訳ジョブを送信するには、StartTextTranslationJob オペレーションを使用します。

バッチ翻訳ジョブのモニタリングと分析

ジョブの ID を使用して、進行状況をモニタリングし、Amazon S3 での出力ドキュメントの場所を取得できます。特定のジョブを監視するには、DescribeTextTranslationJob オペレーションを

使用します。ListTextTranslationJobs オペレーションを使用して、アカウント内のすべての翻訳ジョブに関する情報を取得することもできます。結果を特定の条件に一致するジョブに制限するには、ListTextTranslationJobs オペレーションの `filter` パラメータを使用します。ジョブ名、ジョブステータス、またはジョブが送信された日時で結果をフィルタ処理できます。

Example describe-text-translation-job コマンド

次の例では、 を使用して DescribeTextTranslationJob コマンド AWS CLI を実行して、ジョブのステータスをチェックします。

```
$ aws translate describe-text-translation-job --job-id 1c1838f470806ab9c3e0057f14717bed
```

このコマンドから次の出力が返されます。

```
{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://input-bucket-name/folder"
    },
    "EndTime": 1576551359.483,
    "SourceLanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::012345678901:role/service-role/
AmazonTranslateInputOutputAccess",
    "JobId": "1c1838f470806ab9c3e0057f14717bed",
    "TargetLanguageCodes": [
      "fr"
    ],
    "JobName": "batch-test",
    "SubmittedTime": 1576544017.357,
    "JobStatus": "COMPLETED",
    "Message": "Your job has completed successfully.",
    "JobDetails": {
      "InputDocumentsCount": 77,
      "DocumentsWithErrorsCount": 0,
      "TranslatedDocumentsCount": 77
    },
    "OutputDataConfig": {
      "S3Uri": "s3://bucket-name/output/012345678901-
TranslateText-1c1838f470806ab9c3e0057f14717bed/"
    }
  }
```

```
}
```

StopTextTranslationJob オペレーションを使用して、ステータスが IN_PROGRESS の間にバッチ翻訳ジョブを停止できます。

Example stop-text-translation-job コマンド

次の例では、 を使用して StopTextTranslationJob コマンドを実行して AWS CLI 、 でバッチ翻訳を停止します。

```
$ aws translate stop-text-translation-job --job-id 5236d36ce5192abdb3e2519f3ab8b065
```

このコマンドから次の出力が返されます。

```
{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://input-bucket-name/folder"
    },
    "SourceLanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::012345678901:role/service-role/
AmazonTranslateInputOutputAccess",
    "TargetLanguageCodes": [
      "fr"
    ],
    "JobName": "canceled-test",
    "SubmittedTime": 1576558958.167,
    "JobStatus": "STOP_REQUESTED",
    "JobId": "5236d36ce5192abdb3e2519f3ab8b065",
    "OutputDataConfig": {
      "S3Uri": "s3://output-bucket-name/012345678901-
TranslateText-5236d36ce5192abdb3e2519f3ab8b065/"
    }
  }
}
```

バッチ翻訳結果の取得

ジョブのステータスが COMPLETED または COMPLETED_WITH_ERROR になったら、指定した Amazon S3 フォルダで出力ドキュメントを使用できます。出力ドキュメント名は入力ド

キュメント名と一致し、プレフィックスとしてターゲット言語コードが追加されます。たとえば、mySourceText.txt という名前のドキュメントをフランス語に翻訳した場合、出力ドキュメントは fr.mySourceText.txt という名前になります。

バッチ翻訳ジョブのステータスが FAILED の場合、DescribeTextTranslationJob オペレーションのレスポンスには、ジョブが正常に完了しなかった理由を説明する Message フィールドが含まれます。

各バッチ翻訳ジョブでは、翻訳された文字の合計数や発生したエラーの数など、実行された翻訳に関する情報を含む補助ファイルも生成されます。このファイルは、*target-language-code*.auxiliary-translation-details.json という名前で、出力フォルダの details サブフォルダに生成されます。

以下に示しているのは、バッチ翻訳の補助ファイルの例です。

```
{
  "sourceLanguageCode": "en",
  "targetLanguageCode": "fr",
  "charactersTranslated": "105",
  "documentCountWithCustomerError": "0",
  "documentCountWithServerError": "0",
  "inputDataPrefix": "s3://input-bucket-name/folder",
  "outputDataPrefix": "s3://output-bucket-name/012345678901-
TranslateText-1c1838f470806ab9c3e0057f14717bed/",
  "details": [
    {
      "sourceFile": "mySourceText.txt",
      "targetFile": "fr.mySourceText.txt",
      "auxiliaryData": {
        "appliedTerminologies": [
          {
            "name": "TestTerminology",
            "terms": [
              {
                "sourceText": "Amazon",
                "targetText": "Amazon"
              }
            ]
          }
        ]
      }
    },
    {
      "sourceFile": "batchText.txt",
```

```
      "targetFile": "fr.batchText.txt",
      "auxiliaryData": {
        "appliedTerminologies": [
          {
            "name": "TestTerminology",
            "terms": [
              {
                "sourceText": "Amazon",
                "targetText": "Amazon"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Amazon Translate での翻訳のカスタマイズ

次の設定を使用すれば、Amazon Translate で生成した翻訳をカスタマイズできます。

- Do-not-translateタグ – 開始タグと終了タグを使用して、翻訳しないコンテンツ (HTML コンテンツ) を指定します。
- カスタム用語 – Amazon Translate でブランド名などの特定の用語を翻訳する方法を定義します。
- Brevity – ほとんどの翻訳の翻訳出力の長さを短くします (簡潔でない翻訳出力と比較して) 。リアルタイムテキスト翻訳で簡潔な を使用します。
- 冒涜的 – 翻訳出力で冒涜的な単語やフレーズをマスクします。
- 形式 – 翻訳出力で言語形式のレベルを設定します。
- 並列データ – 指定した翻訳サンプル例のスタイル、トーン、単語の選択を反映するように翻訳出力を調整します。

トピック

- Amazon Translate do-not-translateの使用
- カスタム用語による翻訳のカスタマイズ
- Amazon Translate での簡潔さの使用
- Amazon Translate における不適切な言葉やフレーズのマスキング
- Amazon Translate の形式設定
- 並列データによる翻訳のカスタマイズ (Active Custom Translation)

Amazon Translate do-not-translateの使用

HTML コンテンツの場合、do-not-translateタグを追加できます。この機能は、 コンソールおよび API オペレーションで使用できます。

トピック

- コンソールで翻訳禁止タグを使用する
- API で翻訳禁止タグを使用する

コンソールで翻訳禁止タグを使用する

ソース HTML コンテンツ`translate="no"`で、翻訳しないコンテンツを囲む HTML タグで を指定します。たとえば、次のテキストを英語からドイツ語に翻訳するには、次のようにします。

```
In French, the Louvre Museum is Musée du Louvre.
```

「Musée du Louvre」というテキストはフランス語のままにする必要があるため、スパンタグを使用してこのコンテンツ の翻訳をスキップします。

```
<p>In French, the Louvre Museum  is <span translate="no">Musée du Louvre</span>.</p>
```

この文は、結果としてドイツ語に翻訳されます。

```
<p>Auf Französisch ist <span translate="no">Musée du Louvre</span> das Louvre-Museum.</
p>
```

API で翻訳禁止タグを使用する

do-not-translate は、リアルタイム API オペレーション (`TranslateText` および `TranslateDocument`) および非同期 `StartTextTranslationJob` API オペレーションで使用できます。API リクエストに指定するソーステキスト内で、翻訳する必要のないコンテンツを任意のタイプの HTML 要素で指定することができます。

次の例では、一部のテキストを英語からスペイン語に翻訳しますが、一部のテキストは英語のままにしておきます。

```
aws translate translate-text \
  --source-language-code "en" \
  --target-language-code "es" \
  --region us-west-2 \
  --text "<p>You can translate this paragraph to any language.</p> <p translate=no>But
 do not translate this.</p>"
```

この API リクエストは次のスペイン語の翻訳を返します。

```
{
    "TranslatedText": "<p>Puede traducir este párrafo a cualquier idioma.</p>
```

```
                              <p translate=no>But do not translate this.</p>",
    "SourceLanguageCode": "en",
    "TargetLanguageCode": "es"
}
```

カスタム用語による翻訳のカスタマイズ

翻訳リクエストでカスタム用語を使用すると、ブランド名、キャラクター名、モデル名、およびその他の独自コンテンツを、目的どおりに翻訳することができます。

用語ファイルを作成して Amazon Translate アカウントにアップロードすることができます。ファイルサイズと用語ファイルの数については、「Service Quotas」を参照してください。テキストを翻訳する場合、使用するカスタム用語ファイルを任意で選択できます。Amazon Translate は、ソーステキストと用語ファイルの間で一致を検出すると、用語ファイルにある翻訳を使用します。

次の例を考えてみましょう。Amazon Photos では、Amazon プライム会員に写真と動画の無料ストレージを提供しています。フランス語では、Amazon Photos は翻訳されず原語のまま残されます。

Amazon Translate でコンテキストを追加せずに Amazon Photos をフランス語に翻訳した場合、結果は Photos d'Amazon となりますが、これは望ましい翻訳ではありません。

Amazon Photos という用語にカスタム用語のエントリを追加し、フランス語訳が Amazon Photos であることを指定すると、Amazon Translate はカスタム用語を使用してフレーズを意図したとおりに翻訳します。

Amazon Translate は、すべての翻訳にターゲット用語を使用することを保証するものではありません。カスタム用語は、翻訳コンテキストでソース用語とターゲット用語の意味を使用して、ターゲット用語を使用するかどうかを決定します。詳細については、「ベストプラクティス」を参照してください。

トピック

- カスタム用語の作成
- カスタム用語の使用
- を使用したカスタム用語の例 AWS SDK for Python (Boto)
- 用語の暗号化
- ベストプラクティス

カスタム用語の作成

用語ファイルを作成してカスタム用語を定義します。Amazon Translate は、CSV、TSV、または TMX ファイル形式をサポートしています。ファイル内の各エントリには、ソース用語とそれに対応する (翻訳された) 各ターゲット言語での用語が含まれています。

用語ファイルを作成したら、Amazon Translate アカウントにアップロードします。

> **Important**
>
> カスタム用語のソーステキストでは大文字と小文字が区別されます。Amazon Translate は、翻訳中に入力ドキュメントに完全に一致するものを見つけると、カスタム用語を使用します。

用語ファイルの形式

次の例では、CSV 形式の用語ファイルを示しています。

CSV (カンマ区切り値)

```
en,fr,es
Amazon Photos,Amazon Photos,Amazon Photos
```

次の例では、TMX 形式の用語ファイルを示しています。TMX ファイルは、翻訳ソフトウェアでよく使用される XML 形式を使用しています。

TMX (Translation Memory eXchange)

```
<?xml version="1.0" encoding="UTF-8"?>
 <tmx version="1.4">
  <header
     creationtool="XYZTool" creationtoolversion="0"
     datatype="PlainText" segtype="sentence"
     adminlang="en-us" srclang="en"
     o-tmf="test"/>
  <body>
    <tu>
      <tuv xml:lang="en">
        <seg>Amazon Photos</seg>
      </tuv>
```

```
      <tuv xml:lang="fr">
        <seg>Amazon Photos</seg>
      </tuv>
      <tuv xml:lang="es">
        <seg>Amazon Photos</seg>
      </tuv>
    </tu>
  </body>
</tmx>
```

方向性

カスタム用語ファイルをアップロードするときは、カスタム用語の方向性の値を設定します。方向性とは、用語ファイルが 1 つのソース言語を指定しているのか、複数のソース言語を指定しているのかを示します。

方向性を指定するには、次のいずれかの値を設定します。

Uni-directional (単一方向)

用語ファイルには 1 つのソース言語 (リストの最初の言語) が含まれています。その他の言語はすべてターゲット言語です。

例えば、CSV ファイルの場合、最初の列にソース言語のテキストが記載され、他のすべての列にはターゲット言語のテキストが記載されます。

Multi-directional (多方向)

ファイル内の任意の言語が、ソース言語またはターゲット言語になります。例えば、用語ファイルに英語、フランス語、スペイン語のテキストが含まれている場合、次の言語ペアを翻訳するジョブでファイルを使用できます。

- 英語 → フランス語
- 英語 → スペイン語
- フランス語 → 英語
- フランス語 → スペイン語
- スペイン語 → 英語
- スペイン語 → フランス語

一方で、これら 6 つの翻訳ジョブに対して単一方向の用語ファイルを 3 つ (ソース言語ごとに 1 つずつ) 作成する必要があります 。

カスタム用語の使用

TranslateText オペレーションでテキストを翻訳するときにカスタム用語を使用するには、オプションの TerminologyNames パラメータを含めます。

例えば、Amazon_Family.csv という次の用語ファイルをアカウントにアップロードするとします。

```
    en,fr
    Amazon Family,Amazon Famille
```

以下の CLI コマンドによって、カスタム用語を使用してテキストを翻訳することができます。

> **Note**
>
> この例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

```
aws translate translate-text \
    --region region \
    --source-language-code "en" \
    --target-language-code "fr" \
    --terminology-names "Amazon_Family" \
    --text "Have you ever stored videos in Amazon Family?"
```

ここでは、選択したカスタム用語を使用してこのテキストが「Avez-vous déjà fait des achats avec Amazon Famille?」と訳されます。望ましくない直訳「Avez-vous déjà fait des achats avec Famille Amazon?」にはなりません。

次の例では、同じ用語ファイルを Python で使用する方法を示してます。

```
import boto3

translate = boto3.client(service_name='translate')

print("Translating 'Have you ever shopped with Amazon Family?' from English to French
 with the 'Amazon_Family' custom terminology...")
response = translate.translate_text(Text="Have you ever shopped with Amazon Family?",
 TerminologyNames=["Amazon_Family"], SourceLanguageCode="en", TargetLanguageCode="fr")
```

```
print("Translated text: " + response.get('TranslatedText'))
print("\n")
```

カスタム用語を使った Amazon Translate オペレーションの使用の詳細については、「API オペレーション」を参照してください。

を使用したカスタム用語の例 AWS SDK for Python (Boto)

次の例は、Python でカスタム用語オペレーションを使用する方法を示しています。この例を実行するには、 経由で Python SDK をインストールします AWS CLI。手順については、the section called “のセットアップ AWS CLI” を参照してください。

```
    #!/usr/bin/env python
    # -*- coding: utf-8 -*-

    import boto3

    translate = boto3.client(service_name='translate')

    # The terminology file 'my-first-terminology.csv' has the following contents:
    '''
    en,fr
    Amazon Family,Amazon Famille
    '''

    # Read the terminology from a local file
    with open('/tmp/my-first-terminology.csv', 'rb') as f:
        data = f.read()

    file_data = bytearray(data)

    print("Importing the terminology into Amazon Translate...")
    response = translate.import_terminology(Name='my-first-terminology',
 MergeStrategy='OVERWRITE', TerminologyData={"File": file_data, "Format": 'CSV'})
    print("Terminology imported: "),
    print(response.get('TerminologyProperties'))
    print("\n")

    print("Getting the imported terminology...")
    response = translate.get_terminology(Name='my-first-terminology',
 TerminologyDataFormat='CSV')
    print("Received terminology: "),
    print(response.get('TerminologyProperties'))
```

```
    print("The terminology data file can be downloaded here: " +
 response.get('TerminologyDataLocation').get('Location'))
    print("\n")

    print("Listing the first 10 terminologies for the account...")
    response = translate.list_terminologies(MaxResults=10)
    print("Received terminologies: "),
    print(response.get('TerminologyPropertiesList'))
    print("\n")

    print("Translating 'Amazon Family' from English to French with no terminology...")
    response = translate.translate_text(Text="Amazon Family", SourceLanguageCode="en",
 TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    print("Translating 'Amazon Family' from English to French with the 'my-first-
terminology' terminology...")
    response = translate.translate_text(Text="Amazon Family", TerminologyNames=["my-
first-terminology"], SourceLanguageCode="en", TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    # The terminology file 'my-updated-terminology.csv' has the following contents:
    '''
    en,fr
    Amazon Family,Amazon Famille
    Prime Video, Prime Video
    '''

    # Read the terminology from a local file
    with open('/tmp/my-updated-terminology.csv', 'rb') as f:
        data = f.read()

    file_data = bytearray(data)

    print("Updating the imported terminology in Amazon Translate...")
    response = translate.import_terminology(Name='my-first-terminology',
 MergeStrategy='OVERWRITE', TerminologyData={"File": file_data, "Format": 'CSV'})
    print("Terminology updated: "),
    print(response.get('TerminologyProperties'))
    print("\n")

    print("Translating 'Prime Video' from English to French with no terminology...")
```

```
    response = translate.translate_text(Text="Prime Video", SourceLanguageCode="en",
 TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    print("Translating 'Prime Video' from English to French with the 'my-first-
terminology' terminology...")
    response = translate.translate_text(Text="Prime Video", TerminologyNames=["my-
first-terminology"], SourceLanguageCode="en", TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    print("Cleaning up by deleting 'my-first-terminology'...")
    translate.delete_terminology(Name="my-first-terminology")
    print("Terminology deleted.")
```

用語の暗号化

Amazon Translate はお客様のすべてのデータを保護するべく取り組んでおり、カスタム用語についても同様です。各カスタム用語は、作成時に暗号化され、お客様のみがアクセス可能になります。

使用できる暗号化オプションは 3 つです。

- encryption AWS . AWS encryption は、情報を保護するためのデフォルトのオプションです。
- お客様のアカウントに関連付けられた暗号化キーの使用。コンソールのメニューで、使用する関連付けられた暗号化キーを選択できます。
- お客様のアカウントに関連付けられていない暗号化キーの使用。暗号化キーの Amazon リソースネーム (ARN) を入力する入力フィールドがコンソールに表示されます。

ベストプラクティス

カスタム用語を使用する場合は、次の一般的なベストプラクティスを使用します。

- カスタム用語は整理しておきましょう。翻訳された値を制御する必要がある用語のみを含めてください。
- カスタム用語は大文字と小文字が区別されます。大文字を使ったバージョンと小文字を使ったバージョンに単語を翻訳する必要がある場合は、各バージョンのエントリを含めます。
- カスタム用語は、スペース、句読点、大文字と小文字の使い分けを制御するためのツールではありません。例えば、次のタイプのエントリは避けてください。

  - スペースの追加 – EN: USA FR: U S A
  - 句読点の追加 – EN: USA FR: U.S.A
  - 大文字の変更 – EN: USA FR: Usa
- 同じソースフレーズに対して複数の異なる翻訳を含めないでください。以下に例を示します。
  - エントリ 1 – EN: Amazon、FR: Amazon
  - エントリ 2 – EN: Amazon、FR: Amazone
- Amazon Translate でサポートされているすべての言語のカスタム用語ファイルを作成できます。

Amazon Translate は、カスタム用語がすべての翻訳にターゲット用語を使用することを保証するものではありません。カスタム用語で高い精度を実現するには、用語ファイルのコンテンツを作成するときに以下のベストプラクティスに従ってください。

- カスタム用語は、動詞や同形異義語を含むあらゆる単語に適しています。ブランド名や製品名などの固有名詞は理想的なエントリです。
- 訳語はターゲット言語で流暢に読める必要があります。カスタム用語は、多数の特殊文字やフォーマットを含むターゲット用語には推奨されません。
- 用語ファイルには、複数の単語からなるフレーズや節を含めることができます。ただし、複数の単語を含む用語は、ターゲット言語で流暢に読める可能性が低くなります。
- カスタム用語は、翻訳コンテキストでソース用語とターゲット用語の意味を使用して、ターゲット用語を使用するかどうかを決定します。ターゲット用語が特定の翻訳コンテキストに適切でない場合、Amazon Translate はターゲット用語を使用しないことがあります。

  たとえば、用語ファイルに英語からフランス語への次のエントリが含まれている場合：

  `EN: order, FR: commande` (フランス語の「コマンドャー」に変換される英語の「注文」に基づく)。

  翻訳コンテキストが一致しないため、Amazon Translate は次の文を翻訳するときにこのエントリを使用しません。

  「お客様のために、お客様の名前を共有してください。」

  この種の状況を回避するための提案：

  - 各言語のターゲット用語がソース用語と意味的に同等であることを確認します。
  - 複数の意味を持つソース用語やターゲット用語は避けてください。

Amazon Translate での簡潔さの使用

言語間で翻訳する場合、翻訳出力が目的よりも長い (文字数) ことがあります。出力が長いと、追加の文字用のスペースがない場合、一部のシナリオ (字幕、字幕、見出し、フォームフィールドなど) で問題が発生する可能性があります。

Amazon Translate でリアルタイムテキスト翻訳を実行するときに、簡潔な設定を有効にできます。Brevity は、ほとんどの翻訳の翻訳出力の長さを短くします (簡潔でない翻訳出力と比較して) 。

トピック

- 簡潔さ設定の使用
- サポートされている言語

簡潔さ設定の使用

リアルタイムテキスト翻訳では、簡潔な 設定を使用できます。Amazon Translate は、リアルタイムドキュメント翻訳や非同期翻訳ジョブの簡潔さをサポートしていません。

リアルタイムテキスト翻訳リクエストで簡潔性を使用するには、次のいずれかを実行します。

- コンソール – リアルタイム翻訳ページのテキストタブの「追加設定」で、「詳細度」設定を選択します。
- AWS CLI – `translate-text` コマンドで、 `--settings`パラメータの簡潔さを に設定します`Brevity=ON`。詳細については、AWS CLI コマンドリファレンスの「translate-text」を参照してください。
- AWS API – TranslateText API オペレーションで、設定パラメータで簡潔さを設定します。

サポートされている言語

Amazon Translate では、次の言語ペアの簡潔さがサポートされています。

- 任意のソース言語から次の表のいずれかの言語へ。
- 次の表のいずれかの言語から英語へ。

言語	言語コード
フランス語	fr
ドイツ語	de
イタリア語	it
ポルトガル語 (ブラジル)	pt
スペイン語	es

サポートされていない言語ペアで翻訳の簡潔さをリクエストすると、翻訳は簡潔さの設定をオフにして続行されます。

Amazon Translate における不適切な言葉やフレーズのマスキング

Amazon Translate で翻訳を実行する際に不敬設定を有効にすると、翻訳出力内の不適切な単語とフレーズが隠されます (マスク)。

不適切な単語やフレーズを隠す場合、Amazon Translate ではそれらがグローリクス文字列「?$#@$」に置換されます。この 5 文字のシーケンスは、不適切な各単語または各フレーズに対して、その長さや数に関係なく使用されます。

Amazon Translate では、ソース言語とターゲット言語が同一である翻訳リクエストには不敬マスキングは実行されません。

場合によっては、ソース入力内の不適切な単語が翻訳出力内で自然に無害な表現になることがあります。このような場合、マスキングは適用されません。

Amazon Translate では、文脈上ではなく、文字通り不敬な単語やフレーズを 1 つ 1 つ検出します。つまり、不適切な言葉は、文脈上は無害である場合でも、マスクされる可能性があるということです。たとえば、Amazon Translate により「jerk」(ばか) が不適切な単語として検出される場合、「jerk chicken」(鶏肉料理の一種) というフレーズは、無害な言葉であっても「?$#@$ chicken」と書き込まれます。(ここでは「jerk」をあくまで一例として使用しています。 Amazon Translate ではこの単語は不敬語として検出されません。)

トピック

- 不敬表現設定の使用

- サポートされていない言語

不敬表現設定の使用

不敬設定は、Amazon Translate のリアルタイム翻訳と非同期バッチ処理という翻訳オペレーションの両方で使用できます。

リアルタイム翻訳リクエストで不敬語をマスクするには、次のいずれかの作業を実行します。

- Amazon Translate コンソールの [Real-time translation (リアルタイム翻訳)] ページの [Additional settings (追加設定)] で、[Profanity (不敬)] 設定を有効にします。
- の `translate-text` コマンドで AWS CLI、 `--settings`パラメータを に設定しますProfanity=MASK。詳細については、AWS CLI コマンドリファレンスの「translate-text」を参照してください。
- Amazon Translate API の `TranslateText`アクションで、 `Settings`パラメータを使用して冒涜的なマスキングを設定します。詳細については、 API リファレンスのTranslateText」を参照してください。

非同期バッチ操作で不敬語をマスクするには、バッチ翻訳ジョブの実行 を参照してください。

サポートされていない言語

Amazon Translate でサポートされているターゲット言語を翻訳する場合は不敬表現をマスクできます。ただし、次の例外があります。

言語	言語コード
ベンガル語	bn
ヒンディー語	hi
マラヤーラム語	ml
パンジャブ語	pa
シンハラ語	si
ベトナム語	vi

Amazon Translate でサポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。

Amazon Translate の形式設定

必要に応じて、サポートされているターゲット言語への翻訳に必要な形式のレベルを指定できます。形式設定は、翻訳出力における形式的な言語の使用レベル (敬語または言語使用域とも呼ばれます) を制御します。形式設定は、リアルタイム翻訳および非同期バッチ処理で利用できます。

形式では、以下の値がサポートされています。

- インフォーマル — 翻訳されたテキスト内のすべての文で、インフォーマルなコミュニケーションに関連する言語構造が使用されます。例えば、翻訳されたテキストでは、親しみのある二人称代名詞とその動詞の一致が使用されます (日本語の場合はくだけた表現)。
- フォーマル — 翻訳されたテキスト内のすべての文で、フォーマルで丁寧なコミュニケーションに関連する言語構造が使用されます。例えば、翻訳されたテキストでは、格式ばった二人称代名詞とその動詞の一致が使用されます (日本語の場合は丁寧語)。

「Are you sure? (よろしいですか)」という文を例に挙げます。ドイツ語では 2 つの正しい翻訳が可能です。フォーマルな言語使用域の場合は「Sind Sie sicher?」、インフォーマルな場合は「Bist du sicher?」となります。

Amazon Translate がターゲット言語の形式レベルをサポートしていない場合、または形式パラメータが指定されなかった場合、翻訳ジョブは形式設定を無視します。

トピック

- 形式設定の使用
- サポートされている言語

形式設定の使用

リアルタイムの翻訳リクエストで形式を設定するには、次のいずれかの手順を実行します。

- Amazon Translate コンソールの [リアルタイムでの翻訳] ページで、[追加設定] から [形式] 設定を有効にして、いずれかの値を選択します。
- Amazon Translate API で TranslateText オペレーションの [設定] パラメータを使用します。

- の `translate-text` コマンドで AWS CLI、 `--settings`パラメータを `Formality=FORMAL`または に設定します`Formality=INFORMAL`。詳細については、AWS CLI コマンドリファレンスの「translate-text」を参照してください。

バッチ翻訳リクエストの形式を設定するには、翻訳ジョブを開始するときに [形式] パラメータを設定します。詳細と例については、「バッチ翻訳ジョブの実行」を参照してください。

CLI または API リクエストの場合、レスポンスの `AppliedSettings` フィールドにはリクエストの形式設定 (ある場合) が含まれます。ターゲット言語が形式をサポートしていない場合、レスポンスの `AppliedSettings` の値は NULL です。

サポートされている言語

Amazon Translate は、任意のソース言語から以下のターゲット言語への翻訳の形式設定をサポートしています。

言語	言語コード
オランダ語	nl
フランス語	fr
フランス語 (カナダ)	fr-CA
ドイツ語	de
ヒンディー語	hi
イタリア語	it
日本語	ja
韓国語	ko
ポルトガル語 (ポルトガル)	pt-PT
スペイン語	es
スペイン語 (メキシコ)	es-MX

Amazon Translate でサポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。

並列データによる翻訳のカスタマイズ (Active Custom Translation)

並列データを Amazon Translate に追加してバッチ翻訳ジョブの出力をカスタマイズします。並列データは、テキストのセグメントの翻訳方法を示した例で構成されています。これには、ソース言語でのテキスト例の一群が含まれており、各例については、1 つ以上のターゲット言語での対象の翻訳出力が含まれています。

並列データをバッチ翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。これらのジョブを実行すると、Amazon Translate ではランタイムに並列データが使用されて、カスタマイズされた機械翻訳出力が生成されます。並列データで見つかったスタイル、トーン、単語の選択肢を反映させるために翻訳を適応させます。並列データを使用すると、ライフサイエンス、法律、財務など、特定のドメインに固有の用語やフレーズに合わせて翻訳を調整できます。

> **Note**
>
> Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「Amazon Translate 料金表」を参照してください。

たとえば、次の並列データは CSV ファイルで定義されています。

```
"en","fr"
"How are you?","Comment ça va ?"
```

この例では、英語 (en) がソース言語で、フランス語 (`fr`) がターゲット言語です。この例は、ソースフレーズ "How are you?" が どのようにしてフランス語に翻訳されるかを示しています。この例の入力ファイルは、Amazon Translate にインポートされた後で翻訳ジョブに適用されて、出力に影響を及ぼす可能性があります。このジョブの中で、Amazon Translate では "How are you?" が 日常的な “Comment ça va ?” に翻訳されます。これは格式ばった“Comment allez-vous ?”とは対照的です。たとえば、このジョブで次のソーステキストを受け取ったとします。

```
Hello, how are you?
How are you?
Hi, how are you?
```

```
How are you doing?
```

このテキストから、ジョブでは次の翻訳が生成されます。

```
Bonjour, comment ça va ?
Comment ça va ?
Salut, comment ça va ?
Comment ça va ?
```

対照的に、並列データを使用せずにジョブが実行された場合、出力には硬めの表現である "comment allez-vous" が含まれる可能性があります。

```
Bonjour, comment allez-vous ?
Comment allez-vous ?
Salut, comment allez-vous ?
Comment allez-vous ?
```

並列データを使用してバッチ翻訳ジョブをカスタマイズすることで、翻訳例を使って学習させたカスタム翻訳モデルを使用するのと同じような感覚で、出力に影響を与えることができます。Active Custom Translation にはカスタムモデルのトレーニングは不要であるため、このようなトレーニングにかかる時間と費用を節約できます。翻訳要件の経時的な変化に伴い、並列データを更新して出力を絞り込むことができます。これはカスタムモデルの再トレーニングよりも簡単です。

利用可能なリージョン

Active Custom Translation は以下のリージョンで使用できます。

- 米国東部 (バージニア北部)
- 米国西部 (オレゴン)
- 欧州 (アイルランド)

トピック

- Amazon Translate の並列データ入力ファイル
- Amazon Translate への並列データの追加
- Amazon Translate における並列データの表示と管理

Amazon Translate の並列データ入力ファイル

Amazon Translate で並列データリソースを作成する前に、翻訳例を含む入力ファイルを作成する必要があります。並列データ入力ファイルでは、Amazon Translate でサポートされている言語を使用する必要があります。これらの言語のリストについては、「サポートされている言語と言語コード」を参照してください。

並列データの例

次の表に示すテキストは、並列データ入力ファイルにフォーマットできる翻訳セグメントの例を示しています。

en	es
Amazon Translate is a neural machine translati on service。	Amazon Translate es un servicio de traducción automática basado en redes neuronales。
Neural machine translation is a form of language translation automation that uses deep learning models。	La traducción automática neuronal es una forma de automatizar la traducción de lenguajes utilizando modelos de aprendizaje profundo。
Amazon Translate allows you to localize content for international users。	Amazon Translate le permite localizar contenido para usuarios internacionales。

表の最初の行に表示されているのは言語コードです。1 つ目の言語である英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。1列目にはソーステキストの例が示されています。もう一つの列には翻訳の例が記載されています。この並列データでバッチジョブをカスタマイズすると、Amazon Translate ではその例が反映されて翻訳が適応されます。

入力ファイル形式

Amazon Translate では、並列データ入力ファイルに対して次の形式がサポートされています。

- Translation Memory eXchange (TMX)
- カンマ区切り値 (CSV)
- タブ区切り値 (TSV)

TMX

Example TMX 入力ファイル

次のサンプル TMX ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<tmx version="1.4">
    <header srclang="en"/>
    <body>
        <tu>
            <tuv xml:lang="en">
                <seg>Amazon Translate is a neural machine translation service.</seg>
            </tuv>
            <tuv xml:lang="es">
                <seg>Amazon Translate es un servicio de traducción automática basado
 en redes neuronales.</seg>
            </tuv>
        </tu>
        <tu>
            <tuv xml:lang="en">
                <seg>Neural machine translation is a form of language translation
 automation that uses deep learning models.</seg>
            </tuv>
            <tuv xml:lang="es">
                <seg>La traducción automática neuronal es una forma de automatizar
 la traducción de lenguajes utilizando modelos de aprendizaje profundo.</seg>
            </tuv>
        </tu>
        <tu>
            <tuv xml:lang="en">
                <seg>Amazon Translate allows you to localize content for
 international users.</seg>
            </tuv>
            <tuv xml:lang="es">
                <seg>Amazon Translate le permite localizar contenido para usuarios
 internacionales.</seg>
            </tuv>
        </tu>
```

```
    </body>
</tmx>
```

TMX の要件

TMX ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- Amazon Translate では TMX 1.4b がサポートされています。詳細については、Globalization and Localization Association ウェブサイトの TMX 1.4b の仕様 を参照してください。
- `header` 要素には必ず `srclang` 属性を含めてください。この属性の値によって並列データのソース言語が決まります。
- `body` 要素には必ず、少なくとも 1 つの翻訳単位 (`tu`) 要素を含めてください。
- 各 `tu` 要素には必ず、少なくとも 2 つの翻訳単位バリアント (`tuv`) 要素が含めてください。これらの `tuv` 要素のうちの 1 つに、必ず、`header` 要素の `srclang` 属性に割り当てられている値と値が同じである `xml:lang` 属性を含めてください。
- すべての `tuv` 要素に必ず `xml:lang` 属性を含めてください。
- すべての `tuv` 要素に必ずセグメント (`seg`) 要素を含めてください。
- 入力ファイルの処理中、空である、またはホワイトスペースしかない `seg` 要素が発生した場合は、Amazon Translate では特定の `tu` 要素または `tuv` 要素がスキップされます。
  - `seg` 要素がソース言語に対応している場合は、Amazon Translate では `seg` 要素が占めている `tu` 要素がスキップされます。
  - `seg` 要素がターゲット言語に対応している場合は、Amazon Translate では `seg` 要素が占めている `tuv` 要素のみがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えている `seg` 要素が発生した場合は、Amazon Translate では特定の `tu` 要素または `tuv` 要素がスキップされます。
  - `seg` 要素がソース言語に対応している場合は、Amazon Translate では `seg` 要素が占めている `tu` 要素がスキップされます。
  - `seg` 要素がターゲット言語に対応している場合は、Amazon Translate では `seg` 要素が占めている `tuv` 要素のみがスキップされます。
- ソーステキストが同一である複数の `tu` 要素が入力ファイルに含まれている場合、Amazon Translate では次のいずれかの処理が実行されます。
  - `tu` 要素に `changedate` 属性が含まれている場合は、直近の日付の要素が使用されます。
  - それ以外の場合は、そのファイルの最後に最も近い時期に生じる要素が使用されます。

CSV

次のサンプル CSV ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

Example CSV 入力ファイル

```
en,es
Amazon Translate is a neural machine translation service.,Amazon Translate es un
 servicio de traducción automática basado en redes neuronales.
Neural machine translation is a form of language translation automation that uses
 deep learning models.,La traducción automática neuronal es una forma de automatizar
 la traducción de lenguajes utilizando modelos de aprendizaje profundo.
Amazon Translate allows you to localize content for international users.,Amazon
 Translate le permite localizar contenido para usuarios internacionales.
```

CSV 要件

CSV ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- 1 行目は言語コードで構成されます。最初のコードはソース言語で、後続の各コードはターゲット言語です。
- 1 列目の各フィールドにはソーステキストが記載されています。後続の列の各フィールドにはターゲット翻訳が記載されています。
- いずれかのフィールドのテキストにカンマが含まれている場合、そのテキストを二重引用符 (") で囲む必要があります。
- テキストフィールドを複数の行に渡って表示することはできません。
- フィールドの最初の文字として +、-、=、@ は使用できません。この要件は、フィールドが二重引用符 (") で囲まれているかどうかにかかわらず適用されます。
- フィールド内のテキストに二重引用符 (") が含まれている場合は、二重引用符でエスケープする必要があります。たとえば、次のようなテキストがあるとします。

  ```
  34" monitor
  ```

  次のように記述する必要があります。

```
34"" monitor
```

- 入力ファイルの処理中、空である、またはホワイトスペースしかないフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
  - ソーステキストフィールドが空である場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
  - ターゲット翻訳フィールドが空白である場合、Amazon Translate ではそのフィールドだけがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えているフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
  - あるソーステキストフィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
  - あるターゲット翻訳フィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドだけがスキップされます。
- ソーステキストが同じである複数のレコードが入力ファイルに含まれている場合、Amazon Translate ではそのファイルの最後に最も近いレコードが使用されます。

TSV

次のサンプル TSV ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

Example TSV 入力ファイル

```
en es
Amazon Translate is a neural machine translation service. Amazon Translate es un
 servicio de traducción automática basado en redes neuronales.
Neural machine translation is a form of language translation automation that uses
 deep learning models. La traducción automática neuronal es una forma de automatizar
 la traducción de lenguajes utilizando modelos de aprendizaje profundo.
Amazon Translate allows you to localize content for international users. Amazon
 Translate le permite localizar contenido para usuarios internacionales.
```

TSV 要件

TSV ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- 1 行目は言語コードで構成されます。最初のコードはソース言語で、後続の各コードはターゲット言語です。
- 1 列目の各フィールドにはソーステキストが記載されています。後続の列の各フィールドにはターゲット翻訳が記載されています。
- いずれかのフィールドのテキストにタブ文字が含まれている場合、そのテキストを二重引用符 (") で囲む必要があります。
- テキストフィールドを複数の行に渡って表示することはできません。
- フィールドの最初の文字として +、-、=、@ は使用できません。この要件は、フィールドが二重引用符 (") で囲まれているかどうかにかかわらず適用されます。
- フィールド内のテキストに二重引用符 (") が含まれている場合は、二重引用符でエスケープする必要があります。たとえば、次のようなテキストがあるとします。

  ```
  34" monitor
  ```

  次のように記述する必要があります。

  ```
  34"" monitor
  ```

- 入力ファイルの処理中、空である、またはホワイトスペースしかないフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
  - ソーステキストフィールドが空である場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
  - ターゲット翻訳フィールドが空白である場合、Amazon Translate ではそのフィールドだけがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えているフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
  - あるソーステキストフィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
  - あるターゲット翻訳フィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドだけがスキップされます。

- ソーステキストが同じである複数のレコードが入力ファイルに含まれている場合、Amazon Translate ではそのファイルの最後に最も近いレコードが使用されます。

Amazon Translate への並列データの追加

Amazon Translate に並列データを追加するには、Amazon S3 から並列データ入力ファイルをインポートします。その後、並列データを使用して、バッチ翻訳ジョブによって生成される出力をカスタマイズできます。

> **前提条件**
>
> Amazon Translate に並列データを追加する前に、次の作業を行う必要があります。
>
> - 並列データ入力ファイルを用意してください。作成する場合は「Amazon Translate の並列データ入力ファイル」を参照してください。
> - AWS アカウントに Amazon S3 バケットがある。作成する場合は、Amazon Simple Storage Service ユーザーガイドの「S3 バケットを作成する方法」を参照してください。
> - 入力ファイルを Amazon S3 バケットにアップロードしてください。詳細については、Amazon Simple Storage Service ユーザーガイドの「S3 バケットにファイルとフォルダをアップロードする方法」を参照してください。

並列データの追加 (Amazon Translate コンソール)

Amazon Translate コンソールを使用して並列データを追加するには、並列データページを使用します。

1. Amazon Translate コンソール を開きます。
2. 左側のナビゲーションメニューで、[Customization (カスタマイズ)] を選択し、[Parallel data (並列データ)] を選択します。
3. [Parallel data (並列データ)] ページで [Create parallel data (並列データの作成)] を選択します。コンソールに [Create parallel data (並列データの作成)] ページが表示されます。
4. 以下を指定します。

   Name (名前)

   並列データリソースのカスタム名。アカウントとリージョンで一意の名前を割り当てる必要があります。

Description (説明) - オプション

カスタム説明。

Parallel data location on S3 (S3 での並列データの場所)

Amazon S3 での並列データ入力ファイルの場所。Amazon S3 のファイルに移動して場所を設けるには、[Select file (ファイルを選択)] を選択します。

File format (ファイル形式)

並列データ入力ファイルの形式。サポートされている形式は、Translation Memory eXchange (TMX)、カンマ区切り値 (CSV)、タブ区切り値 (TSV) です。

5. 暗号化キーで、 AWS KMS key を選択して並列データを保護します。これらの KMS キーは AWS Key Management Service () によって管理されますAWS KMS。詳細については AWS KMS、「 AWS Key Management Service デベロッパーガイド」を参照してください。

Use AWS owned key (AWS 所有キーを使用する)

Amazon Translate によって所有され管理されている KMS キーを使用します。これはデフォルトのオプションであり、他の方法を選択しない場合に情報を暗号化するために使用されます。詳細については、AWS Key Management Service デベロッパーガイドの「AWS 所有キー」を参照してください。

Use key from current account (現在のアカウントのキーを使用する)

AWS アカウント AWS KMS で で管理する KMS キーのいずれかを使用します。このオプションを選択すると、選択可能な KMS キーのリストがメニューに表示されます。詳細については、AWS Key Management Service デベロッパーガイドの「カスタマーマネージドキー」を参照してください。

Use key from different account (別のアカウントのキーを使用する)

別の AWS アカウントの AWS KMS で管理されている KMS キーを使用します。このオプションを選択した場合、KMS キーの Amazon リソースネーム (ARN) を入力するためのフィールドがコンソールに表示されます。

暗号化キーの詳細については、AWS Key Management Service デベロッパーガイドを参照してください。

6. [Create parallel data (並列データの作成)] を選択します。

コンソールが [Parallel data (並列データ)] ページに戻ります。このページの上部にあるバナーにインポートステータスが表示されます。数分後、並列データリソースが表に表示されます。[Status (ステータス)] 列の値が [Active (アクティブ)] である場合、並列データがバッチ翻訳ジョブで使用できる状態です。

トラブルシューティング用エラーファイル

Amazon Translate で入力ファイルの処理中にエラーや警告が発生した場合は、コンソールからエラーファイルが提供され、これをダウンロードするとエラーメッセージを確認できます。次に示すのは、エラーファイルのコンテンツの例です。

```
{
  "summary": {
    "record_error_count": 1,
    "record_skipped_count": 0
  },
  "messages": [
    {
      "content": "Number 1 TU element",
      "message": "Invalid TMX format. One tu element should contain exactly one tuv
 element with the source language code: en"
    }
  ]
}
```

並列データの追加 (AWS CLI)

を使用して並列データを追加するには AWS CLI、 `create-parallel-data` コマンドを使用します。

Example create-parallel-data コマンド

次の例では、Amazon S3 から TSV ファイルをインポートして並列データオブジェクトを作成します。

```
$ aws translate create-parallel-data \
> --name my-parallel-data \
> --parallel-data-config S3Uri=s3://input-bucket/parallel-data-file.tsv,Format=TSV
```

コマンドが成功すると、Amazon Translate から、新しい並列データオブジェクトのステータスでの応答があります。

```
{
    "Name": "my-parallel-data",
    "Status": "CREATING"
}
```

`get-parallel-data` コマンドを使用して並列データの進行状況を監視することができます。ステータスが `ACTIVE` である場合、並列データがバッチ翻訳ジョブで使用できる状態です。`get-parallel-data` コマンドの例については「並列データオブジェクトの詳細を表示する方法」を参照してください。

並列データの使用

並列データリソースを作成したので、バッチ翻訳ジョブに適用して出力をカスタマイズすることができます。バッチジョブを実行するには、バッチ翻訳ジョブの実行 を参照してください。

Amazon Translate における並列データの表示と管理

Amazon Translate に追加したすべての並列データリソースを表示でき、各リソースの詳細なサマリーにアクセスできます。翻訳要件の変化に伴い、並列データを更新して翻訳出力を絞り込むことができます。

並列データの表示と管理 (Amazon Translate コンソール)

Amazon Translate コンソールで並列データの表示と管理を行うには、並列データページを使用します。

並列データリソースの一覧を表示する方法

1. Amazon Translate コンソール を開きます。
2. 左側のナビゲーションメニューで、[Customization (カスタマイズ)] を選択し、[Parallel data (並列データ)] を選択します。このページの表には、Amazon Translate に追加した並列データリソースが表示されます。

並列データリソースの詳細を表示する方法

- [Parallel data (並列データ)] ページの [Name (名前)] 列で、並列データリソースの名前を選択します。コンソールで詳細ページを開きます。この詳細ページには、ステータス、最終更新日、ソース言語、ターゲット言語などの情報が含まれています。

並列データリソースを更新する方法

1. 並列データの更新バージョンを、新しい入力ファイルとして Amazon S3 バケットにアップロードします。
2. Amazon Translate コンソールで、[Parallel data (並列データ)] ページに進みます。
3. 更新する並列データを選択し、[Update (更新)] を選択します。コンソールに [Update parallel data (並列データの更新)] ページが表示されます。
4. 以下を指定します。

   Description (説明) - オプション

   更新された説明。

   Parallel data location on S3 (S3 での並列データの場所)

   Amazon S3 での更新済み並列データ入力ファイルの場所。Amazon S3 のファイルに移動して場所を設けるには、[Select file (ファイルを選択)] を選択します。

   Select parallel data file format (並列データファイル形式の選択)

   並列データ入力ファイルの形式。サポートされている形式は、Translation Memory eXchange (TMX)、カンマ区切り値 (CSV)、タブ区切り値 (TSV) です。
5. [Save (保存)] を選択します。Amazon Translate により古い並列データが新しい入力ファイルに置き換えられます。

並列データの表示と管理 (AWS CLI)

を使用して AWS CLI 、並列データリソースを表示および更新できます。

並列データリソースの一覧を表示する方法

Amazon Translate に追加した並列データリソースのリストを表示するには、`list-parallel-data` コマンドを使用します。

Example list-parallel-data コマンド

次の例では並列データリソースとそのプロパティの一覧が返されます。

```
$ aws translate list-parallel-data
```

コマンドが成功すると、Amazon Translate から以下のような配列が返されます。

```
{
    "ParallelDataPropertiesList": [
        {
            "Name": "my-parallel-data",
            "Arn": "arn:aws:translate:us-west-2:111122223333:parallel-data/my-parallel-
data",
            "Status": "ACTIVE",
            "SourceLanguageCode": "en",
            "TargetLanguageCodes": [
                "es",
                "ja",
                "zh"
            ],
            "ParallelDataConfig": {
                "S3Uri": "s3://input-bucket/parallel-data-file.tsv",
                "Format": "TSV"
            },
            "ImportedDataSize": 2283,
            "ImportedRecordCount": 3,
            "FailedRecordCount": 0,
            "CreatedAt": 1598597751.406,
            "LastUpdatedAt": 1598597911.675
        }
    ]
}
```

並列データオブジェクトの詳細を表示する方法

単一の並列データリソースの詳細を調べるには、`get-parallel-data` コマンドを使用します。このコマンドにより、並列データのプロパティと、作成に使用された入力ファイルをダウンロードできる署名付き S3 URL が返されます。

Example get-parallel-data コマンド

次の例では、`my-parallel-data` オブジェクトのプロパティとダウンロード先を取得します。

```
$ aws translate get-parallel-data \
> --name my-parallel-data
```

コマンドが成功すると、Amazon Translate によりプロパティとダウンロード先が返されます。

```
{
    "ParallelDataProperties": {
        "Name": "my-parallel-data",
        "Arn": "arn:aws:translate:us-west-2:111122223333:parallel-data/my-parallel-
data",
        "Status": "ACTIVE",
        "SourceLanguageCode": "en",
        "TargetLanguageCodes": [
            "es",
            "ja",
            "zh"
        ],
        "ParallelDataConfig": {
            "S3Uri": "s3://input-bucket/parallel-data-file.tsv",
            "Format": "TSV"
        },
        "ImportedDataSize": 2283,
        "ImportedRecordCount": 3,
        "FailedRecordCount": 0,
        "CreatedAt": 1598597751.406,
        "LastUpdatedAt": 1598597911.675
    },
    "DataLocation": {
        "RepositoryType": "S3",
        "Location": "pre-signed S3 URL"
    }
}
```

並列データリソースを更新する方法

並列データリソースを更新するには、まず、新しい入力ファイルを Amazon S3 入力バケットにアップロードします。次に、update-parallel-data コマンドを使用し、更新する並列データリソースを指定します。Amazon Translate により古い並列データが新しい入力ファイルの情報に置き換えられます。

Example update-parallel-data コマンド

次のコマンドにより、Amazon S3 の新しい入力ファイルで `my-parallel-data` が更新されます。

```
$ aws translate update-parallel-data \
> --name my-parallel-data \
> --parallel-data-config S3Uri=s3://input-bucket/parallel-data-file.tsv,Format=TSV
```

コマンドが成功すると、Amazon Translate から以下のようなレスポンスが出ます。

```
{
    "Name": "my-parallel-data",
    "Status": "ACTIVE",
    "LatestUpdateAttemptStatus": "UPDATING",
    "LatestUpdateAttemptAt": 1598601455.844
}
```

このレスポンスでは、`Status` フィールドに既存の並列データオブジェクトのステータスが表示され、`LatestUpdateAttemptStatus` フィールドに現在の更新試行のステータスが表示されます。

SDK を使用した Amazon Translate のコード例 AWS SDKs

次のコード例は、 AWS Software Development Kit (SDK) で Amazon Translate を使用する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

「シナリオ」は、1 つのサービス内から、または他の AWS のサービスと組み合わせて複数の関数を呼び出し、特定のタスクを実行する方法を示すコード例です。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

コードの例

- AWS SDKs を使用した Amazon Translate の基本的な例
  - SDK を使用した Amazon Translate のアクション AWS SDKs
    - AWS SDK DescribeTextTranslationJobで を使用する
    - AWS SDK ListTextTranslationJobsで を使用する
    - AWS SDK StartTextTranslationJobで を使用する
    - AWS SDK StopTextTranslationJobで を使用する
    - AWS SDK または CLI TranslateTextで を使用する
- SDK を使用した Amazon Translate のシナリオ AWS SDKs
  - Amazon Transcribe ストリーミングアプリケーションを構築する
  - Amazon Lex チャットボットを作成して、ウェブサイトの訪問者を引き付けましょう
  - メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する
  - 顧客からのフィードバックを分析し、音声を合成するアプリケーションの作成
  - AWS SDK を使用して Amazon Translate ジョブを開始する

AWS SDKs を使用した Amazon Translate の基本的な例

次のコード例は、 SDKs で AWS Amazon Translate の基本を使用する方法を示しています。

例

- SDK を使用した Amazon Translate のアクション AWS SDKs
  - AWS SDK DescribeTextTranslationJobで を使用する
  - AWS SDK ListTextTranslationJobsで を使用する
  - AWS SDK StartTextTranslationJobで を使用する
  - AWS SDK StopTextTranslationJobで を使用する
  - AWS SDK または CLI TranslateTextで を使用する

SDK を使用した Amazon Translate のアクション AWS SDKs

次のコード例は、 AWS SDKs を使用して個々の Amazon Translate アクションを実行する方法を示しています。それぞれの例には、GitHub へのリンクがあり、そこにはコードの設定と実行に関する説明が記載されています。

これらの抜粋は Amazon Translate API を呼び出し、コンテキスト内で実行する必要がある大規模なプログラムからのコード抜粋です。アクションは SDK を使用した Amazon Translate のシナリオ AWS SDKs のコンテキスト内で確認できます。

以下の例には、最も一般的に使用されるアクションのみ含まれています。詳細なリストについては、Amazon Translate API リファレンス」を参照してください。

例

- AWS SDK DescribeTextTranslationJobで を使用する
- AWS SDK ListTextTranslationJobsで を使用する
- AWS SDK StartTextTranslationJobで を使用する
- AWS SDK StopTextTranslationJobで を使用する
- AWS SDK または CLI TranslateTextで を使用する

AWS SDK **`DescribeTextTranslationJob`**で を使用する

次のサンプルコードは、`DescribeTextTranslationJob` を使用する方法を説明しています。

アクション例は、より大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。次のコード例で、このアクションのコンテキストを確認できます。

- 翻訳ジョブを開始する

.NET

SDK for .NET

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// The following example shows how to retrieve the details of
/// a text translation job using Amazon Translate.
/// </summary>
public class DescribeTextTranslation
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();

        // The Job Id is generated when the text translation job is started
        // with a call to the StartTextTranslationJob method.
        var jobId = "1234567890abcdef01234567890abcde";

        var request = new DescribeTextTranslationJobRequest
        {
            JobId = jobId,
        };

        var jobProperties = await DescribeTranslationJobAsync(client,
request);

        DisplayTranslationJobDetails(jobProperties);
    }

    /// <summary>
    /// Retrieve information about an Amazon Translate text translation job.
```

```
        /// </summary>
        /// <param name="client">The initialized Amazon Translate client
 object.</param>
        /// <param name="request">The DescribeTextTranslationJobRequest object.</
param>
        /// <returns>The TextTranslationJobProperties object containing
        /// information about the text translation job..</returns>
        public static async Task<TextTranslationJobProperties>
 DescribeTranslationJobAsync(
            AmazonTranslateClient client,
            DescribeTextTranslationJobRequest request)
        {
            var response = await client.DescribeTextTranslationJobAsync(request);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                return response.TextTranslationJobProperties;
            }
            else
            {
                return null;
            }
        }

        /// <summary>
        /// Displays the properties of the text translation job.
        /// </summary>
        /// <param name="jobProperties">The properties of the text translation
        /// job returned by the call to DescribeTextTranslationJobAsync.</param>
        public static void
 DisplayTranslationJobDetails(TextTranslationJobProperties jobProperties)
        {
            if (jobProperties is null)
            {
                Console.WriteLine("No text translation job properties found.");
                return;
            }

            // Display the details of the text translation job.
            Console.WriteLine($"{jobProperties.JobId}: {jobProperties.JobName}");
        }
    }
```

- API の詳細については、「AWS SDK for .NET API リファレンス」の「DescribeTextTranslationJob」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
   "Gets the properties associated with an asynchronous batch translation job."
   "Includes properties such as name, ID, status, source and target languages,
and input/output Amazon Simple Storage Service (Amazon S3) buckets."
   TRY.
       oo_result = lo_xl8->describetexttranslationjob(      "oo_result is
returned for testing purposes."
         iv_jobid        = iv_jobid ).
       MESSAGE 'Job description retrieved.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
   ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の「DescribeTextTranslationJob」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK **ListTextTranslationJobs**で を使用する

次のサンプルコードは、ListTextTranslationJobs を使用する方法を説明しています。

.NET

SDK for .NET

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// List Amazon Translate translation jobs, along with details about each
job.
/// </summary>
public class ListTranslationJobs
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();
        var filter = new TextTranslationJobFilter
        {
            JobStatus = "COMPLETED",
        };

        var request = new ListTextTranslationJobsRequest
        {
            MaxResults = 10,
            Filter = filter,
        };

        await ListJobsAsync(client, request);
    }
```

```
        /// <summary>
        /// List Amazon Translate text translation jobs.
        /// </summary>
        /// <param name="client">The initialized Amazon Translate client
object.</param>
        /// <param name="request">An Amazon Translate
        /// ListTextTranslationJobsRequest object detailing which text
        /// translation jobs are of interest.</param>
        public static async Task ListJobsAsync(
            AmazonTranslateClient client,
            ListTextTranslationJobsRequest request)
        {
            ListTextTranslationJobsResponse response;

            do
            {
                response = await client.ListTextTranslationJobsAsync(request);

ShowTranslationJobDetails(response.TextTranslationJobPropertiesList);

                request.NextToken = response.NextToken;
            }
            while (response.NextToken is not null);
        }

        /// <summary>
        /// List existing translation job details.
        /// </summary>
        /// <param name="properties">A list of Amazon Translate text
        /// translation jobs.</param>
        public static void
ShowTranslationJobDetails(List<TextTranslationJobProperties> properties)
        {
            properties.ForEach(prop =>
            {
                Console.WriteLine($"{prop.JobId}: {prop.JobName}");
                Console.WriteLine($"Status: {prop.JobStatus}");
                Console.WriteLine($"Submitted time: {prop.SubmittedTime}");
            });
        }
    }
```

- API の詳細については、「AWS SDK for .NET API リファレンス」の「ListTextTranslationJobs」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
   "Gets a list of the batch translation jobs that you have submitted."

   DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.

   "Create an ABAP object for filtering using jobname."
   lo_filter = NEW #( iv_jobname = iv_jobname ).

   TRY.
       oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned
for testing purposes."
         io_filter        = lo_filter ).
       MESSAGE 'Jobs retrieved.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
     CATCH /aws1/cx_xl8invalidfilterex.
       MESSAGE 'The filter specified for the operation is not valid. Specify a
different filter.' TYPE 'E'.
     CATCH /aws1/cx_xl8invalidrequestex.
       MESSAGE 'The request that you made is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
   ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の「ListTextTranslationJobs」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK **StartTextTranslationJob**で を使用する

次のサンプルコードは、`StartTextTranslationJob` を使用する方法を説明しています。

アクション例は、より大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。次のコード例で、このアクションのコンテキストを確認できます。

- 翻訳ジョブを開始する

.NET

SDK for .NET

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// This example shows how to use Amazon Translate to process the files in
/// an Amazon Simple Storage Service (Amazon S3) bucket. The translated
results
/// will also be stored in an Amazon S3 bucket.
/// </summary>
public class BatchTranslate
{
```

```
        public static async Task Main()
        {
            var contentType = "text/plain";

            // Set this variable to an S3 bucket location with a folder."
            // Input files must be in a folder and not at the bucket root."
            var s3InputUri = "s3://amzn-s3-demo-bucket1/FOLDER/";
            var s3OutputUri = "s3://amzn-s3-demo-bucket2/";

            // This role must have permissions to read the source bucket and to
 read and
            // write to the destination bucket where the translated text will be
 stored.
            var dataAccessRoleArn = "arn:aws:iam::0123456789ab:role/
S3TranslateRole";

            var client = new AmazonTranslateClient();

            var inputConfig = new InputDataConfig
            {
                ContentType = contentType,
                S3Uri = s3InputUri,
            };

            var outputConfig = new OutputDataConfig
            {
                S3Uri = s3OutputUri,
            };

            var request = new StartTextTranslationJobRequest
            {
                JobName = "ExampleTranslationJob",
                DataAccessRoleArn = dataAccessRoleArn,
                InputDataConfig = inputConfig,
                OutputDataConfig = outputConfig,
                SourceLanguageCode = "en",
                TargetLanguageCodes = new List<string> { "fr" },
            };

            var response = await StartTextTranslationAsync(client, request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{response.JobId}: {response.JobStatus}");
```

```
            }
        }

        /// <summary>
        /// Start the Amazon Translate text translation job.
        /// </summary>
        /// <param name="client">The initialized AmazonTranslateClient object.</
param>
        /// <param name="request">The request object that includes details such
        /// as source and destination bucket names and the IAM Role that will
        /// be used to access the buckets.</param>
        /// <returns>The StartTextTranslationResponse object that includes the
        /// details of the request response.</returns>
        public static async Task<StartTextTranslationJobResponse>
 StartTextTranslationAsync(AmazonTranslateClient client,
 StartTextTranslationJobRequest request)
        {
            var response = await client.StartTextTranslationJobAsync(request);
            return response;
        }
    }
```

- API の詳細については、「AWS SDK for .NET API リファレンス」の「StartTextTranslationJob」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    "Starts an asynchronous batch translation job."
    "Use batch translation jobs to translate large volumes of text across
 multiple documents at once."
```

```
    DATA lo_inputdataconfig  TYPE REF TO /aws1/cl_xl8inputdataconfig.
    DATA lo_outputdataconfig TYPE REF TO /aws1/cl_xl8outputdataconfig.
    DATA lt_targetlanguagecodes TYPE /aws1/
cl_xl8tgtlanguagecodes00=>tt_targetlanguagecodestrlist.
    DATA lo_targetlanguagecodes TYPE REF TO /aws1/cl_xl8tgtlanguagecodes00.

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).

    "Create an ABAP object for the output data config."
    lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

    "Create an internal table for target languages."
    lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
    INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

    TRY.
        oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned
 for testing purposes."
          io_inputdataconfig = lo_inputdataconfig
            io_outputdataconfig = lo_outputdataconfig
            it_targetlanguagecodes = lt_targetlanguagecodes
            iv_dataaccessrolearn = iv_dataaccessrolearn
            iv_jobname = iv_jobname
            iv_sourcelanguagecode = iv_sourcelanguagecode ).
        MESSAGE 'Translation job started.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE
 'E'.
      CATCH /aws1/cx_xl8invparamvalueex.
        MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
 TYPE 'E'.
      CATCH /aws1/cx_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language
 of the source text into the requested target language.' TYPE 'E'.
    ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の「StartTextTranslationJob」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK **StopTextTranslationJob**で を使用する

次のサンプルコードは、StopTextTranslationJob を使用する方法を説明しています。

.NET

SDK for .NET

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// Shows how to stop a running Amazon Translation Service text translation
/// job.
/// </summary>
public class StopTextTranslationJob
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();
        var jobId = "1234567890abcdef01234567890abcde";

        var request = new StopTextTranslationJobRequest
        {
```

```
                JobId = jobId,
            };

            await StopTranslationJobAsync(client, request);
        }

        /// <summary>
        /// Sends a request to stop a text translation job.
        /// </summary>
        /// <param name="client">Initialized AmazonTrnslateClient object.</param>
        /// <param name="request">The request object to be passed to the
        /// StopTextJobAsync method.</param>
        public static async Task StopTranslationJobAsync(
            AmazonTranslateClient client,
            StopTextTranslationJobRequest request)
        {
            var response = await client.StopTextTranslationJobAsync(request);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{response.JobId} as status:
{response.JobStatus}");
            }
        }
    }
```

- API の詳細については、「AWS SDK for .NET API リファレンス」の「StopTextTranslationJob」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    "Stops an asynchronous batch translation job that is in progress."
```

```
   TRY.
       oo_result = lo_xl8->stoptexttranslationjob(      "oo_result is returned
for testing purposes."
         iv_jobid        = iv_jobid ).
       MESSAGE 'Translation job stopped.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
   ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の「StopTextTranslationJob」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **TranslateText**で を使用する

次のサンプルコードは、`TranslateText` を使用する方法を説明しています。

.NET

SDK for .NET

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
```

```
    using Amazon.S3.Transfer;
    using Amazon.Translate;
    using Amazon.Translate.Model;

    /// <summary>
    /// Take text from a file stored a Amazon Simple Storage Service (Amazon S3)
    /// object and translate it using the Amazon Transfer Service.
    /// </summary>
    public class TranslateText
    {
        public static async Task Main()
        {
            // If the region you want to use is different from the region
            // defined for the default user, supply it as a parameter to the
            // Amazon Translate client object constructor.
            var client = new AmazonTranslateClient();

            // Set the source language to "auto" to request Amazon Translate to
            // automatically detect te language of the source text.

            // You can get a list of the languages supposed by Amazon Translate
            // in the Amazon Translate Developer's Guide here:
            //      https://docs.aws.amazon.com/translate/latest/dg/what-is.html
            string srcLang = "en"; // English.
            string destLang = "fr"; // French.

            // The Amazon Simple Storage Service (Amazon S3) bucket where the
            // source text file is stored.
            string srcBucket = "amzn-s3-demo-bucket";
            string srcTextFile = "source.txt";

            var srcText = await GetSourceTextAsync(srcBucket, srcTextFile);
            var destText = await TranslatingTextAsync(client, srcLang, destLang,
srcText);

            ShowText(srcText, destText);
        }

        /// <summary>
        /// Use the Amazon S3 TransferUtility to retrieve the text to translate
        /// from an object in an S3 bucket.
        /// </summary>
        /// <param name="srcBucket">The name of the S3 bucket where the
        /// text is stored.
```

```
        /// </param>
        /// <param name="srcTextFile">The key of the S3 object that
        /// contains the text to translate.</param>
        /// <returns>A string representing the source text.</returns>
        public static async Task<string> GetSourceTextAsync(string srcBucket,
 string srcTextFile)
        {
            string srcText = string.Empty;

            var s3Client = new AmazonS3Client();
            TransferUtility utility = new TransferUtility(s3Client);

            using var stream = await utility.OpenStreamAsync(srcBucket,
 srcTextFile);

            StreamReader file = new System.IO.StreamReader(stream);

            srcText = file.ReadToEnd();
            return srcText;
        }

        /// <summary>
        /// Use the Amazon Translate Service to translate the document from the
        /// source language to the specified destination language.
        /// </summary>
        /// <param name="client">The Amazon Translate Service client used to
        /// perform the translation.</param>
        /// <param name="srcLang">The language of the source text.</param>
        /// <param name="destLang">The destination language for the translated
        /// text.</param>
        /// <param name="text">A string representing the text to ranslate.</
param>
        /// <returns>The text that has been translated to the destination
        /// language.</returns>
        public static async Task<string>
 TranslatingTextAsync(AmazonTranslateClient client, string srcLang, string
 destLang, string text)
        {
            var request = new TranslateTextRequest
            {
                SourceLanguageCode = srcLang,
                TargetLanguageCode = destLang,
                Text = text,
            };
```

```
            var response = await client.TranslateTextAsync(request);

            return response.TranslatedText;
        }

        /// <summary>
        /// Show the original text followed by the translated text.
        /// </summary>
        /// <param name="srcText">The original text to be translated.</param>
        /// <param name="destText">The translated text.</param>
        public static void ShowText(string srcText, string destText)
        {
            Console.WriteLine("Source text:");
            Console.WriteLine(srcText);
            Console.WriteLine();
            Console.WriteLine("Translated text:");
            Console.WriteLine(destText);
        }
    }
```

- API の詳細については、AWS SDK for .NET API リファレンスの「TranslateText」を参照してください。

PowerShell

Tools for PowerShell V4

例 1: 指定された英語のテキストをフランス語に変換します。変換するテキストは、-Text パラメータとして渡すこともできます。

```
"Hello World" | ConvertTo-TRNTargetLanguage -SourceLanguageCode en -
TargetLanguageCode fr
```

- API の詳細については、「 コマンドレットリファレンス (V4)」のTranslateText」を参照してください。 AWS Tools for PowerShell

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
   "Translates input text from the source language to the target language."
   TRY.
       oo_result = lo_xl8->translatetext(      "oo_result is returned for
testing purposes."
         iv_text        = iv_text
           iv_sourcelanguagecode = iv_sourcelanguagecode
           iv_targetlanguagecode = iv_targetlanguagecode ).
       MESSAGE 'Translation completed.' TYPE 'I'.
     CATCH /aws1/cx_xl8detectedlanguage00.
       MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred.' TYPE 'E'.
     CATCH /aws1/cx_xl8invalidrequestex.
       MESSAGE 'The request that you made is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8serviceunavailex.
       MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE
'E'.
     CATCH /aws1/cx_xl8textsizelmtexcdex.
       MESSAGE 'The size of the text you submitted exceeds the size limit. '
TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
     CATCH /aws1/cx_xl8unsuppedlanguage00.
       MESSAGE 'Amazon Translate does not support translation from the language
of the source text into the requested target language. ' TYPE 'E'.
   ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の「TranslateText」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

SDK を使用した Amazon Translate のシナリオ AWS SDKs

次のコード例は、 AWS SDKs を使用して Amazon Translate で一般的なシナリオを実装する方法を示しています。これらのシナリオでは、Amazon Translate 内で複数の関数を呼び出すか、他の関数と組み合わせて特定のタスクを実行する方法を示します AWS のサービス。各シナリオには、完全なソースコードへのリンクが含まれており、そこからコードの設定方法と実行方法に関する手順を確認できます。

シナリオは、サービスアクションをコンテキストで理解するのに役立つ中級レベルの経験を対象としています。

例

- Amazon Transcribe ストリーミングアプリケーションを構築する
- Amazon Lex チャットボットを作成して、ウェブサイトの訪問者を引き付けましょう
- メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する
- 顧客からのフィードバックを分析し、音声を合成するアプリケーションの作成
- AWS SDK を使用して Amazon Translate ジョブを開始する

Amazon Transcribe ストリーミングアプリケーションを構築する

次のコード例は、ライブ音声をリアルタイムで記録、転写、翻訳し、結果を E メールで送信するアプリケーションを構築する方法を示しています。

JavaScript

SDK for JavaScript (v3)

Amazon Transcribe を使用して、ライブ音声をリアルタイムで記録、転写、翻訳し、Amazon Simple Email Service (Amazon SES) を使用して結果を E メールで送信するアプリケーションを構築する方法について説明します。

完全なソースコードとセットアップおよび実行の手順については、GitHub で完全な例を参照してください。

この例で使用されているサービス

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

Amazon Lex チャットボットを作成して、ウェブサイトの訪問者を引き付けましょう

次のコード例は、Web サイトの訪問者を引き込むチャットボットを作成する方法を示しています。

Java

SDK for Java 2.x

ウェブアプリケーション内に Amazon Lex chatbotを作成して、ウェブサイトの訪問者に対応することができます。

完全なソースコードとセットアップおよび実行の手順については、GitHub で完全な例を参照してください。

この例で使用されているサービス

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

JavaScript

SDK for JavaScript (v3)

ウェブアプリケーション内に Amazon Lex chatbotを作成して、ウェブサイトの訪問者に対応することができます。

完全なソースコードとセットアップと実行の手順については、 デ AWS SDK for JavaScript ベロッパーガイドのAmazon Lexチャットボットの構築」の完全な例を参照してください。

この例で使用されているサービス

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する

次のコード例は、サブスクリプションと発行機能を持ち、メッセージを翻訳するアプリケーションを作成する方法を示しています。

.NET

SDK for .NET

Amazon Simple Notification Service .NET API を使用して、サブスクリプションおよびパブリッシュ機能を持つウェブアプリケーションを作成する方法を説明します。さらに、このサンプルアプリケーションではメッセージを翻訳します。

完全なソースコードとセットアップおよび実行の手順については、GitHub で完全な例を参照してください。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

Java

SDK for Java 2.x

Amazon Simple Notification Service Java API を使用して、サブスクリプションおよびパブリッシュ機能を持つウェブアプリケーションを作成する方法を説明します。さらに、このサンプルアプリケーションではメッセージを翻訳します。

完全なソースコードとセットアップおよび実行の手順については、GitHub で完全な例を参照してください。

Java Async API を使用するサンプルの完全なソースコードとセットアップおよび実行の手順については、GitHub で完全な例を参照してください。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

Kotlin

SDK for Kotlin

Amazon SNS Kotlin API を使用して、サブスクリプションおよび発行機能を持つアプリケーションを作成する方法を示しています。さらに、このサンプルアプリケーションではメッセージを翻訳します。

詳細なソースコードとウェブアプリケーションを作成する手順については、「GitHub」の詳しい例を参照してください。

詳細なソースコードと Android ネイティブアプリケーションを作成する手順については、「GitHub」の詳しい例を参照してください。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

顧客からのフィードバックを分析し、音声を合成するアプリケーションの作成

次のコード例では、顧客のコメントカードを分析し、それを元の言語から翻訳し、顧客の感情を判断し、翻訳されたテキストから音声ファイルを生成するアプリケーションの作成方法を示しています。

.NET

SDK for .NET

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、GitHub のプロジェクトを参照してください。

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK for Java 2.x

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様から

のフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、GitHub のプロジェクトを参照してください。

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK for JavaScript (v3)

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、GitHub のプロジェクトを参照してください。次の抜粋 AWS SDK for JavaScript は、Lambda 関数内で がどのように使用されるかを示しています。

```
import {
  ComprehendClient,
  DetectDominantLanguageCommand,
  DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
 * Determine the language and sentiment of the extracted text.
 *
 * @param {{ source_text: string}} extractTextOutput
 */
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });

  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};
```

```
import {
  DetectDocumentTextCommand,
  TextractClient,
} from "@aws-sdk/client-textract";
```

```
/**
 * Fetch the S3 object from the event and analyze it using Amazon Textract.
 *
 * @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
 eventBridgeS3Event
 */
export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/
textract/latest/dg/API_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );

  return extractedWords.join(" ");
};
```

```
import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string}}
 sourceDestinationConfig
 */
```

```
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};
```

```
import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string}}
 textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});
```

```
  const translateCommand = new TranslateTextCommand({
    SourceLanguageCode: textAndSourceLanguage.source_language_code,
    TargetLanguageCode: "en",
    Text: textAndSourceLanguage.extracted_text,
  });

  const { TranslatedText } = await translateClient.send(translateCommand);

  return { translated_text: TranslatedText };
};
```

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK for Ruby

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、GitHub のプロジェクトを参照してください。

この例で使用されているサービス

- Amazon Comprehend

- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してください[AWS SDK でのこのサービスの使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK を使用して Amazon Translate ジョブを開始する

次のコードサンプルは、以下の操作方法を示しています。

- 非同期バッチ翻訳ジョブを開始します。
- 非同期ジョブが完了するまで待ちます。
- 非同期ジョブを記述します。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。[AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    DATA lo_inputdataconfig  TYPE REF TO /aws1/cl_xl8inputdataconfig.
    DATA lo_outputdataconfig TYPE REF TO /aws1/cl_xl8outputdataconfig.
    DATA lt_targetlanguagecodes TYPE /aws1/
cl_xl8tgtlanguagecodes00=>tt_targetlanguagecodestrlist.
    DATA lo_targetlanguagecodes TYPE REF TO /aws1/cl_xl8tgtlanguagecodes00.

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).
```

```
    "Create an ABAP object for the output data config."
    lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

    "Create an internal table for target languages."
    lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
    INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

    TRY.
        DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
          io_inputdataconfig = lo_inputdataconfig
            io_outputdataconfig = lo_outputdataconfig
            it_targetlanguagecodes = lt_targetlanguagecodes
            iv_dataaccessrolearn = iv_dataaccessrolearn
            iv_jobname = iv_jobname
            iv_sourcelanguagecode = iv_sourcelanguagecode ).
        MESSAGE 'Translation job started.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE
 'E'.
      CATCH /aws1/cx_xl8invparamvalueex.
        MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time. '
 TYPE 'E'.
      CATCH /aws1/cx_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language
 of the source text into the requested target language.' TYPE 'E'.
    ENDTRY.

    "Get the job ID."
    DATA(lv_jobid) = lo_translationjob_result->get_jobid( ).

    "Wait for translate job to complete."
    DATA(lo_des_translation_result) = lo_xl8-
>describetexttranslationjob( iv_jobid = lv_jobid ).
    WHILE lo_des_translation_result->get_textxlationjobproperties( )-
>get_jobstatus( ) <> 'COMPLETED'.
      IF sy-index = 30.
        EXIT.               "Maximum 900 seconds."
      ENDIF.
```

```
      WAIT UP TO 30 SECONDS.
      lo_des_translation_result = lo_xl8->describetexttranslationjob( iv_jobid =
lv_jobid ).
   ENDWHILE.

   TRY.
       oo_result = lo_xl8->describetexttranslationjob(      "oo_result is
returned for testing purposes."
         iv_jobid        = lv_jobid ).
       MESSAGE 'Job description retrieved.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
   ENDTRY.
```

- API の詳細については、「AWS SDK for SAP ABAP API リファレンス」の以下のトピックを参照してください。
  - DescribeTextTranslationJob
  - StartTextTranslationJob

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

リソースのタグ付け

タグは、Amazon Translate リソースに関連付けることができるメタデータです。タグは、キーと値のペアから構成されます。タグは [並列データ] リソースと [カスタム用語] リソースに追加できます。

タグには主に 2 つの機能があります。1 つ目はリソースを整理することと、2 つ目はタグベースのアクセス制御を行うことです。リソースにタグを追加し、そのタグに基づいてリソースへのアクセスを許可または制限する IAM ポリシーを作成できます。

ポリシーでは、リクエストで指定されたタグ (request-tags) または呼び出すリソースに関連するタグ (resource-tags) に基づいてオペレーションを許可または禁止できます。IAM でタグを使用する方法については、「IAM ユーザーガイド」の「タグを使用してアクセスを制御する」を参照してください。

Amazon Translate でタグを使用する際の考慮事項:

- リソースごとに最大 50 個のユーザータグを追加できます。
- リソースの作成時または作成後いつでもタグを追加できます。
- タグキーは必須フィールドですが、タグ値はオプションです。
- タグはリソース間で一意である必要はありませんが、特定のリソースのタグには一意のキーが必要です。
- タグのキーと値では、大文字と小文字が区別されます。
- タグキーは最大 128 文字、タグ値は最大 256 文字です。
- AWS システムタグは、タグキーまたは値のプレフィックス`aws:`で始まります。このプレフィックスが含まれるタグの名前や値は、編集または削除できません。システムタグは、リソースごとのタグクォータには含まれません。

> **Note**
>
> 複数の AWS サービスやリソースでタグ付けスキーマを使用する場合は、他のサービスで許可される文字の要件が異なる場合があることに注意してください。

トピック

- 新しいリソースへのタグ付け

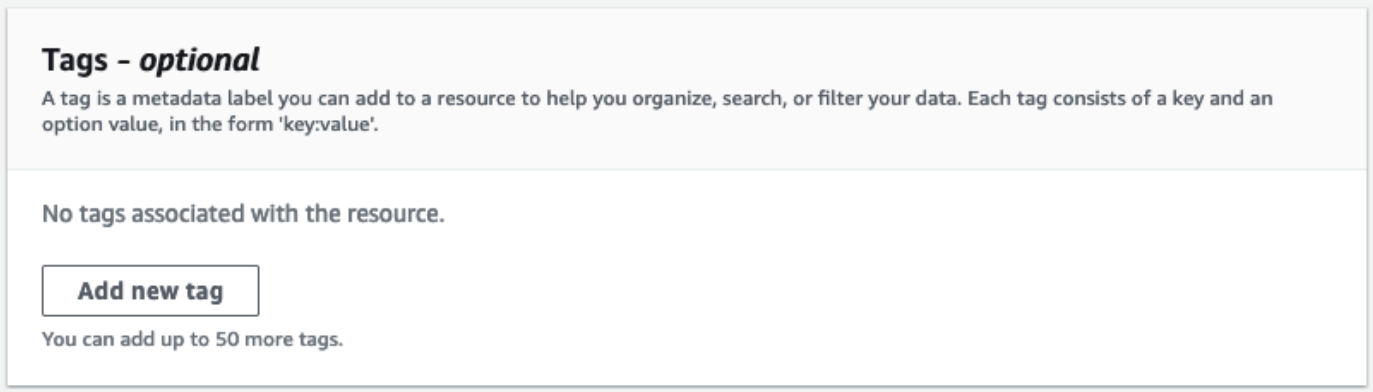
- リソースに関連付けられたタグの表示、更新、削除

新しいリソースへのタグ付け

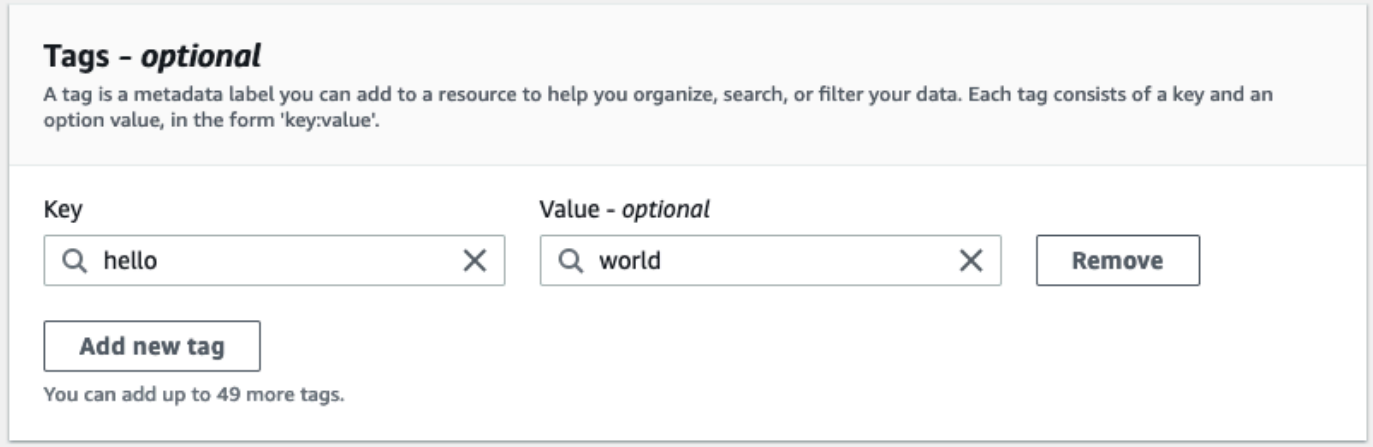
ParallelData リソースまたは カスタム用語 リソースを作成する際に、タグを追加できます。

新しいリソースにタグを追加する (コンソール)

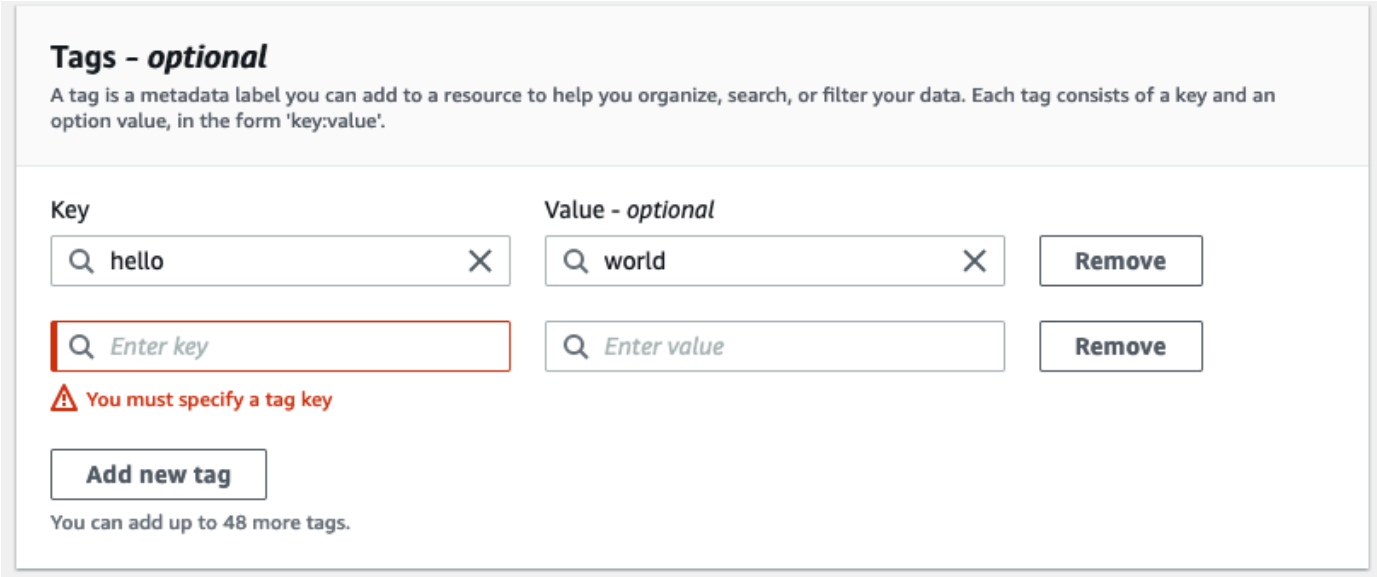
1. Amazon Translate コンソール にサインインします。
2. 左のナビゲーションペインから、作成するリソース (`Parallel data` または `Custom terminology`) を選択します。
3. [並列データの作成] または [用語の作成] を選択します。コンソールに、リソース作成のメインページが表示されます。このページの最後に [タグ - オプション] パネルがあります。



4. [新しいタグを追加] を選択してリソースのタグを追加します。タグキーおよびタグ値 (オプション) を入力します。



5. ステップ 4 を繰り返して、すべてのタグを追加します。各キーは、このリソースに一意である必要があります。



6. [並列データの作成] または [用語の作成] を選択してリソースを作成します。

Amazon Translate の CreateParallelData API オペレーションを使用してタグを追加することもできます。次の例では、create-parallel-data CLI コマンドでタグを追加する方法を示しています。

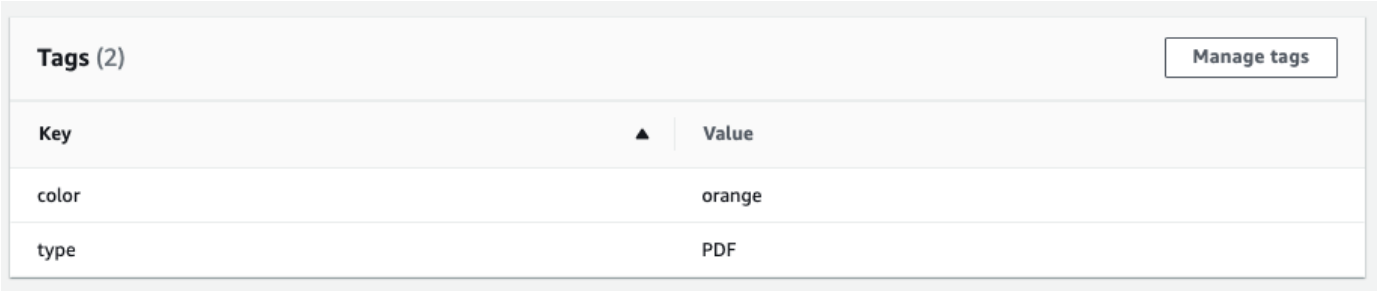
```
aws translate create-parallel-data \
--name "myTest" \
--parallel-data-config "{\"format\": \"CSV\",  \
             "S3Uri\": \"s3://test-input/TEST.csv\"}" \
--tags "[{\"Key\": \"color\",\"Value\": \"orange\"}]"
```

リソースに関連付けられたタグの表示、更新、削除

`Parallel data` または `Custom terminology` リソースに関連付けられているタグを表示、更新、削除できます。

既存のリソースのタグを更新する方法 (コンソール)

1. Amazon Translate コンソール にサインインします。
2. 左のナビゲーションペインで `Parallel data` または `Custom terminology` を選択します。
3. 表示、更新、または削除するタグを含むリソースを選択します。コンソールにリソースの詳細ページが表示されます。
4. [タグ] パネルが表示されるまで下にスクロールします。ここには、選択したリソースに関連付けられているすべてのタグが表示されます。



[タグを管理] を選択して、リソースからタグを編集または削除します。

5. 変更するテキストを選択し、タグを編集します。[削除] を選択してタグを削除することもできます。
6. 新しいタグを追加するには、[新しいタグを追加] を選択し、空白のフィールドにキーと値を入力します。
7. タグの変更が完了したら、[保存] を選択します。

Amazon Translate のセキュリティ

でのクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、最もセキュリティの影響を受けやすい組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS お客様とお客様の間の責任共有です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ – クラウドで AWS AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。 AWS また、 では、安全に使用できるサービスも提供しています。「AWS 」 コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon Translate に適用されるコンプライアンスプログラムについては、「コンプライアンスプログラムによる対象範囲のAWS のサービス」を参照してください。
- クラウド内のセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、 お客様のデータの機密性、企業の要件、および適用可能な法律や規制といった他の要因 についても責任を担います。

このトピックは、 AWSを使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成する AWS ように を設定する方法を示します。また、 AWS リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

- Amazon Translate におけるデータ保護
- Amazon Translate のための Identity and Access Management
- Amazon Translate のモニタリング
- Amazon Translate のコンプライアンス検証
- Amazon Translate の耐障害性
- Amazon Translate のインフラストラクチャセキュリティ
- Amazon Translate とインターフェイス VPC エンドポイント (AWS PrivateLink)

Amazon Translate におけるデータ保護

Amazon Translate は 責任 AWS 共有モデルに準拠しています。 には、データ保護に関する規制とガイドラインが含まれています。 AWS は、すべての AWS サービスを実行するグローバルインフラストラクチャを保護する責任があります。 は、このインフラストラクチャでホストされるデータの制御 AWS を維持します。 お客様のコンテンツと個人データを処理するためのセキュリティ設定コントロールを含めます。 AWS のお客様および APN パートナー、 データコントローラーまたはデータ処理者として動作する は、 AWS クラウドに保存した個人データについて責任を負います。

データ保護の目的で、 AWS アカウント認証情報を保護し、 AWS Identity and Access Management (IAM) でロールを設定することをお勧めします。これにより、各ユーザーに各自の職務を果たすために必要なアクセス許可のみが付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、 AWS サービス内のすべてのデフォルトのセキュリティコントロールを使用します。
- Amazon Simple Storage Service (Amazon S3) に保存されている個人情報の発見と保護を支援する Amazon Macie などの高度なマネージド・ セキュリティ・ サービスを利用します。

顧客のアカウント番号などの機密の識別情報は、[名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、API、 AWS CLIまたは SDK を使用して Amazon Translate または他の AWS のサービスを使用する場合も同様です。 AWS SDKs Amazon Translate や他のサービスに入力したすべてのデータは、診断ログに取り込まれる可能性があります。外部サーバーへの URL を指定するときは、そのサーバーへのリクエストを検証するための認証情報を URL に含めないでください。

データ保護の詳細については、AWS セキュリティブログ のブログ投稿「AWS の責任共有モデルと GDPR」を参照してください。

トピック

- 保管中の暗号化
- 転送中の暗号化

保管中の暗号化

Amazon Translate で実行するバッチ翻訳ジョブについては、翻訳の入力と出力の両方が保存時に暗号化されます。ただし、暗号化方法はそれぞれ異なります。

Amazon Translate では、デフォルトキーで暗号化された Amazon Elastic Block Store (Amazon EBS) ボリュームも使用します。

翻訳入力

Amazon Translate を使用してドキュメントをバッチ単位で翻訳する場合、一連の入力ドキュメントを Amazon S3 バケットに保存します。これらのドキュメントを保存中に暗号化するには、Amazon S3 により提供される SSE-S3 サーバーサイド暗号化オプションを使用できます。このオプションを使用すると、各オブジェクトは、Amazon S3 が管理する一意のキーで暗号化されます。

詳細については、「Amazon Simple Storage Service ユーザーガイド」の「Amazon S3 マネージドキーによるサーバー側の暗号化 (SSE-S3)」を参照してください。

翻訳出力

Amazon Translate がバッチ翻訳ジョブを完了すると、出力は AWS アカウントの Amazon S3 バケットに配置されます。保存時の出力を暗号化するために、Amazon Translate では Amazon S3 により提供される SSE-KMS 暗号化オプションを使用します。このオプションでは、出力は AWS Key Management Service () に保存されているキーで暗号化されますAWS KMS。

詳細については、「Amazon Simple Storage Service ユーザーガイド」の「AWS Key Management Service (SSE-KMS) によるサーバーサイド暗号化を使用したデータの保護」を参照してください。

KMS キーの詳細については、「AWS Key Management Service デベロッパーガイド」の「AWS KMS keys」を参照してください。

この暗号化のために、Amazon Translate では次のいずれかのタイプのキーを使用できます。

AWS マネージドキー

デフォルトでは Amazon Translate では AWS マネージドキー が使用されます。このタイプの KMS キーは、 によって作成 AWS され、アカウントに保存されます。ただし、この KMS キーをユーザー自身が管理することはできません。これは、ユーザーに代わって AWSによってのみ管理と使用が行われます。

カスタマーマネージドキー

オプションで、 AWS アカウントで作成、所有、管理する KMS キーであるカスタマーマネージドキーを使用して出力を暗号化することもできます。

自作の KMS キーを使用できるようにするには、Amazon S3 の出力バケットにアクセスするために Amazon Translate により使用される IAM サービスロールに対して、あらかじめアクセス許可を追加しておく必要があります。別の AWS アカウントにある KMS キーを使用する場合は、キーポリシーも更新する必要があります AWS KMS。詳細については、「暗号化をカスタマイズするための前提となるアクセス許可」を参照してください。

バッチ翻訳ジョブの実行時におけるカスタマーマネージドキーの使用を選択できます。詳細については、「バッチ翻訳ジョブの実行」を参照してください。

転送中の暗号化

Amazon Translate では、転送中のデータを暗号化するために、TLS 1.2 を AWS 証明書とともに使用します。

Amazon Translate のための Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) され、Amazon Translate リソースを使用する認可 を受ける (許可がある) ことができるかを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

トピック

- 対象者
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon Translate で IAM が機能する仕組み
- Amazon Translate のアイデンティティベースのポリシーの例
- AWS Amazon Translate の マネージドポリシー
- Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング

対象者

AWS Identity and Access Management (IAM) の使用方法は、Amazon Translate で行う作業によって異なります。

サービスユーザー – 業務を行うために Amazon Translate サービスを使用する場合は、管理者から必要な認証情報と許可が提供されます。さらに多くの Amazon Translate 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解すると、管理者に適切なアクセス許可をリクエストするのに役に立ちます。Amazon Translate の機能にアクセスできない場合は、「Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング」を参照してください。

サービス管理者 – 社内の Amazon Translate リソースを担当している場合は、通常、Amazon Translate へのフルアクセスがあります。サービスユーザーがアクセスする Amazon Translate 機能やリソースを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社で Amazon Translate と IAM を併用する方法の詳細については、「Amazon Translate で IAM が機能する仕組み」を参照してください。

IAM 管理者 – 管理者は、Amazon Translate へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる Amazon Translate のアイデンティティベースポリシーの例を確認するには、「 Amazon Translate のアイデンティティベースのポリシーの例」を参照してください。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けることによって、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook 認証情報は、フェデレーション ID の例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 AWS サインイン ユーザーガイド」の「 へのサインイン AWS アカウント方法」を参照してください。

AWS プログラムで にアクセスする場合、 はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストを暗号化して署名します。AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。リクエストに自分で署名する推奨方法の使用については、「IAM ユーザーガイド」の「API リクエストに対するAWS Signature Version 4」を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。たとえば、 では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「多要素認証」および「IAM ユーザーガイド」の「IAM のAWS 多要素認証」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して一時的な認証情報 AWS のサービス を使用して にアクセスすることを要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを介して提供された認証情報 AWS のサービス を使用して にアクセスするすべてのユーザーです。フェデレーティッド ID がアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成するか、独自の ID ソースのユーザーとグループのセットに接続して同期し、すべての AWS アカウント とアプリケーションで使用できます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内の ID です。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。で IAM ロールを一時的に引き受けるには AWS Management Console、ユーザーから IAM ロール (コンソール) に切り替えることができます。ロールを引き受けるには、 または AWS API オペレーションを AWS CLI 呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールについては、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する」を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity

Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center User Guide」の「Permission sets」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。
- クロスサービスアクセス — 一部の は他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする と組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除することができます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに許可を委任するロールを作成する」を参照してください。
  - サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。EC2 インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは AWS 、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「管理ポリシーとインラインポリシーのいずれかを選択する」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、および Amazon VPC は AWS WAF、ACLs。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的でない追加のポリシータイプをサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エ

ンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。

- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー (SCP)」を参照してください。
- リソースコントロールポリシー (RCP) – RCP は、所有する各リソースにアタッチされた IAM ポリシーを更新することなく、アカウント内のリソースに利用可能な最大数のアクセス許可を設定するために使用できる JSON ポリシーです。RCP は、メンバーアカウントのリソースのアクセス許可を制限し、組織に属しているかどうかにかかわらず AWS アカウントのルートユーザー、 を含む ID の有効なアクセス許可に影響を与える可能性があります。RCP をサポートする のリストを含む Organizations と RCP の詳細については、AWS Organizations RCPs「リソースコントロールポリシー (RCPs」を参照してください。 AWS のサービス
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「ポリシー評価ロジック」を参照してください。

Amazon Translate で IAM が機能する仕組み

IAM を使用して Amazon Translate へのアクセスを管理する前に、Amazon Translate で使用できる IAM 機能について理解しておく必要があります。

Amazon Translate で使用できる IAM の機能

IAM 機能	Amazon Translate のサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	はい
ポリシー条件キー (サービス固有)	はい
ACL	いいえ
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	あり
転送アクセスセッション (FAS)	あり
サービスロール	はい
サービスリンクロール	いいえ

Amazon Translate およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要については、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

Amazon Translate のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、

ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

Amazon Translate のアイデンティティベースのポリシーの例

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「 Amazon Translate のアイデンティティベースのポリシーの例」を参照してください。

Amazon Translate 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」を参照してください。

Amazon Translate のポリシーアクション

ポリシーアクションのサポート:あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

Amazon Translate アクションのリストを確認するには、「サービス認可リファレンス」の[Amazon Translate で定義されるアクション](#)」を参照してください。

Amazon Translate のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
translate
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
      "translate:ListLanguages",
      "translate:TranslateText"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、`List` という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "translate:List*"
```

1 つのサービスに対してすべてのアクションを指定するワイルドカードを使用しないでください。最小特権を認めるというベストプラクティスに従い、ポリシー内のアクセス許可を絞り込みます。

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「Amazon Translate のアイデンティティベースのポリシーの例」を参照してください。

Amazon Translate のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Resource` JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ステートメントには`Resource` または `NotResource` 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

Amazon Translate リソースタイプとその ARNsAmazon Translate で定義されるリソース」を参照してください。 どのアクションで各リソースの ARN を指定できるかについては、「Amazon Translate で定義されるアクション」を参照してください。

Amazon Translate ポリシーでリソースを使用する方法の例については、ポリシーでのリソースの指定 を参照してください。

Amazon Translate のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素 (または `Condition` ブロック) を使用すると、ステートメントが有効な条件を指定できます。`Condition` 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドのAWS 「グローバル条件コンテキストキー」を参照してください。

Amazon Translate 条件キーのリストを確認するには、「サービス認可リファレンス」のAmazon Translate の条件キー」を参照してください。条件キーを使用できるアクションとリソースについては、「Amazon Translate で定義されるアクション」を参照してください。

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「 Amazon Translate のアイデンティティベースのポリシーの例」を参照してください。

Amazon Translate の ACL

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon Translate での ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

Amazon Translate リソースのタグ付けの詳細については、「リソースのタグ付け」を参照してください。

Amazon Translate での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報と AWS のサービス 連携する などの詳細については、AWS のサービス IAM ユーザーガイドの「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合は、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用してにアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

Amazon Translate の転送アクセスセッション

転送アクセスセッション (FAS) のサポート: あり

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする と組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Amazon Translate のサービスロール

サービスロールのサポート: あり

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに許可を委任するロールを作成する」を参照してください。

> **Warning**
> サービスロールの許可を変更すると、Amazon Translate の機能が破損する可能性があります。Amazon Translate が指示する場合以外は、サービスロールを編集しないでください。

Amazon Translate 非同期オペレーションを使用するには、入力ドキュメントが含まれている Amazon S3 バケットへのアクセスを Amazon Translate に許可する必要があります。これを行うには、Amazon Translate サービスプリンシパルを信頼するように、アカウントにサービスロールを作成します。

ポリシーの例については「バッチ翻訳ジョブの前提条件」を参照してください。

Amazon Translate のサービスにリンクされたロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

Amazon Translate のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには Amazon Translate リソースを作成または変更するアクセス許可がありません。また、 AWS Management Console、 AWS CLI、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、指定されたリソースに対して特定の API オペレーションを実行するために必要なアクセス許可を付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらのアクセス許可が必要な ユーザーまたはロールに、そのポリシーをアタッチします。

JSON ポリシードキュメントのこれらの例を使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「JSON タブでのポリシーの作成」を参照してください。

トピック

- アイデンティティベースのポリシーのベストプラクティス
- Amazon Translate コンソールへのアクセスを許可する
- 自分の権限の表示をユーザーに許可する
- ポリシーでのリソースの指定
- カスタム用語でカスタマーマネージドキーを使用するためのアクセス許可

アイデンティティベースのポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウント内で誰かが Amazon Translate リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めしま

す。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「ジョブ機能のAWS マネージドポリシー」を参照してください。

- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「IAM でのポリシーとアクセス許可」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素:条件」を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「IAM Access Analyzer でポリシーを検証する」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA を使用した安全な API アクセス」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「IAM でのセキュリティのベストプラクティス」を参照してください。

Amazon Translate コンソールへのアクセスを許可する

Amazon Translate コンソールにアクセスするには、許可の最小限のセットが必要です。これらのアクセス許可により、 AWS アカウントの Amazon Translate リソースの詳細を一覧表示および表示できます。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザー、グループ、またはロール) に対してコンソールが意図したとおりに機能しません。

Amazon Translate コンソールのアクセス許可については、エンティティにTranslateFullAccess AWS 管理ポリシーをアタッチできます。詳細については、「AWS Amazon Translate の マネージドポリシー」を参照してください。

また、次のポリシーに示すアクションにアクセス許可を付与する必要があります。これらのアクセス許可は、TranslateFullAccess ポリシーに含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "iam:GetRole",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
             ],
            "Resource": "*"

        }
    ]
}
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。詳細については、「IAM ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

ポリシーでのリソースの指定

多くの Amazon Translate API アクションでは、アクションで許可されている (または許可されていない) リソースを指定することによって、ポリシーの範囲を制限できます。リソースを指定できるアクションのリストについては、「Amazon Translate で定義されるアクション」を参照してください。ポリシーでは以下のリソースを指定できます。

- Custom terminology - 次の ARN 形式を使用します。

  `arn:partition:translate:region:account:terminology/terminology-name/LATEST`

- Parallel data - 次の ARN 形式を使用します。

  `arn:partition:translate:region:account:parallel-data/parallel-data-name`

ワイルドカード文字を使用すると、ポリシーで複数のリソースを指定できます。以下のポリシー例では、すべての Amazon Translate アクションに対してすべてのカスタム用語リソースを許可します。

Example

```
{
        "Sid": "Example1",
        "Effect": "Allow",
        "Action": "translate:*",
        "Resource": [
             "arn:aws:translate:us-west-2:123456789012:terminology/*"
        ]
}
```

次のポリシー例では、GetParallelData アクションの特定の並列データリソースへのアクセスを拒否します。

Example

```
{
        "Sid": "Example2",
        "Effect": "Deny",
        "Action": "translate:GetParallelData",
        "Resource": [
             "arn:aws:translate:us-west-2:123456789012:parallel-data/test-parallel-
data"
        ]
}
```

カスタム用語でカスタマーマネージドキーを使用するためのアクセス許可

Amazon Translate カスタム用語で AWS Key Management Service（AWS KMS) カスタマーマネージドキーを使用する場合は、KMS キーポリシーに追加のアクセス許可が必要になる場合があります。

カスタマーマネージドキーを使用して `ImportTerminology` オペレーションを呼び出すには、既存の KMS キーポリシーに以下のアクセス許可を追加します。

```
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "Allow access for use with Amazon Translate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "IAM USER OR ROLE ARN"
            },
            "Action": [
                "kms:CreateAlias",
                "kms:CreateGrant",
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy",
                "kms:RetireGrant"
            ],
            "Resource": "*"
        }
    ]
}
```

KMS カスタマーマネージドキーでインポートされたカスタム用語に対して `GetTerminology` オペレーションを呼び出すには、KMS キーポリシーに以下のアクセス許可を追加します。

```
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow access for use with Amazon Translate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "IAM USER OR ROLE ARN"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

カスタマーマネージドキーでインポートされたカスタム用語の ListTerminologiesまたは DeleteTermionlogyオペレーションを呼び出すには、特別な AWS KMS アクセス許可は必要ありません。

すべてのカスタム用語オペレーションでカスタマーマネージドキーを使用するには、KMS キーポリシーに以下のアクセス許可を追加します。

```
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow access for use with Amazon Translate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "IAM USER OR ROLE ARN"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy",
                "kms:RetireGrant"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Translate のオペレーションとリソースの詳細については、「サービス認可リファレンス」のAmazon Translate のアクション、リソース、および条件キー」を参照してください。

AWS Amazon Translate の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

トピック

- AWS マネージドポリシー: TranslateFullAccess
- AWS マネージドポリシー: TranslateReadOnly
- AWS マネージドポリシーに対する Amazon Translate の更新

AWS マネージドポリシー: TranslateFullAccess

このポリシーは、Amazon Translate リソース、Amazon Comprehend DetectDominantLanguage API オペレーション、および必要な CloudWatch API オペレーションへのフルアクセスを許可します。このポリシーでは、Amazon S3 バケットと IAM ロールのリスト権限と取得権限も付与されます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "translate:*",
                "comprehend:DetectDominantLanguage",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "iam:ListRoles",
                "iam:GetRole"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*"
        }
    ]
}
```

AWS マネージドポリシー: TranslateReadOnly

このポリシーは、アカウントに関連付けられているリソースを変更しない Amazon Translate API オペレーションへのアクセス許可を付与します。このポリシーでは、Amazon Comprehend DetectDominantLanguage API オペレーションおよび必要な CloudWatch API オペレーションへのアクセス許可も付与されます。

```
{
  "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "translate:TranslateText",
                "translate:TranslateDocument",
                "translate:GetTerminology",
                "translate:ListTerminologies",
                "translate:ListTextTranslationJobs",
                "translate:DescribeTextTranslationJob",
                "translate:GetParallelData",
                "translate:ListParallelData",
                "comprehend:DetectDominantLanguage",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS マネージドポリシーに対する Amazon Translate の更新

このサービスがこれらの変更の追跡を開始してからの Amazon Translate の AWS マネージドポリシーの更新に関する詳細を表示します。このページへの変更に関する自動通知を受け取るに

は、Amazon Translate の「ドキュメント履歴」ページで RSS フィードにサブスクライブしてください。

変更	説明	日付
TranslateReadOnly — 既存のポリシーを更新	Amazon Translate では、`TranslateDocument` TranslateReadOnly ポリシーでアクションが許可されるようになりました。	2023 年 5 月 23 日
Amazon Translate が変更の追跡を開始	Amazon Translate は、 AWS 管理ポリシーの変更の追跡を開始しました。	2023 年 5 月 23 日

Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング

次の情報は、Amazon Translate と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- Amazon Translate でアクションを実行する権限がありません
- iam:PassRole を実行する権限がありません
- 自分の 以外のユーザーに Amazon Translate リソース AWS アカウント へのアクセスを許可したい

Amazon Translate でアクションを実行する権限がありません

アクションを実行する権限がないというエラーが表示された場合は、そのアクションを実行できるようにポリシーを更新する必要があります。

以下のエラー例は、`mateojackson` IAM ユーザーがコンソールを使用して架空の *`my-example-widget`* リソースに関する詳細情報を表示しようとしているが、架空の `translate:`*`GetWidget`* 権限がないという場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 translate:GetWidget on resource: my-example-widget
```

この場合、Mateo のポリシーでは、*my-example-widget* アクションを使用して translate:*GetWidget* リソースにアクセスすることを許可するように更新する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam:PassRole を実行する権限がありません

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Amazon Translate にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Amazon Translate でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに Amazon Translate リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Amazon Translate がこれらの機能をサポートしているかどうかについては、「Amazon Translate で IAM が機能する仕組み」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、「IAM ユーザーガイド」の「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

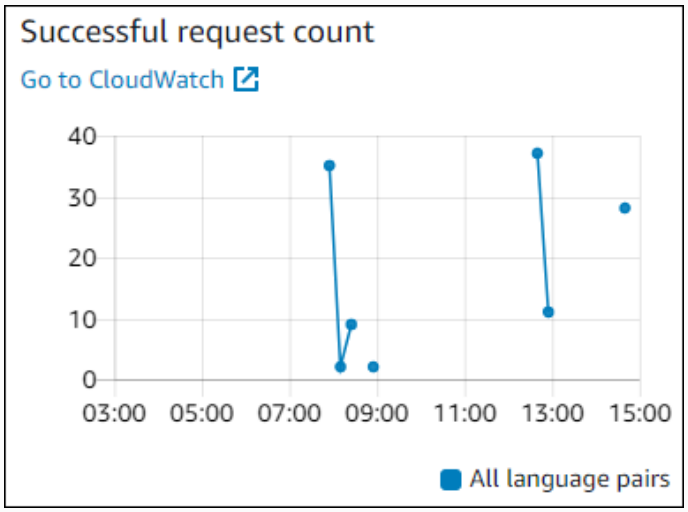
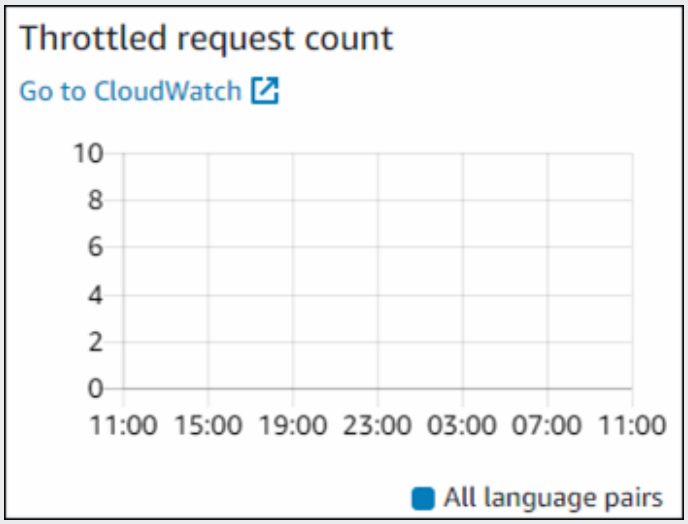
Amazon Translate のモニタリング

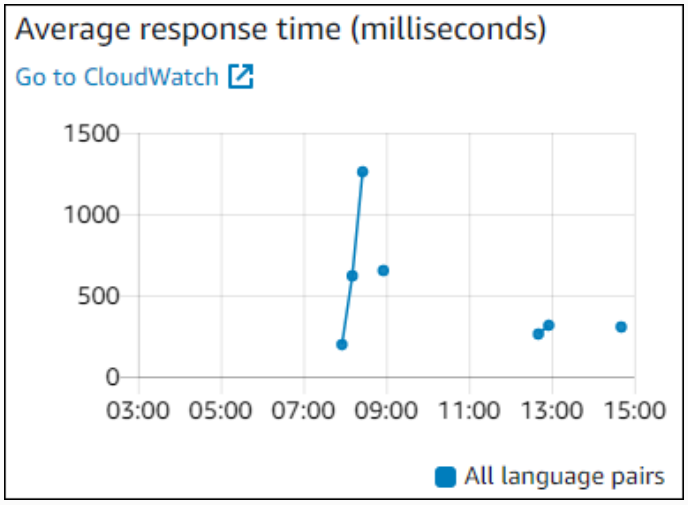
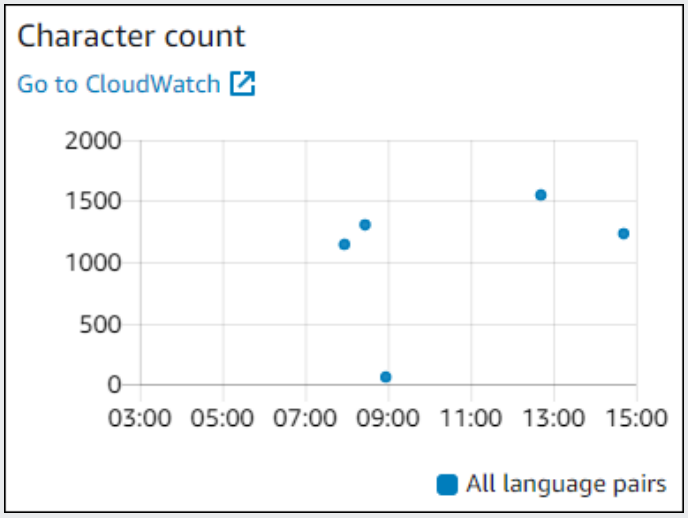
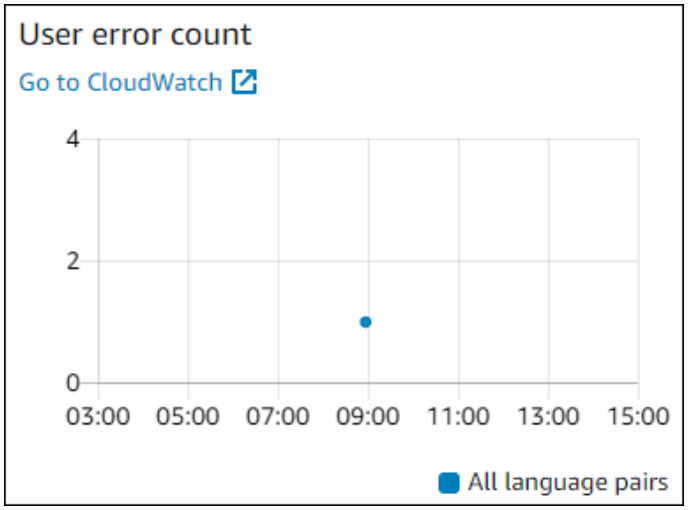
モニタリングは、Amazon Translate と ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。 は、Amazon Translate のモニタリングに使用できるさまざまなツール AWS を提供します。こうしたツールをいくつか設定することで、ソリューションのモニタリングをカスタマイズできます。モニタリングタスクをできるだけ自動化することをお勧めします。

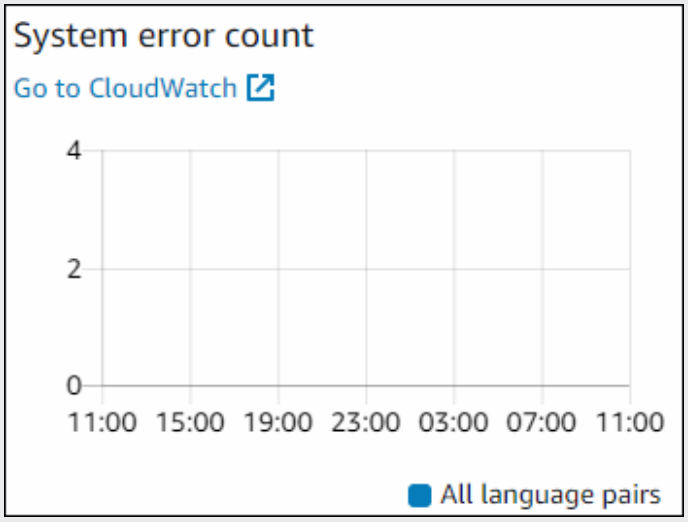
Amazon Translate はソリューションでもっとも重要なメトリクスを表示する事前設定済みのグラフを提供します。各グラフはソリューションのパフォーマンス概要を示しています。ソリューションパフォーマンスの経時的な違いを見るためには、グラフで表示する時間範囲を変更できます。

また、Amazon CloudWatch を使用して、Amazon Translate をモニタリングすることもできます。CloudWatch を使用すると、ソリューションの特定のメトリクスのモニタリングを自動化できます。メトリクスが設定したしきい値の外にある場合は通知が届きます。CloudWatch API を使用して、ご自分のニーズに合わせたカスタムモニタリングアプリケーションを作成できます。詳細については、Amazon CloudWatch ユーザーガイドの「Amazon CloudWatch とは」を参照してください。

次の表は Amazon Translate が提供する事前設定済みのグラフをそれぞれ説明したものです。

グラフ	説明
	成功したリクエストカウント 指定した期間中に成功した Amazon Translate へのリクエスト数です。
	調整済みリクエストカウント 指定した期間中に調整した Amazon Translate へのリクエスト数です。この情報を使用してアプリケーションが Amazon Translate にリクエストを送信するのが早すぎるか判断します。

グラフ	説明
	平均応答時間 Amazon Translate が指定期間中にリクエストを処理した平均時間です。
	文字数カウント 指定した指定期間中に Amazon Translate へ送信した文字数の合計です。この合計数が請求されます。
	ユーザーエラーカウント 指定した期間中に発生したユーザーエラーの数です。ユーザーエラーは、HTTP エラーコード 400 ~ 499 の範囲内です。

グラフ	説明
	システムエラーカウント 指定した期間中に発生したシステムエラーの数です。システムエラーは、HTTP エラーコード 500 ~ 599 の範囲内です。

Amazon Translate のモニタリング

Amazon CloudWatch では、アカウントの個別の Amazon Translate オペレーションまたはグローバルな Amazon Translate メトリクスのメトリクスを取得できます。メトリクスを使用すると、Amazon Translate ソリューションの状態をトラッキングし、定義されているしきい値を 1 つ以上のメトリクスが超えた場合に通知するようにアラームをセットアップできます。たとえば、指定期間中に Amazon Translate に送信されたリクエスト数をモニタリングして、リクエストのレイテンシーを確認し、エラー数がしきい値を超えた場合はアラームを生成できます。

Amazon Translate の CloudWatch メトリクスの概要

Amazon Translate オペレーションのメトリクスを取得するには、以下の情報を指定する必要があります。

- メトリクスディメンション。ディメンションは、メトリクスを識別するための名前と値のペアのセットです。Amazon Translate には 2 つのディメンションがあります。
  - `Operation`
  - `Language pair`
- メトリクス名 (`SuccessfulRequestCount`、`RequestCharacters` など)。メトリクスの一覧については、「Amazon Translate の CloudWatch メトリクス」を参照してください。

Amazon Translate のメトリクスは、 AWS Management Console、、 AWS CLIまたは CloudWatch API を使用して取得できます。CloudWatch API は、いずれかの Amazon AWS Software Development Kit (SDK) または Amazon CloudWatch API ツールでも使用できます。

以下の表は、CloudWatch メトリクスの一般的な利用方法をいくつか示しています。ここで紹介するのは使用開始するための提案事項であり、総括的な一覧ではありません。

どうすればよいか?	モニタリング対象のメトリクス
成功したリクエストの数をトラッキングします	`SuccessfulRequestCount` のメトリクスの sum 統計
アプリケーションが最大しきい値に到達した場合に通知します	`ThrottledCount` のメトリクスの sum 統計
アプリケーションの応答時間を検索します	`ResponseTime` のメトリクスの average 統計
アプリケーションのエラー数を見つけます	`ServerErrorCount` と `UserErrorCount` のメトリクスの sum 統計をモニタリングします。
請求可能な文字の数を検索します	`CharacterCount` のメトリクスの sum 統計

CloudWatch で Amazon Translate をモニタリングするには、CloudWatch への適切なアクセス許可が必要です。詳細については、Amazon CloudWatch ユーザーガイドの「Amazon CloudWatch に対する認証とアクセスコントロール」を参照してください。

Amazon Translate メトリクスの表示

Amazon Translate コンソールにメトリクスを表示します。

メトリクスを表示する方法 (CloudWatch コンソール)

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudwatch/ で CloudWatch コンソールを開きます。
2. [Metrics] で、[All Metrics]、[AWS/Translate] の順に選択します。
3. ディメンションを選択して、メトリクスの名前を選び [Add to graph] を選択します。

4. 日付範囲の値を選択します。指定した日付範囲のメトリクスカウントがグラフに表示されます。

を使用した Amazon Translate API コールのログ記録 AWS CloudTrail

Amazon Translate は AWS CloudTrail、Amazon Translate の IAM エンティティまたはサービスによって実行されたアクションを記録する AWS サービスである Amazon Translate と統合されています。CloudTrail は、Amazon Translate へのすべての API コールをイベントとしてキャプチャします。これには、Amazon Translate コンソールからのコールと、Amazon Translate API オペレーションへのコードコールが含まれます。CloudTrail 追跡を作成すると、Amazon Simple Storage Service (Amazon S3) バケットへの Amazon Translate イベントなど、CloudTrail のイベントの継続的な配信が可能になります。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Amazon Translate に対するリクエスト、リクエスト元の IP アドレス、リクエストの実行者、リクエスト日時などの詳細を把握できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

トピック

- CloudTrail における Amazon Translate の情報
- Amazon Translate ログファイルエントリの概要

CloudTrail における Amazon Translate の情報

CloudTrail は、 AWS アカウントの作成時にアカウントで有効になります。Amazon Translate でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。 AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、CloudTrail イベント履歴でのイベントの表示を参照してください。

Amazon Translate のイベントなど、 AWS アカウント内のイベントの継続的な記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで作成した証跡がすべての AWS リージョンに適用されます。証跡では、 AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した S3 バケットにログファイルが配信されます。CloudTrail ログで収集されたイベントデータをさらに分析して処理するように、他の AWS サービスを設定できます。詳細については、次を参照してください:

- 追跡を作成するための概要

- CloudTrail がサポートされているサービスと統合
- 「CloudTrail の Amazon SNS 通知の設定」
- 複数のリージョンからの CloudTrail ログファイルの受け取りと複数のアカウントからの CloudTrail ログファイルの受け取り

Amazon Translate のすべてのアクションは CloudTrail によってログとして記録され、API リファレンスセクションに記載されます。例えば、`DeleteTerminology`、`ImportTerminology`、`TranslateText` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この情報は以下のことを確認するのに役立ちます:

- リクエストが、ルートユーザーの認証情報で行われたかどうか
- リクエストが、ロールとフェデレーティッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか
- リクエストが別の AWS サービスによって行われたかどうか

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

Amazon Translate ログファイルエントリの概要

「トレイル」は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどの情報を含みます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

以下の例は、`TranslateText` アクションを示す CloudTrail ログエントリです。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/Administrator",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Administrator"
```

```
    },
    "eventTime": "2019-09-03T20:32:50Z",
    "eventSource": "translate.amazonaws.com",
    "eventName": "TranslateText",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.16.207 Python/3.4.7
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 botocore/1.12.197",
    "requestParameters": {
        "text": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "sourceLanguageCode": "en",
        "targetLanguageCode": "fr"
    },
    "responseElements": {
        "translatedText": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "sourceLanguageCode": "en",
        "targetLanguageCode": "fr"
    },
    "requestID": "f56da956-284e-4983-b6fc-59befa20e2bf",
    "eventID": "1dc75278-84d7-4bb2-861a-493d08d67391",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Amazon Translate の CloudWatch メトリクスとディメンション

ソリューションのパフォーマンスをモニタリングするには、Amazon Translate の Amazon CloudWatch メトリクスとディメンションを使用します。

Amazon Translate の CloudWatch メトリクス

メトリクス	説明
`CharacterCount`	リクエスト内の請求可能な文字の数を検索します。 有効なディメンション: 言語ペア、オペレーション 有効な統計: 平均、最大、平均、合計 単位: 数
`ResponseTime`	リクエストに応答するまでにかかった時間です。

メトリクス	説明
	有効なディメンション: 言語ペア、オペレーション 有効な統計: データサンプル、平均 単位: データサンプルの場合は個。平均統計の場合はミリ秒。
`ServerErrorCount`	サーバーエラーの数。サーバーエラーの HTTP レスポンスコード範囲は 500〜599 です。 有効なディメンション: オペレーション 有効な統計: 合計、平均 単位: 数
`SuccessfulRequestCount`	成功した翻訳リクエストの数です。成功したリクエストのレスポンスコードは 200〜299 です。 有効なディメンション: オペレーション 有効な統計: 合計、平均 単位: 数
`ThrottledCount`	スロットリングリクエスト件名の数です。`ThrottledCount` を使用して、アプリケーションによる Amazon Translate へのリクエストの送信速度が、アカウントでの受信設定よりも早いかどうかを判断します。詳細は「Amazon Translate Limits」を「Amazon Web Services 全般のリファレンス」でご覧ください。 有効なディメンション: オペレーション 有効な統計: 合計、平均 単位: 数

メトリクス	説明
`UserErrorCount`	発生したユーザーエラーの数です。ユーザーエラーの HTTP レスポンスコード範囲は 400〜499 です。 有効なディメンション: オペレーション 有効な統計: 合計、平均 単位: 数

Amazon Translate の CloudWatch ディメンション

次のディメンションを使用して Amazon Translate メトリクスをフィルターすることができます。メトリクスはソース言語とターゲット言語でグループ化されます。

ディメンション	説明
`LanguagePair`	特定の言語だけを含むようにメトリクスを制限します。
`Operation`	特定のオペレーションを使用するものだけにメトリクスを制限します。

Amazon EventBridge による Amazon Translate イベントのモニタリング

Amazon Translate は Amazon EventBridge と統合されており、翻訳ジョブや並列データリソースに影響する変更に関する通知が行われます。 AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。たとえば、自動的に開始できるアクションには以下のようなものがあります。

- AWS Lambda 関数の呼び出し
- AWS Systems Manager Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNS トピックまたは Amazon SQS キューの通知

EventBridge ルールの作成の詳細については、Amazon EventBridge ユーザーガイドの「イベントに反応する Amazon EventBridge ルールの作成」を参照してください 。

Amazon Translate イベント

以下に、Amazon Translate のイベントの例を示します。

バッチ翻訳ジョブのイベント

バッチ翻訳ジョブを実行するには、Amazon Translate コンソールまたは StartTextTranslationJob オペレーションを使用します。Amazon Translate では、これらのジョブが正常に完了した時または失敗した時にイベントが送信されます。これらのイベントは次の例のようになります。

```
{
 "version": "0",
 "id": "CWE-event-id",
 "detail-type": "Translate TextTranslationJob State Change",
 "source": "aws.translate",
 "account": "111122223333",
 "time": "2017-04-22T03:31:47Z",
 "region": "us-east-1",
 "resources": [],
 "detail": {
    "jobId": "01234567-0123-0123-0123-012345678901",
    "jobStatus": "STATUS"
 }
}
```

`jobStatus` 属性の値は、Amazon Translate から送信されたイベントのジョブの状態によって異なります。`jobStatus` 値は次のとおりです。

- `COMPLETED` — ジョブが正常に完了し、出力が使用可能になりました。
- `COMPLETED_WITH_ERROR` — ジョブの完了時にエラーが発生しました。エラーは、ジョブの出力で分析できます。
- `STOPPED` — ジョブが停止されました。
- `FAILED` — ジョブが完了しませんでした。詳細を取得するには、DescribeTextTranslationJob オペレーションを使用してください。

並列データリソースのイベント

Amazon Translate を使用して並列データリソースを作成または更新すると、オペレーションが成功したか失敗したかを示すイベントが送信されます。

Amazon Translate コンソールまたは CreateParallelData オペレーションを使用して、並列データリソースを作成します。これを行うと、Amazon Translate により次のようなイベントが送信されます。

```
{
 "version": "0",
 "id": "CWE-event-id",
 "detail-type": "Translate Parallel Data State Change",
 "source": "aws.translate",
 "account": "111122223333",
 "time": "2017-04-22T03:31:47Z",
 "region": "us-east-1",
 "resources": [arn:aws:translate:us-east-1:111122223333:parallel-data/
ExampleParallelData],
 "detail": {
    "operation": "CreateParallelData",
    "name": "ExampleParallelData",
    "status": "STATUS"
 }
}
```

`status` 属性の値は次の通りです。

- `ACTIVE` — `CreateParallelData` オペレーションが成功し、リソースを使用する準備が整いました。
- `FAILED` – `CreateParallelData` オペレーションは失敗しました。

Amazon Translate コンソールまたは UpdateParallelData オペレーションを使用して、並列データリソースを更新します。これを行うと、Amazon Translate により次のようなイベントが送信されます。

```
{
 "version": "0",
 "id": "CWE-event-id",
 "detail-type": "Translate Parallel Data State Change",
```

```
 "source": "aws.translate",
 "account": "111122223333",
 "time": "2017-04-22T03:31:47Z",
 "region": "us-east-1",
 "resources": [arn:aws:translate:us-east-1:111122223333:parallel-data/
ExampleParallelData],
 "detail": {
    "operation": "UpdateParallelData",
    "name": "ExampleParallelData",
    "status": "STATUS",
    "latestUpdateAttemptStatus": "STATUS",
    "latestUpdateAttemptAt": "2017-04-22T03:31:47Z"
 }
}
```

status 属性は、更新により置き換えられる旧バージョンの並列データリソースのステータスを示します。値は次のとおりです。

- ACTIVE — 旧バージョンは正常に作成または更新されました。
- FAILED — 旧バージョンの作成または更新に失敗しました。

latestUpdateAttemptStatus 属性は、更新により作成される新バージョンの並列データリソースのステータスを示します。値は次のとおりです。

- ACTIVE — UpdateParallelData オペレーションは成功し、更新されたリソースを使用する準備が整いました。
- FAILED – UpdateParallelData オペレーションは失敗しました。

Amazon Translate のコンプライアンス検証

サードパーティーの監査者は、複数のコンプライアンスプログラムの一環として Amazon Translate のセキュリティと AWS コンプライアンスを評価します。このプログラムには、PCI、FedRAMP、HIPAA などがあります。を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「<shared id="AWS"/> Artifact のレポートのダウンロード」をご参照ください。

Amazon Translate を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性や貴社のコンプライアンス目的、適用される法律および規制によって決まります。 AWS では、コンプライアンスに役立つ以下のリソースを提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド — これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、機密性とコンプライアンスに焦点を当てたベースライン環境を AWSにデプロイするためのステップを提供します。
- 「Architecting for HIPAA Security and Compliance」ホワイトペーパー – このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS Config – この AWS サービスは、リソース設定が内部プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。
- AWS Security Hub – この AWS サービスは、 内のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準とベストプラクティスへの準拠を確認するのに役立ちます。

特定のコンプライアンスプログラムの対象となる AWS サービスのリストについては、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。一般的な情報については、「AWS コンプライアンスプログラム」を参照してください。

Amazon Translate の耐障害性

AWS グローバルインフラストラクチャは、 AWS リージョンとアベイラビリティーゾーンを中心に構築されています。 AWS リージョンは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーンを提供します。アベイラビリティーゾーンでは、アベイラビリティーゾーン間で中断せずに、自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、およびスケーラビリティが優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、AWS 「 グローバルインフラストラクチャ」を参照してください。

Amazon Translate のインフラストラクチャセキュリティ

マネージドサービスである Amazon Translate は、ホワイトペーパー「Amazon Web Services: セキュリティプロセスの概要」に記載されている AWS グローバルネットワークセキュリティ手順で保護されています。

ネットワーク経由で Amazon Translate にアクセスするには、 AWS 公開された API コールを使用します。クライアントは、TLS 1.2 以降をサポートしている必要があります。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

さらに、 (AWS Identity and Access Management IAM) プリンシパルに関連付けられたアクセスキー ID とシークレットアクセスキーを使用してリクエストに署名する必要があります。またはAWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon Translate とインターフェイス VPC エンドポイント (AWS PrivateLink)

VPC と Amazon Translate とのプライベート接続を確立するには、インターフェイス VPC エンドポイント を作成します。インターフェイスエンドポイントは AWS PrivateLink テクノロジーを利用していますので、Amazon Translate API にプライベートにアクセスできますので、インターネットゲートウェイ、NAT デバイス、VPN 接続、 AWS ダイレクトコネクト接続のいずれも必要としません。VPC のインスタンスは、パブリック IP アドレスがなくても Amazon Translate API と通信できます。VPC と Amazon Translate との間のトラフィックは、Amazon ネットワークを離れません。

各インターフェースエンドポイントは、サブネット内の 1 つ以上の Elastic Network Interface によって表されます。

詳細については、「Amazon VPC ユーザーガイド」の「インターフェイス VPC エンドポイント (AWS PrivateLink)」を参照してください。

Amazon Translate VPC エンドポイントに関する考慮事項

Amazon Translate 用の VPC エンドポイントを設定する前に、Amazon VPC ユーザーガイド の「インターフェイスエンドポイントのプロパティと制限」を確認してください。

Amazon Translate は、VPC からのすべての API アクションの呼び出しをサポートしています。

Amazon Translate 用のインターフェイス VPC エンドポイントの作成

Amazon Translate サービスの VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface () を使用して作成できますAWS CLI。詳細については、 Amazon VPC ユーザーガイド のインターフェイスエンドポイントの作成を参照してください。

以下のサービス名を使用して、Amazon Translate のエンドポイントを作成します。

- com.amazonaws.*region*.translate

エンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`translate.us-east-1.amazonaws.com` など) を使用して、Amazon Translate への API リクエストを実行できます。

詳細については、「Amazon VPC ユーザーガイド」の「インターフェイスエンドポイントを介したサービスへのアクセス」を参照してください。

Amazon Translate 用の VPC エンドポイントポリシーの作成

Amazon Translate へのアクセスをコントロールするVPC エンドポイントにエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、「Amazon VPC ユーザーガイド」の「VPC エンドポイントでサービスへのアクセスを制御する」を参照してください。

例: Amazon Translate のリアルタイム翻訳アクションの VPC エンドポイントポリシー

以下は、Amazon Translate でリアルタイム翻訳を行うためのエンドポイントポリシーの例です。このポリシーは、エンドポイントにアタッチされると、すべてのリソースのすべてのプリンシパルに対して、登録されている Amazon Translate アクションへのアクセスを許可します。

```
{
            "Statement":[
                {
                    "Principal":"*",
                    "Effect":"Allow",
                    "Action":[
                        "translate:TranslateText",
                    ],
                    "Resource":"*"
                }
            ]
```

```
        }
```

例: Amazon Translate バッチ翻訳アクションのための VPC エンドポイントポリシー

Amazon Translateのバッチ翻訳のためのエンドポイントポリシーの例を次に示します。このポリシーは、エンドポイントにアタッチされると、すべてのリソースのすべてのプリンシパルに対して、登録されている Amazon Translate アクションへのアクセスを許可します。

```
{
            "Statement":[
                {
                    "Principal":"*",
                    "Effect":"Allow",
                    "Action":[
                        "translate:StartTextTranslationJob",
                        "iam:PassRole"
                    ],
                    "Resource":"*"
                }
            ]
        }
```

ガイドラインとクォータ

次のセクションには、Amazon Translate のガイドラインとクォータに関する情報が含まれています。

トピック

- サポートしている AWS リージョン
- コンプライアンス
- スロットリング
- ガイドライン
- Service Quotas

サポートしている AWS リージョン

Amazon Translate をサポートする AWS リージョンのリストについては、 AWS 全般のリファレンスのAmazon Translate エンドポイントとクォータ」を参照してください。

コンプライアンス

Amazon Translate コンプライアンスプログラムの詳細については、「AWS コンプライアンス」、「AWS コンプライアンスプログラム」、および「AWS コンプライアンスプログラムによる対象範囲内のサービス」を参照してください。

スロットリング

Amazon Translate は、お客様の運用トラフィックに対応するようにスケールされます。継続的なスロットリングが発生した場合は、AWS Supportにお問い合わせください。

ガイドライン

Amazon Translate では、分析モデルの品質を継続的に向上させるため、お客様のデータを保存する場合があります。詳細については、「Amazon Translate に関するよくある質問」を参照してください。

AWS Support にお問い合わせいただくと、お客様のデータを削除するとともに、お客様のアカウントに関連付けられているデータが今後保存されないようリクエストできます。ただし、お客様のデー

タを削除すると、翻訳の改善に役立つ独自のトレーニングデータも削除される可能性があるため、翻訳の品質が低下することがあります。

Service Quotas

Amazon Translate には次のサービスガイドラインとクォータがあります。

同期リアルタイム翻訳のクォータ

説明	[制限]
文字エンコーディング	UTF-8
最大入力テキスト	10,000 バイト
ドキュメントあたりの最大文字数	100,000
最大ドキュメントサイズ	100,000 bytes

非同期バッチ翻訳のクォータ

説明	[制限]
文字エンコーディング	UTF-8
ドキュメントあたりの最大文字数	1,000,000
ドキュメントあたりの最大サイズ	20 MB
1 つのドキュメント内の翻訳可能なテキストの最大サイズ	1 MB
バッチジョブリクエストのターゲット言語の最大数	10
バッチ内のドキュメントの最大数	1,000,000
バッチ内の合計ドキュメントの最大サイズ	5 GB
同時バッチ翻訳ジョブの最大数	10
キューに入れられたバッチ変換ジョブの最大数	1,000

説明	[制限]
StartTextTranslationJob API アクションの 1 秒あたりのトランザクション	5
DescribeTextTranslationJob API アクションの 1 秒あたりのトランザクション	10
ListTextTranslationJobs API アクションの 1 秒あたりのトランザクション	10
StopTextTranslationJob API アクションのトランザクション/秒 (TPS)	5

カスタム用語のクォータ

説明	[制限]
最大カスタム用語ファイルサイズ	10 MB
各 AWS リージョンの AWS アカウントあたりのカスタム用語の最大数	100
カスタム用語ファイルあたりのターゲット言語の最大数	10
1 カスタム用語あたりのソーステキストとターゲットテキストの最大長	200 バイト
`TranslateText` または `StartTextTranslationJob` リクエストあたりの用語ファイルの最大数。	1
ImportTerminology API アクションの 1 秒あたりのトランザクション	5
GetTerminology API アクションの 1 秒あたりのトランザクション	10
ListTerminologies API アクションの 1 秒あたりのトランザクション	10

説明	[制限]
DeleteTerminology API アクションの 1 秒あたりのトランザクション	5

並列データクォータ

説明	[制限]
各リージョンの AWS アカウントあたりの並列データリソースの最大数	1,000
並列データ入力ファイルの最大サイズ	5 GB
並列データリソース内のソース言語の最大数	1
並列データ入力ファイル内の 1 つのセグメントまたはレコードの最大サイズ	1000 バイト
並列データリソースの作成または更新の同時操作の最大数	1
CreateParallelData API アクションの 1 秒あたりのトランザクション	5
GetParallelData API アクションの 1 秒あたりのトランザクション	10
ListParallelData API アクションの 1 秒あたりのトランザクション	10
UpdateParallelData API アクションの 1 秒あたりのトランザクション	5
DeleteParallelData API アクションの 1 秒あたりのトランザクション	5

Amazon Translate のドキュメント履歴

以下の表は、Amazon Translate の今回のリリースのドキュメント内容をまとめたものです。

変更	説明	日付
新機能: Brevity	Translate では、リアルタイムテキスト翻訳の簡潔さがサポートされるようになりました。Brevity は、ほとんどの翻訳の翻訳出力の長さを短くします (簡潔でない翻訳出力と比較して) 。詳細については、Amazon Translate での簡潔性の使用」を参照してください。	2023 年 10 月 31 日
リアルタイム翻訳へドキュメントを入力する際の言語の自動検出	リアルタイム翻訳 (コンソールまたは API) にドキュメントを入力するときに、言語の自動検出を使用できるようになりました。詳細については、「リアルタイム翻訳」を参照してください。	2023 年 8 月 3 日
リアルタイム翻訳への入力としての Word (.docx) ファイル	テキストファイルと HTML ファイルに加えて、.docx ファイルをリアルタイム翻訳 (コンソールまたは API) への入力として使用できるようになりました。詳細については、「リアルタイム翻訳」を参照してください。	2023 年 7 月 17 日
カスタム用語の強化	Translate は、翻訳の流暢さと正確さを向上させるカスタム用語機能の強化をサポートす	2023 年 6 月 30 日

	るようになりました。詳細については、「カスタム用語による翻訳のカスタマイズ」を参照してください。	
リアルタイム翻訳への入力としてのテキストまたは HTML ファイル	テキストファイルまたは HTML ファイルをリアルタイム翻訳 (コンソールまたは API) への入力とし使用できるようになりました。詳細については、「リアルタイム翻訳」を参照してください。	2023 年 5 月 23 日
TranslateReadOnly ポリシーで許可される新しいアクション	Amazon Translate では、TranslateReadOnly マネージドポリシーで `TranslateDocument` アクションが許可されるようになりました。詳細については、「AWS マネージドポリシー: TranslateReadOnly」を参照してください。	2023 年 5 月 23 日
Translate が、非同期バッチ処理で追加のリージョンをサポートするようになりました。	Translate が、非同期バッチ処理で追加のリージョンをサポートするようになりました。詳細については、「Amazon Translate による非同期バッチ処理」を参照してください。	2023 年 3 月 28 日
リアルタイム翻訳の入力サイズが増えました	リアルタイム翻訳で最大 10,000 文字を入力できるようになりました。詳細については、「Amazon Translate の開始方法 (コンソール)」を参照してください。	2022 年 12 月 16 日

バッチモードにおけるネストされた入力フォルダーのサポート	ネストされた入力フォルダをバッチ翻訳ジョブに提供できるようになりました。詳細については、Amazon Translate での「バッチ翻訳ジョブの実行」を参照してください。	2022 年 11 月 18 日
バッチモードにおける言語の自動検出のサポート	バッチ翻訳ジョブのソース言語を自動検出できるようになりました。その結果、バッチ翻訳ジョブに異なるソース言語の文書を入力することが可能になりました。詳細については、Amazon Translate での「バッチ翻訳ジョブの実行」を参照してください。	2022 年 11 月 18 日
複数のターゲット言語のサポート	バッチ翻訳ジョブで複数のターゲット言語を指定できるようになりました。詳細については、Amazon Translate での「バッチ翻訳ジョブの実行」を参照してください。	2022 年 10 月 10 日
タグのサポート	Amazon Translate で ParallelData リソースとカスタム用語リソースにタグを付けることができるようになりました。詳細については、「Amazon Translate のリソースにタグを付ける」を参照してください。	2022 年 10 月 6 日

さらなる言語での形式のサポート	Amazon Translate では、オランダ語、韓国語、メキシコスペイン語の翻訳形式レベルを設定できるようになりました。詳細については「Amazon Translate の形式設定」を参照してください。	2022 年 10 月 5 日
API リファレンスを分離	「Amazon Translate API リファレンス」は、「デベロッパーガイド」とは別のドキュメントになりました。詳細については、「Amazon Translate API リファレンス」を参照してください。	2022 年 8 月 25 日
新機能	翻訳出力の形式レベルを設定できるようになりました。詳細については「Amazon Translate の形式設定」を参照してください。	2022 年 2 月 22 日
新機能	翻訳出力内の不適切な単語やフレーズをマスクできるようになりました。詳細については「Amazon Translate における不適切な言葉やフレーズのマスキング」を参照してください。	2021 年 11 月 24 日

AWS PrivateLink のサポート	を使用して、VPC と Amazon Translate の間にプライベート接続を確立できるようになりました AWS PrivateLink。詳細については、「Amazon Translate とインターフェイス VPC エンドポイント (AWS PrivateLink)」を参照してください。	2021 年 11 月 24 日
並列データ更新	Amazon Translate でサポートされている言語を使用した並列データリソースを作成できるようになりました。言語の 1 つとして英語を使用する必要はなくなりました。並列データの詳細については、「並列データによる翻訳のカスタマイズ (Active Custom Translation)」を参照してください。	2021 年 11 月 15 日
カスタム用語の方向性	任意の言語をソース言語またはターゲット言語とする多方向の用語集を作成できます。詳細については、「カスタム用語の作成」を参照してください。	2021 年 11 月 11 日

新しい言語	Amazon Translate では、アイルランド語、マラーティー語、ポルトガル語 (ポルトガル)、パンジャブ語がサポートされるようになりました。Amazon Translate でサポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。	2021 年 11 月 10 日
新しいカスタム暗号化設定	AWS Key Management Serviceで管理する独自のカスタマーマネージドキーを使用して、翻訳出力を暗号化できるようになりました。詳細については、「バッチ翻訳ジョブの実行」を参照してください。	2021 年 11 月 5 日
新たなファイル形式サポート	Amazon Translate では、非同期バッチ処理用として XML Localization Interchange File Format (XLIFF) ファイルがサポートされるようになりました。サポートされる形式については、「サポートされるファイル形式」を参照してください。	2021 年 6 月 9 日

EventBridge 統合	翻訳ジョブや並列データリソースに影響する変更に関する通知を行うために、Amazon Translate から Amazon EventBridge にイベントが送信されるようになりました。詳細については「Amazon EventBridge による Amazon Translate イベントのモニタリング」を参照してください。	2021 年 6 月 4 日
新しいクォータ	Amazon Translate では、キューに入れられた最大 1000 個のバッチ翻訳ジョブがサポートされるようになりました。Amazon Translate のすべてのクォータについては、「ガイドラインと制限」を参照してください。	2021 年 4 月 23 日
クォータの増加	並列データ入力ファイルの最大サイズが 1 MB から 5 MB に増加しました。Amazon Translate のすべてのクォータについては、「ガイドラインと制限」を参照してください。	2021 年 3 月 31 日

新しい言語	Amazon Translate では現在、アルメニア語、カタロニア語、ペルシア語 (ペルシア語) 、フィリピン語タガログ語、グジャラート語、ハイチクレオール語、アイスランド語、カンナダ語、カザフ語、リトアニア語、マケドニア語、マラヤーラム語、マルタ語、モンゴル語、シンハラ語、テルグ語、ウズベク語、ウェールズ語がサポートされています。Amazon Translate でサポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。	2020 年 11 月 23 日
新機能	並列データを使用してバッチ翻訳ジョブをカスタマイズできるようになりました。並列データはソーステキストの例とその翻訳例で構成されています。並列データを使用するジョブは、Active Custom Translation (アクティブカスタム翻訳) ジョブと呼ばれます。これらのジョブの実行中、Amazon Translate では並列データ内の例が反映されるように翻訳出力が調整されます。詳細については、「並列データによる翻訳のカスタマイズ (Active Custom Translation)」を参照してください。	2020 年 11 月 23 日

新たなファイル形式サポート	Amazon Translate では、非同期バッチ処理の入力として、Word ドキュメント (.docx)、PowerPoint プレゼンテーション (.pptx)、Excel ワークブック (.xlsx) という Office Open XML ファイル形式がサポートされるようになりました。詳細については、「バッチ翻訳ジョブの開始」を参照してください。	2020 年 7 月 29 日
新しい言語	Amazon Translate で、スペイン語 (メキシコ) の翻訳がサポートされるようになりました。サポートされているすべての言語については、「サポートされている言語と言語コード」を参照してください。	2020 年 4 月 30 日
新しいリージョン	Amazon Translate では 欧州 (ロンドン) リージョンにおける非同期バッチ処理がサポートされています。非同期バッチ処理を利用できるすべての AWS リージョンについては、「利用可能なリージョン」を参照してください。	2020 年 4 月 20 日
新機能	Amazon Translate に非同期バッチ翻訳機能が追加されました。詳細については、「非同期バッチ処理」を参照してください。	2019 年 12 月 23 日

新しいリージョン	Amazon Translate では、アジアパシフィック (香港)、アジアパシフィック (シドニー)、欧州 (ロンドン)、欧州 (パリ)、欧州 (ストックホルム)、米国西部 (北カリフォルニア) の各リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「AWS リージョン表」または「AWS のリージョンとエンドポイント」を参照してください。	2019 年 11 月 25 日
新しい言語	Amazon Translate では、翻訳対象の新しい言語として、アフリカーンス語、アルバニア語、アムハラ語、アゼルバイジャン語、ベンガル語、ボスニア語、ブルガリア語、カナダフランス語、クロアチア語、ダリ語、エストニア語、グルジア語、ハウサ語、ラトビア語、パシュト語、セルビア語、スロバキア語、スロベニア語、ソマリア語、スワヒリ語、タガログ語、タミル語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「サポートされている言語」を参照してください。	2019 年 11 月 25 日

新しい言語	Amazon Translate では、翻訳可能な言語として、ギリシャ語、ハンガリー語、ルーマニア語、タイ語、ウクライナ語、ウルドゥー語、およびベトナム語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「サポートされている言語」を参照してください。	2019 年 10 月 3 日
新機能	Amazon Translate に FedRAMP コンプライアンスが追加されました。詳細については、「コンプライアンス」を参照してください。	2019 年 7 月 31 日
新機能	Amazon Translate に SOC コンプライアンスが追加されました。詳細については、「コンプライアンス」を参照してください。	2019 年 5 月 30 日

新しいリージョン	Amazon Translate では、アジアパシフィック (ムンバイ)、アジアパシフィック (シンガポール)、アジアパシフィック (東京)、およびカナダ (中部) の各リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「AWS リージョン表」または「AWS のリージョンとエンドポイント」を参照してください。	2019 年 5 月 8 日
新しい言語	Amazon Translate では、翻訳対象の新しい言語として、ヒンディー語、マレー語、ノルウェー語、ペルシャ語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「サポートされている言語」を参照してください。	2019 年 5 月 6 日

新しいリージョン	Amazon Translate では、欧州 (フランクフルト) およびアジアパシフィック (ソウル) リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「AWS リージョン表」または「AWS のリージョンとエンドポイント」を参照してください。	2019 年 2 月 28 日
新機能	Amazon Translate に PCI コンプライアンスが追加されました。詳細については、「コンプライアンス」を参照してください。	2018 年 12 月 12 日
新機能	Amazon Translate では、翻訳中により高度な制御を可能にするため、4 つの新しい API とカスタム用語機能が追加されました。翻訳リクエストでカスタム用語を使用すると、ブランド名、キャラクター名、モデル名、およびその他の独自コンテンツを、標準的な翻訳やコンテキストとは無関係に、毎回目的どおりに翻訳することができます。詳細については、「カスタム用語」を参照してください。	2018 年 11 月 27 日

新しい言語	Amazon Translate では、デンマーク語、オランダ語、フィンランド語、ヘブライ語、インドネシア語、韓国語、ポーランド語、およびスウェーデン語でドキュメントを翻訳できるようになりました。Amazon Translate ではサポートされていない言語ペアの数を大幅に減らすことで直接翻訳を継続的に強化しています。Amazon Translate で直接翻訳できる言語の組み合わせについては、「サポートされている言語」を参照してください。	2018 年 11 月 20 日
新機能	Amazon Translate に英語以外のサポート対象言語間の直接翻訳が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせについては、「サポートされている言語」を参照してください。	2018 年 10 月 29 日
新機能	Amazon Translate に HIPAA コンプライアンスが追加されました。詳細については、「コンプライアンス」を参照してください。	2018 年 10 月 25 日

新機能	Amazon Translate では、新たな翻訳言語として、繁体字中国語、チェコ語、イタリア語、日本語、ロシア語、トルコ語が追加されました。Amazon Translate でサポートされている言語のリストについては、「サポートされている言語」を参照してください。	2018 年 7 月 17 日
新機能	Amazon Translate にソース言語の自動検出へのサポートが追加されました。詳細については、「Amazon Translate の仕組み」を参照してください。	2018 年 4 月 4 日
新しいガイド	Amazon Translate デベロッパーガイド の初回リリースです。	2017 年 11 月 29 日

API リファレンス

「Amazon Translate API リファレンス」は独立したドキュメントになりました。詳細については、「Amazon Translate API リファレンス」を参照してください。

AWS 用語集

最新の AWS 用語については、「 AWS の用語集 リファレンス」のAWS 「 用語集」を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。