



デベロッパーガイド

AWS SDK for SAP ABAP



AWS SDK for SAP ABAP: デベロッパーガイド

Copyright © 2026 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

とは AWS SDK for SAP ABAP	1
の機能 AWS SDK for SAP ABAP	1
メンテナンス	1
API リファレンス	2
料金	2
リソース	2
開始方法	3
ステップ 1: AWS アカウントを準備する	3
SAP ユーザーの IAM ロール	3
認証	5
ステップ 2: SDK をインストールする	6
ステップ 3: SDK を設定する	6
ステップ 4: 機能のセットアップ	8
ステップ 5: SAP ユーザーを承認する	10
ステップ 6: コードを記述する	12
ステップ 7: アプリケーションを実行する	15
セットアップ	17
SAP の前提条件	17
SDK for SAP ABAP	17
SDK for SAP ABAP - BTP エディション	21
のインストール AWS SDK for SAP ABAP	22
SDK をダウンロードする	22
ファイルを検証する	22
AWS SDK トランスポート	23
SDK - BTP エディションのインストール	27
SDK for SAP ABAP - BTP エディションをインストールする	27
モジュール	28
SDK for SAP ABAP - BTP エディションへのパッチ適用	29
設定	30
[Global settings] (グローバル設定)	31
技術設定	32
シナリオの設定	32
アプリケーションの設定	33
SDK プロファイル	33

論理リソースリゾルバー	35
例	35
ランタイム設定	35
ログとトレース	36
オプトイン: 拡張テレメトリー	36
アクティブシナリオ	36
高度な接続シナリオ	36
プロキシサーバー経由の接続	37
パケット検査ファイアウォール経由の接続	38
ゲートウェイエンドポイント	38
カスタムインターフェイスのエンドポイント	38
高度なルーティング	40
サービスごとのプロキシサーバーオーバーライド	40
複数のリージョンでエンドポイントにアクセスする	42
サービスプロバイダーの設定	43
更新、トレース、テレメトリー	43
SAP システムの更新	44
トレース	44
Telemetry	45
SDK の使用	47
データの表現	47
データ型	48
AWS データ型	50
プログラムの例	50
前提条件	51
コード	51
コードセクション	52
概念	55
API クラス	55
追加のオブジェクト	55
構造クラス	56
配列	58
マップ	59
高レベルの関数	59
機能	1
プログラムによる設定	60

ウェーター	61
ページネーター	62
再試行動作	63
署名者	64
クロスアカウント IAM ロールの連鎖	64
製品の構築	64
製品 ID を設定する	65
HTTP リクエストを にカスタマイズする AWS	65
エンハンスメントを実装する	65
エンハンスメントをフィルタリングする	65
エンハンスメントをコーディングする	66
制限事項	66
コードの例	68
ACM	70
アクション	70
API ゲートウェイ	78
アクション	70
Application Recovery Controller	87
アクション	70
Aurora	92
アクション	70
Auto Scaling	98
アクション	70
Amazon Bedrock ランタイム	107
Anthropic Claude	108
Stable Diffusion	111
Amazon Bedrock エージェントランタイム	114
アクション	70
CloudFront	115
アクション	70
CloudWatch	118
アクション	70
シナリオ	126
CloudWatch Logs	128
アクション	70
Amazon Cognito ID プロバイダー	131

アクション	70
Amazon Comprehend	136
アクション	70
AWS Config	150
アクション	70
AWS Control Tower	153
アクション	70
Firehose	164
アクション	70
DynamoDB	166
基本	167
アクション	70
Amazon EC2	181
アクション	70
Amazon ECR	204
アクション	70
Amazon EMR	210
アクション	70
EventBridge スケジューラ	216
アクション	70
AWS Glue	222
アクション	70
HealthImaging	235
アクション	70
HealthLake	253
アクション	70
IAM	266
アクション	70
AWS IoT data	292
アクション	70
AWS IoT SiteWise	294
はじめに	294
アクション	70
Amazon Keyspaces	303
アクション	70
Kinesis	311

基本	167
アクション	70
AWS KMS	321
はじめに	294
アクション	70
Lambda	342
基本	167
アクション	70
組織	356
アクション	70
Amazon Pinpoint	361
アクション	70
Amazon Pinpoint SMS および音声 API	365
アクション	70
Amazon Polly	375
アクション	70
Amazon RDS	384
アクション	70
Amazon Redshift	389
アクション	70
Amazon Rekognition	397
アクション	70
Amazon S3	411
基本	167
アクション	70
シナリオ	126
SageMaker AI	438
アクション	70
シナリオ	126
Secrets Manager	456
アクション	70
Amazon SES	458
アクション	70
Amazon SES API v2	475
アクション	70
Amazon SNS	484

アクション	70
シナリオ	126
Amazon SQS	495
アクション	70
シナリオ	126
ステップ関数	505
アクション	70
Systems Manager	513
アクション	70
Amazon Textract	523
アクション	70
シナリオ	126
Amazon Transcribe	534
アクション	70
Amazon Translate	542
アクション	70
シナリオ	126
MCP サーバー	551
利点	551
MCP サーバーのセットアップ	552
MCP サーバーでできること	553
重要な考慮事項	553
セキュリティ	555
システム認証	556
メタデータ認証	556
シークレットアクセスキー認証	557
IAM Roles Anywhere を使用した証明書ベースの認証	557
クロスアカウントアクセスのソースプロファイル認証	558
次のステップ	558
IAM セキュリティのベストプラクティス	558
Amazon EC2 インスタンスプロファイルのベストプラクティス	559
SAP ユーザーの IAM ロール	560
ソースプロファイルのセキュリティに関する考慮事項	563
SAP 認証	564
設定の権限	564
エンドユーザー向けの SAP 認証	565

セキュアなオペレーション	566
保管中のデータの暗号化	566
転送中のデータの暗号化	567
API の使用	2
シークレットキー (SSF)	567
前提条件	567
手順	568
IAM Roles Anywhere	570
前提条件	570
手順	571
ソースプロファイル	64
前提条件	574
手順	575
認証情報ストア	21
設定手順	577
SDK での SAP 認証情報ストアの使用	579
トラブルシューティング	583
インポートの失敗	583
未指定の場所の制約	583
SSL エラー	584
プロフィール設定	585
IAM 認可	586
アクションの承認	586
アクティブシナリオ	36
特殊文字	587
接続	587
その他のトピック	589
リリース	589
リリース戦略	589
ベストプラクティス	558
SDK for SAP ABAP のパッチ適用	590
追加モジュールのインストール	591
SDK for SAP ABAP のアンインストール	591
SAP ライセンス	592
ドキュメント履歴	593
.....	dxcv

とは AWS SDK for SAP ABAP

AWS SDK for SAP ABAP は、ABAP 言語 AWS の が提供する サービスへのインターフェイスを提供します。SDK を使用すると、ABAP BADIs、レポート、トランザクション、OData サービス、および Amazon Simple Storage Service (Amazon S3) AWS のサービス、 、 などの他の ABAP アーティファクト Amazon DynamoDBを に実装できます Amazon Translate。SAP NetWeaver 7.4 から SAP Business Technology Platform 環境で、ABAP ベースのシステム向けに を開発することもできます。詳細については、[AWS 「 SDK for SAP ABAP - BTP エディションのインストール」](#) を参照してください。

トピック

- [の機能 AWS SDK for SAP ABAP](#)
- [SDK メジャーバージョンのメンテナンスとサポート](#)
- [API リファレンス](#)
- [料金](#)
- [その他のリソース](#)

の機能 AWS SDK for SAP ABAP

AWS SDK for SAP ABAP は、SAP 開発者にとって親しみやすく自然に感じられるように設計されています。たとえば、すべての `true` および `false` 文字列 AWS のサービス を使用して XML 構造と JSON 構造でブールデータを表しますが、SDK for SAP ABAP はこれらを ABAP ネイティブ 'X' 値と ' ' 単一文字値に変換します。SDK for SAP ABAP は、データ型やタイムスタンプ形式を含むネイティブ ABAP コンストラクトを可能な限り使用します。そのため、ABAP プログラマーは、基礎となる JSON や XML のシリアル化や API プロトコルのワイヤフォーマットについて心配する必要がありません。

SDK メジャーバージョンのメンテナンスとサポート

SDK メジャーバージョンのメンテナンスとサポート、およびその基礎的な依存関係については、[AWS SDK とツール共有設定および認証情報リファレンスガイド](#)で以下を参照してください。

- [AWS SDKsメンテナンスポリシー](#)
- [AWS SDKsとツール、バージョンサポートマトリックス](#)

API リファレンス

AWS SDK for SAP ABAP APIs 「[AWS SDK for SAP ABAP - API リファレンスガイド](#)」を参照してください。

AWS SDK for SAP ABAP TLAs の完全なモジュールリストを確認するには、「[AWS SDK for SAP ABAP - Module List](#)」を参照してください。

SDK for SAP ABAP - BTP Edition 開発者プレビュー TLAs [AWS 「 SDK for SAP ABAP - BTP Edition - Module List](#)」を参照してください。

料金

AWS SDK for SAP ABAP は追加料金なしで利用できます。SDK で消費する AWS リソースとサービスに対してのみ料金が発生します。

その他のリソース

このガイドに加えて、SDK for SAP ABAP には次のオンラインリソースが利用可能です。

- [AWS SDK for SAP ABAP Knowledge MCP Server](#) – AI 対応 IDE を接続して、AWS 統合用の正確な ABAP コードを生成します。
- [AWS SDK コードサンプルライブラリ](#)
- [SAP on AWS ドキュメント](#)
- [AWS 開発者ブログ](#)
- [AWS 開発者フォーラム](#)
- [@awsdevelopers](#)(Twitter)

の開始方法 AWS SDK for SAP ABAP

このセクションでは、SDK の使用を開始する方法について説明します。SDK のインストール、基本設定の実行、フレーズをある言語から別の言語に翻訳する Hello World コード例の作成に関する情報が含まれています。AWS SDK を初めて使用する場合は、サンドボックス環境でこれらのステップを実行することをお勧めします。

Steps

- [ステップ 1: AWS アカウントを準備する](#)
- [ステップ 2: SDK をインストールする](#)
- [ステップ 3: SDK を設定する](#)
- [ステップ 4: 機能のセットアップ](#)
- [ステップ 5: SAP ユーザーを承認する](#)
- [ステップ 6: コードを記述する](#)
- [ステップ 7: アプリケーションを実行する](#)

ステップ 1: AWS アカウントを準備する

SDK for SAP ABAP の使用を開始するには、アクティブな AWS アカウント が必要です。SAP システムがオンプレミス、SAP Business Technology Platform (BTP)、または別のクラウドプロバイダーでホストされている AWS アカウント 場合でも、 が必要です。

SAP システムが AWS クラウドで実行されている場合、 のサービスを AWS 呼び出します AWS アカウント。

トピック

- [SAP ユーザーの IAM ロール](#)
- [認証](#)

SAP ユーザーの IAM ロール

- 「AWS Identity and Access Management ユーザーガイド」に記載されている手順に従って IAM ロールを作成します。詳細については、「[AWS 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。後で使用できるように IAM ロールの Amazon リソースネーム (ARN) をメモします。

- ユースケースとして Amazon EC2 を選択します。
- ロールの名前として SapDemoTranslate を使用します。
- TranslateReadOnly プロファイルをロールにアタッチします。
- SAP システムがロールを引き受けることができるように、ロールには次のエンティティが必要です。"**111122223333**" を、使用する AWS アカウント番号で置き換えます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "AWS": "111122223333"
      }
    }
  ]
}
```

この例では、のすべてのプリンシパルが AWS アカウント ロールを引き受け"**111122223333**"ることができることを示しています。これは概念実証に適した幅広いアクセス許可です。本番稼働環境には、以下の例のように、より絞り込んだプリンシパルを使用できます。

- 特定のユーザー – SAP システムが次のいずれかを使用している場合。
 - オンプレミス SAP システムから SSF で暗号化された認証情報
 - SAP BTP、ABAP 環境の SAP Credential Store サービスからの認証情報
- 特定のロール – SAP システムが Amazon EC2 にあり、インスタンスプロファイルがある場合。
- Amazon EC2 – SAP システムが Amazon EC2 にあり、インスタンスプロファイルがない場合。

詳細については、「[IAM 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。

認証

認証は SAP システムがホストされている場所によって異なります。

場所

- [AWS クラウド上](#)
- [オンプレミス、SAP BTP、またはその他のクラウド](#)

AWS クラウド上

SAP システムが実行されている EC2 インスタンスに、以下の権限を持つインスタンスプロファイルがあることを確認してください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::111122223333:role/SapDemoTranslate"
    }
  ]
}
```

以前のステップで書き留めた ARN を追加します。

この権限により、SAP システムが ABAP ユーザに代わって SapDemoTranslate ロールを引き受けることができます。

オンプレミス、SAP BTP、またはその他のクラウド

SAP システムがオンプレミス、SAP BTP、またはその他のクラウドにある場合は、次の手順を使用して、シークレットアクセスキーを使用した認証用の接続を確立します。

1. IAM ユーザーを作成します。詳細については、「[IAM ユーザーの作成 \(コンソール\)](#)」を参照してください。

2. IAM ユーザーの名前として SapDemoSID を使用します。SID は、SAP システムのシステム ID です。
3. SapDemoTranslate ロールをこのユーザーに割り当てます。

access_key と を保持しますsecret_access_key。これらの認証情報は SAP システムで設定する必要があります。

Note

SAP システムがオンプレミス、SAP BTP、またはその他のクラウドにある場合は、次のいずれかのオプションを使用して認証できます。

- SSF または SAP 認証情報ストアを使用した[シークレットアクセスキー認証](#)
- [IAM Roles Anywhere での証明書の使用](#)

ステップ 2: SDK をインストールする

インストール手順については、次のタブを参照してください。

SDK for SAP ABAP

SDK for SAP ABAP のトランスポートを SAP システムにインポートします。トランスポートは任意のクライアントにインポートできます。詳細については、[「SDK for SAP ABAP のインストール」](#)を参照してください。

SDK for SAP ABAP - BTP edition

Deploy Product アプリケーションを使用して SDK for SAP ABAP - BTP Edition をインストールします。詳細については、[「SDK for SAP ABAP - BTP エディションのインストール」](#)を参照してください。

ステップ 3: SDK を設定する

SDK を設定する前に、必要な認可があることを確認してください。詳細については、[「SAP 認可」](#)を参照してください。

設定手順については、次のタブを参照してください。

SDK for SAP ABAP

/AWS1/IMG トランザクションを実行して、SDK for SAP ABAP の実装ガイドを開きます。このトランザクションを実行するには、SAP システムのコマンドバーに /n/AWS1/IMG と入力し、[Enter] を選択します。

次の設定を完了します。

- [技術的前提条件] に移動します。
 - 推奨パラメータ <https://docs.aws.amazon.com/sdk-for-sapabap/latest/developer-guide/prerequisites.html#parameters> と [HTTPS 接続](#) を確認します。
- [グローバル設定] → [シナリオの設定] に移動します。
 - [グローバル](#) 設定の推奨事項に従って設定を変更します。
- [グローバル設定] → [技術設定] に移動します。
 - [グローバル](#) 設定の推奨事項に従って設定を変更します。
- [ランタイム設定] → [ログとトレース] に移動します。
 - [新規エントリ] を選択します。
 - トレースレベル: トレースなし。
 - 最大ダンプ行: 100
 - オプトイン: enh telemetry: 空欄のままにしてください。
 - [保存] を選択します。
- [ランタイム設定] → [アクティブシナリオ] に移動します。
 - [新規シナリオ] で DEFAULT を選択します。
 - [シナリオ変更をコミット] を選択します。
 - プロンプトを受け入れます。

オンプレミスシステムの前提条件

SAP システムがオンプレミスまたは別のクラウドで実行されている場合、認証情報は SAP データベースに保存する必要があります。認証情報は SAP SSF を使用して暗号化されるため、SAP の CommonCryptoLib などの設定済みの暗号化ライブラリが必要です。

SDK for SAP ABAP 用に SSF を設定する手順は、/AWS1/IMG トランザクションで説明されています。

Note

SAP システムが Amazon EC2 で実行されている場合、前述の前提条件は適用されません。Amazon EC2 で実行されている SAP システムは、Amazon EC2 インスタンスのメタデータから有効期間が短く、自動的に更新される認証情報を取得します。

SDK for SAP ABAP - BTP edition

ウェブブラウザで ABAP 環境を開き、カスタムビジネス設定アプリケーションに移動します。

次の設定を完了します。

- 「シナリオの設定」を参照してください。
- [グローバル](#)設定の推奨事項に従って設定を変更します。
- 「技術設定」に移動します。
- [グローバル](#)設定の推奨事項に従って設定を変更します。

ステップ 4: 機能のセットアップ

セットアップ手順については、次のタブを参照してください。

SDK for SAP ABAP

トランザクションを実行し /AWS1/IMG (コマンドバー/n/AWS1/IMGに「」と入力し、「Enter」を選択します)、AWS SDK の実装ガイドを開きます。

- [アプリケーション設定] → [SDK プロファイル] に移動します。
- [新規エントリ] を選択します。
- プロファイル: DEMO
- 説明: Demo profile
- [保存] を選択します。
- 作成したエントリを強調表示し、[認証と設定] ツリーブランチをクリックします。
- [新規エントリ] を選択します。
- SID: 現在使用している SAP システムのシステム ID。
- クライアント: 現在使用している SAP システムのクライアント。

- シナリオ ID: Basis 管理者が作成したデフォルトシナリオが表示されるドロップダウンリスト。
- AWS リージョン: 呼び出し先の AWS リージョンを入力します。SAP システムが実行中の場合は AWS、実行中の AWS リージョンを入力します。
- 認証方法:
 - SAP システムが Amazon EC2 で実行されている場合は、[メタデータを介したインスタンスロール] を選択します。
 - SAP システムがオンプレミスまたは他のクラウドで実行されている場合は、[SSF ストレージからの認証情報] を選択します。
 - [認証情報の設定] を選択します。
 - 前のステップで作成したアクセスキー ID とシークレットアクセスキーを入力します。
- [IAM ロールを無効にする] は空白のままにしておきます。
- [保存] を選択します。
- [IAM ロールマッピング] ツリーブランチをクリックします。
 - [新規エントリ] を選択します。
 - [シーケンス番号] に 010 と入力します。
 - [論理 IAM ロール] に TESTUSER と入力します。
 - [IAM ロール ARN] に、前のステップで作成した TranslateReadOnly ポリシーを含む IAM ロールの `arn:aws:` を入力します。

SDK for SAP ABAP - BTP edition

SAP 認証情報ストアを使用して認証を設定します。詳細については、[「SAP 認証情報ストアの使用」](#)を参照してください。

ウェブブラウザで ABAP 環境を開き、カスタムビジネス設定アプリケーションに移動します。

- SDK プロファイルに移動します。
 - 編集 を選択して新しいプロファイルを作成します。
 - プロファイル: DEMO
 - 説明: Demo profile
 - 作成したエントリの横にある右矢印キーを選択して、認証と設定タブに移動します。

[新規エントリ] を選択します。

- SID: 現在使用している SAP システムのシステム ID。
- クライアント: 現在使用している SAP システムのクライアント。
- シナリオ ID: Basis 管理者が作成したデフォルトシナリオが表示されるドロップダウンリスト。
- AWS リージョン: 呼び出し先の AWS リージョンを入力します。SAP システムが実行中の場合は AWS、実行中の AWS リージョンを入力します。
- 認証方法: SAP 認証情報ストアから認証情報を選択します。
- SAP 認証情報ストアに保存されている認証情報の名前空間とキー名を入力します。
- SDK for SAP ABAP - BTP エディションと SAP 認証情報ストア間の通信を確立するために作成された通信協定の名前を入力します。
- [IAM ロールを無効にする] は空白のままにしておきます。
- 作成したエントリの横にある右矢印キーを右クリックして、IAM ロールマッピングタブに移動します。

[新規エントリ] を選択します。

- [シーケンス番号] に 010 と入力します。
- [論理 IAM ロール] に TESTUSER と入力します。
- [IAM ロール ARN] に、前のステップで作成した TranslateReadOnly ポリシーを含む IAM ロールの `arn:aws:` を入力します。

ステップ 5: SAP ユーザーを承認する

SAP ユーザーには、デフォルトで AWS 機能を使用する権限がありません。ユーザーは SAP 認証を使用して明示的に認証を受ける必要があります。詳細については、次のタブを参照してください。

SDK for SAP ABAP

PFCG ロールを作成する

- トランザクション PFCG に移動します
- ロール名 `ZAWS_SDK_DEMO_TESTUSER` を入力し、[単一ロールを作成] を選択します。
 - 説明: Role for demo AWS SDK functionality
 - [認証] タブに移動します。

- 承認データの変更を選択し、情報ポップアップを受け入れます。
- [テンプレートを選択] ポップアップで、[テンプレートを選択しない] を選択します。
- ツールバーから [手動で追加] を選択します。
- 以下の認可オブジェクトを追加します。
 - /AWS1/LROL
 - /AWS1/SESS
- 認可ツリーに次のように入力します。
 - AWS API にアクセスするためのプロファイル: DEMO
 - 論理 IAM ロール: TESTUSER
- [保存] を選択します。
- [生成] を選択します。
- [戻る] を選択します。
- [保存] を選択してロールを保存します。

PFCEG ロールを SAP ユーザーに割り当てる

ZAWS_SDK_DEMO_TESTUSER ロールが割り当てられているユーザーには、AWS SDK プロファイルで設定された設定で DEMO SDK 関数を使用する権限が付与されます。また、承認されたユーザーは、そのプロファイルの TESTUSER 論理 IAM ロールにマップされた IAM ロールも引き受けます。

- トランザクション SU01 を実行します。
 - AWS SDK 機能をテストする SAP ユーザーのユーザー ID を入力します。
 - [変更] を選択します。
 - [ロール] タブに移動し、ユーザーに ZAWS_SDK_DEMO_TESTUSER ロールを割り当てます。
 - [保存] を選択します。

SDK for SAP ABAP - BTP edition

ビジネスロールを作成する

- ウェブブラウザで ABAP 環境を開き、ビジネスロールの維持アプリケーションに移動します。
- テンプレートから作成を選択し、次の詳細を入力します。
 - テンプレート – を選択します **/AWS1/RT_BTP_ENDUSER**。

- 新しいビジネスロール ID – ID を入力します。
- 新しいビジネスロールの説明 – 説明を入力します。
- OK を選択すると、ビジネスロールのページが表示されます。
- 一般ロールの詳細タブで、アクセスカテゴリに移動し、書き込み、読み取り、値ヘルプフィールドを制限付きに設定します。
- 制限の維持を選択し、左側のナビゲーションペインから割り当てられた制限タイプを展開します。「制限と値」セクションの次のフィールドを更新します。
- SDK セッションの選択で、SDK プロファイルの横にある鉛筆アイコンを選択し、範囲タブに移動します。と入力し **DEMO**、追加を選択します。
- 「論理 IAM ロールの選択」で、論理 IAM ロールの横にある鉛筆アイコンを選択し、範囲タブに移動します。と入力し **TESTUSER**、追加を選択します。

SDK プロファイルの横にある鉛筆アイコンを選択し、範囲タブに移動します。を入力し **DEMO**、追加 を選択します。

- ビジネスロールテンプレートに戻り、ビジネスユーザータブを開きます。追加 を選択して、SDK 機能をテストする SAP ビジネスユーザーに新しく作成されたビジネスロールを割り当てます。[保存] を選択します。

作成されたビジネスロールに割り当てられたビジネスユーザーは、AWS SDK プロファイルで設定された設定で DEMO SDK 関数を使用する権限が付与されます。また、承認されたユーザーは、そのプロファイルの TESTUSER 論理 IAM ロールにマップされた IAM ロールも引き受けます。

ステップ 6: コードを記述する

詳細については、次のタブを参照してください。

SDK for SAP ABAP

1. トランザクション SE38 を開きます。
 - プログラム名として ZDEMO_TRANSLATE_HELLO_WORLD を入力します。
 - Create を選択します。
 - タイトルとして AWS SDK Hello World In Any Language を入力します。
 - タイプ: [実行プログラム] を選択します。
 - ステータス: [テストプログラム] を選択します。

- [保存] を選択します。
- プログラムをローカルオブジェクトとして保存します。

次のコードを追加します。

```
*&-----*
*& Report  ZAWS1_DEMO_XL8_SIMPLE
*&
*&-----*
*& A simple demo of language translation with AWS Translate
*&
*&-----*
REPORT zaws1_demo_xl8_simple.

START-OF-SELECTION.
  PARAMETERS pv_text TYPE /aws1/xl8boundedlengthstring DEFAULT 'Hello, World'
  OBLIGATORY.

  PARAMETERS pv_lang1 TYPE languageiso DEFAULT 'EN' OBLIGATORY.
  PARAMETERS pv_lang2 TYPE languageiso DEFAULT 'ES' OBLIGATORY.

  TRY.
    DATA(go_session) = /aws1/cl_rt_session_aws=>create( 'DEMO' ).
    DATA(go_xl8)      = /aws1/cl_xl8_factory=>create( go_session ).
    DATA(lo_output) = go_xl8->translatetext(
      iv_text          = pv_text
      iv_sourcelanguagecode = CONV /aws1/xl8languagecodestring( pv_lang1 )
      iv_targetlanguagecode = CONV /aws1/xl8languagecodestring( pv_lang2 )
    ).

    WRITE: / 'Source Phrase: ', pv_text.
    WRITE: / 'Target Phrase: ', lo_output->get_translatedtext( ).
  CATCH /aws1/cx_xl8unsuppedlanguage00 INTO DATA(lo_lang).
    WRITE: / 'ERROR' COLOR COL_NEGATIVE,
      'Cannot translate from',
      lo_lang->sourcelanguagecode,
      'to',
      lo_lang->targetlanguagecode.
  CATCH cx_root INTO DATA(lo_root).
    WRITE: / 'ERROR' COLOR COL_NEGATIVE, lo_root->get_text( ).
  ENDTRY.
```

SDK for SAP ABAP - BTP edition

1. ABAP クラスを作成する必要があるパッケージを右クリックし、新規 > ABAP クラスを選択します。
2. クラス名 **ZCL_DEMO_XL8_SIMPLE** に を入力し、クラスの説明を追加します。[次へ] を選択してください。
3. トランスポートリクエストを作成または選択します。完了 を選択します。

次のコードを追加します。

```
CLASS zcl_demo_xl8_simple DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_demo_xl8_simple IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

    TRY.
      " input parameters
      DATA(pv_text) = |Hello, World|.
      DATA(pv_lang1) = |EN|.
      DATA(pv_lang2) = |ES|.

      DATA(go_session) = /aws1/cl_rt_session_aws=>create( 'DEMO' ).
      DATA(go_xl8)      = /aws1/cl_xl8_factory=>create( go_session ).
      DATA(lo_output) = go_xl8->translatetext(
        iv_text          = pv_text
        iv_sourcelanguagecode = pv_lang1
        iv_targetlanguagecode = pv_lang2
      ).

      out->write( |Source Phrase: { pv_text }| ).
      out->write( |Target Phrase: { lo_output->get_translatedtext( ) }| ).
```

```
CATCH /aws1/cx_xl8unsuppelanguage00 INTO DATA(lo_lang).
    out->write( |ERROR - Cannot translate from { lo_lang->sourcelanguagecode }
to { lo_lang->targetlanguagecode }| ).
    CATCH cx_root INTO DATA(lo_root).
    out->write( |ERROR - { lo_root->get_text( ) }| ).
ENDTRY.
ENDMETHOD.
ENDCLASS.
```

SDK を使用する ABAP コードを記述する方法の詳細については、[「AWS SDK for SAP ABAP の使用」](#)を参照してください。

ステップ 7: アプリケーションを実行する

詳細については、次のタブを参照してください。

SDK for SAP ABAP

SE38 でアプリケーションを実行します。成功すると、次のような出力が表示されます。

```
Source Phrase: Hello, World
Target Phrase: Hola, mundo
```

認証、設定、または Basis の前提条件を満たしていない場合、エラーメッセージが表示されることがあります。次の例を参照してください。

```
ERROR Could not find configuration under profile DEMO with
scenario DEFAULT for SBX:001
```

SAP ロールで SDK プロファイルの使用と論理 IAM ロールへのマッピングが許可されているのに、SAP システムが IAM ロールを引き受けるように IAM 権限が設定されていない場合、次のような出力になります。

```
ERROR Could not assume role arn:aws:iam::111122223333:role/SapDemoTranslate
```

この場合は、[the section called “ステップ 1: AWS アカウントを準備する”](#) で定義されている IAM ロール、ユーザー、またはその両方で IAM 権限と信頼設定を確認してください。

SDK for SAP ABAP - BTP edition

Eclipse > Run As > ABAP Application (コンソール) でアプリケーションを実行します。成功すると、次のような出力が表示されます。

```
Source Phrase: Hello, World
Target Phrase: Hola, mundo
```

認証、設定、または Basis の前提条件を満たしていない場合、エラーメッセージが表示されることがあります。次の例を参照してください。

```
ERROR Could not find configuration under profile DEMO with
scenario DEFAULT for SBX:001
```

SAP ロールで SDK プロファイルの使用と論理 IAM ロールへのマッピングが許可されているのに、SAP システムが IAM ロールを引き受けるように IAM 権限が設定されていない場合、次のような出力になります。

```
ERROR Could not assume role arn:aws:iam::111122223333:role/SapDemoTranslate
```

この場合は、[the section called “ステップ 1: AWS アカウントを準備する”](#) で定義されている IAM ロール、ユーザー、またはその両方で IAM 権限と信頼設定を確認してください。

セットアップ

このセクションでは、AWS SDK for SAP ABAPを使用する開発環境を設定する方法について説明します。

トピック

- [SAP の前提条件](#)
- [のインストール AWS SDK for SAP ABAP](#)
- [AWS SDK for SAP ABAP - BTP エディションのインストール](#)

SAP の前提条件

SAP システムが でホストされている場合、SDK をインストールするための以下の前提条件が適用されます AWS。

トピック

- [AWS SDK for SAP ABAP の前提条件](#)
- [AWS SDK for SAP ABAP - BTP エディションの前提条件](#)

AWS SDK for SAP ABAP の前提条件

AWS SDK for SAP ABAP の前提条件は次のとおりです。

トピック

- [基本リリース](#)
- [カーネルリリース](#)
- [パラメータ](#)
- [メモ](#)
- [アウトバウンド接続](#)
- [HTTPS 接続](#)
- [Amazon EC2 インスタンスメタデータへのアクセス](#)

基本リリース

SDK for SAP ABAP は SAP NetWeaver 7.4 以降と互換性があります。SDK for SAP ABAP は SAP アプリケーションテーブルには一切影響しません。SAP Enterprise Resource Planning や SAP Landscape Transformation Replication Server などのアプリケーションにはまったく依存しません。

SAP_BASIS 740 でサポートされる最小 SP レベルは SP 0008 です。詳細については、「[SAP Note 1856171 - CL_HTTP_ENTITY 内の同じ名前のフォームフィールドのサポート](#)」を参照してください (SAP ポータルへのアクセスが必要)。次の画像に示すように、ビジネス要件に基づいて、より高い SP レベルを選択できます。

Installed Software Component Versions

Installed Product Versions



Component	Release	SP-Level	Support Package	Short Description of Component
SAP_BASIS	740	0026	SAPKB74026	SAP Basis Component
SAP_ABA	740	0026	SAPKA74026	Cross-Application Component
SAP_GWFND	740	0027	SAPK-74027INSAPGWFND	SAP Gateway Foundation
SAP_UI	754	0008	SAPK-75408INSAPUI	User Interface Technology
DI_BASIS	740	0008	SAPK-74026INDIBASIS	Basic Diagnostics

SAP_BASIS 750 およびそれ以降のリリースには SP レベルの最小要件はありません。

カーネルリリース

SDK for SAP ABAP と HTTP 接続にインターネットコミュニケーションマネージャ (ICM) を使用するツールは、暗号化、HTTP、XML、および JSON 機能を SAP カーネルに依存しています。お使いの SAP NetWeaver プラットフォームと互換性のある最新のカーネルリリースを使用することをお勧めします。最小要件はカーネルリリース 741 です。詳細については、「[SAP Note 2083594 - SAP カーネルバージョンと SAP カーネルパッチレベル](#)」を参照してください (SAP ポータルへのアクセスが必要)。

カーネルリリース 741 または 742 を使用している場合は、以下のパッチレベルが必要です。

- 741 patchno 212
- 742 patchno 111

パラメータ

ご使用の SAP システムは、以下の SAP Notes (SAP ポータルへのアクセスが必要) で説明されているように、Server Name Indication (SNI) をサポートしている必要があります。

- [SAP Note 2124480 - ICM/Web デイスパッチャー: クライアントとしての TLS 拡張 Server Name Indication \(SNI\)](#)
- [SAP Note 2582368 - saphttp、sapkprotp、sldreg による TLS 拡張 SNI のクライアント側送信用の SapSSL アップデート](#)

DEFAULT.PFL ファイル内で次のパラメータを設定します。

```
icm/HTTPS/client_sni_enabled = TRUE
```

メモ

次の SAP Note をシステムに適用してください。

- <https://launchpad.support.sap.com/#/notes/0001856171>
- <https://launchpad.support.sap.com/#/notes/0002619546>

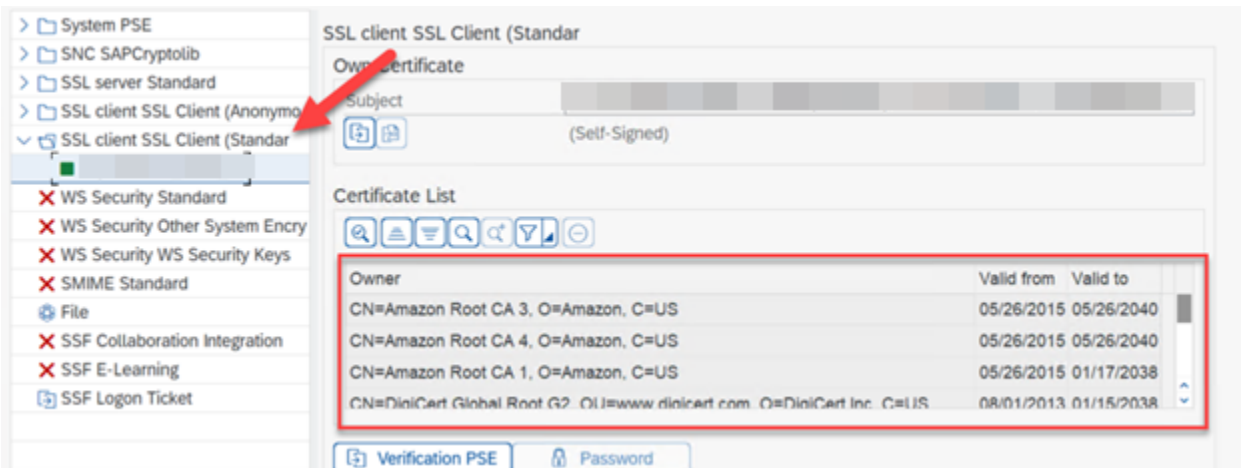
アウトバウンド接続

SDK for SAP ABAP は HTTPS クライアントです。SAP システムは HTTPS メッセージをアウトバウンドに送信します。インバウンド接続は必要ありません。

HTTPS 接続

すべての AWS API コールは、暗号化された HTTPS チャネルで行われます。SAP システムは、アウトバウンド HTTPS 接続を確立するために AWS 証明書を信頼するように設定する必要があります。

1. <https://www.amazontrust.com/repository/> に移動します。
2. [ルート CA] で、PEM リンクを使用してすべての証明書をダウンロードします。
3. 次の図に示すように、これらの証明書を各 SAP システム上の SSL Client (Standard) PSE の STRUST にインポートします。



Amazon EC2 インスタンスメタデータへのアクセス

ABAP システムは、Amazon EC2 インスタンスメタデータを有効にするため、ローカルホスト (<http://169.254.169.254>) への暗号化されていない HTTP 接続を行います。HTTP チャンネルは、ローカルサーバーから AWS 認証情報を取得するためにのみ使用されます。HTTP トラフィックはホスト内にとどまります。

メタデータを使用すると、の SAP システムは AWS、SAP Secure Store にシークレットキーを保存せずに、安全に自身を認証できます。この機能は、Amazon EC2 でホストされる SAP システムにのみ適用されます。

以下のパラメータを使用して DEFAULT.PFL ファイルを設定し、SAP システムが暗号化されていないアウトバウンド HTTP 接続を確立できるようにします。

```
icm/server_port_<xx> = PROT=HTTP,PORT=8000,TIMEOUT=60,PROCTIMEOUT=600
```

次のパラメータを使用して、インバウンド接続を有効にせずにアウトバウンド HTTP 接続を有効にします。

```
icm/server_port_<xx> = PROT=HTTP,PORT=0,TIMEOUT=60,PROCTIMEOUT=600
```

以下の手順で、SAP システムがアウトバウンド HTTP 接続用に設定されていることを確認します。

1. SMICM トランザクションを実行します。
2. [アクティブサービス] に移動します。
3. 次の図に示すように、[アクティブ] 列の [HTTP] 行に緑色のチェックマークが表示されていることを確認します。

No.	Protocol	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv E:
1	HTTPS	50001		60	600	
2	HTTP	0		60	600	✓

AWS SDK for SAP ABAP - BTP エディションの前提条件

AWS SDK for SAP ABAP - BTP エディションの唯一の前提条件は次のとおりです。

トピック

- [SAP Landscape Portal – BTP エディション](#)
- [SAP 認証情報ストア – BTP エディション](#)

SAP Landscape Portal – BTP エディション

この前提条件は、AWS SDK for SAP ABAP - BTP エディションにのみ適用されます。

SAP Landscape Portal は、SAP BTP 環境にアドオンをインストールするためにサポートされている唯一のメカニズムです。このサービスを使用するようにサブスクライブしていることを確認します。詳細については、[「Landscape Portal」](#)を参照してください。

SAP 認証情報ストア – BTP エディション

この前提条件は、AWS SDK for SAP ABAP - BTP エディションにのみ適用されます。

デベロッパープレビューでは、AWS SDK for SAP ABAP - BTP エディションを認証するためにサポートされている唯一のメカニズムはシークレットアクセスキー認証です。SDK は Credential Store から認証情報を読み取り、シークレットアクセスキーを安全に保存します。

以下の前提条件を満たす必要があります。

- Credential Store へのサブスクリプション。
- BTP サブアカウントにエンタイトルメントとして割り当てられた Credential Store。詳細については、[「初期設定」](#)を参照してください。
- Credential Store の標準プランを持つサービスインスタンス。詳細については、[「サービスインスタンスの作成」](#)を参照してください。

詳細については、「[SAP 認証情報ストアの使用](#)」を参照してください。

SAP Credential Store サービスは、ABAP BTP システム外の SAP BTP で実行されます。詳細については、「[SAP 認証情報ストア](#)」を参照してください。

のインストール AWS SDK for SAP ABAP

トピック

- [SDK for SAP ABAP をダウンロードする](#)
- [SDK for SAP ABAP ファイルを検証する \(オプション\)](#)
- [AWS SDK トランスポート](#)

SDK for SAP ABAP をダウンロードする

SDK を <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip> からダウンロードします。

```
curl "https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip" -o "abapsdk-LATEST.zip"
```

ダウンロードが完了したら、ダウンロードしたファイルを /tmp/awssdk などのディレクトリに解凍することをお勧めします。

SDK for SAP ABAP ファイルを検証する (オプション)

SDK ファイルの署名を検証するこのオプションの手順は、SDK が改ざんされていないことを確認するのに役立ちます。SDK ファイルを検証するには、次のステップを行います。

1. 次のコマンドを使って、SDK SIGNATURE ファイルをダウンロードします。

```
curl "https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.sig" -o "abapsdk-LATEST.sig"
```

2. 次のパブリックキーをコピーし、abapsdk-signing-key.pem という名前のファイルに保存します。

```
-----BEGIN PUBLIC KEY-----  
MIICIjANBgkqhkiG9w0BAQEFAA0CAg8AMIICGgKCAgEAmS3oN3wKBh4HJ0Ga0tye  
15RR5909nuw0Jx0vEDCT709wUrxS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
```

```
051sTdEJJpfLEWJJZZNK3v9fGwKyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1LZFbGrn6A5adbwzF0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrXUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWlgbjNmX0hB0Eprg8/nVRHspf3zuisCJ21MPkz0
cH0R31MNsMLzm+d/gVklT31R/JwAcFCkXTWvR8/V0WNGZZXdVUbefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLjPz8dH
Zg25o79U9EJln+lpqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcmzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

- ダウンロードした SDK ZIP ファイルを次のコマンドで検証します。このコマンドには、多くの Linux ディストリビューションの一部である `openssl` が必要です。

```
openssl dgst -sha256 -verify abapsdk-signing-key.pem -keyform PEM -signature
abapsdk-LATEST.sig abapsdk-LATEST.zip
```

- 前述のコマンドの出力が `Verified OK` であることを確認します。
- 出力が `Verification Failure` の場合は、上のステップを繰り返します。失敗した出力が引き続き表示される場合は、SDK をインストールせずに、サポートにお問い合わせください。

AWS SDK トランスポート

トピック

- [内容](#)
- [インポート中](#)
- [名前空間](#)

内容

SDK for SAP ABAP のインストールは ABAP トランスポートを通じて完了します。これらのトランスポートは開発環境またはサンドボックス環境にインポートする必要があります。

SDK for SAP ABAP の各リリースは、以前のリリースと完全に置き換わります。インクリメンタルトランスポートを適用する必要はありません。トランスポートは ZIP ファイルにバンドルされています。ZIP ファイルの構造は次のとおりです。

```
transports/
```

```
transports/core/  
transports/core/Knnnnnnn.AWS  
transports/core/Rnnnnnnn.AWS  
transports/tla1/  
transports/tla1/Knnnnnnn.AWS  
transports/tla1/Rnnnnnnn.AWS  
transports/tla2/  
transports/tla2/Knnnnnnn.AWS  
transports/tla2/Rnnnnnnn.AWS  
.  
.  
.
```

transports フォルダーには core サブフォルダーが含まれています。core サブフォルダーには、コアとなるランタイムトランスポートと、モジュールの 3 文字の略称で名前が付けられた各モジュールのサブフォルダーが含まれます。TLA の完全なモジュールリストについては、「[AWS SDK for SAP ABAP - モジュールリスト](#)」を参照してください。

AWS SDK トランスポートはワークベンチリクエストです。TMS ルートの設定によっては、SDK が以前のシステムにインポートされた後に、品質保証キューや本番稼働環境キューに自動的に転送されない場合があります。各システムのキューに手動で追加する必要があります。

プロジェクトが次のフェーズに進む準備ができたなら、AWS SDK をビジネス機能を持つ独自の Zコードを含む個別のトランスポートと共にインポートできます。SAP Change Request Management (ChaRM) などの変更管理システムを使用している場合は、サードパーティトランスポートの適切な処理について、ChaRM 管理者に相談してください。

インポート中

トピック

- [キーポイント](#)
- [インポートにかかる時間](#)

AWS SDK トランスポートはクライアントに依存しません。コアトランスポートは必須であり、SDK ランタイムコード、 の API AWS Security Token Service、Amazon Simple Storage Service の API が含まれています。残りの SDK モジュールはそれぞれ別のトランスポートで配信されます。システム内の SDK のサイズを小さく保つため、各 SDK モジュールはオプションとなっています。ビジネスロジックに必要な場合は、後で追加のモジュールをインストールできます。

例えば、Amazon S3 と APIs を使用する場合は Amazon Translate、coreトランスポート (コアランタイム、Amazon S3、AWS STS モジュールを含む) とx18トランスポート (モジュールを含む Amazon Translate) をインポートします。

SDK for SAP ABAP の API の完全なリストについては、「[SDK for SAP ABAP - API リファレンスガイド](#)」を参照してください。

AWS SDK トランスポートをインポートする際のキーポイントを次に示します。

- 各トランスポートは、Knnnnnn.AWS と Rnnnnnn.AWS として配信されます。
 - Knnnnnn.AWS は /usr/sap/trans/cofiles にコピーする必要があります。
 - Rnnnnnn.AWS は /usr/sap/trans/data にコピーする必要があります。
- トランスポートをインポートするときは、[インポートトランスポートリクエスト]>[オプション]>[インポートオプション]にある [無効なコンポーネントバージョンを無視] を選択する必要があります。
- 必要なすべてのトランスポートを同時にインポートできます。
- トランスポートを個別にインポートする場合は、最初に core トランスポートをインポートする必要があります。
- すべてのトランスポートのリリースレベルは完全に同一でなければなりません。

インポートにかかる時間

AWS SDK トランスポートのインポートには数分かかる場合があります。STMS が緑 (RC=0) または黄色 (RC=4) のライトを表示していれば、トランスポートは成功です。

- 赤色のライト (RC=8) は、インポートに構文エラーがあったことを示します。
 - [リクエスト] → [表示] → [ログ] を選択し、インポートエラーを調べます。
 - インポート中にインターフェイス IF_SYSTEM_UUID_RFC4122_STATIC がいないためにエラーが発生した場合は、SAP Note 2619546 がシステムに適用されていることを確認してください。詳細については、「[Notes](#)」を参照してください。
 - エラーの原因が不明な場合は、にお問い合わせください サポート。
- 赤い稲妻 (RC=12) は、トランスポートファイルが /usr/sap/trans に正しく読み込まれていないか、必要な権限がないことを示します。

キーポイント

AWS SDK トランスポートをインポートする際のキーポイントを次に示します。

- 各トランスポートは、Knnnnnn.AWS と Rnnnnnn.AWS として配信されます。
 - Knnnnnn.AWS は /usr/sap/trans/cofiles にコピーする必要があります。
 - Rnnnnnn.AWS は /usr/sap/trans/data にコピーする必要があります。
- トランスポートをインポートするときは、[インポートトランスポートリクエスト]>[オプション]>[インポートオプション]にある [無効なコンポーネントバージョンを無視] を選択する必要があります。
- 必要なすべてのトランスポートを同時にインポートできます。
- トランスポートを個別にインポートする場合は、最初に core トランスポートをインポートする必要があります。
- すべてのトランスポートのリリースレベルは完全に同一でなければなりません。

インポートにかかる時間

AWS SDK トランスポートのインポートには数分かかる場合があります。STMS が緑 (RC=0) または黄色 (RC=4) のライトを表示していれば、トランスポートは成功です。

- 赤色のライト (RC=8) は、インポートに構文エラーがあったことを示します。
 - [リクエスト] → [表示] → [ログ] を選択し、インポートエラーを調べます。
 - インポート中にインターフェイス IF_SYSTEM_UUID RFC4122_STATIC がいないためにエラーが発生した場合は、SAP Note 2619546 がシステムに適用されていることを確認してください。詳細については、「[Notes](#)」を参照してください。
 - エラーの原因が不明な場合は、[お問い合わせ](#)ください サポート。
- 赤い稲妻 (RC=12) は、トランスポートファイルが /usr/sap/trans に正しく読み込まれていないか、必要な権限がないことを示します。

名前空間

SDK for SAP ABAP は /AWS1/ 名前空間を使用し、以下の例外を除いてシステム内の SAP オブジェクトやその他のオブジェクトは変更しません。

- AWS auth オブジェクトは、認証オブジェクトクラスにあります。認証オブジェクトクラスは 4 文字までに制限されており、名前空間をサポートしていません。SDK for SAP ABAP が使用する認

証オブジェクトクラスは YAW1 です。トランザクション YAW1に認証オブジェクトクラスが既にある場合はSU21、インストール サポート 前に [お問い合わせ](#) してください。

AWS SDK for SAP ABAP - BTP エディションのインストール

BTP エディションはデベロッパープレビューであり、プレビューに参加してインストールできます。SDK をインストールするには、[AWS SDK for SAP ABAP - BTP Edition 開発者プレビュー](#)の参加フォームに入力します。

SDK for SAP ABAP - BTP エディションをインストールする前に、必要な前提条件を満たしていることを確認してください。詳細については、「[SAP Landscape Portal](#)」と「[SAP Credential Store](#)」を参照してください。

トピック

- [SDK for SAP ABAP - BTP エディションをインストールする](#)
- [モジュール](#)
- [SDK for SAP ABAP - BTP エディションへのパッチ適用](#)

SDK for SAP ABAP - BTP エディションをインストールする

1. SAP Landscape Portal インスタンスに移動し、Deploy Product fiori アプリケーションを起動します。
2. 製品 で、パートナー製品 **/AWS1/SDK_OMNI** で を選択します。

デベロッパープレビューで承諾された**/AWS1/SDK_OMNI**後に が表示され サポート ない場合は、 [お問い合わせ](#) してください。

3. ターゲットバージョンで、システムにインストールする SDK for SAP ABAP - BTP エディションのバージョンを選択します。
4. 使用可能なシステムで、SDK をインストールするすべての SIDs のチェックボックスをオンにします。
5. デプロイを選択し、スケジュールの詳細を入力し、スケジュールを選択します。製品バージョンのデプロイステータスの進行状況をモニタリングできます。

インストールには 30~45 分かかる場合があります、これにはシステムのダウンタイムが含まれます。詳細については、「[製品のデプロイ](#)」を参照してください。

モジュール

以下のモジュールは、AWS SDK for SAP ABAP - BTP エディションの開発者プレビューに含まれています。

- [Amazon API Gateway \[agw\]](#)
- [Amazon Athena \[ath\]](#)
- [Amazon Bedrock ランタイム \[bdr\]](#)
- [Amazon Comprehend \[cpd\]](#)
- [Amazon EventBridge \[evb\]](#)
- [Amazon Forecast \[fcs\]](#)
- [Amazon Kinesis \[kns\]](#)
- [Amazon Data Firehose \[frh\]](#)
- [Amazon SageMaker AI \[sgm\]](#)
- [Amazon Simple Notification Service \[sns\]](#)
- [Amazon Simple Queue Service \[sqs\]](#)
- [Amazon Simple Storage Service \[s3\]](#)
- [AWS Systems Manager \[ssm\]](#)
- [Amazon Textract \[tex\]](#)
- [Amazon Transcribe \[tnb\]](#)
- [Amazon Translate \[x18\]](#)
- [AWS CloudTrail \[tr1\]](#)
- [AWS IoT \[iot\]](#)
- [AWS KMS \[kms\]](#)
- [AWS Lambda \[lmd\]](#)
- [AWS Secrets Manager \[smr\]](#)
- [AWS Security Token Service \[sts\]](#)
- [AWS Transfer Family \[trn\]](#)
- [IAM Roles Anywhere \[r1a\]](#)
- [Amazon Redshift Data API \[rsd\]](#)

SDK for SAP ABAP - BTP エディションへのパッチ適用

SDK for SAP ABAP - BTP エディションのパッチ適用プロセスは、インストールプロセスと似ています。古いバージョンが既にインストールされているシステムに SDK をインストールすると、選択した新しいバージョンに SDK にパッチが適用されます。

の設定 AWS SDK for SAP ABAP

を使用する前に AWS SDK for SAP ABAP、SDK オペレーションに必要な技術的および機能的な設定で SDK を設定する必要があります。設定には転送可能なものもあれば、ランタイム設定もあります。設定の多くは、他の SDK の .INI ファイルに定義されている設定と直接似ています。

ランタイム設定を除く SDK 設定は、開発環境で完了する必要があります。構成は、通常の転送ルールおよび変更管理ルールに従って QA および本番稼働環境に転送できます。本番環境では、転送可能な設定は推奨されません。

AWS SDK を設定するアクセス許可がない場合は、[「SAP 認可」](#)を参照してください。

の設定 AWS SDK for SAP ABAP

構成トランザクションを実行するには、SAPGUI コマンドバーに /n/AWS1/IMG と入力します。

AWS SDK for SAP ABAP の設定 - BTP エディション

SDK for SAP ABAP - BTP エディションを設定するには、次の手順に従います。

1. ウェブブラウザで ABAP 環境を開きます。
2. カスタムビジネス設定アプリケーションに移動します。

Export Customizing Transports アプリケーションを使用してカスタマイズリクエストを作成するには、[「Export Customizing Transports App - Create Request」](#)の「Working」を参照してください。

カスタムビジネス設定アプリケーションでは、SDK 設定のタイプに基づいて設定をグループ化できます。設定をグループ化するには、次のステップを使用します。

1. ウェブブラウザで ABAP 環境を開き、カスタムビジネス設定アプリケーションに移動します。
2. 設定 > グループを選択し、ドロップダウンリストから設定グループを選択します。[OK] を選択します。
3. イメージに表示される階層構造で設定を使用できるようになりました。ビューを保存するには、[「ビュー \(バリエーション管理\) - コンポーネント」](#)を参照してください。

Custom Business Configurations (4)

Name	Description	
Application Configuration		
SDK Profile	Maintain AWS SDK Profile	>
Logical Resource Resolver	Maintain Logical Resource Resolution	>
Global Settings		
Technical Settings	Maintain Technical Settings	>
Configure Scenarios	Configure Scenarios	>

このセクションでは、次のトピックについて説明します。

トピック

- [\[Global settings\] \(グローバル設定\)](#)
- [アプリケーションの設定](#)
- [ランタイム設定](#)
- [高度な接続シナリオ](#)
- [サービスプロバイダーの設定](#)
- [の更新、トレース、テレメトリのトピック AWS SDK for SAP ABAP](#)

[Global settings] (グローバル設定)

AWS SDK for SAP ABAP には /n/AWS1/IMG IMG トランスアセンテーションを使用し、AWS SDK for SAP ABAP - BTP エディションにはカスタムビジネス設定アプリケーションを使用してグローバル設定を設定します。このトピックでは、IMG とカスタムビジネス設定を同じ意味で使用します。

このセクションでは、次のトピックについて説明します。

トピック

- [技術設定](#)
- [シナリオの設定](#)

技術設定

/AWS1/IMG トランザクションのグローバル設定は SDK 全体の動作に影響します。これらの設定は通常、Basis 管理者が設定します。これらの値は次の推奨設定に設定できます。

- [新規エントリ] を選択します。
 - S3 リージョン化: s3.amazonaws.com を使用して us-east-1 バケットにアクセスします。
 - STS のリージョン化: グローバルエンドポイントを使用して STS にアクセスします。
 - EC2 メタデータを無効にする: このフィールドは空白のままにしてください。このフィールドは BTP エディションでは読み取り専用で、デフォルトで「はい」に設定されています。
 - メタデータエンドポイントモード: IPv4 メタデータエンドポイントを使用します。このフィールドは BTP エディションでは読み取り専用で、自動更新されます。
 - メタデータエンドポイント URL: このフィールドは空白のままにしてください。このフィールドは BTP エディションでは読み取り専用です。
- [保存] を選択します。

シナリオの設定

シナリオにより、AWS SDK はマルチリージョンの災害テストまたはディザスタリカバリテストのシナリオ中に設定をより効率的に切り替えることができます。この機能は必要ない場合があり、その場合は、次のデフォルトシナリオを設定するだけで済みます。

- [新規エントリ] を選択します。
 - シナリオ ID: DEFAULT
 - シナリオの説明: デフォルトシナリオ
- [保存] を選択します。

マルチリージョンのディザスタリカバリのセットアップや、設定をすばやく変更する必要があるその他の特殊なケースでは、複数のシナリオを構成できます。

- DEFAULT - 標準オペレーション。
- DR - 災害時にシステム全体を別のリージョンに移動する必要がある場合の特別な設定。
- DR_TEST - 災害をシミュレートするための特別な設定 (本番稼働環境の一時的なクローン作成など)。

アプリケーションの設定

SDK for SAP ABAP の設定は、他の ABAP ベースのアプリケーションの設定と同様です。さまざまなシナリオの設定をグループ化するために、さまざまなプロファイルにまとめられています。ABAP SDK プロファイルは、特定のアプリケーションシナリオに必要な設定を定義します。たとえば、トランザクション ZVA01、ZVA02ZVA03が、Amazon S3 AWS のサービス、などの請求書関連のトランザクションが拡張され AWS Lambda、で実行される場合 Amazon SageMaker AI、という SDK プロファイルZINVOICEを作成できます。Amazon S3 このプロファイルでは、請求書関連機能の技術設定、SAP 権限、IAM ロールマッピングをグループ化できます。

AWS SDK for SAP ABAP の/n/AWS1/IMG変換と AWS SDK for SAP ABAP のカスタムビジネス設定アプリケーション - BTP エディションを使用して、グローバル設定を設定します。このトピックでは、IMG とカスタムビジネス設定を同じ意味で使用します。

トピック

- [SDK プロファイル](#)
- [論理リソースリゾルバー](#)
- [例](#)

SDK プロファイル

ABAP SDK プロファイルは、SID とクライアントごとに以下を定義します。

Note


SAP BTP、ABAP 環境では、クライアントは常に 100 です。

- すべての API コールのデフォルト AWS リージョン。例えば、SAP システムが us-east-1リージョンで実行されている場合、他の AWS リソースも同じリージョンにある可能性があります。こ

れはデフォルトのリージョンである必要があります。ABAP コードでデフォルトリージョンを上書きできます。

- 認証方法

- Amazon EC2 で実行されている SAP システムでは、有効期間が短く、自動的に更新される認証情報を活用できるインスタンスロールメタデータを選択することを強くお勧めします。
- SAP システムがオンプレミスまたは他のクラウドで実行されている場合は、SSF ストレージから取得した認証情報を選択する必要があります。
- SAP BTP で実行されている ABAP システムでは、SAP 認証情報ストアから認証情報を選択する必要があります。詳細については、[「認証に SAP 認証情報ストアを使用する」](#)を参照してください。
- クロスアカウントロールの連鎖シナリオでは、ソースプロファイルを選択し、ソースプロファイル ID を指定します。これにより、ロールの引き受けチェーンの自動解決と実行が可能になります。詳細については、[「クロスアカウントアクセスのソースプロファイルの使用」](#)を参照してください。
- 論理 IAM ロールの IAM ロールへのマッピング。
 - このマッピングは優先度の降順でソートされます。
 - PFCG ロールでユーザーに権限が与えられている優先度が最も高い IAM ロールが、そのユーザーに自動的に選択されます。
- カスタムエンドポイントへのサービスのオプションのマッピング。この設定については、[「高度な接続シナリオ」](#)で説明します。

 Note

PFCG ロールは、SAP BTP、ABAP 環境でビジネスロールと呼ばれます。

ABAP プログラムが に接続する場合 AWS のサービス、必要な設定をプルする ABAP SDK プロファイルを指定します。AUTHORIZATION-CHECK は、ユーザに SDK プロファイルへのアクセス権限があることを確認するために実行されます。SAP セキュリティ管理者は、適切なユーザーにアクセス権を付与する PFCG ロールを定義できます。

論理リソースリゾルバー

論理リソースリゾルバーを使用すると、リソース名を標準的な場所に保存できます。リゾルバーは、SDK for SAP ABAP に付属しています。その動作は、FILE トランザクションが論理ファイル名を物理ファイル名にマップする方法と似ています。

論理リソースは、請求書を保持する Amazon S3 バケットなど、AWS リソースの概念を定義します。この論理リソースには、例えば ZINVOICES_OUTBOUND という名前を付け、SAP システムが開発、QA、または本番稼働のいずれであるかに応じて、異なる物理バケット名にマッピングできます。

SDK for SAP ABAP は、システムが本番稼働環境から更新された後でも、QA システムが論理リソースを QA 物理リソースに解決するように設定されています。すべてのシステムのリソースマッピングは開発用 SAP システムで定義され、転送されます。このアプローチは、マッピングがマスターデータとして扱われ、各システムで設定される SAP システムの通常の設定とは異なります。SDK for SAP ABAP が提供する論理リソースリゾルバーの利点は、システム更新後に誤って転送される可能性がほとんどないことです。

例

4 つの Amazon S3 バケットがあり、それぞれ開発用、本番稼働用、QA 用、および回帰テスト用の 2 つ目の QA バケットです。

SDK は、ZINVOICE_OUTBOUND などの論理リソースを物理リソースに解決するときに、SY-SYSID と SY-MANDT をチェックして、どの SID とクライアントで実行しているかを尋ね、正しい物理リソースを自動的に選択します。

本番稼働環境のリソースのマッピングを変更する必要がある場合は、開発システムの IMG 内のマッピングを変更して転送する必要があります。これにより、SAP システムへの AWS リソースの再割り当ては、他のトランスポートと同様に変更管理の対象となります。

Note

SDK の設定はクライアントに依存するため、リソースの再割り当てはカスタマイズリクエストで転送され、トランスポートは各クライアントにインポートする必要があります。

ランタイム設定

このセクションでは、次のトピックについて説明します。

Note

これらの設定は転送できず、各 SAP システムに限定されます。

トピック

- [ログとトレース](#)
- [オプション: 拡張テレメトリー](#)
- [アクティブシナリオ](#)

ログとトレース

トレースをデバッグ目的でアクティブ化できます。技術的な問題を診断する場合以外は、トレースレベルを [トレースなし] に保つことをお勧めします。詳細については、「[セキュアなオペレーション](#)」を参照してください。

これらの設定は、SDK for SAP ABAP - BTP エディションには適用されません。

オプション: 拡張テレメトリー

すべての SDKs、サポート AWS 目的でテレメトリー情報を に送信します。拡張テレメトリーをオプションできます。これは、サポート に連絡して特定の API コールのソースを特定する場合に特に便利です。詳細については、「[トレース](#)」と「[テレメトリー](#)」を参照してください。

これらの設定は、SDK for SAP ABAP - BTP エディションには適用されません。

アクティブシナリオ

このトランザクションで DEFAULT シナリオを有効化してください。この有効化はシステムごとに 1 回だけ必要で、システムがマルチリージョンのディザスタリカバリを受ける場合以外に変更しないでください。マルチリージョンのセットアップでは、この設定を使用して SAP システムをディザスタリカバリ環境またはディザスタリカバリテストシナリオに切り替えることができます。

高度な接続シナリオ

AWS SDK for SAP ABAP は AWS 、エンドポイントへの HTTPS 呼び出しを行うこと AWS のサービスで を消費します。一般に、AWS エンドポイントは、インターネットからアクセス可能で

す。SAP システムは、これらのアウトバウンド接続を確立するためにインターネットに接続できる必要があります。SDK for SAP ABAP では、インターネットから SAP システムへのインバウンド接続は一切必要ありません。

以下のシナリオでは、アウトバウンド接続を確立するためのさまざまな方法を示します。

シナリオ

- [プロキシサーバー経由の接続](#)
- [パケット検査ファイアウォール経由の接続](#)
- [ゲートウェイエンドポイント](#)
- [カスタムインターフェイスのエンドポイント](#)
- [高度なルーティング](#)
- [サービスごとのプロキシサーバーオーバーライド](#)
- [複数のリージョンでエンドポイントにアクセスする](#)

プロキシサーバー経由の接続

プロキシサーバー経由で接続を確立するには、以下の手順に従います。

1. SDK で、トランザクション **SICF** に移動します。
2. [実行] を選択してください。
3. メニューで [クライアント] > [プロキシサーバー] を選択します。
4. [プロキシ設定] を [アクティブ] に設定します。
5. 例外があれば、[次のアドレスにはプロキシなし] フィールドにセミコロンで区切って記載します。
6. [HTTP プロトコル] と [HTTPS プロトコル] フィールドに、プロキシサーバーの接続の詳細を指定します。

SDK はプロキシサーバーを認識しないため、SAP システムのプロキシサーバー設定を使用するための設定は必要ありません。

Note

[Amazon EC2 インスタンスメタデータ認証](#)を使用する場合、SAP システムはプロキシサーバーを使用して `http://169.254.169.254` のローカルインスタンスメタデー

タにアクセスすることはできません。[次のアドレスにはプロキシなし] フィールドに 169.254.169.254 を含める必要があります。

Note

プロキシサーバーの動作は、の高度なルーティングセクションでサービスごとに上書きできます/AWS1/IMG。詳細については、「[the section called “サービスごとのプロキシサーバーオーバーライド”](#)」を参照してください。

パケット検査ファイアウォール経由の接続

アウトバウンド接続用のパケット検査ファイアウォールを設定できます。これらのファイアウォールは SSL トラフィックを復号化し、エンドポイントに渡す前に再暗号化します。この設定では、通常、ファイアウォールが AWS のサービスを使用している SAP システムに自身の証明書を発行する必要があります。ファイアウォールの CA 証明書を STRUST にインストールする必要があります。詳細については、「[HTTPS 接続](#)」を参照してください。

ゲートウェイエンドポイント

一部の AWS のサービスは、インターネットなしで高性能アクセスを VPC に提供するゲートウェイエンドポイントを提供します。これらのエンドポイントは SDK for SAP ABAP に対して透過的であり、設定は一切必要ありません。

詳細については、「[ゲートウェイエンドポイント](#)」を参照してください。

カスタムインターフェイスのエンドポイント

デフォルトのエンドポイント解決をカスタムエンドポイントでオーバーライドする必要がある場合は、インターフェイスエンドポイントを使用してインターネットなしで VPC に高性能アクセスを提供できます。詳細については、「[インターフェイスエンドポイントを設定する](#)」を参照してください。

プライベート DNS を使用しない場合、これらのエンドポイントには独自の DNS アドレスが割り当てられるため、ABAP プログラムは通常のエンドポイント解決ロジックを明示的にオーバーライドする必要があります。詳細については、AWS re:Post 「」を参照してください。[インターフェイス VPC エンドポイントのサービスドメイン名を解決できないのはなぜですか？](#)

次の例では、AWS STS とのインターフェイスエンドポイントが作成されます Amazon Translate。SAP システムはプライベート DNS を使用せず、カスタムエンドポイントを使用してサービスを呼び出します。/AWS1/IMG で定義されている論理リソースは、vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com などの物理インターフェイスのエンドポイントアドレスを表します。これにより、DNS をコード内でハードコーディングする必要がなくなります。

次のコードでは、/AWS1/IMG 内の論理リソースが最初に物理エンドポイント名に変換されます。次に、AWS セッションクラスのファクトリメソッド (IAM ロールを引き受け AWS STS するために使用するメソッド) に提供され、API クラスを変換します。

```
" This example assumes we have defined our logical endpoints in /AWS1/IMG
" as logical resources so that we don't hardcode our endpoints in code.
" The endpoints may be different in Dev, QA and Prod environments.
DATA(lo_config) = /aws1/cl_rt_config=>create( 'DEMO' ).
DATA(lo_resolver) = /aws1/cl_rt_lresource_resolver=>create( lo_config ).

" logical resource STS_ENDPOINT should resolve to the interface endpoint
" for example vpce-0123456789-abcdefg.sts.us-west-2.vpce.amazonaws.com
DATA(lv_sts_endpoint) = lo_resolver->resolve_lresource( 'STS_ENDPOINT' ).

" logical resource XL8_ENDPOINT should resolve to the interface endpoint
" e.g. vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com
DATA(lv_xl8_endpoint) = lo_resolver->resolve_lresource( 'XL8_ENDPOINT' ).

" the session itself uses the sts service to assume a role, so the
" session creation process requires a custom endpoint, specified here
DATA(lo_session) = /aws1/cl_rt_session_aws=>create(
  iv_profile_id = 'DEMO'
  iv_custom_sts_endpoint = |https://{ lv_sts_endpoint }|
).

" now we create an API object, and override the default endpoint with
" the custom endpoint
DATA(lo_xl8) = /aws1/cl_xl8_factory=>create(
  io_session = lo_session
  iv_custom_endpoint = |https://{ lv_xl8_endpoint }| " provide custom endpoint
).
" now calls to lo_xl8 go to custom endpoint...
```

例に示すように、go_xl8 上のメソッド呼び出しはすべてエンドポイント https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com に移

動します。次のセクションに示すように、コードではなく IMG 設定でルーティングカスタムエンドポイントを定義することもできます。

高度なルーティング

前のセクションでは、SDK モジュールのファクトリメソッドの `iv_custom_endpoint` 引数でカスタムエンドポイントを指定する方法を示しました。SDK を使用する ABAP プログラムの数が増えるにつれて、管理が困難になる可能性があります。から SDK プロファイルのカスタムエンドポイント AWS のサービス へのマッピングを設定できます。SID、クライアント、シナリオごとに、サービス 3 文字の省略形 (TLA) をエンドポイント URL にマッピングできます。

TLA	カスタムエンドポイント URL
BDR	<code>https://vpce-23456789abcdef012-3c4d5e6f.bedrock-runtime.us-east-1.vpce.amazonaws.com</code>
LMD	<code>https://vpce-123456789abcdef01-2b3c4d5e.lambda.us-east-1.vpce.amazonaws.com</code>
S3	<code>https://vpce-0123456789abcdef0-1a2b3c4d.s3.us-east-1.vpce.amazonaws.com</code>

この設定では、ファクトリメソッド呼び出し `iv_custom_endpoint` で を指定する必要はありません。カスタムエンドポイントは、設定テーブルから自動的に選択されます。この設定は SDK プロファイルに固有であるため、ニーズに合わせて異なるルーティングで複数のプロファイルを作成できます。他の SDK プロファイル設定と同様に、ルーティングは SID とクライアント固有であるため、異なるシステムに個別のルーティングを定義できます。

サービスごとのプロキシサーバーオーバーライド

デフォルトでは、SDK はトランザクションで設定されたプロキシサーバー設定を使用します SICF (「」を参照 [the section called “プロキシサーバー経由の接続”](#))。のプロキシ設定は、SAP システムからのすべてのアウトバウンド HTTP 接続にグローバル SICF に適用されます。環境によっては、プロキシサーバー AWS のサービス を使用すると直接接続する をより細かく制御する必要がある場合があります。

の高度なルーティングセクションでは/AWS1/IMG、サービスごとにプロキシサーバーの使用を設定できます。この設定は、グローバルプロキシのアクティベーションまたはフィルター設定に関係なくSICF、SDK が で定義されたプロキシサーバーを介してそのサービスのリクエストをルーティングするかどうかを制御します。

次の値を使用できます。

- デフォルト – で設定されているプロキシ動作を使用しますSICF。プロキシがアクティブで、エンドポイントがフィルターによって除外されていない場合、プロキシが使用されます。これがデフォルトの動作です。
- Always – グローバルアクティベーションまたはフィルター設定に関係なくSICF、常に で定義されたプロキシサーバーを介してこのサービスのリクエストをルーティングします。
- Never – グローバルアクティベーションまたはフィルター設定に関係なく、プロキシサーバーを介してこのサービスのリクエストをルーティングしないでください。これは、VPC エンドポイントを使用する場合など、プロキシなしでサービスエンドポイントに直接到達できる場合に使用します。

例えば、プロキシを (VPC ゲートウェイエンドポイントを介してアクセスされるため) 使用しない AWS STS ように Amazon S3 と を設定できますが、Amazon SNS はインターネット経由でのみアクセスできるため、常にプロキシを使用します。

TLA	プロキシサーバーを使用する
EC2	デフォルト
S3	なし
SNS	常に
STS	なし

この設定は、SDK プロファイル、SID、クライアント、シナリオごとに定義されます。異なる環境やユースケースに合わせて、異なるプロキシルーティングで複数のプロファイルを作成できます。

Note

プロキシサーバーのホストとポートは常に、トランザクションのプロキシ設定の HTTPS プロトコルタブで定義されます。SICF。サービスごとのオーバーライドは、どのプロキシサーバーを使用するかではなく、プロキシを使用するかどうかのみを制御します。

複数のリージョンでエンドポイントにアクセスする

AWS エンドポイントは、SDK プロファイルで AWS リージョン 定義されているデフォルトから自動的に決定されます。プログラムによってリージョンを指定して、デフォルトのリージョンをオーバーライドすることもできます。これはファクトリー CREATE() メソッドでオーバーライドすることも、後で SDK の設定オブジェクトでオーバーライドすることもできます。詳細については、「[プログラムによる設定](#)」を参照してください。

次の例では、ファクトリー CREATE() メソッドを使用してリージョンを設定し、us-east-1 リージョンと us-west-2 リージョンの両方の Amazon SQS キューを一覧表示します。

```
REPORT zdemo_sqs_queue_list.
parameters: profile type /AWS1/RT_PROFILE_ID OBLIGATORY.

START-OF-SELECTION.
DATA(go_session) = /aws1/cl_rt_session_aws=>create( profile ).
data(lt_region) = VALUE stringtab(
  ( |us-east-1| )
  ( |us-west-2| )
).

LOOP AT lt_region INTO DATA(lv_region).
  DATA(go_sqs) = /aws1/cl_sqs_factory=>create(
    io_session = go_session
    iv_region = conv /AWS1/RT_REGION_ID( lv_region )
  ).
  WRITE: / lv_region COLOR COL_HEADING.
  LOOP AT go_sqs->listqueues( )->get_queueurls( ) INTO DATA(lo_url).
    WRITE: / lo_url->get_value( ).
  ENDLLOOP.
ENDLOOP.
```

サービスプロバイダーの設定

基本管理者は、クライアント からシステム全体で SDK の特定の機能を制御する必要がある場合があります。これは、顧客に代わって自分の AWS アカウントでシステムを運用するホスティングおよびサービスプロバイダーの一般的なシナリオです。AWS SDK for SAP ABAP は、サービスプロバイダー設定をサポートしています。これらの設定はクライアント で設定され、すべてのクライアントで SDK に影響します。サービスプロバイダー設定は、SDK for SAP ABAP - BTP エディションではサポートされていません。

サービスプロバイダーの設定はトランザクション で設定され、AWS1/IMG、クライアント で設定する必要があります。他のクライアントのサービスプロバイダー設定は無視されます。クライアントの設定はすべてのクライアントで有効になり、競合が発生した場合は他のIMG設定よりも優先されます。

次の手順を使用して、クライアント でサービスプロバイダーの設定を行います。

1. トランザクション のサービスプロバイダー設定ブランチを展開します/AWS1/IMG。
2. サービスプロバイダーガードレールを選択する
3. 新しいエントリを選択し、ビジネス要件に基づいて設定を調整します。
 - EC2 メタデータを無効にする – SDK プロファイルが EC2 インスタンスメタデータを使用して認証するように設定されている場合でも、SDK がすべてのクライアントの EC2 インスタンスメタデータにアクセスできないようにします。SDK は、ABAP プログラムが SDK を使用してインスタンスメタデータにアクセスしようとすると、例外を発生させます。
4. [保存] を選択します。

の更新、トレース、テレメトリのトピック AWS SDK for SAP ABAP

このセクションでは、次のトピックについて説明します。

トピック

- [SAP システムの更新](#)
- [トレース](#)
- [Telemetry](#)

SAP システムの更新

システム更新後、Basis 管理者にとっての主な課題は、別々のシステムが互いのリソースにアクセスしないようにすることです。例えば、QA SAP システムが本番稼働ランドスケープの S3 バケットなどのリソースにアクセスしないようにしたい場合があります。

SDK for SAP ABAP では、この課題に対して論理リソースの安全性を配慮したアプローチをとっています。ビジネスアナリストは次のステップを実行できます。

1. ZINVOICE_OUTBOUND などの論理リソースを定義します。
2. 開発システム内のすべてのシステムとクライアントをマッピングします。
3. すべてのシステムの構成を本番稼働ランドスケープまで転送します。

更新後の基本ステップ

1. 認証を確認する

- システムがシークレットアクセスキー認証を使用している場合、SSF で暗号化された認証情報はマスターデータに保存されるため無効になります。認証情報を再入力する必要があります。そのためには、<https://console.aws.amazon.com/iam/> で新しいシークレットアクセスキーを再生成する必要がある場合があります。
- システムが EC2 インスタンスのメタデータで認証されている場合は、手順は必要ありません。

トレース設定を確認する

- /AWS1/IMG で、トレース設定が希望どおりであることを確認します。これらの設定は転送できません。

トレース

トレース出力は IMG ランタイム設定で制御されます。

使用できるトレースレベルは以下のとおりです。

- トレースなし
- API コールをトレース
- API コールとペイロードをトレース

このオプションには、暗号化されていないペイロード情報が含まれます。

- API コール、ペイロード、内部 XML 変換をトレース

このオプションには、暗号化されていないペイロード情報が含まれます。

API トレースが有効になっている場合、トレースは `aws1_trace-YYYY-MM-DD.log` ファイルの `DIR_WORK` に書き込まれます。

ペイロードトレースをさらに有効にすると、呼び出しとペイロードコンポーネントごとに `aws1_payload_*` というタイトルの付いた追加ファイルが作成されます。ペイロードトレースの長さは、個々のペイロードトレースが失敗するたびに長さ制限を適用して制限できます。

ペイロードトレースは主に、シリアル化エラー サポート が発生した場合に に提供される情報を収集することを目的としています。SDK エラーを診断する場合以外は、[トレースなし] を選択することをお勧めします。

Note

ペイロードトレースには、暗号化されていないビジネス情報が含まれている場合があります。これらのトレースは、トラブルシューティングに役立つ AWS サポートからのリクエストに対してのみオンにすることをお勧めします。これらのトレースは解決後にオフにできます。トレースは自動的に削除されないため、不要になったらシステム管理者が削除する必要があります。

これらの設定は、SDK for SAP ABAP - BTP エディションには適用されません。

Telemetry

SDKs テレメトリ情報を に送信します サポート。SDK for SAP ABAP は以下の情報を収集します。

- OS リリースとパッチレベル
- SAP_BASIS リリースとパッチレベル
- SAP カーネルリリースとパッチレベル

次の情報を サポートに送信するようにオプトインできます。

- SAP SID とインスタンス名 (host_sid_nn)
- SAP クライアント (SY-MANDT)
- トランザクションコード (SY-TCODE) とレポート (SY-REPID)

追加情報により サポート、 はユーザーをより適切に支援できます。サポートは特定の API コールが行われた理由を検出し、SAP システムで関連するトランザクションをさらに見つけることができます。

テレメトリは、SDK for SAP ABAP - BTP エディションの SDK および API バージョンに制限されています。

の使用 AWS SDK for SAP ABAP

SDK for SAP ABAP には 2 つの主要なコンポーネントがあります。

- SDK ランタイム (パッケージ /AWS1/RT) — セキュリティ、認証、トレース、設定、データ変換、およびその他のクロス API 機能を支えるオブジェクトのセットです。Amazon S3、AWS STS、IAM Roles Anywhere、Secrets Manager の API モジュールは必須です。
- API (パッケージ /AWS1/API とそのサブパッケージ) — 各 API のサブパッケージ。各 API のオブジェクトは互いに完全に独立しているため、ある API を変更したことで別の API が壊れることはありません。AWS SDK for SAP ABAP API の完全なリストについては、「[AWS SDK for SAP ABAP - API リファレンスガイド](#)」を参照してください。

このセクションでは、次のトピックについて説明します。

トピック

- [ABAP でのデータの表現](#)
- [Amazon S3 プログラムの例](#)
- [SDK for SAP ABAP の概念](#)
- [AWS SDK for SAP ABAP features](#)
- [SDK による製品の構築](#)
- [HTTP リクエストをカスタマイズする AWS](#)
- [制限事項](#)

ABAP でのデータの表現

このセクションでは、次のトピックについて説明します。

トピック

- [データ型](#)
- [AWS データ型](#)

データ型

AWS のサービスには、ABAP データ型にマッピングする必要がある標準のデータ型のセットがあります。詳細については、以下の表をご参照ください。

AWS データ型	ABAP データ型	コメント
boolean	C	単一文字 "X" および " "
String	STRING	
バイト	INT2	INT2 の範囲は 0 ~ 255 よりも大きくなります。ほとんどの AWS のサービスはオーバーフローを切り捨てますが、この動作は正式に定義されていません。
ショート	INT2	
整数	INT4	
Long	DEC19	INT8 は ABAP 750 まででは使用できません。DEC19 は、サポートされているすべての ABAP プラットフォームでの互換性と一貫性を保つために使用されています。
blob	XSTRING	バイナリデータを表します
浮動小数点数	STRING	ABAP は DECFLOATs をサポートしていますが、NaN、Infinity、-Infinity などの値を表すことはできません。AWS SDK はこれらを内部的に STRINGs として表し、実行時に DECFLOAT16 に変換します。NaN、無限大、ま
倍精度	STRING	

AWS データ型	ABAP データ型	コメント
		たは正の無限大が表現されている場合、開発者は特別な例外またはマッピングのセットに対応してこれら进行处理できます。
bigInteger	STRING	これらの値は ABAP では表現できない無限長の数値を表し、bigInteger の代わりに文字列が使用されます。
bigDecimal	STRING	
Timestamp	TZNTSTMP5	TZNTSTMP5 により、ネイティブ ABAP タイムスタンプ関数による処理が可能になります。

AWS のサービスは、次の集計データ型も返します。

AWS データ型	ABAP データ型	コメント
構造	Class	
Union	Class	Union は構造体と同じですが、Union は複数のフィールドセットを持たない点が異なります。その他のフィールドはすべて [値なし] に設定されます。
配列	STANDARD TABLE	
ハッシュ	HASHED TABLE	ハッシュテーブルには、KEY (文字列) と VALUE (クラス) の 2 つの列しかありません。

AWS データ型

ABAP AWS のサービスで をサポートするために、以下のアプローチが統合されています。

- 特定の AWS データ型を ABAP で表現することはできません。例えば、ABAP の float データ型は、NaN、Infinity、または -Infinity の値をサポートしていません。そのため、float データ型は STRING として表現され、実行時に DECFLOAT16 に変換されます。
- AWS データはワイヤ上で JSON または XML として表され、値はオプションです。たとえば、JSON AWS のサービスで から返される次の例を参照してください。

```
Fullname: {
  Firstname: "Ana",
  Middlename: "Carolina",
  Lastname: "Silva"
}
```

Ana にミドルネームがない場合、サービスは次の出力を返します。

```
Fullname: {
  Firstname: "Ana",
  Lastname: "Silva"
}
```

ABAP は長さ 0 の文字列と値のない文字列を区別しません。他の言語では、文字列に NULL 値を割り当てたり、文字列をコンストラクト (Java の `Optional<>` ラッパーなど) でラップしたりする場合があります。これらは ABAP ではサポートされていません。そのため、SDK for SAP ABAP には getter メソッドのバリエーションが用意されており、値の区別が容易になっています。

Amazon S3 プログラムの例

このセクションでは、`ListObjectsV2` を呼び出して Amazon S3 バケットの内容を一覧表示する簡単なサンプルプログラムについて説明します。

トピック

- [前提条件](#)
- [コード](#)

• [コードセクション](#)

前提条件

このサンプルプログラムの実行には、次の前提条件を満たす必要があります。

- Amazon S3 バケットがある。このチュートリアルでは、バケットの名前は demo-invoices.customer.com です。
- トランザクション /AWS1/IMG に
 - DEMO_S3 という定義済みの SDK プロファイルがある。
 - SDK プロファイルでは、論理 IAM ロール TESTUSER は IAM ロールにマッピングする必要があります。例えば、arn:aws:iam::111122223333:role/SapDemoFinance は、Amazon S3 バケットの内容を一覧表示する s3:ListBucket アクセス権を付与します。
 - SAP システムの SID とクライアントを使用して Amazon S3 バケットにマッピングされる DEMO_BUCKET という名前の論理リソースがある。
- ユーザーに以下のことを行う PFCG ロールがある。
 - 認証オブジェクト /AWS1/SESS を介して DEMO_S3 SDK プロファイルへのアクセスをユーザーに許可する。
 - 認証オブジェクト /AWS1/LROL を介して論理 IAM ロールの TESTUSER アクセスをユーザーに許可する。
- SAP システムは、SDK プロファイルで定義された メソッド AWS を使用して、 に対して自身を認証できます。
- Amazon EC2 インスタンスプロファイルが、SDK プロファイルにマッピングされた IAM ロール arn:aws:iam::111122223333:role/SapDemoFinance の sts:assumeRole に対する権限を SAP システムに付与する。

コード

次のコードブロックは、コードがどのようになるかを示しています。

```
REPORT  zdemo_s3_listbuckets.

START-OF-SELECTION.
  PARAMETERS pv_lres TYPE  /aws1/rt_resource_logical
```

```

                DEFAULT 'DEMO_BUCKET' OBLIGATORY.

DATA(go_session) = /aws1/cl_rt_session_aws=>create( 'DEMO_S3' ).
DATA(gv_bucket)  = go_session->resolve_lresource( pv_lres ).

DATA(go_s3)      = /aws1/cl_s3_factory=>create( go_session ).

TRY.
    DATA(lo_output) = go_s3->listobjectsv2(
        iv_bucket = CONV string( gv_bucket )
        iv_maxkeys = 100
    ).
    LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).
        DATA lv_mdate TYPE datum.
        CONVERT TIME STAMP lo_object->get_lastmodified( )
            TIME ZONE 'UTC'
            INTO DATE lv_mdate.
        WRITE: / CONV text30( lo_object->get_key( ) ),
            lv_mdate, lo_object->get_size( ).
    ENDLLOOP.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_msg) = lo_ex->if_message~get_text( ).
        MESSAGE lv_msg TYPE 'I'.
ENDTRY.

```

コードセクション

以下は、コードをセクションごとにまとめたものです。

```

PARAMETERS pv_lres TYPE /aws1/rt_resource_logical
            DEFAULT 'DEMO_BUCKET' OBLIGATORY.

```

ユーザーは物理バケット名を指定できません。論理バケットを指定し、システム管理者 (特にビジネスアナリスト) は AWS 管理者と連携して論理バケットを /AWS1/IMG 内の物理バケットにマッピングします。ほとんどのビジネスシナリオでは、ユーザーは論理バケットを選択する機会がありません。論理リソース ID はコードにハードコーディングされているか、カスタム設定テーブルで設定されます。

```

DATA(go_session) = /aws1/cl_rt_session_aws=>create( 'DEMO_S3' ).

```

この行はセキュリティセッションを確立し、この ABAP プログラムが DEMO_S3 SDK プロファイルを使用することを想定していることを宣言します。この呼び出しは SDK 設定への接続であり、デフォルトのリージョン、認証設定、および必要な IAM ロールを取得します。認可オブジェクトが /AWS1/SESS 確実に満たされるように、の呼び出し AUTHORIZATION-CHECK が自動的に行われます。さらに、承認オブジェクト /AWS1/LROL に基づいて、ユーザーに許可されている最も強力な (シーケンス番号が小さい) 論理 IAM ロールを決定するための AUTHORIZATION-CHECK 呼び出しが行われます。SDK は、IAM ロールが SID とクライアントの論理 IAM ロールにマッピングされていると想定します。次に、セッションオブジェクトは IMG のトレース設定に基づいてトレースを有効にします。

リクエストされた SDK プロファイルまたは使用可能な論理 IAM ロールに対する権限がユーザーにない場合、例外が発生します。

```
DATA(gv_bucket) = go_session->resolve_lresource( pv_lres ).
```

この行は論理リソースを物理バケット名に解決します。この SID/クライアントの組み合わせに対するマッピングが構成に含まれていないために論理リソースを解決できない場合、例外が発生します。

```
DATA(go_s3) = /aws1/cl_s3_factory=>create( go_session ).
```

この行では、/aws1/cl_s3_factory の create() メソッドを使用して Amazon S3 の API オブジェクトを作成します。返されるオブジェクトは、Amazon S3 API のインターフェイスである /aws1/if_s3 型です。サービスごとに個別の API オブジェクトを作成する必要があります。たとえば、ABAP プログラムが Amazon S3、AWS Lambda、DynamoDB を使用している場合、/aws1/cl_s3_factory、/aws1/cl_lmd_factory および /aws1/cl_dyn_factory から API オブジェクトを作成します。

コンストラクタには、IMG に設定されているデフォルトリージョンをオーバーライドしたい場合にリージョンを指定できるオプションパラメータがいくつかあります。このように、あるリージョンのバケットから別のリージョンのバケットにオブジェクトをコピーする場合、/aws1/if_s3 の 2 つのインスタンス (us-east-1 用と us-west-2 用) を使用できます。同様に、財務関連のバケットからレポートを読み込み、ロジスティクス関連のバケットにオブジェクトを書き込む必要がある場合は、2 つの異なるセキュリティセッションオブジェクトを作成し、それらを使用して /aws1/cl_s3 の 2 つの個別のインスタンスを作成できます。

```
DATA(lo_output) = go_s3->listobjectsv2(  
    iv_bucket = CONV string( gv_bucket )  
    iv_maxkeys = 100  
).
```

この行は ListObjectsV2 への呼び出しです。単純な入力引数が必要で、1つのオブジェクトを返します。これらのオブジェクトは、ABAP オブジェクト指向構造に逆シリアル化された深い JSON データや XML データを表す場合があります。場合によっては、かなり複雑になることがあります。あとは、出力を処理してバケットの内容を一覧表示するだけです。

```
LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).  
    DATA lv_mdate TYPE datum.  
    CONVERT TIME STAMP lo_object->get_lastmodified( )  
        TIME ZONE 'UTC'  
        INTO DATE lv_mdate.  
    WRITE: / CONV text30( lo_object->get_key( ) ),  
        lv_mdate, lo_object->get_size( ).  
ENDLOOP.
```

データには、データの内部表現を隠す GET...() スタイルメソッドを使用してアクセスします。GET_CONTENTS() は ABAP テーブルを返し、各行自体には単一の Amazon S3 エントリを表すオブジェクトが含まれます。ほとんどの場合、AWS SDK はこのオブジェクト指向のアプローチを取り、すべてのデータはオブジェクトとテーブルとして表されます。LastModifiedフィールドはタイムスタンプとして表され、ABAP ネイティブ CONVERT TIME STAMP コマンドを使用して日付に変換できます。は、計算とフォーマットの操作を容易に INT4 するために GET_SIZE() を返します。

```
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_msg) = lo_ex->if_message~get_text( ).  
    MESSAGE lv_msg TYPE 'I'.
```

接続、4xx クライアント、5xx サーバー、または認可エラーや設定エラーなどの ABAP エラーを含むすべてのエラーは例外として表されます。それぞれの例外に個別に対処できます。例外を情報エラー、再試行、警告、シヨートダンプ、またはその他の処理として処理するかどうかを選択できます。

SDK for SAP ABAP の概念

このセクションでは、 の基本概念について説明します AWS SDK for SAP ABAP。

トピック

- [API クラス](#)
- [追加のオブジェクト](#)
- [構造クラス](#)
- [配列](#)
- [マップ](#)
- [高レベルの関数](#)

API クラス

各には 3 文字の頭字語または AWS のサービス が割り当てられます TLA。サービスは /AWS1/IF_<TLA> 形式のインターフェイスで表されます。これをサービスインターフェイスと呼びます。API クラスは /AWS1/API_<TLA> パッケージにあります。サービスインターフェイスは AWS、オペレーションごとに 1 つのメソッドで構成されます (これらのメソッドをオペレーションメソッドと呼びます)。AWS SDK for SAP ABAP TLAs の完全なモジュールリストを確認するには、「[AWS SDK for SAP ABAP - Module List](#)」を参照してください。

各オペレーションメソッドには、いくつかの IMPORTING 引数と、最大 1 つの RETURNING 引数があります。多くの場合、これらの引数は複雑なコンストラクターと長い GET...() メソッドセットを持つオブジェクトです。多くの場合、オブジェクトには入れ子オブジェクト、再帰リファレンス、オブジェクトのテーブル、テーブルのテーブルなどが含まれます。これは、AWS のサービスがディープ XML 構造と JSON 構造を渡すためです。これは、引数のフラットセットでは表現できません。

RETURNING 引数は、クラスに 1 つの属性のみが含まれている場合でも、常にクラスです。

追加のオブジェクト

各 API パッケージには、プライマリ API クラスに加えて、関連するさまざまなリポジトリとデータディクショナリオブジェクトが含まれています。

- 各構造型オブジェクトのクラス。

- テーブルに表示されるあらゆるプリミティブデータ型のクラス。例えば、サービスが文字列のテーブルを返す場合、ABAP API はそれをオブジェクトのテーブルとして表現します。各オブジェクトは文字列をカプセル化するラッパークラスです。これは、ABAP でネイティブに表現できない null 文字列を表現する際に、ラッパークラスが詳細を隠すことができるようにするためです。
- サービスによって定義された特定のエラーの例外クラス。
- 各プリミティブデータ型のデータ要素。各データ型には、自己文書化できるように独自のデータ要素があります。
- XML や JSON ペイロードをシリアル化および逆シリアル化するための XSLT 変換など、内部処理用の追加オブジェクト。

構造クラス

サービスによって送受信されるほとんどの AWS データは、AWS SDK によってクラスとして表されます。これらのクラスはデータの構造を表し、ストレージの内部の詳細を隠します。特に、このフィールドには値がないことを SDK が表現する方法がクラスによって隠されています。

構造クラスの各フィールドには、3 つのメソッドがあります。

GET_field()

GET_field() メソッド

- フィールドの値が返されるか、または
- フィールドに値がない場合は、オプションパラメータとして設定できるデフォルト値が返されます。

例えば、バケットの場所の制約を出力する次のコードを考えてみます。

```
DATA(lo_location) = go_s3->getbucketlocation( iv_bucket = CONV string( gv_bucket ) ).  
WRITE: / 'Bucket Location: ',  
       lo_location->get_locationconstraint( ).
```

バケットに場所の制約がまったくない場合 (us-east-1 の場合のように)、GET_LOCATIONCONSTRAINT() は空の文字列を返します。フィールドに値がまったくない場合は、この動作をオーバーライドして目的の値を指定できます。

```
DATA(lo_location) = go_s3->getbucketlocation( iv_bucket = CONV string( gv_bucket ) ).  
WRITE: / 'Bucket Location: ',  
       lo_location->get_locationconstraint( iv_value_if_missing = 'assuming us-east-1' ).
```

これで、`getbucketlocation()` の結果で場所が返されない場合、プログラムは `Bucket Location: assuming us-east-1` と記述するようになります。

要求された値が完全に欠落している場合、`GET()` メソッドに特定の結果を返すように要求できません。次のコード例を参照してください。

```
data(lo_location) = go_s3->GETBUCKETLOCATION(  
    new /AWS1/CL_S3_GET_BUCKET_LOC_REQ( iv_bucket = gv_bucket )  
).  
write: / 'Location constraint: ',  
       lo_location->GET_LOCATIONCONSTRAINT( 'NopeNopeNope' ).
```

この場合、場所の制約がなければ、`GET_LOCATIONCONSTRAINT()` は `NopeNopeNope` を返します。

HAS_field()

`HAS_field()` メソッドはフィールドに値があるかどうかを調べる方法です。次の例を参照してください。

```
if NOT lo_location->HAS_LOCATIONCONSTRAINT( ).  
    write: / 'There is no location constraint'.  
endif.
```

あるフィールドに必ず値があることがわかっている場合、`HAS_field()` メソッドはありません。

ASK_field()

`ASK_field()` メソッドはフィールドの値を返すか、値がない場合は例外を発生させます。これは、多数のフィールドを処理し、値のないフィールドがある場合にロジックを回避して別のアプローチを取る場合に便利な方法です。

```
TRY.
```

```
WRITE: / 'Location constraint: ', lo_location->ask_locationconstraint( ).
CATCH /aws1/cx_rt_value_missing.
WRITE: / 'Never mind, there is no location constraint'.
ENDTRY.
```

/AWS1/CX_RT_VALUE_MISSING は静的な例外であり、これをキャッチしないことを選択すると警告が表示されることに注意してください。

ベストプラクティス

一般に、GET_field() メソッドは NULL 文字列を空の文字列として扱い、3つのオプションの中で最も ABAP に似ているため、使用できます。ただし、この方法では、フィールドの値が空白の状況とフィールドに値がない状況を簡単に区別することはできません。ビジネスロジックが欠損データと空白データの区別に依存している場合は、HAS または ASK メソッドを使用してこれらのケースに対処できます。

配列

配列は ABAP 標準オブジェクトテーブルとして表されます。

JSON 配列には、['cat', 'dog', null, 'horse'] の配列のような NULL 値を含めることができます。これはスパース配列と呼ばれます。ABAP ではオブジェクトリファレンスの内部テーブルとして表され、null 値はテーブル内では真の ABAP null 値として表されます。スパーステーブル内を繰り返し処理する場合、null オブジェクトにアクセスして CX_SY_REF_IS_INITIAL 例外が発生しないように、null 値をチェックする必要があります。実際には、スパース配列は AWS サービスではまれです。

オブジェクトの配列を初期化するには、ABAP 7.40 の新しいコンストラクトを使用すると便利です。複数のセキュリティグループが割り当てられた Amazon EC2 インスタンスの起動を考えてみます。

```
ao_ec2->runinstances(
  iv_imageid           = lo_latest_ami->get_imageid( )
  iv_instancetype     = 't2.micro'
  iv_maxcount         = 1
  iv_mincount         = 1
  it_securitygroupids = VALUE /aws1/
cl_ec2secgrpiddstrlist_w=>tt_securitygroupidstringlist(
  ( NEW /aws1/
cl_ec2secgrpiddstrlist_w( 'sg-12345678' ) )
```

```
        ( NEW /aws1/  
cl_ec2secgrpiddstrlist_w( 'sg-55555555' ) )  
        ( NEW /aws1/  
cl_ec2secgrpiddstrlist_w( 'sg-99999999' ) )  
    )  
    iv_subnetid          = ao_snet->get_subnetid( )  
    it_tagspecifications = make_tag_spec( 'instance' )  
)
```

マップ

JSON マップは ABAP では Hashed Tables として表され、各テーブル行には 2 つのコンポーネントのみが含まれます。

- KEY — テーブルの UNIQUE KEY である文字列。
- VALUE — 値を含むオブジェクト。

マップは、AWS SDK がクラスではなく真の構造を使用する数少ないケースの 1 つです。これは、ABAP ハッシュテーブルがキーフィールドとしてオブジェクト参照を持つことができず、AWS マップキーは常に null 以外の文字列であるために必要です。

高レベルの関数

前のセクションで [API クラス](#) 説明した は、AWS サービス APIs し、これらの APIs として表します。場合によっては、SDK には、特定のオペレーションを簡素化するために API クラス上に構築される高レベルの関数も含まれています。プログラマーの利便性のために上位レベルの関数が含まれており、下位レベルの API クラスを置き換えるものではありません。

SDK にモジュールの上位レベルの関数が含まれている場合、それらは同じトランスポートに含まれ、というファクトリクラスを介してアクセスできます /AWS1/CL_TLA_L2_FACTORY。ファクトリクラスには、API [ドキュメント](#) で残りの API とともに文書化された、モジュールのさまざまな上位レベルのクライアントを作成するメソッドが含まれています。

AWS SDK for SAP ABAP features

AWS SDK for SAP ABAP には以下の機能があります。

トピック

- [プログラムによる設定](#)
- [ウェーター](#)
- [ページネーター](#)
- [再試行動作](#)
- [署名者](#)
- [クロスアカウント IAM ロールの連鎖](#)

プログラムによる設定

AWS SDK for SAP ABAP には `/n/AWS1/IMG IMG` 変換を使用し、プログラムによる設定には AWS SDK for SAP ABAP - BTP エディションにはカスタムビジネス設定アプリケーションを使用します。

プログラムによる設定を開始するには、まず `get_config()` コマンドで設定オブジェクトを取得します。

```
data(lo_config) = lo_s3->get_config( ).
```

各設定オブジェクトは、IMG に対応する GETter と SETter を含む `/AWS1/IF_RT_CONFIG` インターフェイスを実装します。例えば、デフォルトリージョンはオーバーライドできます。次のコマンド例を参照してください。

```
lo_s3->get_config( )->/aws1/if_rt_config~set_region( 'us-east-1' ).
```

設定オブジェクトの中には、IMG 表現がなく、最大再試行回数など、プログラムでしか設定できないものもあります。次のコマンド例を参照してください。

```
lo_s3->get_config( )->/aws1/if_rt_config~set_max_attempts( 10 ).
```

の設定オブジェクト AWS のサービス には、 で表されていないサービス固有のメソッドを含めることもできます `/aws1/if_rt_config`。例えば、Amazon S3 は、 `foobucket.s3.region.amazonaws.com` 仮想エンドポイントまたは `s3.region.amazonaws.com/foobucket` パススタイルを使用して `foobucket` という名前のバケットをアドレス指定できます。以下のコマンド例でパススタイルの使用を強制できます。

```
lo_s3->get_config( )->set_forcepathstyle( abap_true ).
```

サービス設定の詳細については、[AWS SDK for SAP ABAP 「 - API リファレンスガイド」](#)を参照してください。

ウェーター

非同期 AWS APIs を使用する場合は、特定のリソースが利用可能になるまで待つから、さらにアクションを実行する必要があります。たとえば、 の `CREATETABLE()` API は、テーブルステータスですぐに Amazon DynamoDB 応答します `CREATING`。読み取りまたは書き込み操作は、テーブルのステータスが `ACTIVE` に変更された後にのみ開始できます。ウェーターを使用すると、リソースに対してアクションを実行する前に、AWS リソースが特定の状態にあることを確認できます。

ウェーターは、サービスオペレーションを使用して、AWS リソースが目的の状態に達するか、リソースが目的の状態に達しないと判断されるまで、リソースのステータスをポーリングします。AWS リソースを継続的にポーリングするコードを書くのは時間がかかり、エラーが発生しやすくなります。ウェーターは、ユーザーに代わってポーリングを実施する責任を負うことで、この複雑さを簡素化するのに役立ちます。

ウェーターを使用した次の Amazon S3 の例を参照してください。

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

" Create a bucket - initiates the process of creating an S3 bucket and might return
before the bucket exists
lo_s3#createbucket( iv_bucket = |amzn-s3-demo-bucket| ).

" Wait until the newly created bucket becomes available
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
    iv_bucket = |amzn-s3-demo-bucket|
).
```

- 次の例では、Amazon S3 クライアントを使用してバケットを作成します。 `get_waiter()` コマンドは、いつ `bucketexists` を指定するように実装されています。
- ウェーターごとに `iv_max_wait_time` パラメータを指定する必要があります。これは、ウェーターが完了するまで待たなければならない合計時間を表します。前述の例では、ウェーターは 200 秒間実行できます。

- 必須パラメータには追加の入力が必要な場合があります。前の例では、iv_bucket パラメータには Amazon S3 バケット名が必要です。
- /AWS1/CX_RT_WAITER_FAILURE 例外は、ウェーターが iv_max_wait_time パラメータで指定された最大時間を超えたことを示します。
- /AWS1/CX_RT_WAITER_TIMEOUT 例外は、ウェーターが目的の状態に達しなかったために停止したことを示します。

ページネーター

一部の AWS のサービス オペレーションでは、ページ分割されたレスポンスが提供されます。応答ごとに固定量のデータを返すようにページ分割されます。結果セット全体を取得するには、トークンまたはマーカを使用して後続のリクエストを行う必要があります。例えば、ListObjectsV2 Amazon S3 オペレーションは、一度に最大で 1,000 個のオブジェクトを返します。結果の次のページを取得するには、適切なトークンを使用して後続のリクエストを行う必要があります。

ページ分割とは、連続してリクエストを送信して、前のリクエストが中断したところから再開するプロセスです。ページネーターは SDK for SAP ABAP によって提供される結果のイテレーターです。ページ分割された API は、ページ分割トークンを使用する API の基本的なメカニズムを理解していなくても簡単に使用できます。

ページネーターの操作

ページネーターオブジェクトを返す `get_paginator()` メソッドを使用してページネーターを作成できます。ページネーターオブジェクトは、ページ分割中のオペレーションを呼び出します。ページネーターオブジェクトは、基になる API に提供される必須パラメータを受け入れます。このプロセスは、`has_next()` および `get_next()` メソッドを使用して、ページ分割された結果を繰り返し処理できるイテレーターオブジェクトを返します。

- `has_next()` は、呼び出されたオペレーションで利用できる応答またはページが他にもあるかどうかを示すブール値を返します。
- `get_next()` は、オペレーションレスポンスを返します。

次の例では、ページネーターを使用して取得した S3 バケット内のすべてのオブジェクトを一覧表示しています。

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( 'DEMO' ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).
```

```
TRY.  
  DATA(lo_paginator) = lo_s3->get_paginator( ).  
  DATA(lo_iterator) = lo_paginator->listobjectsv2(  
    iv_bucket = 'example_bucket'  
  ).  
  WHILE lo_iterator->has_next( ).  
    DATA(lo_output) = lo_iterator->get_next( ).  
    LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).  
      WRITE: / lo_object->get_key( ), lo_object->get_size( ).  
    ENDLOOP.  
  ENDWHILE.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  MESSAGE lo_ex->if_message~get_text( ) TYPE 'I'.  
ENDTRY.
```

再試行動作

SDK for SAP ABAP では、スロットリングまたは一時的なエラーにより失敗した AWS のサービスへのリクエストの最大再試行回数を設定できます。サービスクライアントレベルで許可される再試行の回数、つまり失敗して例外が発生する前に SDK がオペレーションを再試行する回数は、サービス設定オブジェクトの AV_MAX_ATTEMPTS 属性によって指定されます。サービスクライアントオブジェクトが作成されると、SDK は AV_MAX_ATTEMPTS 属性をデフォルト値の 3 に設定します。サービス設定オブジェクトを使用すると、最大再試行回数をプログラムで希望の値に設定できます。詳細については、次の例を参照してください。

```
" Retrieve configuration object using Amazon S3 service's get_config( ) method  
DATA(lo_config) = lo_s3->get_config( ).  
  
" Set the maximum number of retries to 5  
lo_config->/aws1/if_rt_config~set_max_attempts( 5 ).  
  
" Get the value of the maximum retry attempt.  
DATA(lv_max_retry_attempts) = lo_config->/aws1/if_rt_config~get_max_attempts( ).
```

Note

設定オブジェクト ABAP SDK では /AWS1/IF_RT_CONFIG~SET_RETRY_MODE() メソッドで再試行モードを設定できますが、SDK は standard 再試行モードのみをサポートします。詳細については、「SDK およびツールリファレンスガイド」の「[Retry behavior](#)」を参照してください。AWS SDKs

署名者

署名付き URLs を使用して、一部の に時間制限付きアクセスを許可できます AWS のサービス。署名付き URL は、ブラウザに入力することも、プログラムで使用してサービスオペレーションを実行することもできます。署名付き URL は、有効期限日時まで複数回使用できます。詳細については、[URLs SDK for SAP ABAP クライアントの使用](#)には、そのサービスの署名者を作成GET_PRESIGNER()するための特別なメソッドが呼び出されます。次に、API クライアントのメソッドに対応する署名者のメソッドを呼び出します。ただし、実際にオペレーションを実行するのではなく、署名付き URL を返す点が異なります。

```
" Retrieve a presigner for Amazon S3
DATA(lo_presigner) = lo_s3->get_presigner( iv_expires_sec = 600 ).

" the presigner getobject() method has the same signature as
" lo_s3->getobject(), but it doesn't actually make the call.
" to the service. It just prepares a presigned URL for a future call
DATA(lo_presigned_req) = lo_presigner->getobject( iv_bucket = iv_bucket_name iv_key =
  iv_key ).

" You can provide this URL to a web page, user, email etc so they
" can retrieve the file. The URL will expire in 10 minutes.
ov_url = lo_presigned_req->get_url( ).
```

クロスアカウント IAM ロールの連鎖

クロスアカウント IAM ロール連鎖のサポートにより、ソースプロファイル設定を通じて複数の AWS アカウントにわたるリソースへのシームレスなアクセスが可能になります。この機能を使用すると、複数のロールの前提条件を設定できます。1つのプロファイルはロールを引き受け、別のロールを引き受けて、複雑なクロスアカウントアクセスパターンを有効にします。

詳細については、[「クロスアカウントアクセスのソースプロファイルの使用」](#)を参照してください。

SDK による製品の構築

が消費する製品または ABAP アドオン AWS のサービスは、SDK の機能を強化および拡張できます。このような製品を構築して SDK と共に使用することができます。

トピック

- [製品 ID を設定する](#)

製品 ID を設定する

製品 ID は、製品またはアドオン内でセッションを確立するときに設定することをお勧めします。詳細については、次の例を参照してください。

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( 'DEMO' ).  
lo_session->set_product_id( 'INVOICE_ANALYZER' ).
```

製品 ID には、文字、数字、アンダースコアのみを使用でき、スペースや特殊文字を使用することはできません。製品 ID は、製品の技術名やその他の ID に対応させることができます。複数の製品やアドオンを開発する場合、製品 ID は製品ごとに固有である必要があります。例えば、Invoice Analyzer、Tax Calculator、Pricing Engine の各製品の製品 ID は、INVOICE_ANALYZER、TAX_CALCULATOR、PRICING_ENGINE とすることができます。

セッションに製品 ID を追加すると、サービス呼び出し AWS ごとに送信されるテレメトリが強化されます。呼び出しを行うオブジェクトの製品 ID と名前空間がテレメトリに含まれます。このテレメトリを使用すると、SDK で顧客が問題に直面した場合に、が通話を行っている製品を特定サポートできます。これにより、呼び出しが実際には製品によって行われているものであり、顧客のコードによるものではないことが明確になります。

HTTP リクエストを にカスタマイズする AWS

は、HTTP リクエストの作成、ペイロードの送信、レスポンスの受信のプロセス AWS SDK for SAP ABAP を処理します。HTTP リクエストの動作またはコンテンツをカスタマイズして、独自の IT 要件を満たすことができます。SDK は、HTTP 通信を強化するための中心的な場所/AWS1/RT_EHN_HTTP_CLIENTとしてエンハンスメントスポットを定義します。エンハンスメントスポットは、へのリクエストへの HTTP ヘッダーの追加をサポートします AWS。

エンハンスメントを実装する

SAP では、エンハンスメントスポットを実装するための以下の手順が用意されています。

- [クラシック ABAP](#)
- [BTP ABAP](#)

エンハンスメントをフィルタリングする

エンハンスメントスポットは、同時にアクティブにできる複数の実装をサポートします。エンハンスメントが特定の AWS サービスまたは API アクションへの呼び出しでのみ実行されるようにする必要があります。次の属性に基づいて BAdi の実行をフィルタリングできます。

- TLA - サービスの 3 文字の省略形。大文字で表記します。
- OPERATION - API アクション名。たとえば、S3 バケットからオブジェクトを取得するオペレーションは [GetObject](#) です。アクション名では大文字と小文字が区別され、ABAP メソッド名と完全に一致しない場合があります。

エンハンスメントをコーディングする

拡張機能では、次の方法が提供されます。

MODIFY_REQ_HEADERS

```
CHANGING CT_HEADERS TYPE /AWS1/RT_STRINGMAP_TT
```

CT_HEADERS 内部テーブルにヘッダーを追加および変更できます。ヘッダーを変更することはお勧めしません。これにより、AWS サービスが使用するデータが変更されるためです。追加したヘッダーは AWS サービスによって無視されますが、プロキシサーバーやその他のミドルウェアなどの IT インフラストラクチャで処理できます。

エンハンスメントスポットは、認証ヘッダーとテレメトリヘッダーの計算前に呼び出されるため、エンハンスメントによって変更することはできません。

実装例を次に示します。

```
METHOD /aws1/if_rt_badi_http_client~modify_req_headers.  
  APPEND VALUE /aws1/rt_stringpair_ts( name = 'x-test-example' value = 'value' )  
  TO ct_headers.  
ENDMETHOD.
```

制限事項

AWS SDK for SAP ABAP には、すべての用の SDK モジュールが含まれています AWS のサービス。これらのモジュールの中には、ここで説明するように、制限があるものもあります。

- などのMQTTプロトコルバイインディングに依存するモジュールは機能しません。MQTTは HTTP ベースのプロトコルではなく iotevents、現在 ではサポートされていません AWS SDK for SAP ABAP。
- イベントストリームを返すオペレーションはサポートされますが、 はストリーム全体が受信されるまで待ってから ABAP に制御を返します。Amazon Bedrock エージェントランタイム InvokeAgent と AWS Lambda の例 InvokeWithResponseStream
- 基になる ABAP プラットフォームの制限により、 イベントストリームを受信するオペレーションはサポートされていません。Amazon Q Business Chat と Amazon Lex の例 StartConversation

AWS SDK for SAP ABAP には以下の機能制限があります。

- 以下の Amazon S3 機能はまだサポートされていません。
 - マルチリージョンアクセスポイント
 - Amazon S3 クライアント側の暗号化

AWS SDK for SAP ABAP - BTP エディションには、デベロッパープレビュー中に以下の制限があります。

- 一部のモジュールは使用できない場合があります。
- アンインストールすることはできません。
- 更新頻度は低くなります。

SDK for SAP ABAP のコード例

このトピックのコード例は、で AWS SDK for SAP ABAP を使用方法を示しています AWS。

基本は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

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。

一部のサービスには、サービス固有のライブラリや関数の活用方法を示す追加のカテゴリ例が含まれています。

サービス

- [SDK for SAP ABAP を使用した ACM の例](#)
- [SDK for SAP ABAP を使用した API Gateway の例](#)
- [SDK for SAP ABAP を使用した Application Recovery Controller の例](#)
- [SDK for SAP ABAP を使用した Aurora の例](#)
- [SDK for SAP ABAP を使用した Auto Scaling の例](#)
- [SDK for SAP ABAP を使用した Amazon Bedrock ランタイムの例](#)
- [SDK for SAP ABAP を使用した Amazon Bedrock エージェントランタイムの例](#)
- [SDK for SAP ABAP を使用した CloudFront の例](#)
- [SDK for SAP ABAP を使用した CloudWatch の例](#)
- [SDK for SAP ABAP を使用した CloudWatch Logs の例](#)
- [SDK for SAP ABAP を使用した Amazon Cognito ID プロバイダーの例](#)
- [SDK for SAP ABAP を使用した Amazon Comprehend の例](#)
- [AWS Config SDK for SAP ABAP を使用した例](#)
- [AWS Control Tower SDK for SAP ABAP を使用した例](#)
- [SDK for SAP ABAP を使用した Firehose の例](#)
- [SDK for SAP ABAP を使用した DynamoDB の例](#)

- [SDK for SAP ABAP を使用した Amazon EC2 の例](#)
- [SDK for SAP ABAP を使用した Amazon ECR の例](#)
- [SDK for SAP ABAP を使用した Amazon EMR の例](#)
- [SDK for SAP ABAP を使用した EventBridge スケジューラの例](#)
- [AWS Glue SDK for SAP ABAP を使用した例](#)
- [SDK for SAP ABAP を使用した HealthImaging の例](#)
- [SDK for SAP ABAP を使用した HealthLake の例](#)
- [SDK for SAP ABAP を使用した IAM の例](#)
- [AWS IoT data SDK for SAP ABAP を使用した例](#)
- [AWS IoT SiteWise SDK for SAP ABAP を使用した例](#)
- [SDK for SAP ABAP を使用した Amazon Keyspaces の例](#)
- [SDK for SAP ABAP を使用した Kinesis の例](#)
- [AWS KMS SDK for SAP ABAP を使用した例](#)
- [SDK for SAP ABAP を使用した Lambda の例](#)
- [SDK for SAP ABAP を使用した Organizations の例](#)
- [SDK for SAP ABAP を使用した Amazon Pinpoint の例](#)
- [SDK for SAP ABAP を使用した Amazon Pinpoint SMS および音声 API の例](#)
- [SDK for SAP ABAP を使用した Amazon Polly の例](#)
- [SDK for SAP ABAP を使用した Amazon RDS の例](#)
- [SDK for SAP ABAP を使用した Amazon Redshift の例](#)
- [SDK for SAP ABAP を使用した Amazon Rekognition の例](#)
- [SDK for SAP ABAP を使用した Amazon S3 の例](#)
- [SDK for SAP ABAP を使用した SageMaker AI の例](#)
- [SDK for SAP ABAP を使用した Secrets Manager の例](#)
- [SDK for SAP ABAP を使用した Amazon SES の例](#)
- [SDK for SAP ABAP を使用した Amazon SES API v2 の例](#)
- [SDK for SAP ABAP を使用した Amazon SNS の例](#)
- [SDK for SAP ABAP を使用した Amazon SQS の例](#)

- [SDK for SAP ABAP を使用した Step Functions の例](#)
- [SDK for SAP ABAP を使用した Systems Manager の例](#)
- [SDK for SAP ABAP を使用した Amazon Textract の例](#)
- [SDK for SAP ABAP を使用した Amazon Transcribe の例](#)
- [SDK for SAP ABAP を使用した Amazon Translate の例](#)

SDK for SAP ABAP を使用した ACM の例

次のコード例は、ACM で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AddTagsToCertificate

次の例は、AddTagsToCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
" iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
lo_acm->addtagstocertificate(
  iv_certificatearn = iv_certificate_arn
  it_tags = it_tags
).
MESSAGE 'Tags added to certificate successfully.' TYPE 'I'.
CATCH /aws1/cx_acminvalidarnex.
MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
CATCH /aws1/cx_acmresourcenotfoundex.
MESSAGE 'Certificate not found.' TYPE 'I'.
CATCH /aws1/cx_acminvalidtagex.
MESSAGE 'Invalid tag provided.' TYPE 'I'.
CATCH /aws1/cx_acmtoomanytagsex.
MESSAGE 'Too many tags for certificate.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[AddTagsToCertificate](#)」を参照してください。

DeleteCertificate

次の例は、DeleteCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
" iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
lo_acm->deletecertificate( iv_certificatearn = iv_certificate_arn ).
MESSAGE 'Certificate deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_acminvalidarnex.
MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
CATCH /aws1/cx_acmresourcenotfoundex.
```

```
MESSAGE 'Certificate not found.' TYPE 'I'.
CATCH /aws1/cx_acmresourceinuseex.
MESSAGE 'Certificate is in use and cannot be deleted.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteCertificate](#)を参照してください。

DescribeCertificate

次の例は、DescribeCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
  oo_result = lo_acm->describecertificate( iv_certificatearn =
iv_certificate_arn ).
  MESSAGE 'Certificate details retrieved.' TYPE 'I'.
CATCH /aws1/cx_acminvalidarnex.
  MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
CATCH /aws1/cx_acmresourcefoundex.
  MESSAGE 'Certificate not found.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeCertificate](#)」を参照してください。

GetCertificate

次の例は、GetCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/  
certificate-id'  
    oo_result = lo_acm->getcertificate( iv_certificatearn =  
iv_certificate_arn ).  
    MESSAGE 'Certificate body and chain retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_acminvalidarnex.  
    MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.  
    CATCH /aws1/cx_acmresourcenotfoundex.  
    MESSAGE 'Certificate not found.' TYPE 'I'.  
    CATCH /aws1/cx_acmrequestinprgssex.  
    MESSAGE 'Certificate request is in progress.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetCertificate](#)」を参照してください。

ImportCertificate

次の例は、ImportCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " Only pass certificate chain if it's provided (it's optional)
```

```
IF iv_certificate_chain IS NOT INITIAL.  
  DATA(lo_result) = lo_acm->importcertificate(  
    iv_certificate = iv_certificate  
    iv_privatekey = iv_private_key  
    iv_certificatechain = iv_certificate_chain  
  ).  
ELSE.  
  lo_result = lo_acm->importcertificate(  
    iv_certificate = iv_certificate  
    iv_privatekey = iv_private_key  
  ).  
ENDIF.  
ov_certificate_arn = lo_result->get_certificatearn( ).  
MESSAGE 'Certificate imported successfully.' TYPE 'I'.  
CATCH /aws1/cx_acminvalidparameterex.  
  MESSAGE 'Invalid parameter provided.' TYPE 'I'.  
CATCH /aws1/cx_acmlimitexceededex.  
  MESSAGE 'Certificate limit exceeded.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの [ImportCertificate](#) を参照してください。

ListCertificates

次の例は、ListCertificates を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_acm->listcertificates(  
    iv_maxitems = iv_max_items  
    it_certificatestatuses = it_statuses  
    io_includes = io_includes
```

```
    ).  
    MESSAGE 'Certificates listed successfully.' TYPE 'I'.  
CATCH /aws1/cx_acminvalidargsex.  
    MESSAGE 'Invalid arguments provided.' TYPE 'I'.  
CATCH /aws1/cx_acmvalidationex.  
    MESSAGE 'Validation error occurred.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListCertificates](#)」を参照してください。

ListTagsForCertificate

次の例は、ListTagsForCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/  
certificate-id'  
    DATA(lo_result) = lo_acm->listtagsforcertificate(  
        iv_certificatearn = iv_certificate_arn  
    ).  
    ot_tags = lo_result->get_tags( ).  
    MESSAGE 'Certificate tags retrieved successfully.' TYPE 'I'.  
CATCH /aws1/cx_acminvalidarnex.  
    MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.  
CATCH /aws1/cx_acmresourcenotfoundex.  
    MESSAGE 'Certificate not found.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListTagsForCertificate](#)」を参照してください。

RemoveTagsFromCertificate

次の例は、RemoveTagsFromCertificate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/  
certificate-id'  
    lo_acm->removetagsfromcertificate(  
        iv_certificatearn = iv_certificate_arn  
        it_tags = it_tags  
    ).  
    MESSAGE 'Tags removed from certificate successfully.' TYPE 'I'.  
CATCH /aws1/cx_acminvalidarnex.  
    MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.  
CATCH /aws1/cx_acmresourcenotfoundex.  
    MESSAGE 'Certificate not found.' TYPE 'I'.  
CATCH /aws1/cx_acminvalidtagex.  
    MESSAGE 'Invalid tag provided.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[RemoveTagsFromCertificate](#)」を参照してください。

RequestCertificate

次の例は、RequestCertificate を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
  " iv_domain_name = 'example.com'
  " iv_validation_method = 'DNS' or 'EMAIL'
  DATA(lo_result) = lo_acm->requestcertificate(
    iv_domainname = iv_domain_name
    it_subjectalternativenames = COND #( WHEN it_alternate_domains IS NOT
INITIAL
                                     THEN it_alternate_domains )
    iv_validationmethod = iv_validation_method
  ).
  ov_certificate_arn = lo_result->get_certificatearn( ).
  MESSAGE 'Certificate requested successfully.' TYPE 'I'.
CATCH /aws1/cx_acminvalidparameterex.
  MESSAGE 'Invalid parameter provided.' TYPE 'I'.
CATCH /aws1/cx_acmlimitexceededx.
  MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
CATCH /aws1/cx_acminvdomvationoptsex.
  MESSAGE 'Invalid domain validation options.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[RequestCertificate](#)を参照してください。

ResendValidationEmail

次の例は、ResendValidationEmail を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
  " iv_domain = 'example.com'
  " iv_validation_domain = 'example.com'
  lo_acm->resendvalidationemail(
    iv_certificatearn = iv_certificate_arn
    iv_domain = iv_domain
    iv_validationdomain = iv_validation_domain
  ).
  MESSAGE 'Validation email resent successfully.' TYPE 'I'.
CATCH /aws1/cx_acminvalidarnex.
  MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
CATCH /aws1/cx_acmresourcenotfoundex.
  MESSAGE 'Certificate not found.' TYPE 'I'.
CATCH /aws1/cx_acminvalidstateex.
  MESSAGE 'Certificate is not in a valid state.' TYPE 'I'.
CATCH /aws1/cx_acminvdomvationoptsex.
  MESSAGE 'Invalid domain validation options.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ResendValidationEmail](#)を参照してください。

SDK for SAP ABAP を使用した API Gateway の例

次のコード例は、API Gateway で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateDeployment

次の例は、CreateDeployment を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->createdeployment(  
        iv_restapiid = iv_rest_api_id  
        iv_stagename = iv_stage_name  
        iv_description = 'Deployment created by ABAP SDK' ).  
    DATA(lv_deployment_id) = oo_result->get_id( ).  
    MESSAGE 'Deployment created with ID: ' && lv_deployment_id TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
    MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
    MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
    MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many.
```

```
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateDeployment](#)」を参照してください。

CreateResource

次の例は、CreateResource を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->createresource(  
        iv_restapiid = iv_rest_api_id  
        iv_parentid = iv_parent_id  
        iv_pathpart = iv_resource_path ).  
    DATA(lv_resource_id) = oo_result->get_id( ).  
    MESSAGE 'Resource created with ID: ' && lv_resource_id TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
    MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
    MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
    MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateResource](#)」を参照してください。

CreateRestApi

次の例は、CreateRestApi を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->createrestapi(  
        iv_name = iv_api_name  
        iv_description = 'Sample REST API created by ABAP SDK' ).  
    DATA(lv_api_id) = oo_result->get_id( ).  
    MESSAGE 'REST API created with ID: ' && lv_api_id TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
    MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
    MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many.  
CATCH /aws1/cx_agwunauthorizedex INTO DATA(lo_unauthorized).  
    MESSAGE lo_unauthorized->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_unauthorized.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateRestApi](#)」を参照してください。

DeleteRestApi

次の例は、DeleteRestApi を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_agw->deleterestapi(  
    iv_restapiid = iv_rest_api_id ).  
  MESSAGE 'REST API deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
  MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
  MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
  RAISE EXCEPTION lo_not_found.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
  MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
  RAISE EXCEPTION lo_too_many.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteRestApi](#)を参照してください。

GetResources

次の例は、GetResources を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```

oo_result = lo_agw->getresources(
  iv_restapiid = iv_rest_api_id ).
DATA(lt_resources) = oo_result->get_items( ).
DATA(lv_count) = lines( lt_resources ).
MESSAGE 'Found ' && lv_count && ' resources' TYPE 'I'.
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).
MESSAGE lo_bad_request->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_bad_request.
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).
MESSAGE lo_not_found->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_not_found.
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).
MESSAGE lo_too_many->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many.
ENDTRY.

```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[GetResources](#)」を参照してください。

GetRestApis

次の例は、GetRestApis を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  oo_result = lo_agw->getrestapis( ).
  DATA(lt_apis) = oo_result->get_items( ).
  DATA(lv_count) = lines( lt_apis ).
  MESSAGE 'Found ' && lv_count && ' REST APIs' TYPE 'I'.
  CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).
  MESSAGE lo_bad_request->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request.
  CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).

```

```
MESSAGE lo_too_many->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetRestApis](#)」を参照してください。

PutIntegration

次の例は、PutIntegration を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_agw->putintegration(
    iv_restapiid = iv_rest_api_id
    iv_resourceid = iv_resource_id
    iv_httpmethod = iv_http_method
    iv_type = 'AWS_PROXY'
    iv_integrationhttpmethod = 'POST'
    iv_uri = iv_integration_uri ).
  MESSAGE 'Integration configured for method' TYPE 'I'.
  CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).
  MESSAGE lo_bad_request->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request.
  CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).
  MESSAGE lo_not_found->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_not_found.
  CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).
  MESSAGE lo_too_many->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_too_many.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutIntegration](#)」を参照してください。

PutIntegrationResponse

次の例は、PutIntegrationResponse を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->putintegrationresponse(  
        iv_restapiid = iv_rest_api_id  
        iv_resourceid = iv_resource_id  
        iv_httpmethod = iv_http_method  
        iv_statuscode = '200' ).  
    MESSAGE 'Integration response configured for status 200' TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
    MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
    MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
    MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutIntegrationResponse](#)」を参照してください。

PutMethod

次の例は、PutMethod を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->putmethod(  
        iv_restapiid = iv_rest_api_id  
        iv_resourceid = iv_resource_id  
        iv_httpmethod = iv_http_method  
        iv_authorizationtype = 'NONE' ).  
    MESSAGE 'Method ' && iv_http_method && ' added to resource' TYPE 'I'.  
    CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
        MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_bad_request.  
    CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
        MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_not_found.  
    CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
        MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_too_many.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutMethod](#)」を参照してください。

PutMethodResponse

次の例は、PutMethodResponse を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_agw->putmethodresponse(  
        iv_restapiid = iv_rest_api_id  
        iv_resourceid = iv_resource_id  
        iv_httpmethod = iv_http_method  
        iv_statuscode = '200' ).  
    MESSAGE 'Method response configured for status 200' TYPE 'I'.  
CATCH /aws1/cx_agwbadrequestex INTO DATA(lo_bad_request).  
    MESSAGE lo_bad_request->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_agwnotfoundexception INTO DATA(lo_not_found).  
    MESSAGE lo_not_found->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found.  
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).  
    MESSAGE lo_too_many->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutMethodResponse](#)を参照してください。

SDK for SAP ABAP を使用した Application Recovery Controller の例

次のコード例は、Application Recovery Controller で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

GetRoutingControlState

次の例は、GetRoutingControlState を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA lo_exception TYPE REF TO /aws1/cx_rt_generic.
DATA lo_session TYPE REF TO /aws1/cl_rt_session_base.
DATA lo_client TYPE REF TO /aws1/if_r5v.
DATA lt_endpoints TYPE TABLE OF string.
DATA lv_endpoint TYPE string.
DATA lv_region TYPE /aws1/rt_region_id.

" Parse the comma-separated cluster endpoints
" Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-east-1"
SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

" As a best practice, shuffle cluster endpoints to distribute load
" For more information, see https://docs.aws.amazon.com/r53recovery/latest/dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
" For simplicity, we'll try them in order (shuffling can be added if needed)

" Try each endpoint in order
LOOP AT lt_endpoints INTO lv_endpoint.
  TRY.
    " Parse endpoint and region from the format "url|region"
    DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
    IF lv_pos > 0.
      DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
      lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
    ELSE.
      " If no region specified, use default
```

```
        lv_url = lv_endpoint.  
        lv_region = 'us-east-1'.  
    ENDIF.  
  
    " Create session for this region  
    lo_session = /aws1/cl_rt_session_aws=>create( cv_pfl ).  
  
    " Create client with the specific endpoint  
    lo_client = create_recovery_client(  
        iv_endpoint = lv_url  
        iv_region    = lv_region  
        io_session   = lo_session ).  
  
    " Try to get the routing control state  
    oo_result = lo_client->getroutingcontrolstate(  
        iv_routingcontrolarn = iv_routing_control_arn ).  
  
    " If successful, return the result  
    RETURN.  
  
CATCH /aws1/cx_r5vendpttmpyunavailex INTO DATA(lo_endpoint_ex).  
    " This endpoint is temporarily unavailable, try the next one  
    lo_exception = lo_endpoint_ex.  
    CONTINUE.  
  
CATCH /aws1/cx_r5vaccessdeniedex  
        /aws1/cx_r5vinternalserverex  
        /aws1/cx_r5vresourcenotfoundex  
        /aws1/cx_r5vthrottlingex  
        /aws1/cx_r5vvalidationex  
        /aws1/cx_rt_generic INTO lo_exception.  
    " For other errors, re-raise immediately  
    RAISE EXCEPTION lo_exception.  
ENDTRY.  
ENDLOOP.  
  
" If we get here, all endpoints failed - re-raise the last exception  
IF lo_exception IS BOUND.  
    RAISE EXCEPTION lo_exception.  
ENDIF.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetRoutingControlState](#)」を参照してください。

UpdateRoutingControlState

次の例は、UpdateRoutingControlState を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA lo_exception TYPE REF TO /aws1/cx_rt_generic.
DATA lo_session TYPE REF TO /aws1/cl_rt_session_base.
DATA lo_client TYPE REF TO /aws1/if_r5v.
DATA lt_endpoints TYPE TABLE OF string.
DATA lv_endpoint TYPE string.
DATA lv_region TYPE /aws1/rt_region_id.

" Parse the comma-separated cluster endpoints
" Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

" As a best practice, shuffle cluster endpoints to distribute load
" For more information, see https://docs.aws.amazon.com/r53recovery/latest/dg/
route53-arc-best-practices.html#route53-arc-best-practices.regional
" For simplicity, we'll try them in order (shuffling can be added if needed)

" Try each endpoint in order
LOOP AT lt_endpoints INTO lv_endpoint.
  TRY.
    " Parse endpoint and region from the format "url|region"
    DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
    IF lv_pos > 0.
      DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
      lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
    ELSE.
```

```
" If no region specified, use default
lv_url = lv_endpoint.
lv_region = 'us-east-1'.
ENDIF.

" Create session for this region
lo_session = /aws1/cl_rt_session_aws=>create( cv_pfl ).

" Create client with the specific endpoint
lo_client = create_recovery_client(
  iv_endpoint = lv_url
  iv_region   = lv_region
  io_session  = lo_session ).

" Try to update the routing control state
oo_result = lo_client->updateroutingcontrolstate(
  iv_routingcontrolarn      = iv_routing_control_arn
  iv_routingcontrolstate   = iv_routing_control_state
  it_safetyrulestooverride = it_safety_rules_override ).

" If successful, return the result
RETURN.

CATCH /aws1/cx_r5vendpttmpyunavailex INTO DATA(lo_endpoint_ex).
" This endpoint is temporarily unavailable, try the next one
lo_exception = lo_endpoint_ex.
CONTINUE.

CATCH /aws1/cx_r5vaccessdeniedex
      /aws1/cx_r5vconflictexception
      /aws1/cx_r5vinternalserverex
      /aws1/cx_r5vresourcenotfoundex
      /aws1/cx_r5vthrottlingex
      /aws1/cx_r5vvalidationex
      /aws1/cx_rt_generic INTO lo_exception.
" For other errors, re-raise immediately
RAISE EXCEPTION lo_exception.

ENDTRY.
ENDLOOP.

" If we get here, all endpoints failed - re-raise the last exception
IF lo_exception IS BOUND.
  RAISE EXCEPTION lo_exception.
ENDIF.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの [UpdateRoutingControlState](#) を参照してください。

SDK for SAP ABAP を使用した Aurora の例

次のコード例は、Aurora で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateDBClusterParameterGroup

次の例は、CreateDBClusterParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
DATA(lo_output) = lo_rds->createdbclusterparamgroup(  
    iv_dbclusterparamgroupname = iv_param_group_name  
    iv_dbparametergroupfamily = iv_param_group_family  
    iv_description = iv_description
```

```

    ).
    oo_result = lo_output->get_dbclusterparametergroup( ).
  CATCH /aws1/cx_rdsdbparmgralrexfault.
    " Re-raise exception - parameter group already exists
    RAISE EXCEPTION TYPE /aws1/cx_rdsdbparmgralrexfault.
  CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
    " Re-raise exception - quota exceeded
    RAISE EXCEPTION TYPE /aws1/cx_rdsdbprmgrquotaexcd00.
  ENDTRY.

```

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

DeleteDBClusterParameterGroup

次の例は、DeleteDBClusterParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

  TRY.
    lo_rds->deletedbclusterparamgroup(
      iv_dbclusterparamgroupname = iv_param_group_name
    ).
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    " Re-raise exception - parameter group not found
    RAISE EXCEPTION TYPE /aws1/cx_rdsdbprmgrnotfndfault.
  CATCH /aws1/cx_rdsinvdbprmgrstatef00.
    " Re-raise exception - invalid state
    RAISE EXCEPTION TYPE /aws1/cx_rdsinvdbprmgrstatef00.
  ENDTRY.

```

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

DescribeDBClusterParameterGroups

次の例は、DescribeDBClusterParameterGroups を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_output) = lo_rds->describedbclusterparamgroups(  
    iv_dbclusterparamgroupname = iv_param_group_name  
  ).  
  DATA(lt_param_groups) = lo_output->get_dbclusterparametergroups( ).  
  IF lines( lt_param_groups ) > 0.  
    oo_result = lt_param_groups[ 1 ].  
  ENDIF.  
  CATCH /aws1/cx_rdsdbprimgrnotfndfault.  
ENDTRY.
```

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

DescribeDBClusterParameters

次の例は、DescribeDBClusterParameters を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
DATA lv_marker TYPE /aws1/rdsstring VALUE ''.  
DATA lt_all_parameters TYPE /aws1/cl_rdsparameter=>tt_parameterslist.  
  
DO.  
  DATA(lo_output) = lo_rds->describedbclusterparameters(  
    iv_dbclusterparamgroupname = iv_param_group_name  
    iv_source = iv_source  
    iv_marker = lv_marker  
  ).  
  
  LOOP AT lo_output->get_parameters( ) INTO DATA(lo_param).  
    IF iv_name_prefix IS INITIAL OR  
      lo_param->get_parametername( ) CP |{ iv_name_prefix }*|.  
      APPEND lo_param TO lt_all_parameters.  
    ENDIF.  
  ENDLLOOP.  
  
  lv_marker = lo_output->get_marker( ).  
  IF lv_marker IS INITIAL.  
    EXIT.  
  ENDIF.  
ENDDO.  
  
ot_parameters = lt_all_parameters.  
CATCH /aws1/cx_rdsdbprimgrnotfndfault.  
  " Re-raise exception - parameter group not found  
  RAISE EXCEPTION TYPE /aws1/cx_rdsdbprimgrnotfndfault.  
ENDTRY.
```

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

DescribeDBEngineVersions

次の例は、DescribeDBEngineVersions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_engine          = 'mysql'
" iv_dbparametergroupfamily = 'mysql8.0' (optional - filters by parameter group
family)
TRY.
  oo_result = lo_rds->describedbengineversions(
    iv_engine          = iv_engine
    iv_dbparametergroupfamily = iv_dbparametergroupfamily ).
  DATA(lv_version_count) = lines( oo_result->get_dbengineversions( ) ).
  MESSAGE |Retrieved { lv_version_count } engine versions.| TYPE 'I'.
ENDTRY.
```

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

DescribeOrderableDBInstanceOptions

次の例は、DescribeOrderableDBInstanceOptions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_engine          = 'mysql'
" iv_engineversion   = '8.0.35'
TRY.
  oo_result = lo_rds->descrororderabledbinoptions(
```

```

        iv_engine          = iv_engine
        iv_engineversion = iv_engineversion ).
    DATA(lv_option_count) = lines( oo_result->get_orderabledbinoptions( ) ).
    MESSAGE |Retrieved { lv_option_count } orderable DB instance options.| TYPE
'I'.
    ENDTRY.

```

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

ModifyDBClusterParameterGroup

次の例は、ModifyDBClusterParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    oo_result = lo_rds->modifydbclusterparamgroup(
        iv_dbclusterparamgroupname = iv_param_group_name
        it_parameters = it_update_parameters
    ).
CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    " Re-raise exception - parameter group not found
    RAISE EXCEPTION TYPE /aws1/cx_rdsdbprmgrnotfndfault.
CATCH /aws1/cx_rdsinldbprmgrstatef00.
    " Re-raise exception - invalid state
    RAISE EXCEPTION TYPE /aws1/cx_rdsinldbprmgrstatef00.
ENDTRY.

```

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

SDK for SAP ABAP を使用した Auto Scaling の例

次のコード例は、Auto Scaling で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateAutoScalingGroup

次の例は、CreateAutoScalingGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_launch_template TYPE REF TO /aws1/cl_aslaunchtemplatespec.

" Example: iv_group_name = 'my-auto-scaling-group'
" Example: iv_launch_template_name = 'my-launch-template'
" Example: iv_min_size = 1
" Example: iv_max_size = 3
" Example: iv_vpc_zone_identifier = 'subnet-12345,subnet-67890' (for VPC)

TRY.
    " Create launch template specification
```

```

lo_launch_template = NEW /aws1/cl_asclaunchtemplatespec(
  iv_launchtemplatename = iv_launch_template_name
  iv_version = '$Default' ).

" Create the Auto Scaling group
" Use VPCZoneIdentifier for VPC subnets, or AvailabilityZones for EC2-
Classic
IF iv_vpc_zone_identifier IS NOT INITIAL.
  " VPC-based deployment - use subnet IDs
  ao_asc->createautoscalinggroup(
    iv_autoscalinggroupname = iv_group_name
    iv_vpczoneidentifier = iv_vpc_zone_identifier
    io_launchtemplate = lo_launch_template
    iv_minsize = iv_min_size
    iv_maxsize = iv_max_size ).
ELSE.
  " EC2-Classic or default VPC - use availability zones
  ao_asc->createautoscalinggroup(
    iv_autoscalinggroupname = iv_group_name
    it_availabilityzones = it_group_zones
    io_launchtemplate = lo_launch_template
    iv_minsize = iv_min_size
    iv_maxsize = iv_max_size ).
ENDIF.

" Wait for the group to be created (simplified - in production use proper
polling)
WAIT UP TO 10 SECONDS.

MESSAGE 'Auto Scaling group created successfully' TYPE 'I'.

CATCH /aws1/cx_ascalreadyexistsfault INTO DATA(lo_already_exists).
  RAISE EXCEPTION lo_already_exists.
CATCH /aws1/cx_asclimitexceededfault INTO DATA(lo_limit_exceeded).
  RAISE EXCEPTION lo_limit_exceeded.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateAutoScalingGroup](#)」を参照してください。

DeleteAutoScalingGroup

次の例は、DeleteAutoScalingGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'

TRY.
  ao_asc->deleteautoscalinggroup(
    iv_autoscalinggroupname = iv_group_name ).

    " Wait for the group to be deleted (simplified - in production use proper
polling)
    WAIT UP TO 10 SECONDS.

    MESSAGE 'Auto Scaling group deleted successfully' TYPE 'I'.


CATCH /aws1/cx_ascscaactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_ascresourceinusefault INTO DATA(lo_resource_in_use).
  RAISE EXCEPTION lo_resource_in_use.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteAutoScalingGroup](#)を参照してください。

DescribeAutoScalingGroups

次の例は、DescribeAutoScalingGroups を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_group_names TYPE /aws1/
cl_ascautoscgroupnames_w=>tt_autoscalinggroupnames.
DATA lo_group_name TYPE REF TO /aws1/cl_ascautoscgroupnames_w.

" Example: iv_group_name = 'my-auto-scaling-group'

TRY.
  " Build group names parameter
  CREATE OBJECT lo_group_name
    EXPORTING
      iv_value = iv_group_name.
  APPEND lo_group_name TO lt_group_names.

  " Describe the Auto Scaling group
  DATA(lo_output) = ao_asc->describeautoscalinggroups(
    it_autoscalinggroupnames = lt_group_names ).

  " Return the first (and only) group in the result
  DATA(lt_groups) = lo_output->get_autoscalinggroups( ).
  IF lines( lt_groups ) > 0.
    READ TABLE lt_groups INDEX 1 INTO DATA(lo_group).
    oo_output = lo_group.
  ENDIF.

  MESSAGE 'Auto Scaling group information retrieved successfully' TYPE 'I'.

  CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
    RAISE EXCEPTION lo_contention.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
    RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeAutoScalingGroups](#)」を参照してください。

DescribeAutoScalingInstances

次の例は、DescribeAutoScalingInstances を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: it_instance_ids contains a list of instance IDs

TRY.
  DATA(lo_output) = ao_asc->describeautoscalinginstances(
    it_instanceids = it_instance_ids ).

  ot_output = lo_output->get_autoscalinginstances( ).

  MESSAGE 'Auto Scaling instances information retrieved successfully' TYPE
'I'.


CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeAutoScalingInstances](#)」を参照してください。

DescribeScalingActivities

次の例は、DescribeScalingActivities を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'

TRY.
  DATA(lo_output) = ao_asc->describescalingactivities(
    iv_autoscalinggroupname = iv_group_name ).

  ot_output = lo_output->get_activities( ).

  MESSAGE 'Scaling activities retrieved successfully' TYPE 'I'.


CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeScalingActivities](#)」を参照してください。

DisableMetricsCollection

次の例は、DisableMetricsCollection を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'

TRY.
  ao_asc->disablemetricscollection(
    iv_autoscalinggroupname = iv_group_name ).

  MESSAGE 'Metrics collection disabled successfully' TYPE 'I'.

CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DisableMetricsCollection](#)を参照してください。

EnableMetricsCollection

次の例は、EnableMetricsCollection を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'
" Example: it_metrics contains list of metrics like 'GroupMinSize',
'GroupMaxSize', etc.

TRY.
  ao_asc->enablemetricscollection(
    iv_autoscalinggroupname = iv_group_name
    it_metrics = it_metrics
    iv_granularity = '1Minute' ).

  MESSAGE 'Metrics collection enabled successfully' TYPE 'I'.
```

```
CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[EnableMetricsCollection](#)」を参照してください。

SetDesiredCapacity

次の例は、SetDesiredCapacity を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'
" Example: iv_capacity = 2

TRY.
  ao_asc->setdesiredcapacity(
    iv_autoscalinggroupname = iv_group_name
    iv_desiredcapacity = iv_capacity
    iv_honorcooldown = abap_false ).

  MESSAGE 'Desired capacity set successfully' TYPE 'I'.

CATCH /aws1/cx_ascscaactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[SetDesiredCapacity](#)」を参照してください。

TerminateInstanceInAutoScalingGroup

次の例は、`TerminateInstanceInAutoScalingGroup` を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_instance_id = 'i-1234567890abcdef0'
" Example: iv_decrease_capacity = abap_true

TRY.
  DATA(lo_output) = ao_asc->terminateinstinautoscggroup(
    iv_instanceid = iv_instance_id
    iv_shoulddecrementdesiredcap = iv_decrease_capacity ).

  oo_output = lo_output->get_activity( ).

  MESSAGE 'Instance terminated successfully' TYPE 'I'.

CATCH /aws1/cx_ascscaactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[TerminateInstanceInAutoScalingGroup](#)」を参照してください。

UpdateAutoScalingGroup

次の例は、UpdateAutoScalingGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Example: iv_group_name = 'my-auto-scaling-group'
" Example: iv_max_size = 5

TRY.
  ao_asc->updateautoscalinggroup(
    iv_autoscalinggroupname = iv_group_name
    iv_maxsize = iv_max_size
    iv_minsize = iv_min_size ).

  MESSAGE 'Auto Scaling group updated successfully' TYPE 'I'.

CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_ascscaactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[UpdateAutoScalingGroup](#)を参照してください。

SDK for SAP ABAP を使用した Amazon Bedrock ランタイムの例

次のコード例は、Amazon Bedrock ランタイムで AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

各例には完全なソースコードへのリンクが含まれており、コンテキスト内でコードを設定および実行する方法の手順を確認できます。

トピック

- [Anthropic Claude](#)
- [Stable Diffusion](#)

Anthropic Claude

InvokeModel

次のコード例は、Invoke Model API を使用して Anthropic Claude にテキストメッセージを送信する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

Anthropic Claude 2 基盤モデルを呼び出して、テキストを生成します。この例では、一部の NetWeaver バージョンでは使用できない場合がある /US2/CL_JSON の機能を使用しています。

```
"Claude V2 Input Parameters should be in a format like this:
* {
*   "prompt": "\n\nHuman:\nTell me a joke\n\nAssistant:\n",
*   "max_tokens_to_sample": 2048,
*   "temperature": 0.5,
*   "top_k": 250,
*   "top_p": 1.0,
*   "stop_sequences": []
* }

DATA: BEGIN OF ls_input,
      prompt                TYPE string,
      max_tokens_to_sample TYPE /aws1/rt_shape_integer,
      temperature           TYPE /aws1/rt_shape_float,
      top_k                 TYPE /aws1/rt_shape_integer,
```

```

        top_p                TYPE /aws1/rt_shape_float,
        stop_sequences       TYPE /aws1/rt_stringtab,
    END OF ls_input.

    "Leave ls_input-stop_sequences empty.
    ls_input-prompt = |\n\nHuman:\n{ iv_prompt }\n\nAssistant:\n|.
    ls_input-max_tokens_to_sample = 2048.
    ls_input-temperature = '0.5'.
    ls_input-top_k = 250.
    ls_input-top_p = 1.

    "Serialize into JSON with /ui2/cl_json -- this assumes SAP_UI is installed.
    DATA(lv_json) = /ui2/cl_json=>serialize(
        data = ls_input
        pretty_name = /ui2/cl_json=>pretty_mode-low_case ).

    TRY.
        DATA(lo_response) = lo_bdr->invokemodel(
            iv_body = /aws1/cl_rt_util=>string_to_xstring( lv_json )
            iv_modelid = 'anthropic.claude-v2'
            iv_accept = 'application/json'
            iv_contenttype = 'application/json' ).

        "Claude V2 Response format will be:
        * {
        *   "completion": "Knock Knock...",
        *   "stop_reason": "stop_sequence"
        * }
        DATA: BEGIN OF ls_response,
            completion TYPE string,
            stop_reason TYPE string,
        END OF ls_response.

        /ui2/cl_json=>deserialize(
            EXPORTING jsonx = lo_response->get_body( )
            pretty_name = /ui2/cl_json=>pretty_mode-camel_case
            CHANGING data = ls_response ).

        DATA(lv_answer) = ls_response-completion.
        CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

```

```
ENDTRY.
```

Anthropic Claude 2 基盤モデルを呼び出し、L2 高レベルクライアントを使用してテキストを生成します。

```
TRY.
    DATA(lo_bdr_l2_claude) = /aws1/cl_bdr_l2_factory=>create_claude_2( lo_bdr ).
    " iv_prompt can contain a prompt like 'tell me a joke about Java
    programmers'.
    DATA(lv_answer) = lo_bdr_l2_claude->prompt_for_text( iv_prompt ).
    CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
    WRITE / lo_ex->get_text( ).
    WRITE / |Don't forget to enable model access at https://
    console.aws.amazon.com/bedrock/home?#/modelaccess|.

ENDTRY.
```

Anthropic Claude 3 基盤モデルを呼び出し、L2 高レベルクライアントを使用してテキストを生成します。

```
TRY.
    " Choose a model ID from Anthropic that supports the Messages API -
    currently this is
    " Claude v2, Claude v3 and v3.5. For the list of model ID, see:
    " https://docs.aws.amazon.com/bedrock/latest/userguide/model-ids.html

    " for the list of models that support the Messages API see:
    " https://docs.aws.amazon.com/bedrock/latest/userguide/model-parameters-
    anthropic-claude-messages.html
    DATA(lo_bdr_l2_claude) = /aws1/cl_bdr_l2_factory=>create_anthropic_msg_api(
        io_bdr = lo_bdr
        iv_model_id = 'anthropic.claude-3-sonnet-20240229-v1:0' ). " choosing
    Claude v3 Sonnet
    " iv_prompt can contain a prompt like 'tell me a joke about Java
    programmers'.
    DATA(lv_answer) = lo_bdr_l2_claude->prompt_for_text( iv_prompt = iv_prompt
        iv_max_tokens = 100 ).

    CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
    WRITE / lo_ex->get_text( ).
    WRITE / |Don't forget to enable model access at https://
    console.aws.amazon.com/bedrock/home?#/modelaccess|.
```

```
ENDTRY.
```

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

Stable Diffusion

InvokeModel

次のコード例は、Amazon Bedrock で Stability.ai Stable Diffusion XL を呼び出して画像を生成する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

Stable Diffusion で画像を作成します。

```
"Stable Diffusion Input Parameters should be in a format like this:
*  {
*    "text_prompts": [
*      {"text":"Draw a dolphin with a mustache"},
*      {"text":"Make it photorealistic"}
*    ],
*    "cfg_scale":10,
*    "seed":0,
*    "steps":50
*  }
TYPES: BEGIN OF prompt_ts,
        text TYPE /aws1/rt_shape_string,
        END OF prompt_ts.

DATA: BEGIN OF ls_input,
        text_prompts TYPE STANDARD TABLE OF prompt_ts,
        cfg_scale   TYPE /aws1/rt_shape_integer,
```

```

        seed          TYPE /aws1/rt_shape_integer,
        steps        TYPE /aws1/rt_shape_integer,
    END OF ls_input.

APPEND VALUE prompt_ts( text = iv_prompt ) TO ls_input-text_prompts.
ls_input-cfg_scale = 10.
ls_input-seed = 0. "or better, choose a random integer.
ls_input-steps = 50.

DATA(lv_json) = /ui2/cl_json=>serialize(
    data = ls_input
        pretty_name   = /ui2/cl_json=>pretty_mode-low_case ).

TRY.
    DATA(lo_response) = lo_bdr->invokemodel(
        iv_body = /aws1/cl_rt_util=>string_to_xstring( lv_json )
        iv_modelid = 'stability.stable-diffusion-xl-v1'
        iv_accept = 'application/json'
        iv_contenttype = 'application/json' ).

    "Stable Diffusion Result Format:
    *
    * {
    *   "result": "success",
    *   "artifacts": [
    *     {
    *       "seed": 0,
    *       "base64": "iVBORw0KGgoAAAANSUhEUgAAAgAAA...
    *       "finishReason": "SUCCESS"
    *     }
    *   ]
    * }
    TYPES: BEGIN OF artifact_ts,
            seed          TYPE /aws1/rt_shape_integer,
            base64        TYPE /aws1/rt_shape_string,
            finishreason TYPE /aws1/rt_shape_string,
    END OF artifact_ts.

    DATA: BEGIN OF ls_response,
            result TYPE /aws1/rt_shape_string,
            artifacts TYPE STANDARD TABLE OF artifact_ts,
    END OF ls_response.

    /ui2/cl_json=>deserialize(
        EXPORTING jsonx = lo_response->get_body( )

```

```
        pretty_name = /ui2/cl_json=>pretty_mode-camel_case
    CHANGING data = ls_response ).
    IF ls_response-artifacts IS NOT INITIAL.
        DATA(lv_image) =
    cl_http_utility=>if_http_utility~decode_x_base64( ls_response-artifacts[ 1 ]-
base64 ).
        ENDIF.
    CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

Stability.ai Stable Diffusion XL 基盤モデルを呼び出し、L2 高レベルクライアントを使用して画像を生成します。

```
    TRY.
        DATA(lo_bdr_l2_sd) = /aws1/
    cl_bdr_l2_factory=>create_stable_diffusion_xl_1( lo_bdr ).
        " iv_prompt contains a prompt like 'Show me a picture of a unicorn reading
an enterprise financial report'.
        DATA(lv_image) = lo_bdr_l2_sd->text_to_image( iv_prompt ).
        CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
            WRITE / lo_ex->get_text( ).
            WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

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

SDK for SAP ABAP を使用した Amazon Bedrock エージェントランタイムの例

次のコード例は、Amazon Bedrock エージェントランタイムで AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

InvokeAgent

次のコード例は、InvokeAgent を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_result) = lo_bdz->invokeagent(  
  iv_agentid      = iv_agentid  
  iv_agentaliasid = iv_agentaliasid  
  iv_enabletrace  = abap_true  
  iv_sessionid    = CONV #( cl_system_uuid=>create_uuid_c26_static( ) )  
  iv_inputtext    = |Let's play "rock, paper, scissors". I choose rock.| ).  
DATA(lo_stream) = lo_result->get_completion( ).
```

```
TRY.  
  " loop while there are still events in the stream  
  WHILE lo_stream->/aws1/if_rt_stream_reader~data_available( ) = abap_true.  
    DATA(lo_evt) = lo_stream->read( ).  
    " each /AWS1/CL_BDZRESPONSESTREAM_EV event contains exactly one member  
    " all others are INITIAL. For each event, process the non-initial  
    " member if desired  
    IF lo_evt->get_chunk( ) IS NOT INITIAL.  
      " Process a Chunk event  
      DATA(lv_xstr) = lo_evt->get_chunk( )->get_bytes( ).  
      DATA(lv_answer) = /aws1/cl_rt_util=>xstring_to_string( lv_xstr ).  
      " the answer says something like "I chose paper, so you lost"  
    ELSEIF lo_evt->get_files( ) IS NOT INITIAL.  
      " process a Files event if desired  
    ELSEIF lo_evt->get_returncontrol( ) IS NOT INITIAL.  
      " process a ReturnControl event if desired  
    ELSEIF lo_evt->get_trace( ) IS NOT INITIAL.  
      " process a Trace event if desired  
    ENDIF.  
  ENDWHILE.  
  " the stream of events can possibly contain an exception  
  " which will be raised to break the loop  
  " catch /AWS1/CX_BDZACCESSDENIEDEX.  
  " catch /AWS1/CX_BDZINTERNALSERVEREX.  
  " catch /AWS1/CX_BDZMODELNOTREADYEX.  
  " catch /AWS1/CX_BDZVALIDATIONEX.  
  " catch /AWS1/CX_BDZTHROTTLINGEX.  
  " catch /AWS1/CX_BDZDEPENDENCYFAILEDEX.  
  " catch /AWS1/CX_BDZBADGATEWAYEX.  
  " catch /AWS1/CX_BDZRESOURCENOTFOUNDEX.  
  " catch /AWS1/CX_BDZSERVICEQUOTAEXCDEX.  
  " catch /AWS1/CX_BDZCONFLICTEXCEPTION.  
ENDTRY.
```

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

SDK for SAP ABAP を使用した CloudFront の例

次のコード例は、CloudFront で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

ListDistributions

次の例は、ListDistributions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_fnt->listdistributions( ). " oo_result is returned for  
testing purposes. "  
    MESSAGE 'Retrieved list of CloudFront distributions.' TYPE 'I'.  
    CATCH /aws1/cx_fntinvalidargument.  
        MESSAGE 'Invalid argument provided.' 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「[ListDistributions](#)」を参照してください。

UpdateDistribution

次の例は、UpdateDistribution を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Get the current distribution configuration and ETag "
  DATA(lo_distribution_config_result) = lo_fnt->getdistributionconfig( iv_id =
iv_distribution_id ).
  DATA(lo_old_config) = lo_distribution_config_result-
>get_distributionconfig( ).
  DATA(lv_etag) = lo_distribution_config_result->get_etag( ).

  " Create a new distribution config with the updated comment "
  " Since the config object is immutable, we need to create a new one with all
existing values "
  DATA(lo_new_config) = NEW /aws1/cl_fntdistributionconfig(
  iv_callerreference = lo_old_config->get_callerreference( )
  io_aliases = lo_old_config->get_aliases( )
  iv_defaultrootobject = lo_old_config->get_defaultrootobject( )
  io_origins = lo_old_config->get_origins( )
  io_oringroups = lo_old_config->get_oringroups( )
  io_defaultcachebehavior = lo_old_config->get_defaultcachebehavior( )
  io_cachebehaviors = lo_old_config->get_cachebehaviors( )
  io_customerrorresponses = lo_old_config->get_customerrorresponses( )
  iv_comment = iv_comment
  io_logging = lo_old_config->get_logging( )
  iv_priceclass = lo_old_config->get_priceclass( )
  iv_enabled = lo_old_config->get_enabled( )
  io_viewercertificate = lo_old_config->get_viewercertificate( )
  io_restrictions = lo_old_config->get_restrictions( )
  iv_webaclid = lo_old_config->get_webaclid( )
  iv_httpversion = lo_old_config->get_httpversion( )
  iv_isipv6enabled = lo_old_config->get_isipv6enabled( ) ).

  " Update the distribution with the modified configuration "
  lo_fnt->updatedistribution(
  io_distributionconfig = lo_new_config
  iv_id = iv_distribution_id
```

```
        iv_ifmatch = lv_etag ).
    MESSAGE 'CloudFront distribution updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_fntnosuchdistribution.
    MESSAGE 'Distribution does not exist.' TYPE 'E'.
    CATCH /aws1/cx_fntpreconditionfailed.
    MESSAGE 'Precondition failed - ETag mismatch.' TYPE 'E'.
    CATCH /aws1/cx_fntinvalidifmatchvrs.
    MESSAGE 'Invalid If-Match version.' TYPE 'E'.
ENDTRY.
```

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

SDK for SAP ABAP を使用した CloudWatch の例

次のコード例は、CloudWatch で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

DeleteAlarms

次のコード例は、DeleteAlarms を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_cwt->deletealarms(  
        it_alarmnames = it_alarm_names ).  
    MESSAGE 'Alarms deleted.' TYPE 'I'.  
CATCH /aws1/cx_cwtresourcenotfound.  
    MESSAGE 'Resource being accessed is not found.' 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「[DeleteAlarms](#)」を参照してください。

DescribeAlarms

次のコード例は、DescribeAlarms を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cwt->describealarms(                                " oo_result is returned  
for testing purposes. "  
        it_alarmnames = it_alarm_names ).  
    MESSAGE 'Alarms retrieved.' TYPE 'I'.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-  
>av_err_msg }|. 
```

```
MESSAGE lv_error 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「[DescribeAlarms](#)」を参照してください。

DescribeAlarmsForMetric

次の例は、DescribeAlarmsForMetric を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cwt->describealarmsformetric(           " oo_result is  
returned for testing purposes. "  
    iv_namespace   = iv_namespace  
    iv_metricname  = iv_metric_name  
    it_dimensions  = it_dimensions  
    iv_statistic   = iv_statistic  
    iv_period      = iv_period  
    iv_unit        = iv_unit ).  
    MESSAGE 'Alarms for metric retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-  
>av_err_msg }|.  
    MESSAGE lv_error 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「[DescribeAlarmsForMetric](#)」を参照してください。

DisableAlarmActions

次の例は、DisableAlarmActions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Disables actions on the specified alarm. "  
TRY.  
    lo_cwt->disablealarmactions(  
        it_alarmnames = it_alarm_names ).  
    MESSAGE 'Alarm actions disabled.' TYPE 'I'.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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「[DisableAlarmActions](#)」を参照してください。

EnableAlarmActions

次のコード例は、EnableAlarmActions を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

"Enable actions on the specified alarm."
TRY.
  lo_cwt->enablealarmactions(
    it_alarmnames = it_alarm_names ).
  MESSAGE 'Alarm actions enabled.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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「[EnableAlarmActions](#)」を参照してください。

GetMetricStatistics

次の例は、GetMetricStatistics を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  oo_result = lo_cwt->getmetricstatistics(
returned for testing purposes. "
    iv_namespace   = iv_namespace
    iv_metricname  = iv_metric_name
    iv_starttime   = iv_start_time
    iv_endtime     = iv_end_time
    iv_period      = iv_period
    it_statistics  = it_statistics ).
  MESSAGE 'Metric statistics retrieved.' TYPE 'I'.
CATCH /aws1/cx_cwtinparamvalueex.
  MESSAGE 'The specified argument was not valid.' 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「[GetMetricStatistics](#)」を参照してください。

ListMetrics

次の例は、ListMetrics を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
"The following list-metrics example displays the metrics for Amazon CloudWatch."
TRY.
    oo_result = lo_cwt->listmetrics(           " oo_result is returned for
testing purposes. "
    iv_namespace = iv_namespace ).
    DATA(lt_metrics) = oo_result->get_metrics( ).
    MESSAGE 'Metrics retrieved.' TYPE 'I'.
    CATCH /aws1/cx_cwtinvparamvalueex.
    MESSAGE 'The specified argument was not valid.' 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「[ListMetrics](#)」を参照してください。

PutMetricAlarm

次のコード例は、PutMetricAlarm を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  lo_cwt->putmetricalarm(  
    iv_alarmname           = iv_alarm_name  
    iv_comparisonoperator  = iv_comparison_operator  
    iv_evaluationperiods   = iv_evaluation_periods  
    iv_metricname         = iv_metric_name  
    iv_namespace          = iv_namespace  
    iv_statistic          = iv_statistic  
    iv_threshold          = iv_threshold  
    iv_actionsenabled     = iv_actions_enabled  
    iv_alarmdescription   = iv_alarm_description  
    iv_unit               = iv_unit  
    iv_period             = iv_period  
    it_dimensions        = it_dimensions ).  
  MESSAGE 'Alarm created.' TYPE 'I'.  
CATCH /aws1/cx_cwtlimitexceededfault.  
  MESSAGE 'The request processing has exceeded the limit' 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「[PutMetricAlarm](#)」を参照してください。

PutMetricData

次の例は、PutMetricData を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_metricdata TYPE /aws1/cl_cwtmetricdatum=>tt_metricdata.

"Create metric data object.
DATA(lo_metricdatum) = NEW /aws1/cl_cwtmetricdatum(
  iv_metricname = iv_metric_name
  iv_value      = iv_value
  iv_unit       = iv_unit ).

INSERT lo_metricdatum INTO TABLE lt_metricdata.

TRY.
  lo_cwt->putmetricdata(
    iv_namespace = iv_namespace
    it_metricdata = lt_metricdata ).
  MESSAGE 'Metric data added.' TYPE 'I'.
CATCH /aws1/cx_cwtinparamvalueex.
  MESSAGE 'The specified argument was not valid.' TYPE 'E'.
ENDTRY.
```

一連のデータを CloudWatch メトリクスに配置します。

```
DATA lt_metricdata TYPE /aws1/cl_cwtmetricdatum=>tt_metricdata.

"Create metric data object with values and counts.
DATA(lo_metricdatum) = NEW /aws1/cl_cwtmetricdatum(
  iv_metricname = iv_metric_name
  iv_timestamp  = iv_timestamp
  iv_unit       = iv_unit
  it_values     = it_values
  it_counts    = it_counts ).
```

```
INSERT lo_metricdatum INTO TABLE lt_metricdata.

TRY.
  lo_cwt->putmetricdata(
    iv_namespace = iv_namespace
    it_metricdata = lt_metricdata ).
  MESSAGE 'Metric data set added.' TYPE 'I'.
CATCH /aws1/cx_cwtinparamvalueex.
  MESSAGE 'The specified argument was not valid.' 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「[PutMetricData](#)」を参照してください。

シナリオ

アラームの使用を開始

次のコード例は、以下の操作方法を示しています。

- アラームを作成します。
- アラームアクションの無効化。
- アラームの記述。
- アラームの削除。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

"Create an alarm"
```

```

TRY.
  lo_cwt->putmetricalarm(
    iv_alarmname           = iv_alarm_name
    iv_comparisonoperator  = iv_comparison_operator
    iv_evaluationperiods   = iv_evaluation_periods
    iv_metricname          = iv_metric_name
    iv_namespace           = iv_namespace
    iv_statistic           = iv_statistic
    iv_threshold           = iv_threshold
    iv_actionsenabled      = iv_actions_enabled
    iv_alarmdescription    = iv_alarm_description
    iv_unit                = iv_unit
    iv_period              = iv_period
    it_dimensions          = it_dimensions ).
  MESSAGE 'Alarm created' TYPE 'I'.
CATCH /aws1/cx_cwtlimitexceededfault.
  MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.

"Create an ABAP internal table for the created alarm."
lo_alarmname = NEW #( iv_value = iv_alarm_name ).
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."
TRY.
  lo_cwt->disablealarmactions(
    it_alarmnames          = lt_alarmnames ).
  MESSAGE 'Alarm actions disabled' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
  DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception->av_err_code }"
- { lo_disablealarm_exception->av_err_msg }|.
  MESSAGE lv_disablealarm_error TYPE 'E'.
ENDTRY.

"Describe alarm using the same ABAP internal table."
TRY.
  oo_result = lo_cwt->describealarms(
returned for testing purpose "                                " oo_result is
    it_alarmnames          = lt_alarmnames ).
  MESSAGE 'Alarms retrieved' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
  DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
  MESSAGE lv_describealarms_error TYPE 'E'.

```

```
ENDTRY.  
  
"Delete alarm."  
TRY.  
    lo_cwt->deletealarms(  
        it_alarmnames = lt_alarmnames ).  
    MESSAGE 'Alarms deleted' TYPE 'I'.  
    CATCH /aws1/cx_cwtresourcenotfound.  
        MESSAGE 'Resource being access is not found.' TYPE 'E'.  
ENDTRY.
```

- APIの詳細については、「AWS SDK for SAP ABAP API リファレンス」の次のトピックを参照してください。
 - [DeleteAlarms](#)
 - [DescribeAlarms](#)
 - [DisableAlarmActions](#)
 - [PutMetricAlarm](#)

SDK for SAP ABAP を使用した CloudWatch Logs の例

次のコード例は、CloudWatch Logs で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック


- [アクション](#)

アクション

GetQueryResults

次の例は、GetQueryResults を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cwl->getqueryresults(
    iv_queryid = iv_query_id ).

  " Display query status and result count
  DATA(lv_status) = oo_result->get_status( ).
  DATA(lt_results) = oo_result->get_results( ).
  DATA(lv_result_count) = lines( lt_results ).


  MESSAGE |Query status: { lv_status }. Retrieved { lv_result_count } log
event(s).| TYPE 'I'.
  CATCH /aws1/cx_cwlinvalidparameterex.
    MESSAGE 'Invalid parameter.' TYPE 'E'.
  CATCH /aws1/cx_cwlresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
  CATCH /aws1/cx_cwlserviceunavailex.
    MESSAGE 'Service unavailable.' 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[GetQueryResults](#)を参照してください。

StartQuery

次の例は、StartQuery を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_log_group_name = '/aws/lambda/my-function'
  " iv_query_string = 'fields @timestamp, @message | sort @timestamp desc |
limit 20'
  " iv_start_time and iv_end_time must be in Unix epoch milliseconds (ms since
Jan 1, 1970 00:00:00 UTC)
  oo_result = lo_cwl->startquery(
    iv_loggroupname = iv_log_group_name
    iv_starttime    = iv_start_time
    iv_endtime      = iv_end_time
    iv_querystring  = iv_query_string
    iv_limit        = iv_limit ).

  " Display the query ID for tracking
  DATA(lv_query_id) = oo_result->get_queryid( ).
  MESSAGE |Query started successfully with ID: { lv_query_id }| TYPE 'I'.
  CATCH /aws1/cx_cwlinvalidparameterex.
    MESSAGE 'Invalid parameter.' TYPE 'E'.
  CATCH /aws1/cx_cwllimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_cwlmalformedqueryex.
    MESSAGE 'Malformed query.' TYPE 'E'.
  CATCH /aws1/cx_cwlresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
  CATCH /aws1/cx_cwlserviceunavailex.
    MESSAGE 'Service unavailable.' 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[StartQuery](#)を参照してください。

SDK for SAP ABAP を使用した Amazon Cognito ID プロバイダーの例

次のコード例は、Amazon Cognito ID プロバイダーで AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AdminInitiateAuth

次の例は、AdminInitiateAuth を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " Set up authentication parameters  
    DATA(lt_auth_params) = VALUE /aws1/  
c1_cgpathparamstype_w=>tt_authparameterstype(  
    ( VALUE /aws1/c1_cgpathparamstype_w=>ts_authparameterstype_maprow(  
        key = 'USERNAME'  
        value = NEW /aws1/c1_cgpathparamstype_w( iv_user_name ) ) )  
    ( VALUE /aws1/c1_cgpathparamstype_w=>ts_authparameterstype_maprow(  
        key = 'SECRET_KEY'  
        value = NEW /aws1/c1_cgpathparamstype_w( iv_secret_key ) ) ) ) )
```

```
        key = 'PASSWORD'
        value = NEW /aws1/cl_cgpaauthparamstype_w( iv_password ) ) )
    ).

" Add SECRET_HASH if provided
IF iv_secret_hash IS NOT INITIAL.
    INSERT VALUE #(
        key = 'SECRET_HASH'
        value = NEW /aws1/cl_cgpaauthparamstype_w( iv_secret_hash )
    ) INTO TABLE lt_auth_params.
ENDIF.

oo_result = lo_cgp->admininitiateauth(
    iv_userpoolid = iv_user_pool_id
    iv_clientid = iv_client_id
    iv_authflow = 'ADMIN_USER_PASSWORD_AUTH'
    it_authparameters = lt_auth_params
).

DATA(lv_challenge) = oo_result->get_challengename( ).

IF lv_challenge IS INITIAL.
    MESSAGE 'User successfully signed in.' TYPE 'I'.
ELSE.
    MESSAGE |Authentication challenge required: { lv_challenge }.| TYPE 'I'.
ENDIF.

CATCH /aws1/cx_cgpusernotfoundex INTO DATA(lo_user_ex).
    MESSAGE |User { iv_user_name } not found.| TYPE 'E'.


CATCH /aws1/cx_cgpnnotauthorizedex INTO DATA(lo_auth_ex).
    MESSAGE 'Not authorized. Check credentials.' 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[AdminInitiateAuth](#)を参照してください。

AdminRespondToAuthChallenge

次の例は、AdminRespondToAuthChallenge を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Build challenge responses
  DATA(lt_challenge_responses) = VALUE /aws1/
cl_cgpchallengerspstyp00=>tt_challengeresponsestype(
  ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
    key = 'USERNAME'
    value = NEW /aws1/cl_cgpchallengerspstyp00( iv_user_name ) ) )
  ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
    key = 'SOFTWARE_TOKEN_MFA_CODE'
    value = NEW /aws1/cl_cgpchallengerspstyp00( iv_mfa_code ) ) )
  ).

  " Add SECRET_HASH if provided
  IF iv_secret_hash IS NOT INITIAL.
    INSERT VALUE #(
      key = 'SECRET_HASH'
      value = NEW /aws1/cl_cgpchallengerspstyp00( iv_secret_hash )
    ) INTO TABLE lt_challenge_responses.
  ENDIF.

  DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(
    iv_userpoolid = iv_user_pool_id
    iv_clientid = iv_client_id
    iv_challenge_name = 'SOFTWARE_TOKEN_MFA'
    it_challengeresponses = lt_challenge_responses
    iv_session = iv_session
  ).

  oo_auth_result = lo_result->get_authenticationresult( ).

  IF oo_auth_result IS BOUND.
    MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.
  ELSE.
    " Another challenge might be required
```

```
DATA(lv_next_challenge) = lo_result->get_challengename( ).
MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE 'I'.
ENDIF.

CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

CATCH /aws1/cx_cgpxpiredcodeex INTO DATA(lo_expired_ex).
MESSAGE 'MFA code has expired.' TYPE 'E'.

CATCH /aws1/cx_cgpnauthorizedex INTO DATA(lo_auth_ex).
MESSAGE 'Not authorized. Check MFA configuration.' 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[AdminRespondToAuthChallenge](#)」を参照してください。

AssociateSoftwareToken

次の例は、AssociateSoftwareToken を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
DATA(lo_result) = lo_cgp->associatesoftwaretoken(
    iv_session = iv_session
).

ov_secret_code = lo_result->get_secretcode( ).

MESSAGE 'MFA secret code generated successfully.' TYPE 'I'.

CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
MESSAGE 'Session not found or expired.' TYPE 'E'.
```

```
CATCH /aws1/cx_cgpnnotauthorizedex INTO DATA(lo_auth_ex).
  MESSAGE 'Not authorized to associate software token.' 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[AssociateSoftwareToken](#)を参照してください。

ListUsers

次の例は、ListUsers を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_result) = lo_cgp->listusers(
    iv_userpoolid = iv_user_pool_id
  ).

  ot_users = lo_result->get_users( ).

  MESSAGE |Found { lines( ot_users ) } users in the pool.| TYPE 'I'.

CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
  MESSAGE |User pool { iv_user_pool_id } not found.| TYPE 'E'.

CATCH /aws1/cx_cgpnnotauthorizedex INTO DATA(lo_auth_ex).
  MESSAGE 'Not authorized to list users.' 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「[ListUsers](#)」を参照してください。

VerifySoftwareToken

次の例は、VerifySoftwareToken を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_result) = lo_cgp->verifysoftwaretoken(
    iv_session = iv_session
    iv_usercode = iv_user_code
  ).

  ov_status = lo_result->get_status( ).

  IF ov_status = 'SUCCESS'.
    MESSAGE 'MFA token verified successfully.' TYPE 'I'.
  ELSE.
    MESSAGE |MFA verification status: { ov_status }.| TYPE 'I'.
  ENDIF.

  CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
    MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

  CATCH /aws1/cx_cgpenbsoftwaretokmf00 INTO DATA(lo_enabled_ex).
    MESSAGE 'Software token MFA is already enabled.' 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[VerifySoftwareToken](#)を参照してください。

SDK for SAP ABAP を使用した Amazon Comprehend の例

次のコード例は、Amazon Comprehend で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateDocumentClassifier

次の例は、CreateDocumentClassifier を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->createdocumentclassifier(
    iv_documentclassifiername = iv_classifier_name
    iv_languagecode = iv_language_code
    io_inputdataconfig = NEW /aws1/cl_cpddocclifierinpdat00(
      iv_s3uri = iv_training_s3_uri
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_mode = iv_mode
  ).
  MESSAGE 'Document classifier creation started.' TYPE 'I'.
CATCH /aws1/cx_cpdivalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdrresrclimitexcdex.
  MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
```

```
MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdingernalserverex.
MESSAGE 'Internal server error occurred.' 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「[CreateDocumentClassifier](#)」を参照してください。

DeleteDocumentClassifier

次の例は、DeleteDocumentClassifier を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_cpd->deletedocumentclassifier(
        iv_documentclassifierarn = iv_classifier_arn
    ).
    MESSAGE 'Document classifier deleted.' TYPE 'I'.
CATCH /aws1/cx_cpdingvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdrourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cpdrresourceinuseex.
    MESSAGE 'Resource in use.' TYPE 'E'.
CATCH /aws1/cx_cpdingernalserverex.
    MESSAGE 'Internal server error occurred.' 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 [DeleteDocumentClassifier](#) を参照してください。

DescribeDocumentClassificationJob

次の例は、DescribeDocumentClassificationJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cpd->describedocclassificationjob(  
        iv_jobid = iv_job_id  
    ).  
    MESSAGE 'Document classification job described.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdjobnotfoundex.  
    MESSAGE 'Job not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' 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「[DescribeDocumentClassificationJob](#)」を参照してください。

DescribeDocumentClassifier

次の例は、DescribeDocumentClassifier を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cpd->describedocumentclassifier(  
        iv_documentclassifierarn = iv_classifier_arn  
    ).  
    MESSAGE 'Document classifier described.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpresourcefoundex.  
    MESSAGE 'Resource not found.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' 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「[DescribeDocumentClassifier](#)」を参照してください。

DescribeTopicsDetectionJob

次の例は、DescribeTopicsDetectionJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_cpd->describetopicdetectionjob(
  iv_jobid = iv_job_id
).
MESSAGE 'Topics detection job described.' TYPE 'I'.
CATCH /aws1/cx_cpinvalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdjobnotfoundex.
  MESSAGE 'Job not found.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
  MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
  MESSAGE 'Internal server error occurred.' 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「[DescribeTopicsDetectionJob](#)」を参照してください。

DetectDominantLanguage

次の例は、DetectDominantLanguage を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->detectdominantlanguage( iv_text = iv_text ).
  MESSAGE 'Languages detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextrsizeexceedex.
  MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_cpinvalidrequestex.
  MESSAGE 'Invalid request.' 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[DetectDominantLanguage](#)」を参照してください。

DetectEntities

次の例は、DetectEntities を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_cpd->detectentities(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Entities detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrisizelmtexcdex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundsuppedlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinvalidrequestex.  
    MESSAGE 'Invalid request.' 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[DetectEntities](#)」を参照してください。

DetectKeyPhrases

次の例は、DetectKeyPhrases を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_cpd->detectkeyphrases(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Key phrases detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrisizeexceedsex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundisupportedlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinvalidrequestex.  
    MESSAGE 'Invalid request.' 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[DetectKeyPhrases](#)」を参照してください。

DetectPiiEntities

次の例は、DetectPiiEntities を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->detectpiientities(
    iv_text = iv_text
    iv_languagecode = iv_language_code
  ).
  MESSAGE 'PII entities detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextrisizelmtexcdex.
  MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdundsuppedlanguageex.
  MESSAGE 'Unsupported language.' TYPE 'E'.
CATCH /aws1/cx_cpdiinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_cpdivalidrequestex.
  MESSAGE 'Invalid request.' 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[DetectPiiEntities](#)」を参照してください。

DetectSentiment

次の例は、DetectSentiment を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->detectsentiment(
    iv_text = iv_text
    iv_languagecode = iv_language_code
  ).
  MESSAGE 'Sentiment detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextrisizelmtexcdex.
  MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdundsuppedlanguageex.
```

```
MESSAGE 'Unsupported language.' TYPE 'E'.
CATCH /aws1/cx_cpinternalserverex.
MESSAGE 'Internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_cpinvalidrequestex.
MESSAGE 'Invalid request.' 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[DetectSentiment](#)を参照してください。

DetectSyntax

次の例は、DetectSyntax を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->detectsyntax(
    iv_text = iv_text
    iv_languagecode = iv_language_code
  ).
  MESSAGE 'Syntax tokens detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextrsizefmtexcdex.
  MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpunsuppdedlanguageex.
  MESSAGE 'Unsupported language.' TYPE 'E'.
CATCH /aws1/cx_cpinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_cpinvalidrequestex.
  MESSAGE 'Invalid request.' 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[DetectSyntax](#)」を参照してください。

ListDocumentClassificationJobs

次の例は、ListDocumentClassificationJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cpd->listdocclassificationjobs( ).  
    MESSAGE 'Document classification jobs listed.' TYPE 'I'.  
CATCH /aws1/cx_cpdivalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdivalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpdivernalserverex.  
    MESSAGE 'Internal server error occurred.' 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[ListDocumentClassificationJobs](#)」を参照してください。

ListDocumentClassifiers

次の例は、ListDocumentClassifiers を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cpd->listdocumentclassifiers( ).  
    MESSAGE 'Document classifiers listed.' TYPE 'I'.  
CATCH /aws1/cx_cpdivalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdivalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpdivernalserverex.  
    MESSAGE 'Internal server error occurred.' 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[ListDocumentClassifiers](#)」を参照してください。

ListTopicsDetectionJobs

次の例は、ListTopicsDetectionJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_cpd->listtopicsdetectionjobs( ).  
    MESSAGE 'Topics detection jobs listed.' TYPE 'I'.  
ENDTRY.
```

```
CATCH /aws1/cx_cpinvalidrequestex.  
  MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
  MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidfilterex.  
  MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
  MESSAGE 'Internal server error occurred.' 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[ListTopicsDetectionJobs](#)」を参照してください。

StartDocumentClassificationJob

次の例は、StartDocumentClassificationJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_cpd->startdocclassificationjob(  
    iv_jobname = iv_job_name  
    iv_documentclassifierarn = iv_classifier_arn  
    io_inputdataconfig = NEW /aws1/cl_cpinputdataconfig(  
      iv_s3uri = iv_input_s3_uri  
      iv_inputformat = iv_input_format  
    )  
    io_outputdataconfig = NEW /aws1/cl_cpoutputdataconfig(  
      iv_s3uri = iv_output_s3_uri  
    )  
    iv_dataaccessrolearn = iv_data_access_role_arn  
  ).  
  MESSAGE 'Document classification job started.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.
```

```
MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdtresourcenotfoundex.
MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cpdtresourceunavailex.
MESSAGE 'Resource unavailable.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
MESSAGE 'KMS key validation error.' TYPE 'E'.
CATCH /aws1/cx_cpdttoomanytagsex.
MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdtresrclimitexcdex.
MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdtinternalserverex.
MESSAGE 'Internal server error occurred.' 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[StartDocumentClassificationJob](#)を参照してください。

StartTopicsDetectionJob

次の例は、StartTopicsDetectionJobを使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWSコード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_cpd->starttopicsdetectionjob(
    iv_jobname = iv_job_name
    io_inputdataconfig = NEW /aws1/cl_cpdtinputdataconfig(
      iv_s3uri = iv_input_s3_uri
      iv_inputformat = iv_input_format
    )
    io_outputdataconfig = NEW /aws1/cl_cpdtoutputdataconfig(
```

```
        iv_s3uri = iv_output_s3_uri
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
).
MESSAGE 'Topics detection job started.' TYPE 'I'.
CATCH /aws1/cx_cpinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
    MESSAGE 'KMS key validation error.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
    MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdrresrclimitexcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpinternalserverex.
    MESSAGE 'Internal server error occurred.' 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[StartTopicsDetectionJob](#)を参照してください。

AWS Config SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Config。

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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

DeleteConfigRule

次の例は、DeleteConfigRule を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
lo_cfs->deleteconfigrule( iv_rule_name ).  
MESSAGE 'Deleted AWS Config rule.' TYPE 'I'.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteConfigRule](#)を参照してください。

DescribeConfigRules

次の例は、DescribeConfigRules を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_result) = lo_cfs->describeconfigrules(  
  it_configrulenames = VALUE /aws1/cl_cfsconfigrulenames_w=>tt_configrulenames(  
    ( NEW /aws1/cl_cfsconfigrulenames_w( iv_rule_name ) )  
  )  
).  
ot_cfg_rules = lo_result->get_configrules( ).  
MESSAGE 'Retrieved AWS Config rule data.' TYPE 'I'.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeConfigRules](#)」を参照してください。

PutConfigRule

次の例は、PutConfigRule を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create a config rule for S3 bucket public read prohibition
lo_cfs->putconfigrule(
  io_configrule = NEW /aws1/cl_cfssconfigrule(
    iv_configrulename = iv_rule_name
    iv_description = |S3 Public Read Prohibited Bucket Rule|
    io_scope = NEW /aws1/cl_cfssscope(
      it_complianceresourcetypes = VALUE /aws1/
cl_cfsclnresrctypes_w=>tt_complianceresourcetypes(
        ( NEW /aws1/cl_cfsclnresrctypes_w( |AWS::S3::Bucket| ) )
      )
    )
    io_source = NEW /aws1/cl_cfsssource(
      iv_owner = |AWS|
      iv_sourceidentifier = |S3_BUCKET_PUBLIC_READ_PROHIBITED|
    )
    iv_inputparameters = '{}'
    iv_configrulestate = |ACTIVE|
  )
).
MESSAGE 'Created AWS Config rule.' TYPE 'I'.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutConfigRule](#)」を参照してください。

AWS Control Tower SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Control Tower。

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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

DisableBaseline

次の例は、DisableBaseline を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Disable the baseline  
  DATA(lo_output) = io_ctt->disablebaseline(  
    iv_enabledbaselineidentifier = iv_enabled_baseline_identifier  
  ).  
  
  DATA(lv_operation_id) = lo_output->get_operationidentifier( ).  
  
  " Wait for operation to complete  
  DATA lv_status TYPE /aws1/cttbaselineopstatus.
```

```
DO 100 TIMES.
  lv_status = get_baseline_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Baseline operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

  " Wait 30 seconds
  WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Baseline disabled successfully.' TYPE 'I'.
CATCH /aws1/cx_cttconflictexception INTO DATA(lo_conflict).
  " Log conflict but don't fail - return empty operation ID
  DATA(lv_msg2) = |Conflict disabling baseline: { lo_conflict->get_text( ) }.
  Skipping disable step.|.
  MESSAGE lv_msg2 TYPE 'I'.
  CLEAR ov_operation_id.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DisableBaseline](#)を参照してください。

DisableControl

次の例は、DisableControl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Disable the control
DATA(lo_output) = io_ctt->disablecontrol(
  iv_controlidentifier = iv_control_arn
  iv_targetidentifier  = iv_target_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttcontrolopstatus.
DO 100 TIMES.
  lv_status = get_control_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

DATA(lv_msg) = |Control operation status: { lv_status }|.
MESSAGE lv_msg TYPE 'I'.

IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
  EXIT.
ENDIF.

" Wait 30 seconds
WAIT UP TO 30 SECONDS.
ENDDO.


ov_operation_id = lv_operation_id.
MESSAGE 'Control disabled successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DisableControl](#)を参照してください。

EnableBaseline

次の例は、EnableBaseline を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Prepare parameters for enabling baseline
DATA lt_parameters TYPE /aws1/
cl_cttenbdbaselineparam=>tt_enabledbaselineparameters.

" Add Identity Center baseline parameter if provided
IF iv_identity_center_baseline IS NOT INITIAL.
  " Create a JSON document with the baseline ARN value
  DATA(lv_json) = |\{ "IdentityCenterEnabledBaselineArn":
"{ iv_identity_center_baseline }" \}|.
  DATA(lo_param) = NEW /aws1/cl_cttenbdbaselineparam(
    iv_key = 'IdentityCenterEnabledBaselineArn'
    io_value = /aws1/cl_rt_document=>from_json_str( lv_json )
  ).
  APPEND lo_param TO lt_parameters.
ENDIF.

" Enable the baseline
DATA(lo_output) = io_ctt->enablebaseline(
  iv_baselineidentifier = iv_baseline_identifier
  iv_baselineversion    = iv_baseline_version
  iv_targetidentifier   = iv_target_identifier
  it_parameters        = lt_parameters
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttbaselineopstatus.
DO 100 TIMES.
  lv_status = get_baseline_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).
```

```
DATA(lv_msg) = |Baseline operation status: { lv_status }|.
MESSAGE lv_msg TYPE 'I'.

IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
  EXIT.
ENDIF.

" Wait 30 seconds
WAIT UP TO 30 SECONDS.
ENDDO.

ov_enabled_baseline_arn = lo_output->get_arn( ).
MESSAGE 'Baseline enabled successfully.' TYPE 'I'.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[EnableBaseline](#)」を参照してください。

EnableControl

次の例は、EnableControl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Enable the control
DATA(lo_output) = io_ctt->enablecontrol(
  iv_controlidentifier = iv_control_arn
  iv_targetidentifier  = iv_target_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttcontrolopstatus.
DO 100 TIMES.
```

```
lv_status = get_control_operation(  
    io_ctt = io_ctt  
    iv_operation_id = lv_operation_id  
).  
  
DATA(lv_msg) = |Control operation status: { lv_status }|.  
MESSAGE lv_msg TYPE 'I'.  
  
IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.  
    EXIT.  
ENDIF.  
  
" Wait 30 seconds  
WAIT UP TO 30 SECONDS.  
ENDDO.  
  
ov_operation_id = lv_operation_id.  
MESSAGE 'Control enabled successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[EnableControl](#)を参照してください。

GetBaselineOperation

次の例は、GetBaselineOperation を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_output) = io_ctt->getbaselineoperation(  
    iv_operationidentifier = iv_operation_id  
).  
  
ov_status = lo_output->get_baselineoperation( )->get_status( ).
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetBaselineOperation](#)」を参照してください。

GetControlOperation

次の例は、GetControlOperation を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_output) = io_ctt->getcontroloperation(  
    iv_operationidentifier = iv_operation_id  
    ).  
  
ov_status = lo_output->get_controloperation( )->get_status( ).
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetControlOperation](#)」を参照してください。

ListBaselines

次の例は、ListBaselines を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_baselines TYPE /aws1/cl_cttbaselinessummary=>tt_baselines.  
DATA lv_nexttoken TYPE /aws1/cttstring.
```

```

" List all baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listbaselines(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_baselines( ) TO lt_baselines.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_baselines = lt_baselines.
MESSAGE 'Listed baselines successfully.' TYPE 'I'.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListBaselines](#)」を参照してください。

ListEnabledBaselines

次の例は、ListEnabledBaselines を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

DATA lt_enabled_baselines TYPE /aws1/
cl_cttenbdbaselinesumm=>tt_enabledbaselines.
DATA lv_nexttoken TYPE /aws1/cttlstenbdbaselinesnex00.

" List all enabled baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listenabledbaselines(

```

```
        iv_nexttoken = lv_nexttoken
    ).

    APPEND LINES OF lo_output->get_enabledbaselines( ) TO lt_enabled_baselines.

    lv_nexttoken = lo_output->get_nexttoken( ).
    IF lv_nexttoken IS INITIAL.
        EXIT.
    ENDIF.
ENDDO.

ot_enabled_baselines = lt_enabled_baselines.
MESSAGE 'Listed enabled baselines successfully.' TYPE 'I'.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ListEnabledBaselines](#)」を参照してください。

ListEnabledControls

次の例は、ListEnabledControls を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_enabled_controls TYPE /aws1/
cl_cttenabledcontrolsumm=>tt_enabledcontrols.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all enabled controls using pagination
DO.
    DATA(lo_output) = io_ctt->listenabledcontrols(
        iv_targetidentifier = iv_target_identifier
        iv_nexttoken        = lv_nexttoken
    ).
```

```
APPEND LINES OF lo_output->get_enabledcontrols( ) TO lt_enabled_controls.

lv_nexttoken = lo_output->get_nexttoken( ).
IF lv_nexttoken IS INITIAL.
  EXIT.
ENDIF.
ENDDO.

ot_enabled_controls = lt_enabled_controls.
MESSAGE 'Listed enabled controls successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListEnabledControls](#)を参照してください。

ListLandingZones

次の例は、ListLandingZones を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_landing_zones TYPE /aws1/
cl_cttlandingzonessummary=>tt_landingzonessummaries.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all landing zones using pagination
DO.
  DATA(lo_output) = io_ctt->listlandingzones(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_landingzones( ) TO lt_landing_zones.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
```

```
EXIT.  
ENDIF.  
ENDDO.  
  
ot_landing_zones = lt_landing_zones.  
MESSAGE 'Listed landing zones successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListLandingZones](#)」を参照してください。

ResetEnabledBaseline

次の例は、ResetEnabledBaseline を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Reset the enabled baseline  
DATA(lo_output) = io_ctt->resetenabledbaseline(  
  iv_enabledbaselineidentifier = iv_enabled_baseline_identifier  
)  
  
DATA(lv_operation_id) = lo_output->get_operationidentifier( ).  
  
" Wait for operation to complete  
DATA lv_status TYPE /aws1/cttbaselineopstatus.  
DO 100 TIMES.  
  lv_status = get_baseline_operation(  
    io_ctt = io_ctt  
    iv_operation_id = lv_operation_id  
  ).  
  
DATA(lv_msg) = |Baseline operation status: { lv_status }|.  
MESSAGE lv_msg TYPE 'I'.
```

```
IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
  EXIT.
ENDIF.

" Wait 30 seconds
WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Baseline reset successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ResetEnabledBaseline](#)を参照してください。

SDK for SAP ABAP を使用した Firehose の例

次のコード例は、Firehose で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック


- [アクション](#)

アクション

PutRecord

次の例は、PutRecord を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  DATA(lo_record) = NEW /aws1/cl_frhrecord( iv_data = iv_data ).  
  
  DATA(lo_result) = lo_frh->putrecord(  
    iv_deliverystreamname = iv_deliv_stream_name  
    io_record              = lo_record ).  
  
  MESSAGE 'Record sent to Firehose delivery stream.' TYPE 'I'.  
CATCH /aws1/cx_frhresourcenotfoundex.  
  MESSAGE 'Delivery stream not found.' TYPE 'E'.  
CATCH /aws1/cx_frhinvalidargumentex.  
  MESSAGE 'Invalid argument provided.' TYPE 'E'.  
CATCH /aws1/cx_frhserviceunavailex.  
  MESSAGE 'Service temporarily unavailable.' 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「[PutRecord](#)」を参照してください。

PutRecordBatch

次の例は、PutRecordBatch を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_result) = lo_frh->putrecordbatch(
    iv_deliverystreamname = iv_deliv_stream_name
    it_records             = it_records ).

  DATA(lv_failed_count) = lo_result->get_failedputcount( ).

  IF lv_failed_count > 0.
    MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
  ELSE.
    MESSAGE 'All records sent successfully to Firehose delivery stream.' TYPE
'I'.
  ENDIF.
CATCH /aws1/cx_frhresourcenotfoundex.
  MESSAGE 'Delivery stream not found.' TYPE 'E'.
CATCH /aws1/cx_frhinvalidargumentex.
  MESSAGE 'Invalid argument provided.' TYPE 'E'.
CATCH /aws1/cx_frhserviceunavailex.
  MESSAGE 'Service temporarily unavailable.' 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「[PutRecordBatch](#)」を参照してください。

SDK for SAP ABAP を使用した DynamoDB の例

次のコード例は、DynamoDB で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

基本は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

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。

各例には、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。

トピック

- [基本](#)

• [アクション](#)

基本

基本を学ぶ

次のコード例は、以下の操作方法を示しています。

- 映画データを保持できるテーブルを作成します。
- テーブルに1つの映画を入れ、取得して更新する。
- サンプル JSON ファイルから映画データをテーブルに書き込む。
- 特定の年にリリースされた映画を照会する。
- 一定期間内に公開された映画をスキャンします。
- テーブルから映画を削除し、テーブルを削除します。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create an Amazon Dynamo DB table.

TRY.
  DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
  DATA(lo_dyn) = /aws1/cl_dyn_factory=>create( lo_session ).
  DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
    ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                          iv_keytype = 'HASH' ) )
    ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                          iv_keytype = 'RANGE' ) ) ).
  DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
    ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                     iv_attributetype = 'N' ) )
    ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                     iv_attributetype = 'S' ) ) ).
```

```

" Adjust read/write capacities as desired.
DATA(lo_dynprovthroughput) = NEW /aws1/cl_dynprovthroughput(
  iv_readcapacityunits = 5
  iv_writecapacityunits = 5 ).
DATA(oo_result) = lo_dyn->createtable(
  it_keyschema = lt_keyschema
  iv_tablename = iv_table_name
  it_attributedefinitions = lt_attributedefinitions
  io_provisionedthroughput = lo_dynprovthroughput ).
" Table creation can take some time. Wait till table exists before
returning.
lo_dyn->get_waiter( )->tableexists(
  iv_max_wait_time = 200
  iv_tablename      = iv_table_name ).
MESSAGE 'DynamoDB Table' && iv_table_name && 'created.' TYPE 'I'.
" It throws exception if the table already exists.
CATCH /aws1/cx_dynresourceinuseex INTO DATA(lo_resourceinuseex).
  DATA(lv_error) = |"{ lo_resourceinuseex->av_err_code }" -
{ lo_resourceinuseex->av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

" Describe table
TRY.
  DATA(lo_table) = lo_dyn->describetable( iv_tablename = iv_table_name ).
  DATA(lv_tablename) = lo_table->get_table( )->ask_tablename( ).
  MESSAGE 'The table name is ' && lv_tablename TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table does not exist' TYPE 'E'.
ENDTRY.

" Put items into the table.
TRY.
  DATA(lo_resp_putitem) = lo_dyn->putitem(
    iv_tablename = iv_table_name
    it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
  ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
    key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
  ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
    key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) ) )

```

```

        ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
          key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
      ) ).
    lo_resp_putitem = lo_dyn->putitem(
      iv_tablename = iv_table_name
      it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
Wars' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) )
      ) ).
    lo_resp_putitem = lo_dyn->putitem(
      iv_tablename = iv_table_name
      it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Speed' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
      ) ).
    " TYPE REF TO /AWSEX/CL_AWS1_dyn_PUT_ITEM_OUTPUT
    MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
    CATCH /aws1/cx_dyncondalcheckfaile00.
    MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
    CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dyntransactconflictex.
    MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

    " Get item from table.

```

```

TRY.
  DATA(lo_resp_getitem) = lo_dyn->getitem(
    iv_tablename          = iv_table_name
    it_key                = VALUE /aws1/cl_dynattributevalue=>tt_key(
      ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
'1975' ) ) )
    ) ).
  DATA(lt_attr) = lo_resp_getitem->get_item( ).
  DATA(lo_title) = lt_attr[ key = 'title' ]-value.
  DATA(lo_year) = lt_attr[ key = 'year' ]-value.
  DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Query item from table.
TRY.
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
    ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
  DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
    ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
      key = 'year'
      value = NEW /aws1/cl_dyncondition(
        it_attributelist = lt_attributelist
        iv_comparisonoperator = |EQ|
      ) ) ) ).
  DATA(lo_query_result) = lo_dyn->query(
    iv_tablename = iv_table_name
    it_keyconditions = lt_keyconditions ).
  DATA(lt_items) = lo_query_result->get_items( ).
  READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.

```

```

    MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Scan items from table.
TRY.
  DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
  lt_items = lo_scan_result->get_items( ).
  " Read the first item and display the attributes.
  READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Update items from table.
TRY.
  DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
    ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
      key = 'rating' value = NEW /aws1/cl_dynattrvalueupdate(
        io_value = NEW /aws1/cl_dynattributevalue( iv_n = '7.6' )
        iv_action = |PUT| ) ) ) ).
  DATA(lt_key) = VALUE /aws1/cl_dynattributevalue=>tt_key(
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
      key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) )
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
      key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'1980' ) ) ) ) ).
  DATA(lo_resp) = lo_dyn->updateitem(
    iv_tablename      = iv_table_name
    it_key            = lt_key
    it_attributeupdates = lt_attributeupdates ).
  MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.
  CATCH /aws1/cx_dyncondalcheckfaile00.
    MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
  CATCH /aws1/cx_dynresourcenotfoundex.

```

```
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  CATCH /aws1/cx_dyntransactconflictex.
    MESSAGE 'Another transaction is using the item' TYPE 'E'.
  ENDTRY.

" Delete table.
TRY.
  lo_dyn->deletetable( iv_tablename = iv_table_name ).
  lo_dyn->get_waiter( )->tablenotexists(
    iv_max_wait_time = 200
    iv_tablename      = iv_table_name ).
  MESSAGE 'DynamoDB Table deleted.' TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  CATCH /aws1/cx_dynresourceinuseex.
    MESSAGE 'The table cannot be deleted as it is in use' 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。
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)
 - [Scan](#)
 - [UpdateItem](#)

アクション

CreateTable

次のコード例は、CreateTable を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
    ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                          iv_keytype = 'HASH' ) )
    ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                          iv_keytype = 'RANGE' ) ) ).
  DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
    ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                     iv_attributetype = 'N' ) )
    ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                     iv_attributetype = 'S' ) ) ).

  " Adjust read/write capacities as desired.
  DATA(lo_dynprovthroughput) = NEW /aws1/cl_dynprovthroughput(
    iv_readcapacityunits = 5
    iv_writecapacityunits = 5 ).
  oo_result = lo_dyn->createtable(
    it_keyschema = lt_keyschema
    iv_tablename = iv_table_name
    it_attributedefinitions = lt_attributedefinitions
    io_provisionedthroughput = lo_dynprovthroughput ).
  " Table creation can take some time. Wait till table exists before
  returning.
  lo_dyn->get_waiter( )->tableexists(
    iv_max_wait_time = 200
    iv_tablename      = iv_table_name ).
  MESSAGE 'DynamoDB Table' && iv_table_name && 'created.' TYPE 'I'.
  " This exception can happen if the table already exists.
  CATCH /aws1/cx_dynresourceinuseex INTO DATA(lo_resourceinuseex).
  DATA(lv_error) = |"{ lo_resourceinuseex->av_err_code }" -
{ lo_resourceinuseex->av_err_msg }|.
  MESSAGE lv_error 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「[CreateTable](#)」を参照してください。

DeleteItem

次のコード例は、DeleteItem を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    DATA(lo_resp) = lo_dyn->deleteitem(  
        iv_tablename          = iv_table_name  
        it_key                 = it_key_input ).  
    MESSAGE 'Deleted one item.' TYPE 'I'.  
CATCH /aws1/cx_dyncondalcheckfaile00.  
    MESSAGE 'A condition specified in the operation could not be evaluated.'  
TYPE 'E'.  
CATCH /aws1/cx_dynresourcenotfoundex.  
    MESSAGE 'The table or index does not exist' TYPE 'E'.  
CATCH /aws1/cx_dyntransactconflictex.  
    MESSAGE 'Another transaction is using the item' 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「[DeleteItem](#)」を参照してください。

DeleteTable

次のコード例は、DeleteTable を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_dyn->deletetable( iv_tablename = iv_table_name ).  
  " Wait till the table is actually deleted.  
  lo_dyn->get_waiter( )->tablenotexists(  
    iv_max_wait_time = 200  
    iv_tablename      = iv_table_name ).  
  MESSAGE 'Table ' && iv_table_name && ' deleted.' TYPE 'I'.  
CATCH /aws1/cx_dynresourcenotfoundex.  
  MESSAGE 'The table ' && iv_table_name && ' does not exist' TYPE 'E'.  
CATCH /aws1/cx_dynresourceinuseex.  
  MESSAGE 'The table cannot be deleted since it is in use' 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「[DeleteTable](#)」を参照してください。

DescribeTable

次のコード例は、DescribeTable を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_dyn->describetable( iv_tablename = iv_table_name ).  
  DATA(lv_tablename) = oo_result->get_table( )->ask_tablename( ).
```

```
DATA(lv_tablearn) = oo_result->get_table( )->ask_tablearn( ).
DATA(lv_tablestatus) = oo_result->get_table( )->ask_tablestatus( ).
DATA(lv_itemcount) = oo_result->get_table( )->ask_itemcount( ).
MESSAGE 'The table name is ' && lv_tablename
        && '. The table ARN is ' && lv_tablearn
        && '. The tablestatus is ' && lv_tablestatus
        && '. Item count is ' && lv_itemcount TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table ' && lv_tablename && ' does not exist' 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「[DescribeTable](#)」を参照してください。

GetItem

次のコード例は、GetItem を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_item = lo_dyn->getitem(
        iv_tablename      = iv_table_name
        it_key            = it_key ).
    DATA(lt_attr) = oo_item->get_item( ).
    DATA(lo_title) = lt_attr[ key = 'title' ]-value.
    DATA(lo_year) = lt_attr[ key = 'year' ]-value.
    DATA(lo_rating) = lt_attr[ key = 'rating' ]-value.
    MESSAGE 'Movie name is: ' && lo_title->get_s( )
            && 'Movie year is: ' && lo_year->get_n( )
            && 'Moving rating is: ' && lo_rating->get_n( ) TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' 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「[GetItem](#)」を参照してください。

ListTables

次のコード例は、ListTables を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_dyn->listtables( ).  
    " You can loop over the oo_result to get table properties like this.  
    LOOP AT oo_result->get_tablenames( ) INTO DATA(lo_table_name).  
        DATA(lv_tablename) = lo_table_name->get_value( ).  
    ENDLOOP.  
    DATA(lv_tablecount) = lines( oo_result->get_tablenames( ) ).  
    MESSAGE 'Found ' && lv_tablecount && ' tables' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.  
        MESSAGE lv_error 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「[ListTables](#)」を参照してください。

PutItem

次のコード例は、PutItem を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_resp) = lo_dyn->putitem(  
    iv_tablename = iv_table_name  
    it_item      = it_item ).  
  MESSAGE '1 row inserted into DynamoDB Table' && iv_table_name TYPE 'I'.  
CATCH /aws1/cx_dyncondalcheckfaile00.  
  MESSAGE 'A condition specified in the operation could not be evaluated.'  
TYPE 'E'.  
CATCH /aws1/cx_dynresourcenotfoundex.  
  MESSAGE 'The table or index does not exist' TYPE 'E'.  
CATCH /aws1/cx_dyntransactconflictex.  
  MESSAGE 'Another transaction is using the item' 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「[PutItem](#)」を参照してください。

Query

次のコード例は、Query を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Query movies for a given year .
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
  ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_year }| ) ) ).
  DATA(lt_key_conditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
  ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
  key = 'year'
  value = NEW /aws1/cl_dyncondition(
  it_attributelist = lt_attributelist
  iv_comparisonoperator = |EQ|
  ) ) ) ).
  oo_result = lo_dyn->query(
  iv_tablename = iv_table_name
  it_keyconditions = lt_key_conditions ).
  DATA(lt_items) = oo_result->get_items( ).
  "You can loop over the results to get item attributes.
  LOOP AT lt_items INTO DATA(lt_item).
  DATA(lo_title) = lt_item[ key = 'title' ]-value.
  DATA(lo_year) = lt_item[ key = 'year' ]-value.
  ENDLLOOP.
  DATA(lv_count) = oo_result->get_count( ).
  MESSAGE 'Item count is: ' && lv_count TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' 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「[Query](#)」を参照してください。

Scan

次のコード例は、Scan を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Scan movies for rating greater than or equal to the rating specified
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
  ( NEW /aws1/cl_dynattributevalue( iv_n = |[ iv_rating ]| ) ) ).
  DATA(lt_filter_conditions) = VALUE /aws1/
cl_dyncondition=>tt_filterconditionmap(
  ( VALUE /aws1/cl_dyncondition=>ts_filterconditionmap_maprow(
    key = 'rating'
    value = NEW /aws1/cl_dyncondition(
      it_attributelist = lt_attributelist
      iv_comparisonoperator = |GE|
    ) ) ) ).
  oo_scan_result = lo_dyn->scan( iv_tablename = iv_table_name
    it_scanfilter = lt_filter_conditions ).
  DATA(lt_items) = oo_scan_result->get_items( ).
  LOOP AT lt_items INTO DATA(lo_item).
    " You can loop over to get individual attributes.
    DATA(lo_title) = lo_item[ key = 'title' ]-value.
    DATA(lo_year) = lo_item[ key = 'year' ]-value.
  ENDLLOOP.
  DATA(lv_count) = oo_scan_result->get_count( ).
  MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' 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「[Scan](#)」を参照してください。

UpdateItem

次のコード例は、UpdateItem を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_output = lo_dyn->updateitem(  
        iv_tablename      = iv_table_name  
        it_key            = it_item_key  
        it_attributeupdates = it_attribute_updates ).  
    MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.  
CATCH /aws1/cx_dyncondalcheckfaile00.  
    MESSAGE 'A condition specified in the operation could not be evaluated.'  
TYPE 'E'.  
CATCH /aws1/cx_dynresourcenotfoundex.  
    MESSAGE 'The table or index does not exist' TYPE 'E'.  
CATCH /aws1/cx_dyntransactconflictex.  
    MESSAGE 'Another transaction is using the item' 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「[UpdateItem](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon EC2 の例

次のコード例は、Amazon EC2 で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AllocateAddress

次のコード例は、AllocateAddress を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ec2->allocateaddress( iv_domain = 'vpc' ).    " oo_result is  
returned for testing purposes. "  
    MESSAGE 'Allocated an Elastic IP address.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-  
>av_err_msg }|.  
    MESSAGE lv_error 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「[AllocateAddress](#)」を参照してください。

AssociateAddress

次のコード例は、AssociateAddress を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ec2->associateaddress(                                " oo_result is  
returned for testing purposes. "  
        iv_allocationid = iv_allocation_id  
        iv_instanceid = iv_instance_id ).
```

```

    MESSAGE 'Associated an Elastic IP address with an EC2 instance.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
    ENTRY.

```

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

AuthorizeSecurityGroupIngress

次の例は、AuthorizeSecurityGroupIngress を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

" Create IP permissions for SSH access (port 22)
" iv_cidr_ip = '192.0.2.0/24'
DATA lt_ip_permissions TYPE /aws1/cl_ec2ippermission=>tt_ippermissionlist.
DATA(lo_ip_permission) = NEW /aws1/cl_ec2ippermission(
  iv_ipprotocol = 'tcp'
  iv_fromport = 22
  iv_toport = 22
  it_ipranges = VALUE /aws1/cl_ec2iprange=>tt_iprangelist(
    ( NEW /aws1/cl_ec2iprange( iv_cidrip = iv_cidr_ip ) )
  )
).
APPEND lo_ip_permission TO lt_ip_permissions.

TRY.
  oo_result = lo_ec2->authsecuritygroupingress(          " oo_result is
returned for testing purposes. "
  iv_groupid = iv_group_id
  it_ippermissions = lt_ip_permissions ).

```

```
MESSAGE 'Authorized ingress rule for security group.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
MESSAGE lv_error 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[AuthorizeSecurityGroupIngress](#)を参照してください。

CreateKeyPair

次の例は、CreateKeyPair を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
  oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
  " oo_result is returned for testing purposes. "
  MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
MESSAGE lv_error 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「[CreateKeyPair](#)」を参照してください。

CreateSecurityGroup

次のコード例は、CreateSecurityGroup を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
    oo_result = lo_ec2->createsecuritygroup(
        iv_description = 'Security group example'
        iv_groupname = iv_security_group_name
        iv_vpcid = iv_vpc_id ).
    MESSAGE 'Security group created.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
    >av_err_msg }|.
    MESSAGE lv_error 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「[CreateSecurityGroup](#)」を参照してください。

CreateVpc

次の例は、CreateVpc を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_cidr_block = '10.0.0.0/16'
TRY.
```

```

        oo_result = lo_ec2->createvpc( iv_cidrblock = iv_cidr_block ).
oo_result is returned for testing purposes. "
        DATA(lv_vpc_id) = oo_result->get_vpc( )->get_vpcid( ).
        MESSAGE 'Created VPC.' TYPE 'I'.
        CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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「[CreateVpc](#)」を参照してください。

CreateVpcEndpoint

次の例は、CreateVpcEndpoint を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

" iv_vpc_id = 'vpc-abc123'
" iv_service_name = 'com.amazonaws.region.service'
TRY.
    oo_result = lo_ec2->createvpcendpoint(
for testing purposes. "
        iv_vpcid = iv_vpc_id
        iv_servicename = iv_service_name
        it_routetableids = it_route_table_ids ).
    DATA(lv_vpc_endpoint_id) = oo_result->get_vpcendpoint( )-
>get_vpcendpointid( ).
    MESSAGE 'Created VPC endpoint.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[CreateVpcEndpoint](#)」を参照してください。

DeleteKeyPair

次の例は、DeleteKeyPair を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ec2->deletekeypair( iv_keyname = iv_key_name ).  
    MESSAGE 'Amazon EC2 key pair deleted.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.  
    MESSAGE lv_error 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「[DeleteKeyPair](#)」を参照してください。

DeleteSecurityGroup

次のコード例は、DeleteSecurityGroup を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ec2->deletesecuritygroup( iv_groupid = iv_security_group_id ).  
    MESSAGE 'Security group deleted.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.  
    MESSAGE lv_error 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「[DeleteSecurityGroup](#)」を参照してください。

DeleteVpc

次の例は、DeleteVpc を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ec2->deletevpc( iv_vpcid = iv_vpc_id ).  
    MESSAGE 'Deleted VPC.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|. 
```

```
MESSAGE lv_error 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[DeleteVpc](#)」を参照してください。

DeleteVpcEndpoints

次の例は、DeleteVpcEndpoints を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    lo_ec2->deletevpcendpoints( it_vpcendpointids = it_vpc_endpoint_ids ).  
    MESSAGE 'Deleted VPC endpoint(s).' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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[DeleteVpcEndpoints](#)」を参照してください。

DescribeAddresses

次の例は、DescribeAddresses を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_ec2->describeaddresses( ). " oo_result  
is returned for testing purposes. "  
    DATA(lt_addresses) = oo_result->get_addresses( ).  
    MESSAGE 'Retrieved information about Elastic IP addresses.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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「[DescribeAddresses](#)」を参照してください。

DescribeAvailabilityZones

次のコード例は、DescribeAvailabilityZones を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ec2->describeavailabilityzones( ). "  
oo_result is returned for testing purposes. "  
    DATA(lt_zones) = oo_result->get_availabilityzones( ).
```

```
MESSAGE 'Retrieved information about Availability Zones.' TYPE 'I'.

CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.
  MESSAGE lv_error 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「[DescribeAvailabilityZones](#)」を参照してください。

DescribeImages

次の例は、DescribeImages を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_ec2->describeimages( it_imageids = it_image_ids ).
  " oo_result is returned for testing purposes. "
  DATA(lt_images) = oo_result->get_images( ).
  MESSAGE 'Retrieved information about Amazon Machine Images (AMIs).' TYPE
'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.
  MESSAGE lv_error 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「[DescribeImages](#)」を参照してください。

DescribeInstanceTypes

次の例は、DescribeInstanceTypes を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create filters for architecture and instance type patterns
" iv_architecture = 'x86_64'
DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
APPEND NEW /aws1/cl_ec2filter(
  iv_name = 'processor-info.supported-architecture'
  it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
    ( NEW /aws1/cl_ec2valuestringlist_w( iv_architecture ) )
  )
) TO lt_filters.
" Filter for instance type patterns like '*.micro', '*.small'
APPEND NEW /aws1/cl_ec2filter(
  iv_name = 'instance-type'
  it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
    ( NEW /aws1/cl_ec2valuestringlist_w( '*.micro' ) )
    ( NEW /aws1/cl_ec2valuestringlist_w( '*.small' ) )
  )
) TO lt_filters.

TRY.
  oo_result = lo_ec2->describeinstancetypes( it_filters = lt_filters ).
  " oo_result is returned for testing purposes. "
  DATA(lt_instance_types) = oo_result->get_instancetypes( ).
  MESSAGE 'Retrieved information about EC2 instance types.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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「[DescribeInstanceTypes](#)」を参照してください。

DescribeInstances

次の例は、DescribeInstances を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_ec2->describeinstances( ).
    " oo_result
    is returned for testing purposes. "

    " Retrieving details of EC2 instances. "
    DATA: lv_instance_id    TYPE /aws1/ec2string,
           lv_status        TYPE /aws1/ec2instancename,
           lv_instance_type TYPE /aws1/ec2instancetype,
           lv_image_id      TYPE /aws1/ec2string.
    LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
        LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
            lv_instance_id = lo_instance->get_instanceid( ).
            lv_status = lo_instance->get_state( )->get_name( ).
            lv_instance_type = lo_instance->get_instancetype( ).
            lv_image_id = lo_instance->get_imageid( ).
        ENDLOOP.
    ENDLOOP.
    MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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「[DescribeInstances](#)」を参照してください。

DescribeKeyPairs

次のコード例は、DescribeKeyPairs を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ec2->describekeypairs( ). " oo_result  
is returned for testing purposes. "  
    DATA(lt_key_pairs) = oo_result->get_keypairs( ).  
    MESSAGE 'Retrieved information about key pairs.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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「[DescribeKeyPairs](#)」を参照してください。

DescribeRegions

次のコード例は、DescribeRegions を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_ec2->describeregions( ).
    " oo_result
is returned for testing purposes. "
    DATA(lt_regions) = oo_result->get_regions( ).
    MESSAGE 'Retrieved information about Regions.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[DescribeRegions](#)」を参照してください。

DescribeRouteTables

次の例は、DescribeRouteTables を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create filter for VPC ID
" iv_vpc_id = 'vpc-abc123'
DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
APPEND NEW /aws1/cl_ec2filter(
```

```

    iv_name = 'vpc-id'
    it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
      ( NEW /aws1/cl_ec2valuestringlist_w( iv_vpc_id ) )
    )
  ) TO lt_filters.

  TRY.
    oo_result = lo_ec2->describeroutetables( it_filters = lt_filters ).
    " oo_result is returned for testing purposes. "
    DATA(lt_route_tables) = oo_result->get_routetables( ).
    MESSAGE 'Retrieved information about route tables.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[DescribeRouteTables](#)」を参照してください。

DescribeSecurityGroups

次の例は、DescribeSecurityGroups を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

  TRY.
    DATA lt_group_ids TYPE /aws1/cl_ec2groupidstrlist_w=>tt_groupidstringlist.
    APPEND NEW /aws1/cl_ec2groupidstrlist_w( iv_value = iv_group_id ) TO
lt_group_ids.
    oo_result = lo_ec2->describesecuritygroups( it_groupids = lt_group_ids ).
    " oo_result is returned for testing purposes. "
    DATA(lt_security_groups) = oo_result->get_securitygroups( ).
    MESSAGE 'Retrieved information about security groups.' TYPE 'I'.
  TRY.

```

```

CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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「[DescribeSecurityGroups](#)」を参照してください。

MonitorInstances

次のコード例は、MonitorInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
  APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
  lt_instance_ids.

  "Perform dry run"
  TRY.
    " DryRun is set to true. This checks for the required permissions to monitor
the instance without actually making the request. "
    lo_ec2->monitorinstances(
      it_instanceids = lt_instance_ids
      iv_dryrun = abap_true ).
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
required permissions to monitor this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
      MESSAGE 'Dry run to enable detailed monitoring completed.' TYPE 'I'.
    " DryRun is set to false to enable detailed monitoring. "

```

```

    lo_ec2->monitorinstances(
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_false ).
    MESSAGE 'Detailed monitoring enabled.' TYPE 'I'.
    " If the error code returned is `UnauthorizedOperation`, then you don't
    have the required permissions to monitor this instance. "
    ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
        MESSAGE 'Dry run to enable detailed monitoring failed. User does not have
        the permissions to monitor the instance.' TYPE 'E'.
    ELSE.
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
        >av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDIF.
ENDTRY.

```

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

RebootInstances

次のコード例は、RebootInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    DATA lt_instance_ids TYPE /aws1/
    cl_ec2instidstringlist_w=>tt_instanceidstringlist.
    APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
    lt_instance_ids.

    "Perform dry run"
    TRY.
        " DryRun is set to true. This checks for the required permissions to reboot
        the instance without actually making the request. "

```

```
lo_ec2->rebootinstances(  
    it_instanceids = lt_instance_ids  
    iv_dryrun = abap_true ).  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    " If the error code returned is `DryRunOperation`, then you have the  
    required permissions to reboot this instance. "  
    IF lo_exception->av_err_code = 'DryRunOperation'.  
        MESSAGE 'Dry run to reboot instance completed.' TYPE 'I'.  
        " DryRun is set to false to make a reboot request. "  
        lo_ec2->rebootinstances(  
            it_instanceids = lt_instance_ids  
            iv_dryrun = abap_false ).  
        MESSAGE 'Instance rebooted.' TYPE 'I'.  
        " If the error code returned is `UnauthorizedOperation`, then you don't  
        have the required permissions to reboot this instance. "  
        ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.  
            MESSAGE 'Dry run to reboot instance failed. User does not have permissions  
            to reboot the instance.' TYPE 'E'.  
        ELSE.  
            DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
            MESSAGE lv_error TYPE 'E'.  
        ENDIF.  
    ENDTRY.
```

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

ReleaseAddress

次のコード例は、ReleaseAddress を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  lo_ec2->releaseaddress( iv_allocationid = iv_allocation_id ).
  MESSAGE 'Elastic IP address released.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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「[ReleaseAddress](#)」を参照してください。

RunInstances

次のコード例は、RunInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

" Create tags for resource created during instance launch. "
DATA lt_tag specifications TYPE /aws1/
cl_ec2tag specification=>tt_tag specification list.
DATA ls_tag specifications LIKE LINE OF lt_tag specifications.
ls_tag specifications = NEW /aws1/cl_ec2tag specification(
  iv_resource type = 'instance'
  it_tags = VALUE /aws1/cl_ec2tag=>tt_tag list(
    ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
  ) ).
APPEND ls_tag specifications TO lt_tag specifications.

TRY.
  " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
  oo_result = lo_ec2->run instances( " oo_result is
returned for testing purposes. "

```

```

        iv_imageid = iv_ami_id
        iv_instancetype = 't3.micro'
        iv_maxcount = 1
        iv_mincount = 1
        it_tagspecifications = lt_tagspecifications
        iv_subnetid = iv_subnet_id ).
    MESSAGE 'EC2 instance created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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「[RunInstances](#)」を参照してください。

StartInstances

次のコード例は、StartInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
    APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
lt_instance_ids.

    "Perform dry run"
    TRY.
        " DryRun is set to true. This checks for the required permissions to start
the instance without actually making the request. "
        lo_ec2->startinstances(
            it_instanceids = lt_instance_ids

```

```
        iv_dryrun = abap_true ).
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
    required permissions to start this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
        MESSAGE 'Dry run to start instance completed.' TYPE 'I'.
        " DryRun is set to false to start instance. "
        oo_result = lo_ec2->startinstances(          " oo_result is returned for
testing purposes. "
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_false ).
        MESSAGE 'Successfully started the EC2 instance.' TYPE 'I'.
        " If the error code returned is `UnauthorizedOperation`, then you don't
        have the required permissions to start this instance. "
        ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
            MESSAGE 'Dry run to start instance failed. User does not have permissions
to start the instance.' TYPE 'E'.
        ELSE.
            DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
            MESSAGE lv_error TYPE 'E'.
        ENDIF.
    ENDTRY.
```

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

StopInstances

次のコード例は、StopInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
  APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
  lt_instance_ids.

  "Perform dry run"
  TRY.
    " DryRun is set to true. This checks for the required permissions to stop
    the instance without actually making the request. "
    lo_ec2->stopinstances(
      it_instanceids = lt_instance_ids
      iv_dryrun = abap_true ).
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
    required permissions to stop this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
      MESSAGE 'Dry run to stop instance completed.' TYPE 'I'.
      " DryRun is set to false to stop instance. "
      oo_result = lo_ec2->stopinstances( " oo_result is returned for
      testing purposes. "
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_false ).
      MESSAGE 'Successfully stopped the EC2 instance.' TYPE 'I'.
      " If the error code returned is `UnauthorizedOperation`, then you don't
      have the required permissions to stop this instance. "
      ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
        MESSAGE 'Dry run to stop instance failed. User does not have permissions
        to stop the instance.' TYPE 'E'.
      ELSE.
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
        >av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
      ENDIF.
    ENDTRY.
  ENDTRY.

```

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

SDK for SAP ABAP を使用した Amazon ECR の例

次のコード例は、Amazon ECR で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateRepository

次の例は、CreateRepository を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_repository_name = 'my-repository'
  oo_result = lo_ecr->createrepository(
    iv_repositoryname = iv_repository_name ).
  DATA(lv_repository_uri) = oo_result->get_repository( )-
>get_repositoryuri( ).
  MESSAGE |Repository created with URI: { lv_repository_uri }| TYPE 'I'.
CATCH /aws1/cx_ecrrepositoryalrexex.
  " If repository already exists, retrieve it
  DATA lt_repo_names TYPE /aws1/
cl_ecrrepositorynamels00=>tt_repositorynamelist.
```

```

        APPEND NEW /aws1/cl_ecrrepositorynames00( iv_value = iv_repository_name )
    TO lt_repo_names.
    DATA(lo_describe_result) = lo_ecr->describerepositories( it_repositorynames
= lt_repo_names ).
    DATA(lt_repos) = lo_describe_result->get_repositories( ).
    IF lines( lt_repos ) > 0.
        READ TABLE lt_repos INDEX 1 INTO DATA(lo_repo).
        oo_result = NEW /aws1/cl_ecrcrrepositoryrsp( io_repository = lo_repo ).
        MESSAGE |Repository { iv_repository_name } already exists.| TYPE 'I'.
    ENDIF.
ENDTRY.

```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateRepository](#)」を参照してください。

DeleteRepository

次の例は、DeleteRepository を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " iv_repository_name = 'my-repository'
    lo_ecr->deleterepository(
        iv_repositoryname = iv_repository_name
        iv_force = abap_true ).
    MESSAGE |Repository { iv_repository_name } deleted.| TYPE 'I'.
CATCH /aws1/cx_ecrrepositorynotfound.
    MESSAGE 'Repository not found.' TYPE 'I'.
ENDTRY.

```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteRepository](#)」を参照してください。

DescribeImages

次の例は、DescribeImages を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_repository_name = 'my-repository'
  " it_image_ids = VALUE #( ( NEW /aws1/cl_ecrimageidentifier( iv_imagetag =
'latest' ) ) )
  IF it_image_ids IS NOT INITIAL.
    oo_result = lo_ecr->describeimages(
      iv_repositoryname = iv_repository_name
      it_imageids = it_image_ids ).
  ELSE.
    oo_result = lo_ecr->describeimages(
      iv_repositoryname = iv_repository_name ).
  ENDIF.
  DATA(lt_image_details) = oo_result->get_imagedetails( ).
  MESSAGE |Found { lines( lt_image_details ) } images in repository.| TYPE
'I'.
  CATCH /aws1/cx_ecrrepositorynotfoundex.
    MESSAGE 'Repository not found.' TYPE 'I'.
  CATCH /aws1/cx_ecrimagenotfoundex.
    MESSAGE 'Image not found.' TYPE 'I'.
  CATCH /aws1/cx_ecrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeImages](#)」を参照してください。

DescribeRepositories

次の例は、DescribeRepositories を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " it_repository_names = VALUE #( ( NEW /aws1/  
cl_ocrrepositorynames00( iv_value = 'my-repository' ) ) )  
    oo_result = lo_ocr->describerepositories(  
        it_repository_names = it_repository_names ).  
    DATA(lt_repositories) = oo_result->get_repositories( ).  
    MESSAGE |Found { lines( lt_repositories ) } repositories.| TYPE 'I'.  
CATCH /aws1/cx_ocrrepositorynotfound.  
    MESSAGE 'Repository not found.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeRepositories](#)」を参照してください。

GetAuthorizationToken

次の例は、GetAuthorizationToken を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_ecr->getauthorizationtoken( ).
DATA(lt_auth_data) = oo_result->get_authorizationdata( ).
IF lines( lt_auth_data ) > 0.
  READ TABLE lt_auth_data INDEX 1 INTO DATA(lo_auth_data).
  DATA(lv_token) = lo_auth_data->get_authorizationtoken( ).
  MESSAGE 'Authorization token retrieved.' TYPE 'I'.
ENDIF.
CATCH /aws1/cx_ecrserverexception.
  MESSAGE 'Server exception occurred.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetAuthorizationToken](#)を参照してください。

GetRepositoryPolicy

次の例は、GetRepositoryPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_repository_name = 'my-repository'
  oo_result = lo_ecr->getrepositorypolicy(
    iv_repositoryname = iv_repository_name ).
  DATA(lv_policy_text) = oo_result->get_policytext( ).
  MESSAGE 'Repository policy retrieved.' TYPE 'I'.
CATCH /aws1/cx_ecrrepositorynotfound.
  MESSAGE 'Repository not found.' TYPE 'I'.
CATCH /aws1/cx_ecrrepositoryplynot00.
  MESSAGE 'Repository policy not found.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetRepositoryPolicy](#)」を参照してください。

PutLifecyclePolicy

次の例は、PutLifecyclePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
  " iv_repository_name = 'my-repository'
  " iv_lifecycle_policy_text = '{"rules":
[{"rulePriority":1,"description":"Expire images older than 14 days",...}]}'
  lo_ecr->putlifecyclepolicy(
    iv_repositoryname = iv_repository_name
    iv_lifecyclepolicytext = iv_lifecycle_policy_text ).
  MESSAGE |Lifecycle policy set for repository { iv_repository_name }.| TYPE
'I'.
  CATCH /aws1/cx_ecrrepositorynotfoundex.
    MESSAGE 'Repository not found.' TYPE 'I'.
  CATCH /aws1/cx_ecrvalidationex.
    MESSAGE 'Invalid lifecycle policy format.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[PutLifecyclePolicy](#)」を参照してください。

SetRepositoryPolicy

次の例は、SetRepositoryPolicy を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_repository_name = 'my-repository'  
  " iv_policy_text = '{"Version":"2012-10-17",          "Statement":[...]}'  
  lo_ecr->setrepositorypolicy(  
    iv_repositoryname = iv_repository_name  
    iv_policytext = iv_policy_text ).  
  MESSAGE |Policy set for repository { iv_repository_name }.| TYPE 'I'.  
CATCH /aws1/cx_ecrrepositorynotfound.  
  MESSAGE 'Repository not found.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[SetRepositoryPolicy](#)を参照してください。

SDK for SAP ABAP を使用した Amazon EMR の例

次のコード例は、Amazon EMR で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AddJobFlowSteps

次の例は、AddJobFlowSteps を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Build args list for Spark submit
  DATA lt_args TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( 'spark-submit' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( '--deploy-mode' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( 'cluster' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_script_uri ) TO lt_args.
  APPEND LINES OF it_script_args TO lt_args.

  " Create step configuration
  DATA(lo_hadoop_jar_step) = NEW /aws1/cl_emrhadoopjarstepcfg(
    iv_jar = 'command-runner.jar'
    it_args = lt_args
  ).

  DATA(lo_step_config) = NEW /aws1/cl_emrstepconfig(
    iv_name = iv_name
    iv_actiononfailure = 'CONTINUE'
    io_hadoopjarstep = lo_hadoop_jar_step
  ).

  DATA lt_steps TYPE /aws1/cl_emrstepconfig=>tt_stepconfiglist.
  APPEND lo_step_config TO lt_steps.

  DATA(lo_result) = lo_emr->addjobflowsteps(
    iv_jobflowid = iv_cluster_id
    it_steps = lt_steps
  ).
```

```

" Get first step ID
DATA(lt_step_ids) = lo_result->get_stepids( ).
READ TABLE lt_step_ids INDEX 1 INTO DATA(lo_step_id_obj).
IF sy-subrc = 0.
    ov_step_id = lo_step_id_obj->get_value( ).
    MESSAGE |Step added with ID { ov_step_id }| TYPE 'I'.
ENDIF.
CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
DATA(lv_error) = lo_internal_error->if_message~get_text( ).
MESSAGE lv_error 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「[AddJobFlowSteps](#)」を参照してください。

DescribeCluster

次の例は、DescribeCluster を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    oo_result = lo_emr->describecluster(
        iv_clusterid = iv_cluster_id
    ).
    DATA(lo_cluster) = oo_result->get_cluster( ).
    DATA(lv_cluster_name) = lo_cluster->get_name( ).
    MESSAGE |Retrieved cluster information for { lv_cluster_name }| TYPE 'I'.
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
    lv_error = lo_invalid_error->if_message~get_text( ).

```

```
MESSAGE lv_error 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「[DescribeCluster](#)」を参照してください。

DescribeStep

次の例は、DescribeStep を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_emr->describestep(  
        iv_clusterid = iv_cluster_id  
        iv_stepid = iv_step_id  
    ).  
    DATA(lo_step) = oo_result->get_step( ).  
    DATA(lv_step_name) = lo_step->get_name( ).  
    MESSAGE |Retrieved step information for { lv_step_name }| TYPE 'I'.  
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).  
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).  
    MESSAGE lv_error TYPE 'E'.  
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).  
    lv_error = lo_invalid_error->if_message~get_text( ).  
    MESSAGE lv_error 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「[DescribeStep](#)」を参照してください。

ListSteps

次の例は、ListSteps を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_emr->liststeps(  
        iv_clusterid = iv_cluster_id  
    ).  
    DATA(lt_steps) = oo_result->get_steps( ).  
    DATA(lv_step_count) = lines( lt_steps ).  
    MESSAGE |Retrieved { lv_step_count } steps for cluster| TYPE 'I'.  
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).  
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).  
    MESSAGE lv_error TYPE 'E'.  
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).  
    lv_error = lo_invalid_error->if_message~get_text( ).  
    MESSAGE lv_error 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[ListSteps](#)」を参照してください。

RunJobFlow

次の例は、RunJobFlow を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create instances configuration
  DATA(lo_instances) = NEW /aws1/cl_emrjobflowinstsconfig(
    iv_masterinstancetype = 'm5.xlarge'
    iv_slaveinstancetype = 'm5.xlarge'
    iv_instancecount = 3
    iv_keepjobflowalivewhennos00 = iv_keep_alive
    iv_emrmanagedmastersecgroup = iv_primary_sec_grp
    iv_emrmanagedslavesecgroup = iv_secondary_sec_grp
  ).

  DATA(lo_result) = lo_emr->runjobflow(
    iv_name = iv_name
    iv_loguri = iv_log_uri
    iv_releaselabel = 'emr-5.30.1'
    io_instances = lo_instances
    it_steps = it_steps
    it_applications = it_applications
    iv_jobflowrole = iv_job_flow_role
    iv_servicerole = iv_service_role
    iv_ebsrootvolumesize = 10
    iv_visibletoallusers = abap_true
  ).

  ov_cluster_id = lo_result->get_jobflowid( ).
  MESSAGE 'EMR cluster created successfully.' TYPE 'I'.
CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
CATCH /aws1/cx_emrclientexc INTO DATA(lo_client_error).
  lv_error = lo_client_error->if_message~get_text( ).
  MESSAGE lv_error 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[RunJobFlow](#)」を参照してください。

TerminateJobFlows

次の例は、TerminateJobFlows を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA lt_cluster_ids TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.  
  APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_cluster_id ) TO lt_cluster_ids.  
  
  lo_emr->terminatejobflows(  
    it_jobflowids = lt_cluster_ids  
  ).  
  MESSAGE 'EMR cluster terminated successfully.' TYPE 'I'.  
CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).  
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).  
  MESSAGE lv_error 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[TerminateJobFlows](#)」を参照してください。

SDK for SAP ABAP を使用した EventBridge スケジューラの例

次のコード例は、EventBridge スケジューラで AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateSchedule

次の例は、CreateSchedule を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Constants for time calculations
  DATA lv_start_date TYPE /aws1/scdstartdate.
  DATA lv_end_date TYPE /aws1/scdenddate.
  DATA lv_start_timestamp TYPE timestamp.
  DATA lv_end_timestamp TYPE timestamp.
  DATA lv_hours_to_run TYPE i VALUE 1.

  " Get current timestamp
  GET TIME STAMP FIELD lv_start_timestamp.

  " Add 1 hour to the current timestamp using CL_ABAP_TSTMP
  lv_end_timestamp = cl_abap_tstmp=>add(
    tstmp = lv_start_timestamp
    secs = lv_hours_to_run * 3600 ).

  " Convert timestamps to decimal format for AWS API
```

```
lv_start_date = lv_start_timestamp.
lv_end_date = lv_end_timestamp.

" Prepare flexible time window configuration
DATA lo_flexible_time_window TYPE REF TO /aws1/cl_scdflexibletimewindow.
IF iv_use_flexible_time_win = abap_true.
  " iv_use_flexible_time_win = ABAP_TRUE
  " Example: Set MaximumWindowInMinutes to 10 for flexible window
  lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
    iv_mode = 'FLEXIBLE'
    iv_maximumwindowinminutes = 10 ).
ELSE.
  lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
    iv_mode = 'OFF' ).
ENDIF.

" Prepare target configuration
" Example iv_target_arn = 'arn:aws:sqs:us-east-1:123456789012:my-queue'
" Example iv_role_arn = 'arn:aws:iam::123456789012:role/SchedulerRole'
" Example iv_input = '{"message": "Hello from EventBridge Scheduler"}'
DATA(lo_target) = NEW /aws1/cl_scdtarget(
  iv_arn = iv_target_arn
  iv_rolearn = iv_role_arn
  iv_input = iv_input ).

" Set action after completion if needed
DATA lv_action_after_completion TYPE /aws1/scdactionaftercompletion.
IF iv_delete_after_completion = abap_true.
  " iv_delete_after_completion = ABAP_TRUE
  lv_action_after_completion = 'DELETE'.
ELSE.
  lv_action_after_completion = 'NONE'.
ENDIF.

" Create the schedule
" Example iv_name = 'my-schedule'
" Example iv_schedule_expression = 'rate(15 minutes)'
" Example iv_schedule_group_name = 'my-schedule-group'
DATA(lo_result) = lo_scd->createschedule(
  iv_name = iv_name
  iv_scheduleexpression = iv_schedule_expression
  iv_groupname = iv_schedule_group_name
  io_target = lo_target
  io_flexibletimewindow = lo_flexible_time_window
```

```
        iv_startdate = lv_start_date
        iv_enddate = lv_end_date
        iv_actionaftercompletion = lv_action_after_completion ).

        ov_schedule_arn = lo_result->get_scheduledarn( ).
        MESSAGE 'Schedule created successfully.' TYPE 'I'.

        CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
        DATA(lv_error) = |Conflict creating schedule: { lo_conflict_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error creating schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateSchedule](#)」を参照してください。

CreateScheduleGroup

次の例は、CreateScheduleGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
    TRY.
        " Example iv_name = 'my-schedule-group'
        DATA(lo_result) = lo_scd->createschedulegroup(
            iv_name = iv_name ).

        ov_schedule_group_arn = lo_result->get_schedulegrouparn( ).
        MESSAGE 'Schedule group created successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
  DATA(lv_error) = |Conflict creating schedule group: { lo_conflict_ex-
>if_message~get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
  DATA(lv_generic_error) = |Error creating schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
  MESSAGE lv_generic_error TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateScheduleGroup](#)」を参照してください。

DeleteSchedule

次の例は、DeleteSchedule を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Example iv_name = 'my-schedule'
  " Example iv_schedule_group_name = 'my-schedule-group'
  lo_scd->deleteschedule(
    iv_name = iv_name
    iv_groupname = iv_schedule_group_name ).
  MESSAGE 'Schedule deleted successfully.' TYPE 'I'.

  CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
    DATA(lv_error) = |Schedule not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
    DATA(lv_generic_error) = |Error deleting schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
```

```
MESSAGE lv_generic_error TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteSchedule](#)」を参照してください。

DeleteScheduleGroup

次の例は、DeleteScheduleGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Example iv_name = 'my-schedule-group'
  lo_scd->deleteschedulegroup(
    iv_name = iv_name ).
  MESSAGE 'Schedule group deleted successfully.' TYPE 'I'.

  CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
    DATA(lv_error) = |Schedule group not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
    DATA(lv_generic_error) = |Error deleting schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_generic_error TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteScheduleGroup](#)」を参照してください。

AWS Glue SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Glue。

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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateCrawler

次の例は、CreateCrawler を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_crawler_name = 'my-crawler'
  " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
  " iv_database_name = 'my-database'
  " iv_table_prefix = 'test_'
  " iv_s3_target = 's3://example-bucket/data/'

  DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
    ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

  DATA(lo_targets) = NEW /aws1/cl_glucrawlertargets(
```

```
        it_s3targets = lt_s3_targets ).

lo_glu->createcrawler(
    iv_name = iv_crawler_name
    iv_role = iv_role_arn
    iv_databasename = iv_database_name
    iv_tableprefix = iv_table_prefix
    io_targets = lo_targets ).
MESSAGE 'Crawler created successfully.' TYPE 'I'.
CATCH /aws1/cx_glualreadyexistsex.
    MESSAGE 'Crawler already exists.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
    DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
    MESSAGE lv_limit_error 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「[CreateCrawler](#)」を参照してください。

CreateJob

次の例は、CreateJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    " iv_job_name = 'my-etl-job'
    " iv_description = 'ETL job for data transformation'
```

```
" iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
" iv_script_location = 's3://example-bucket/scripts/my-script.py'

DATA(lo_command) = NEW /aws1/cl_glujobcommand(
  iv_name = 'glueet1'
  iv_scriptlocation = iv_script_location
  iv_pythonversion = '3' ).

lo_glu->createjob(
  iv_name = iv_job_name
  iv_description = iv_description
  iv_role = iv_role_arn
  io_command = lo_command
  iv_glueversion = '3.0' ).
MESSAGE 'Job created successfully.' TYPE 'I'.
CATCH /aws1/cx_glualreadyexistsex.
  MESSAGE 'Job already exists.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
  DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
  MESSAGE lv_limit_error 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「[CreateJob](#)」を参照してください。

DeleteCrawler

次の例は、DeleteCrawler を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_crawler_name = 'my-crawler'  
  lo_glu->deletecrawler( iv_name = iv_crawler_name ).  
  MESSAGE 'Crawler deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_gluclawerrunningex.  
  MESSAGE 'Crawler is currently running.' TYPE 'E'.  
CATCH /aws1/cx_gluentitynotfoundex.  
  MESSAGE 'Crawler does not exist.' TYPE 'E'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error TYPE 'E'.  
CATCH /aws1/cx_gluschdrtransingex.  
  MESSAGE 'Scheduler is transitioning.' 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「[DeleteCrawler](#)」を参照してください。

DeleteDatabase

次の例は、DeleteDatabase を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
" iv_database_name = 'my-database'
lo_glu->deletedatabase( iv_name = iv_database_name ).
MESSAGE 'Database deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
MESSAGE 'Database does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
MESSAGE lv_timeout_error 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「[DeleteDatabase](#)」を参照してください。

DeleteJob

次の例は、DeleteJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
" iv_job_name = 'my-etl-job'
lo_glu->deletejob( iv_jobname = iv_job_name ).
MESSAGE 'Job deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
```

```
MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error 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「[DeleteJob](#)」を参照してください。

DeleteTable

次の例は、DeleteTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_database_name = 'my-database'
  " iv_table_name = 'my-table'
  lo_glu->deletetable(
    iv_databasename = iv_database_name
    iv_name = iv_table_name ).
  MESSAGE 'Table deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Table or database does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error 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「[DeleteTable](#)」を参照してください。

GetCrawler

次の例は、GetCrawler を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_crawler_name = 'my-crawler'  
  oo_result = lo_glu->getcrawler( iv_name = iv_crawler_name ).  
  DATA(lo_crawler) = oo_result->get_crawler( ).  
  MESSAGE 'Crawler information retrieved.' TYPE 'I'.  
CATCH /aws1/cx_gluentitynotfoundex.  
  MESSAGE 'Crawler does not exist.' TYPE 'I'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error 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「[GetCrawler](#)」を参照してください。

GetDatabase

次の例は、GetDatabase を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_database_name = 'my-database'  
  oo_result = lo_glu->getdatabase( iv_name = iv_database_name ).  
  DATA(lo_database) = oo_result->get_database( ).  
  MESSAGE 'Database information retrieved.' TYPE 'I'.  
CATCH /aws1/cx_gluentitynotfoundex.  
  MESSAGE 'Database does not exist.' TYPE 'E'.  
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).  
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).  
  MESSAGE lv_invalid_error TYPE 'E'.  
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).  
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).  
  MESSAGE lv_internal_error TYPE 'E'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error 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「[GetDatabase](#)」を参照してください。

GetJobRun

次の例は、GetJobRun を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_job_name = 'my-etl-job'
  " iv_run_id = 'jr_abcd1234567890abcdef1234567890abcdef12345678'
  oo_result = lo_glu->getjobrun(
    iv_jobname = iv_job_name
    iv_runid = iv_run_id ).
  DATA(lo_job_run) = oo_result->get_jobrun( ).
  MESSAGE 'Job run information retrieved.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Job or job run does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error 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「[GetJobRun](#)」を参照してください。

GetJobRuns

次の例は、GetJobRuns を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_job_name = 'my-etl-job'
  oo_result = lo_glu->getjobruns( iv_jobname = iv_job_name ).
  DATA(lt_job_runs) = oo_result->get_jobruns( ).
```

```

    MESSAGE 'Job runs retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error 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「[GetJobRuns](#)」を参照してください。

GetTables

次の例は、GetTables を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " iv_database_name = 'my-database'
  oo_result = lo_glu->gettables( iv_databasename = iv_database_name ).
  DATA(lt_tables) = oo_result->get_tablelist( ).
  MESSAGE 'Tables retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).

```

```
DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
MESSAGE lv_timeout_error 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「[GetTables](#)」を参照してください。

ListJobs

次の例は、ListJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_glu->listjobs( ).
  DATA(lt_job_names) = oo_result->get_jobnames( ).
  MESSAGE 'Job list retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'No jobs found.' TYPE 'I'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error 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「[ListJobs](#)」を参照してください。

StartCrawler

次の例は、StartCrawler を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  " iv_crawler_name = 'my-crawler'  
  lo_glu->startcrawler( iv_name = iv_crawler_name ).  
  MESSAGE 'Crawler started successfully.' TYPE 'I'.  
  CATCH /aws1/cx_gluclawerrunningex.  
  MESSAGE 'Crawler is already running.' TYPE 'I'.  
  CATCH /aws1/cx_gluentitynotfoundex.  
  MESSAGE 'Crawler does not exist.' TYPE 'E'.  
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error 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「[StartCrawler](#)」を参照してください。

StartJobRun

次の例は、StartJobRun を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_job_name = 'my-etl-job'
  " iv_input_database = 'my-database'
  " iv_input_table = 'my-table'
  " iv_output_bucket_url = 's3://example-output-bucket/'

  DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
  lt_arguments = VALUE #(
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--input_database'
      value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_database ) ) )
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--input_table'
      value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--output_bucket_url'
      value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

  DATA(oo_result) = lo_glu->startjobrun(
    iv_jobname = iv_job_name
    it_arguments = lt_arguments ).
  ov_job_run_id = oo_result->get_jobrunid( ).
  MESSAGE 'Job run started successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluconcurrentrunsex00.
  MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
  CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Job does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
```

```
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
CATCH /aws1/cx_gluressrcnumlmtexcdex INTO DATA(lo_limit_ex).
  DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
  MESSAGE lv_limit_error 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「[StartJobRun](#)」を参照してください。

SDK for SAP ABAP を使用した HealthImaging の例

次のコード例は、HealthImaging で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CopyImageSet


次の例は、CopyImageSet を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_source_image_set_id = '1234567890123456789012345678901234567890'
  " iv_source_version_id = '1'
```

```
" iv_destination_image_set_id =
'1234567890123456789012345678901234567890' (optional)
" iv_destination_version_id = '1' (optional)
" iv_force = abap_false
DATA(lo_source_info) = NEW /aws1/cl_migcpsrcimagesetinf00(
  iv_latestversionid = iv_source_version_id ).
DATA(lo_copy_info) = NEW /aws1/cl_migcpimagesetinfmtion(
  io_sourceimageset = lo_source_info ).
IF iv_destination_image_set_id IS NOT INITIAL AND
  iv_destination_version_id IS NOT INITIAL.
  DATA(lo_dest_info) = NEW /aws1/cl_migcopydstimageset(
    iv_imagesetid = iv_destination_image_set_id
    iv_latestversionid = iv_destination_version_id ).
  lo_copy_info = NEW /aws1/cl_migcpimagesetinfmtion(
    io_sourceimageset = lo_source_info
    io_destinationimageset = lo_dest_info ).
ENDIF.
oo_result = lo_mig->copyimageset(
  iv_datastoreid = iv_datastore_id
  iv_sourceimagesetid = iv_source_image_set_id
  io_copyimagesetinformatoin = lo_copy_info
  iv_force = iv_force ).
DATA(lo_dest_props) = oo_result->get_dstimagesetproperties( ).
DATA(lv_new_id) = lo_dest_props->get_imagesetid( ).
MESSAGE |Image set copied with new ID: { lv_new_id }.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migservicequotaexcdex.
  MESSAGE 'Service quota exceeded.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[CopyImageSet](#)」を参照してください。

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

CreateDatastore

次の例は、CreateDatastore を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_name = 'my-datastore-name'  
  oo_result = lo_mig->createdatastore( iv_datastorename = iv_datastore_name ).  
  DATA(lv_datastore_id) = oo_result->get_datastoreid( ).  
  MESSAGE 'Data store created.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict. Data store may already exist.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migservicequotaexcdex.  
  MESSAGE 'Service quota exceeded.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateDatastore](#)」を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

DeleteDatastore

次の例は、DeleteDatastore を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'  
  oo_result = lo_mig->deletedatastore( iv_datastoreid = iv_datastore_id ).  
  MESSAGE 'Data store deleted.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict. Data store may contain resources.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Data store not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteDatastore](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

DeleteImageSet

次の例は、DeleteImageSet を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_image_set_id = '1234567890123456789012345678901234567890'
  oo_result = lo_mig->deleteimageset(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id ).
  MESSAGE 'Image set deleted.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteImageSet](#)」を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetDICOMImportJob

次の例は、GetDICOMImportJob を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_job_id = '12345678901234567890123456789012'
  oo_result = lo_mig->getdicomimportjob(
    iv_datastoreid = iv_datastore_id
    iv_jobid = iv_job_id ).
  DATA(lo_job_props) = oo_result->get_jobproperties( ).
  DATA(lv_job_status) = lo_job_props->get_jobstatus( ).
  MESSAGE |Job status: { lv_job_status }.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Job not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetDICOMImportJob](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetDatastore

次の例は、GetDatastore を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
```

```
" iv_datastore_id = '1234567890123456789012345678901234567890'
oo_result = lo_mig->getdatastore( iv_datastoreid = iv_datastore_id ).
DATA(lo_properties) = oo_result->get_datastoreproperties( ).
DATA(lv_name) = lo_properties->get_datastorename( ).
DATA(lv_status) = lo_properties->get_datastorestatus( ).
MESSAGE 'Data store properties retrieved.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Data store not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetDatastore](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetImageFrame

次の例は、GetImageFrame を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_image_set_id = '1234567890123456789012345678901234567890'
  " iv_image_frame_id = '1234567890123456789012345678901234567890'
  oo_result = lo_mig->getimageframe(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id
    io_imageframeinformation = NEW /aws1/cl_migimageframeinfmktion(
```

```
        iv_imageframeid = iv_image_frame_id ) ).
    DATA(lv_frame_blob) = oo_result->get_imageframeblob( ).
    MESSAGE 'Image frame retrieved.' TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image frame not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetImageFrame](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetImageSet

次の例は、GetImageSet を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    " iv_version_id = '1' (optional)
    IF iv_version_id IS NOT INITIAL.
        oo_result = lo_mig->getimageset(
            iv_datastoreid = iv_datastore_id
            iv_imagesetid = iv_image_set_id
            iv_versionid = iv_version_id ).
```

```
ELSE.  
    oo_result = lo_mig->getimageset(  
        iv_datastoreid = iv_datastore_id  
        iv_imagesetid = iv_image_set_id ).  
ENDIF.  
DATA(lv_state) = oo_result->get_imagesetstate( ).  
MESSAGE |Image set retrieved with state: { lv_state }.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
    MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
    MESSAGE 'Conflict error.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
    MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
    MESSAGE 'Image set not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
    MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
    MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetImageSet](#)を参照してください。

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetImageSetMetadata

次の例は、GetImageSetMetadata を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'  
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'  
    " iv_version_id = '1' (optional)  
    IF iv_version_id IS NOT INITIAL.
```

```
oo_result = lo_mig->getimagesetmetadata(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id
    iv_versionid = iv_version_id ).
ELSE.
oo_result = lo_mig->getimagesetmetadata(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id ).
ENDIF.
DATA(lv_metadata_blob) = oo_result->get_imagesetmetadatablob( ).
MESSAGE 'Image set metadata retrieved.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetImageSetMetadata](#)を参照してください。

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWSコード例リポジトリ](#)での設定と実行の方法を確認してください。

ListDICOMImportJobs

次の例は、ListDICOMImportJobsを使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
```

```
" iv_datastore_id = '1234567890123456789012345678901234567890'
oo_result = lo_mig->listdicomimportjobs( iv_datastoreid = iv_datastore_id ).
DATA(lt_jobs) = oo_result->get_jobsummaries( ).
DATA(lv_count) = lines( lt_jobs ).
MESSAGE |Found { lv_count } DICOM import jobs.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListDICOMImportJobs](#)を参照してください。

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

ListDatastores

次の例は、ListDatastores を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  oo_result = lo_mig->listdatastores( ).
  DATA(lt_datastores) = oo_result->get_datastoresummaries( ).
  DATA(lv_count) = lines( lt_datastores ).
  MESSAGE |Found { lv_count } data stores.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
```

```
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListDatastores](#)」を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

ListImageSetVersions

次の例は、ListImageSetVersions を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '1234567890123456789012345678901234567890'  
  " iv_image_set_id = '1234567890123456789012345678901234567890'  
  oo_result = lo_mig->listimagesetversions(  
    iv_datastoreid = iv_datastore_id  
    iv_imagesetid = iv_image_set_id ).  
  DATA(lt_versions) = oo_result->get_imagesetpropertieslist( ).  
  DATA(lv_count) = lines( lt_versions ).  
  MESSAGE |Found { lv_count } image set versions.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict error.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Image set not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.
```

```
MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListImageSetVersions](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


ListTagsForResource

次の例は、ListTagsForResource を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012: datastore/12345678901234567890123456789012'
  oo_result = lo_mig->listtagsforresource( iv_resourcearn = iv_resource_arn ).
  DATA(lt_tags) = oo_result->get_tags( ).
  DATA(lv_count) = lines( lt_tags ).
  MESSAGE |Found { lv_count } tags for resource.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ListTagsForResource](#)を参照してください。

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

SearchImageSets

次の例は、SearchImageSets を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'  
  oo_result = lo_mig->searchimagesets(  
    iv_datastoreid = iv_datastore_id  
    io_searchcriteria = io_search_criteria ).  
  DATA(lt_imagesets) = oo_result->get_imagesetsmetadatasums( ).  
  DATA(lv_count) = lines( lt_imagesets ).  
  MESSAGE |Found { lv_count } image sets.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict error.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Resource not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[SearchImageSets](#)を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


StartDICOMImportJob

次の例は、StartDICOMImportJob を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_job_name = 'import-job-1'
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_role_arn = 'arn:aws:iam::123456789012:role/ImportJobRole'
  " iv_input_s3_uri = 's3://my-bucket/input/'
  " iv_output_s3_uri = 's3://my-bucket/output/'
  oo_result = lo_mig->startdicomimportjob(
    iv_jobname = iv_job_name
    iv_datastoreid = iv_datastore_id
    iv_dataaccessrolearn = iv_role_arn
    iv_inputs3uri = iv_input_s3_uri
    iv_outputs3uri = iv_output_s3_uri ).
  DATA(lv_job_id) = oo_result->get_jobid( ).
  MESSAGE |DICOM import job started with ID: { lv_job_id }.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migservicequotaexcdex.
  MESSAGE 'Service quota exceeded.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[StartDICOMImportJob](#)」を参照してください。

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TagResource

次の例は、TagResource を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
    " iv_resource_arn = 'arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012'  
    lo_mig->tagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tags = it_tags ).  
    MESSAGE 'Resource tagged successfully.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
    MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
    MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
    MESSAGE 'Resource not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
    MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
    MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[TagResource](#)」を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

UntagResource

次の例は、UntagResource を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.  
    " iv_resource_arn = 'arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012'  
    lo_mig->untagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tagkeys = it_tag_keys ).  
    MESSAGE 'Resource untagged successfully.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
    MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
    MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcefoundex.  
    MESSAGE 'Resource not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
    MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
    MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [UntagResource](#)」を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

UpdateImageSetMetadata

次の例は、UpdateImageSetMetadata を使用する方法を説明しています。

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
  " iv_latest_version_id = '1'
  " iv_force = abap_false
  oo_result = lo_mig->updateimagesetmetadata(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id
    iv_latestversionid = iv_latest_version_id
    io_updateimagesetmetupdates = io_metadata_updates
    iv_force = iv_force ).
  DATA(lv_new_version) = oo_result->get_latestversionid( ).
  MESSAGE |Image set metadata updated to version: { lv_new_version }.| TYPE
'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migservicequotaexcdex.
  MESSAGE 'Service quota exceeded.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの [UpdateImageSetMetadata](#) を参照してください。

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

SDK for SAP ABAP を使用した HealthLake の例

次のコード例は、HealthLake で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateFHIRDatastore

次の例は、CreateFHIRDatastore を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
" iv_datastore_name = 'MyHealthLakeDataStore'  
oo_result = lo_hll->createfhirdatastore(  
    iv_datastorename = iv_datastore_name
```

```

        iv_datastoretypeversion = 'R4'
    ).
    MESSAGE 'Data store created successfully.' TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).
        lv_error = |Internal server error: { lo_internal_ex->av_err_code }-
{ lo_internal_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateFHIRDatastore](#)」を参照してください。

DeleteFHIRDatastore

次の例は、DeleteFHIRDatastore を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->deletefhirdatastore(
        iv_datastoreid = iv_datastore_id
    ).

```

```
    MESSAGE 'Data store deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
{ lo_access_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
    CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
    lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
{ lo_conflict_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_conflict_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteFHIRDatastore](#)を参照してください。

DescribeFHIRDatastore

次の例は、DescribeFHIRDatastore を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->describefhirdatastore(
        iv_datastoreid = iv_datastore_id
    ).
    DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).
    IF lo_datastore_properties IS BOUND.
```

```

        DATA(lv_datastore_name) = lo_datastore_properties->get_datastorename( ).
        DATA(lv_datastore_status) = lo_datastore_properties-
>get_datastorestatus( ).
        MESSAGE 'Data store described successfully.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeFHIRDatastore](#)」を参照してください。

DescribeFHIRExportJob

次の例は、DescribeFHIRExportJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->describefhirexportjob(
        iv_datastoreid = iv_datastore_id
        iv_jobid = iv_job_id
    ).
    DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).

```

```

IF lo_export_job_properties IS BOUND.
  DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).
  MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.
ENDIF.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
  DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_notfound_ex.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeFHIRExportJob](#)」を参照してください。

DescribeFHIRImportJob

次の例は、DescribeFHIRImportJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  oo_result = lo_hll->describefhirimportjob(
    iv_datastoreid = iv_datastore_id
    iv_jobid = iv_job_id
  ).
  DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
  IF lo_import_job_properties IS BOUND.

```

```

        DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
        MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeFHIRImportJob](#)」を参照してください。

ListFHIRDatastores

次の例は、ListFHIRDatastores を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    TRY.
        oo_result = lo_hll->listfhirdatastores( ).
        DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
        DATA(lv_datastore_count) = lines( lt_datastores ).
        MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    TRY.

```

```

CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
  lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_throttling_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListFHIRDatastores](#)」を参照してください。

ListFHIRExportJobs

次の例は、ListFHIRExportJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  IF iv_submitted_after IS NOT INITIAL.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
      iv_submittedafter = iv_submitted_after
    ).
  ELSE.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
    ).
  ENDIF.
  DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
  DATA(lv_job_count) = lines( lt_export_jobs ).
  MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.

```

```
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
  lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListFHIRExportJobs](#)」を参照してください。

ListFHIRImportJobs

次の例は、ListFHIRImportJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  IF iv_submitted_after IS NOT INITIAL.
    oo_result = lo_hll->listfhirimportjobs(
      iv_datastoreid = iv_datastore_id
      iv_submittedafter = iv_submitted_after
    ).
  ELSE.
    oo_result = lo_hll->listfhirimportjobs(
      iv_datastoreid = iv_datastore_id
    ).
  ENDIF.
  DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).
  DATA(lv_job_count) = lines( lt_import_jobs ).
  MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
```

```
DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListFHIRImportJobs](#)を参照してください。

ListTagsForResource

次の例は、ListTagsForResource を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  DATA(lo_result) = lo_hll->listtagsforresource(
    iv_resourcearn = iv_resource_arn
  ).
  ot_tags = lo_result->get_tags( ).
  DATA(lv_tag_count) = lines( ot_tags ).
  MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
```

```

    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
      lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListTagsForResource](#)を参照してください。

StartFHIRExportJob

次の例は、StartFHIRExportJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " iv_job_name = 'MyExportJob'
  " iv_output_s3_uri = 's3://my-bucket/export/output/'
  " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
  " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
  oo_result = lo_hll->startfhirexportjob(
    iv_jobname = iv_job_name
    io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
      io_s3configuration = NEW /aws1/cl_hlls3configuration(
        iv_s3uri = iv_output_s3_uri
        iv_kmskeyid = iv_kms_key_id
      )
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_datastoreid = iv_datastore_id
  ).

```

```

    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[StartFHIRExportJob](#)を参照してください。

StartFHIRImportJob

次の例は、StartFHIRImportJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " iv_job_name = 'MyImportJob'
    " iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'
    " iv_job_output_s3_uri = 's3://my-bucket/import/output/'
    " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'

```

```

    " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeImportRole'
    oo_result = lo_hll->startfhirimportjob(
        iv_jobname = iv_job_name
        io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =
iv_input_s3_uri )
        io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
            io_s3configuration = NEW /aws1/cl_hlls3configuration(
                iv_s3uri = iv_job_output_s3_uri
                iv_kmskeyid = iv_kms_key_id
            )
        )
        iv_dataaccessrolearn = iv_data_access_role_arn
        iv_datastoreid = iv_datastore_id
    ).
    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
ENDTRY.


```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[StartFHIRImportJob](#)を参照してください。

TagResource

次の例は、TagResource を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    lo_hll->tagresource(
        iv_resourcearn = iv_resource_arn
        it_tags = it_tags
    ).
    MESSAGE 'Resource tagged successfully.' TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[TagResource](#)」を参照してください。

UntagResource

次の例は、UntagResource を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/  
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
    lo_hll->untagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tagkeys = it_tag_keys  
    ).  
    MESSAGE 'Resource untagged successfully.' TYPE 'I'.  
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [UntagResource](#)」を参照してください。

SDK for SAP ABAP を使用した IAM の例

次のコード例は、IAM で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AttachRolePolicy

次の例は、AttachRolePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->attachrolepolicy(  
    iv_rolename = iv_role_name  
    iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy attached to role successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Role or policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
  MESSAGE 'Policy attachment limit exceeded.' 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「[AttachRolePolicy](#)」を参照してください。

AttachUserPolicy

次の例は、AttachUserPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->attachuserpolicy(  
    iv_username = iv_user_name  
    iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy attached to user successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'User or policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
  MESSAGE 'Policy attachment limit exceeded.' 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「[AttachUserPolicy](#)」を参照してください。

CreateAccessKey

次の例は、CreateAccessKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_iam->createaccesskey(  
    iv_username = iv_user_name ).  
  MESSAGE 'Access key created successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'User does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Maximum number of access keys reached.' 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「[CreateAccessKey](#)」を参照してください。

CreateAccountAlias

次の例は、CreateAccountAlias を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_iam->createaccountalias(  
        iv_accountalias = iv_account_alias ).  
    MESSAGE 'Account alias created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'Account alias already exists.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Account alias limit exceeded.' 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「[CreateAccountAlias](#)」を参照してください。

CreatePolicy

次の例は、CreatePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->createpolicy(  
        iv_policyname = iv_policy_name  
        iv_policydocument = iv_policy_document  
        iv_description = iv_description ).  
    MESSAGE 'Policy created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'Policy already exists.' TYPE 'E'.  
CATCH /aws1/cx_iammalformedplydocex.  
    MESSAGE 'Policy document is malformed.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Policy limit exceeded.' 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「[CreatePolicy](#)」を参照してください。

CreatePolicyVersion

次の例は、CreatePolicyVersion を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->createpolicyversion(  

```

```
    iv_policyarn = iv_policy_arn
    iv_policydocument = iv_policy_document
    iv_setasdefault = iv_set_as_default ).
MESSAGE 'Policy version created successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_iammalformedplydocex.
MESSAGE 'Policy document is malformed.' TYPE 'E'.
CATCH /aws1/cx_iamlimitexceeddex.
MESSAGE 'Policy version limit exceeded.' 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「[CreatePolicyVersion](#)」を参照してください。

CreateRole

次の例は、CreateRole を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_iam->createrole(
        iv_rolename = iv_role_name
        iv_assumerolepolicydocument = iv_assume_role_policy_document ).
MESSAGE 'Role created successfully.' TYPE 'I'.
CATCH /aws1/cx_iamentityalrddyexex.
MESSAGE 'Role already exists.' TYPE 'E'.
CATCH /aws1/cx_iammalformedplydocex.
MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
CATCH /aws1/cx_iamlimitexceeddex.
MESSAGE 'Role limit exceeded.' 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「[CreateRole](#)」を参照してください。

CreateServiceLinkedRole

次の例は、CreateServiceLinkedRole を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listpolicyversions(  
        iv_policyarn = iv_policy_arn ).  
    MESSAGE 'Retrieved policy versions list.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing policy versions.' 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「[CreateServiceLinkedRole](#)」を参照してください。

CreateUser

次の例は、CreateUser を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->createuser(  
        iv_username = iv_user_name ).  
    MESSAGE 'User created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'User already exists.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceedex.  
    MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Entity does not exist.' 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「[CreateUser](#)」を参照してください。

DeleteAccessKey

次の例は、DeleteAccessKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_iam->deleteaccesskey(  
        iv_accesskeyid = iv_access_key_id  
        iv_username = iv_user_name ).  
    MESSAGE 'Access key deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Access key or user does not exist.' 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「DeleteAccessKey」を参照してください。

DeleteAccountAlias

次の例は、DeleteAccountAlias を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->deleteaccountalias(  
    iv_accountalias = iv_account_alias ).  
  MESSAGE 'Account alias deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Account alias does not exist.' 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「[DeleteAccountAlias](#)」を参照してください。

DeletePolicy

次の例は、DeletePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->deletepolicy( iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy deleted successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
  MESSAGE 'Policy cannot be deleted due to attachments.' 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「[DeletePolicy](#)」を参照してください。

DeletePolicyVersion

次の例は、DeletePolicyVersion を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->deletepolicyversion(  
    iv_policyarn = iv_policy_arn  
    iv_versionid = iv_version_id ).  
  MESSAGE 'Policy version deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Policy or version does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
  MESSAGE 'Cannot delete default policy version.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceedex.  
  MESSAGE 'Limit exceeded.' 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「[DeletePolicyVersion](#)」を参照してください。

DeleteRole

次の例は、DeleteRole を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_iam->deleterole( iv_rolename = iv_role_name ).  
    MESSAGE 'Role deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
    MESSAGE 'Role cannot be deleted due to attached resources.' 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「[DeleteRole](#)」を参照してください。

DeleteUser

次の例は、DeleteUser を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_iam->deleteuser( iv_username = iv_user_name ).
```

```
MESSAGE 'User deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'User does not exist.' TYPE 'E'.
CATCH /aws1/cx_iamdeleteconflictex.
MESSAGE 'User cannot be deleted due to attached resources.' 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「[DeleteUser](#)」を参照してください。

DetachRolePolicy

次の例は、DetachRolePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_iam->detachrolepolicy(
    iv_rolename = iv_role_name
    iv_policyarn = iv_policy_arn ).
MESSAGE 'Policy detached from role successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Role or policy does not exist.' 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「[DetachRolePolicy](#)」を参照してください。

DetachUserPolicy

次の例は、DetachUserPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->detachuserpolicy(  
    iv_username = iv_user_name  
    iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy detached from user successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'User or policy does not exist.' 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「[DetachUserPolicy](#)」を参照してください。

GenerateCredentialReport

次の例は、GenerateCredentialReport を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_iam->generatecredentialreport( ).  
  MESSAGE 'Credential report generation started.' TYPE 'I'.  
CATCH /aws1/cx_iamlimitexceededex.  
  MESSAGE 'Report generation limit exceeded.' TYPE 'E'.  
CATCH /aws1/cx_iamservicefailureex.
```

```
MESSAGE 'Service failure when generating credential report.' 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「[GenerateCredentialReport](#)」を参照してください。

GetAccessKeyLastUsed

次の例は、GetAccessKeyLastUsed を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getaccesskeylastused(  
        iv_accesskeyid = iv_access_key_id ).  
    MESSAGE 'Retrieved access key last used information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Access key does not exist.' 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「[GetAccessKeyLastUsed](#)」を参照してください。

GetAccountAuthorizationDetails

次の例は、GetAccountAuthorizationDetails を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getaccountauthdetails( ).  
    MESSAGE 'Retrieved account authorization details.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting account authorization details.' 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「[GetAccountAuthorizationDetails](#)」を参照してください。

GetAccountPasswordPolicy

次の例は、GetAccountPasswordPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getaccountpasswordpolicy( ).  
    MESSAGE 'Retrieved account password policy.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'No password policy exists.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting password policy.' 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「[GetAccountPasswordPolicy](#)」を参照してください。

GetAccountSummary

次の例は、GetAccountSummary を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getaccountsummary( ).  
    MESSAGE 'Retrieved account summary.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting account summary.' 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「[GetAccountSummary](#)」を参照してください。

GetCredentialReport

次の例は、GetCredentialReport を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getcredentialreport( ).  
    MESSAGE 'Retrieved credential report.' TYPE 'I'.  
CATCH /aws1/cx_iamcredrptnotpresen00.  
    MESSAGE 'Credential report not present.' TYPE 'E'.  
CATCH /aws1/cx_iamcredrptexpiredex.  
    MESSAGE 'Credential report expired.' TYPE 'E'.  
CATCH /aws1/cx_iamcredrptnotreadyex.  
    MESSAGE 'Credential report not ready.' TYPE 'E'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting credential report.' 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「[GetCredentialReport](#)」を参照してください。

GetPolicy

次の例は、GetPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getpolicy( iv_policyarn = iv_policy_arn ).  
    MESSAGE 'Retrieved policy information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Policy does not exist.' 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「[GetPolicy](#)」を参照してください。

GetPolicyVersion

次の例は、GetPolicyVersion を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->getpolicyversion(  
        iv_policyarn = iv_policy_arn  
        iv_versionid = iv_version_id ).  
    MESSAGE 'Retrieved policy version information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iaminvalidinputex.  
    MESSAGE 'Invalid input provided.' 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「[GetPolicyVersion](#)」を参照してください。

GetRole

次の例は、GetRole を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_iam->getrole( iv_rolename = iv_role_name ).  
MESSAGE 'Retrieved role information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
MESSAGE 'Role does not exist.' 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「[GetRole](#)」を参照してください。

ListAccessKeys

次の例は、ListAccessKeys を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
oo_result = lo_iam->listaccesskeys(  
iv_username = iv_user_name ).  
MESSAGE 'Retrieved access key list.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
MESSAGE 'User does not exist.' 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「[ListAccessKeys](#)」を参照してください。

ListAccountAliases

次の例は、ListAccountAliases を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listaccountaliases( ).  
    MESSAGE 'Retrieved account alias list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing account aliases.' 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「[ListAccountAliases](#)」を参照してください。

ListAttachedRolePolicies

次の例は、ListAttachedRolePolicies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listattachedrolepolicies(  
        iv_rolename = iv_role_name ).  
    MESSAGE 'Retrieved attached policy list for role.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' 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「[ListAttachedRolePolicies](#)」を参照してください。

ListGroups

次の例は、ListGroups を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listgroups( ).  
    MESSAGE 'Retrieved group list.' TYPE 'I'.  
    CATCH /aws1/cx_iamservicefailureex.  
        MESSAGE 'Service failure when listing groups.' 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「[ListGroups](#)」を参照してください。

ListPolicies

次の例は、ListPolicies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_iam->listpolicies( iv_scope = iv_scope ).
MESSAGE 'Retrieved policy list.' TYPE 'I'.
CATCH /aws1/cx_iamservicefailureex.
MESSAGE 'Service failure when listing policies.' 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「[ListPolicies](#)」を参照してください。

ListPolicyVersions

次の例は、ListPolicyVersions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
oo_result = lo_iam->listpolicyversions(
iv_policyarn = iv_policy_arn ).
MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_iamservicefailureex.
MESSAGE 'Service failure when listing policy versions.' 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「[ListPolicyVersions](#)」を参照してください。

ListRolePolicies

次の例は、ListRolePolicies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listrolepolicies(  
        iv_rolename = iv_role_name ).  
    MESSAGE 'Retrieved inline policy list for role.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' 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「[ListRolePolicies](#)」を参照してください。

ListRoles

次の例は、ListRoles を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listroles( ).  
    MESSAGE 'Retrieved role list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing roles.' 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「[ListRoles](#)」を参照してください。

ListSAMLProviders

次の例は、ListSAMLProviders を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_iam->listsamlproviders( ).  
    MESSAGE 'Retrieved SAML provider list.' TYPE 'I'.  
    CATCH /aws1/cx_iamservicefailureex.  
        MESSAGE 'Service failure when listing SAML providers.' 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「[ListSAMLProviders](#)」を参照してください。

ListUsers

次の例は、ListUsers を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_iam->listusers( ).
MESSAGE 'Retrieved user list.' TYPE 'I'.
CATCH /aws1/cx_iamservicefailureex.
MESSAGE 'Service failure when listing users.' 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「[ListUsers](#)」を参照してください。

SetDefaultPolicyVersion

次の例は、SetDefaultPolicyVersion を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_iam->setdefaultpolicyversion(
    iv_policyarn = iv_policy_arn
    iv_versionid = iv_version_id ).
  MESSAGE 'Default policy version set successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
  MESSAGE 'Policy or version does not exist.' TYPE 'E'.
CATCH /aws1/cx_iaminvalidinputex.
  MESSAGE 'Invalid input provided.' TYPE 'E'.
CATCH /aws1/cx_iamlimitexceededex.
  MESSAGE 'Limit exceeded.' 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「[SetDefaultPolicyVersion](#)」を参照してください。

UpdateAccessKey

次の例は、UpdateAccessKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_iam->updateaccesskey(  
    iv_accesskeyid = iv_access_key_id  
    iv_status = iv_status  
    iv_username = iv_user_name ).  
  MESSAGE 'Access key updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Access key or user does not exist.' 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「[UpdateAccessKey](#)」を参照してください。

UpdateUser

次の例は、UpdateUser を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
lo_iam->updateuser(  
    iv_username = iv_user_name  
    iv_newusername = iv_new_user_name ).  
MESSAGE 'User updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
MESSAGE 'User does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamentityalrddyexex.  
MESSAGE 'New user name already exists.' 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「[UpdateUser](#)」を参照してください。

AWS IoT data SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT data。

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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

GetThingShadow

次の例は、GetThingShadow を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    DATA(lo_result) = lo_iop->getthingsshadow( iv_thingname = iv_thing_name ).  
  
    " Convert xstring payload to JSON string  
    DATA(lv_payload) = lo_result->get_payload( ).  
    ov_shadow = /aws1/cl_rt_util=>xstring_to_string( lv_payload ).  
    MESSAGE 'Thing shadow retrieved successfully.' TYPE 'I'.  
CATCH /aws1/cx_iopresourcenotfoundex.  
    MESSAGE 'Thing shadow not found.' TYPE 'E'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |{ lo_exception->get_text( ) }|.  
    MESSAGE lv_error 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[GetThingShadow](#)」を参照してください。

UpdateThingShadow

次の例は、UpdateThingShadow を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
" Convert JSON string to xstring for payload
DATA(lv_payload) = /aws1/cl_rt_util=>string_to_xstring( iv_shadow_state ).

lo_iop->updatethingshadow(
  iv_thingname = iv_thing_name
  iv_payload = lv_payload ).
MESSAGE 'Thing shadow updated successfully.' TYPE 'I'.
CATCH /aws1/cx_iopresourcenotfoundex.
  MESSAGE 'Thing not found.' TYPE 'E'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).
  DATA(lv_error) = |{ lo_exception->get_text( ) }|.
  MESSAGE lv_error 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[UpdateThingShadow](#)」を参照してください。

AWS IoT SiteWise SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT SiteWise。

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。

各例には完全なソースコードへのリンクが含まれており、コンテキスト内でコードを設定および実行する方法の手順を確認できます。

トピック

- [はじめに](#)
- [アクション](#)

はじめに

こんにちは AWS IoT SiteWiseは

次のコード例は、AWS IoT SiteWiseの使用を開始する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing  
purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset models.' 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[ListAssetModels](#)」を参照してください。

アクション

BatchPutAssetPropertyValue

次の例は、BatchPutAssetPropertyValue を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ios->batchputassetpropertyvalue(  
        it_entries = it_entries  
    ).
```

```
MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
MESSAGE 'Asset does not exist.' 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[BatchPutAssetPropertyValue](#)を参照してください。

CreateAsset

次の例は、CreateAsset を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_ios->createasset(
    iv_assetname = iv_asset_name
    iv_assetmodelid = iv_asset_model_id
  ). " oo_result is returned for testing purposes. "
  MESSAGE 'IoT SiteWise asset created' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
  MESSAGE 'Asset model does not exist.' 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「[CreateAsset](#)」を参照してください。

CreateAssetModel

次の例は、CreateAssetModel を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_ios->createassetmodel(  
    iv_assetmodelname = iv_asset_model_name  
    iv_assetmodeldescription = 'This is a sample asset model description.'  
    it_assetmodelproperties = it_properties  
  ). " oo_result is returned for testing purposes. "  
  MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.  
CATCH /aws1/cx_iosresrcalrddyexistsex.  
  MESSAGE 'Asset model already exists.' 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「[CreateAssetModel](#)」を参照してください。

CreateGateway

次の例は、CreateGateway を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_ios->creategateway(  
    iv_gatewayname = iv_gateway_name  
    io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(  
      io_greenrassv2 = NEW /aws1/cl_iosgreenrassv2(  
        iv_gatewayname = iv_gateway_name  
        iv_gatewayplatform = iv_gatewayplatform  
        iv_gatewayplatformname = iv_gatewayplatform_name  
        iv_gatewayplatformdescription = iv_gatewayplatform_description  
        iv_gatewayplatformproperties = iv_gatewayplatform_properties  
        iv_gatewayplatformdescription = iv_gatewayplatform_description  
        iv_gatewayplatformproperties = iv_gatewayplatform_properties  
      )  
    )  
  )
```

```

        iv_coredevicethingname = iv_core_device_thing_name
    )
)
it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
    (
        VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
            key = 'Environment'
            value = NEW /aws1/cl_iostagmap_w( 'Production' )
        )
    )
)
). " oo_result is returned for testing purposes. "
MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
CATCH /aws1/cx_iosresrcalrddyexistsex.
    MESSAGE 'Gateway already exists.' 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「[CreateGateway](#)」を参照してください。

DeleteAsset

次の例は、DeleteAsset を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    lo_ios->deleteasset(
        iv_assetid = iv_asset_id
    ).
    MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to delete asset.' 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[DeleteAsset](#)」を参照してください。

DeleteAssetModel

次の例は、DeleteAssetModel を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ios->deleteassetmodel(  
        iv_assetmodelid = iv_asset_model_id  
    ).  
    MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to delete asset model.' 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[DeleteAssetModel](#)」を参照してください。

DeleteGateway

次の例は、DeleteGateway を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ios->deletegateway(  
    iv_gatewayid = iv_gateway_id  
  ).  
  MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
  MESSAGE 'Gateway does not exist.' 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[DeleteGateway](#)」を参照してください。

DescribeGateway

次の例は、DescribeGateway を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_ios->describegateway(  
    iv_gatewayid = iv_gateway_id  
  ). " oo_result is returned for testing purposes. "  
  MESSAGE 'Retrieved gateway description.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.
```

```
MESSAGE 'Gateway does not exist.' 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「[DescribeGateway](#)」を参照してください。

GetAssetPropertyValue

次の例は、GetAssetPropertyValue を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ios->getassetpropertyvalue(  
        iv_assetid = iv_asset_id  
        iv_propertyid = iv_property_id  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
    MESSAGE 'Asset or property does not exist.' 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[GetAssetPropertyValue](#)」を参照してください。

ListAssetModelProperties

次の例は、ListAssetModelProperties を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ios->listassetmodelproperties(  
        iv_assetmodelid = iv_asset_model_id  
    ). " oo_result is returned for testing purposes. "  
    DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).  
    MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset model properties.' 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[ListAssetModelProperties](#)」を参照してください。

ListAssetModels

次の例は、ListAssetModels を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing  
purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.
```

```
MESSAGE 'Unable to list asset models.' 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[ListAssetModels](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon Keyspaces の例

次のコード例は、Amazon Keyspaces で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateKeyspace

次の例は、CreateKeyspace を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kys->createkeyspace(  
        iv_keyspacename = iv_keyspace_name ).
```

```

    MESSAGE 'Keyspace created successfully.' TYPE 'I'.
  CATCH /aws1/cx_kysconflictexception.
    MESSAGE 'Keyspace already exists.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[CreateKeyspace](#)」を参照してください。

CreateTable

次の例は、CreateTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Define schema with columns
  DATA(lt_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
    ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'title' iv_type = 'text' ) )
    ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'year' iv_type = 'int' ) )
    ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'release_date' iv_type =
'timestamp' ) )
    ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'plot' iv_type = 'text' ) )
  ).

  " Define partition keys
  DATA(lt_partition_keys) = VALUE /aws1/
cl_kyspartitionkey=>tt_partitionkeylist(
    ( NEW /aws1/cl_kyspartitionkey( iv_name = 'year' ) )
    ( NEW /aws1/cl_kyspartitionkey( iv_name = 'title' ) )
  ).

```

```

    ).

    " Create schema definition
    DATA(lo_schema) = NEW /aws1/cl_kyssschemadefinition(
        it_allcolumns = lt_columns
        it_partitionkeys = lt_partition_keys ).

    " Enable point-in-time recovery
    DATA(lo_pitr) = NEW /aws1/cl_kyspointintimerec(
        iv_status = 'ENABLED' ).

    oo_result = lo_kys->createtable(
        iv_keyspacename = iv_keyspace_name
        iv_tablename = iv_table_name
        io_schemadefinition = lo_schema
        io_pointintimerecovery = lo_pitr ).
    MESSAGE 'Table created successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[CreateTable](#)」を参照してください。

DeleteKeyspace

次の例は、DeleteKeyspace を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    TRY.
        lo_kys->deletekeyspace(

```

```
        iv_keyspacename = iv_keyspace_name ).
    MESSAGE 'Keyspace deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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[DeleteKeyspace](#)」を参照してください。

DeleteTable

次の例は、DeleteTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_kys->deletetable(
        iv_keyspacename = iv_keyspace_name
        iv_tablename = iv_table_name ).
    MESSAGE 'Table deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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「[DeleteTable](#)」を参照してください。

GetKeyspace

次の例は、GetKeyspace を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kys->getkeyspace(  
        iv_keyspacename = iv_keyspace_name ).  
    MESSAGE 'Keyspace retrieved successfully.' TYPE 'I'.  
    CATCH /aws1/cx_kysresourcenotfoundex.  
        MESSAGE 'Keyspace does not exist.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
        MESSAGE lv_error 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[GetKeyspace](#)」を参照してください。

GetTable

次の例は、GetTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_kys->gettable(
    iv_keyspacename = iv_keyspace_name
    iv_tablename = iv_table_name ).
  MESSAGE 'Table information retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_kysresourcenotfoundex.
  MESSAGE 'Table does not exist.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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[GetTable](#)を参照してください。

ListKeyspaces

次の例は、ListKeyspaces を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_kys->listkeyspaces(
    iv_maxresults = iv_max_results ).
  MESSAGE 'Keyspaces listed successfully.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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[ListKeyspaces](#)」を参照してください。

ListTables

次の例は、ListTables を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kys->listtables(  
        iv_keyspacename = iv_keyspace_name ).  
    MESSAGE 'Tables listed successfully.' TYPE 'I'.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.  
    MESSAGE lv_error 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「[ListTables](#)」を参照してください。

RestoreTable

次の例は、RestoreTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_kys->restoretable(
    iv_sourcekeyspacename = iv_source_keyspace_name
    iv_sourcetablename = iv_source_table_name
    iv_targetkeyspacename = iv_target_keyspace_name
    iv_targettablename = iv_target_table_name
    iv_restoretimestamp = iv_restore_timestamp ).
  MESSAGE 'Table restore initiated successfully.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error 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[RestoreTable](#)」を参照してください。

UpdateTable

次の例は、UpdateTable を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Add a new column to track watched movies
  DATA(lt_add_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
  ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'watched' iv_type =
'boolean' ) )
  ).

  oo_result = lo_kys->updatetable(
    iv_keyspacename = iv_keyspace_name
    iv_tablename = iv_table_name
```

```
        it_addcolumns = lt_add_columns ).
    MESSAGE 'Table updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error 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[UpdateTable](#)」を参照してください。

SDK for SAP ABAP を使用した Kinesis の例

次のコード例は、Kinesis で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

基本は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

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。

各例には、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。

トピック

- [基本](#)
- [アクション](#)


基本

基本を学ぶ

次のコード例は、以下の操作方法を示しています。

- ストリームを作成し、そのストリームにレコードを挿入します。
- シャードイテレーターを作成します。
- レコードを読み取り、リソースをクリーンアップします。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_stream_describe_result TYPE REF TO /aws1/cl_knsdescrstreamoutput.
DATA lo_stream_description TYPE REF TO /aws1/cl_knsstreamdescription.
DATA lo_sharditerator TYPE REF TO /aws1/cl_knsgetsharditerator01.
DATA lo_record_result TYPE REF TO /aws1/cl_knsputrecordoutput.

"Create stream."
TRY.
    lo_kns->createstream(
        iv_streamname = iv_stream_name
        iv_shardcount = iv_shard_count ).
    MESSAGE 'Stream created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knslimitexceededex.
    MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex.
    MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.

"Wait for stream to becomes active."
lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
WHILE lo_stream_description->get_streamstatus( ) <> 'ACTIVE'.
    IF sy-index = 30.
        EXIT.                "maximum 5 minutes"
    ENDIF.
    WAIT UP TO 10 SECONDS.
    lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
    lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
```

```
ENDWHILE.

"Create record."
TRY.
    lo_record_result = lo_kns->putrecord(
        iv_streamname = iv_stream_name
        iv_data        = iv_data
        iv_partitionkey = iv_partition_key ).
    MESSAGE 'Record created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex.
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex.
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired.
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmssthrrottlingex.
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Create a shard iterator in order to read the record."
TRY.
    lo_sharditerator = lo_kns->getsharditerator(
        iv_shardid = lo_record_result->get_shardid( )
        iv_sharditeratortype = iv_sharditeratortype
        iv_streamname = iv_stream_name ).
    MESSAGE 'Shard iterator created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
```

```
CATCH /aws1/cx_sgmresourcenotfound.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Read the record."
TRY.
  oo_result = lo_kns->getrecords(                                " oo_result is returned
for testing purposes. "
  iv_sharditerator = lo_sharditerator->get_sharditerator( ) ).
  MESSAGE 'Shard iterator created.' TYPE 'I'.
CATCH /aws1/cx_knsexpirediteratorex.
  MESSAGE 'Iterator expired.' TYPE 'E'.
CATCH /aws1/cx_knsinvalidargumentex.
  MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
  MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.
  MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex.
  MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex.
  MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired.
  MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmsstrottlingex.
  MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
  MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Delete stream."
TRY.
  lo_kns->deletestream(
    iv_streamname = iv_stream_name ).
  MESSAGE 'Stream deleted.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceededex.
  MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex.
```

```
MESSAGE 'The request processing has failed because the resource is in use.'  
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。
 - [CreateStream](#)
 - [DeleteStream](#)
 - [GetRecords](#)
 - [GetShardIterator](#)
 - [PutRecord](#)

アクション

CreateStream

次のコード例は、CreateStream を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_kns->createstream(  
        iv_streamname = iv_stream_name  
        iv_shardcount = iv_shard_count ).  
    MESSAGE 'Stream created.' TYPE 'I'.  
CATCH /aws1/cx_knsinvalidargumentex.  
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.  
CATCH /aws1/cx_knslimitexceededex.  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
CATCH /aws1/cx_knsresourceinuseex.
```

```
MESSAGE 'The request processing has failed because the resource is in use.'  
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「[CreateStream](#)」を参照してください。

DeleteStream

次のコード例は、DeleteStream を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_kns->deletestream(  
        iv_streamname = iv_stream_name ).  
    MESSAGE 'Stream deleted.' TYPE 'I'.  
    CATCH /aws1/cx_knslimitexceedex.  
        MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
    CATCH /aws1/cx_knsresourceinuseex.  
        MESSAGE 'The request processing has failed because the resource is in use.'  
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「[DeleteStream](#)」を参照してください。

DescribeStream

次のコード例は、DescribeStream を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kns->describestream(  
        iv_streamname = iv_stream_name ).  
    DATA(lt_stream_description) = oo_result->get_streamdescription( ).  
    MESSAGE 'Streams retrieved.' TYPE 'I'.  
CATCH /aws1/cx_knslimitexceededException.  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
CATCH /aws1/cx_knsresourceNotFoundEx.  
    MESSAGE 'Resource being accessed is not found.' 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「[DescribeStream](#)」を参照してください。

GetRecords

次のコード例は、GetRecords を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kns->getrecords(           " oo_result is returned for  
testing purposes. "  
        iv_sharditerator = iv_shard_iterator ).
```

```
DATA(lt_records) = oo_result->get_records( ).
MESSAGE 'Record retrieved.' TYPE 'I'.
CATCH /aws1/cx_knsexpirediteratorex.
MESSAGE 'Iterator expired.' TYPE 'E'.
CATCH /aws1/cx_knsinvalidargumentex.
MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.
MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex.
MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex.
MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired.
MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmsstrottlingex.
MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
MESSAGE 'Resource being accessed is not found.' 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「[GetRecords](#)」を参照してください。

ListStreams

次のコード例は、ListStreams を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    oo_result = lo_kns->liststreams(           " oo_result is returned for testing
purposes. "
        "Set Limit to specify that a maximum of streams should be returned."
        iv_limit = iv_limit ).
    DATA(lt_streams) = oo_result->get_streamnames( ).
    MESSAGE 'Streams listed.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceeddex.
    MESSAGE 'The request processing has failed because of a limit exceed
exception.' 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「[ListStreams](#)」を参照してください。

PutRecord

次のコード例は、PutRecord を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    oo_result = lo_kns->putrecord(           " oo_result is returned for
testing purposes. "
        iv_streamname = iv_stream_name
        iv_data        = iv_data
        iv_partitionkey = iv_partition_key ).
    MESSAGE 'Record created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.

```

```

    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
    CATCH /aws1/cx_knskmsinvalidstateex.
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
    CATCH /aws1/cx_knskmsnotfoundex.
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
    CATCH /aws1/cx_knskmsoptinrequired.
    MESSAGE 'KMS key option is required.' TYPE 'E'.
    CATCH /aws1/cx_knskmssthrrottlingex.
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' 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「[PutRecord](#)」を参照してください。

RegisterStreamConsumer

次のコード例は、RegisterStreamConsumer を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    oo_result = lo_kns->registerstreamconsumer(      " oo_result is returned
for testing purposes. "
    iv_streamarn = iv_stream_arn
    iv_consumername = iv_consumer_name ).
    MESSAGE 'Stream consumer registered.' TYPE 'I'.
    CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.

```

```
CATCH /aws1/cx_sgmresource-limit-excd.  
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.  
CATCH /aws1/cx_sgmresource-in-use.  
  MESSAGE 'Resource being accessed is in use.' TYPE 'E'.  
CATCH /aws1/cx_sgmresource-not-found.  
  MESSAGE 'Resource being accessed is not found.' 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「[RegisterStreamConsumer](#)」を参照してください。

AWS KMS SDK for SAP ABAP を使用した例

次のコード例は、で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS KMS。

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。

各例には完全なソースコードへのリンクが含まれており、コンテキスト内でコードを設定および実行する方法の手順を確認できます。

トピック

- [はじめに](#)
- [アクション](#)

はじめに

こんにちは AWS KMSは

次のコード例は、AWS Key Management Serviceの使用を開始する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kms->listkeys( ).  
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[ListKeys](#)」を参照してください。

アクション

CreateAlias

次の例は、CreateAlias を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_alias_name = 'alias/my-key-alias'  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->createalias(  
        iv_aliasname = iv_alias_name  
        iv_targetkeyid = iv_key_id
```

```
    ).  
    MESSAGE 'Alias created successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsalreadyexistsex.  
    MESSAGE 'Alias already exists.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsinvalidaliasnameex.  
    MESSAGE 'Invalid alias name.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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「[CreateAlias](#)」を参照してください。

CreateGrant

次の例は、CreateGrant を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'  
    " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'  
    oo_result = lo_kms->creategrant(  
        iv_keyid = iv_key_id  
        iv_granteeprincipal = iv_grantee_principal  
        it_operations = it_operations  
    ).  
    MESSAGE 'Grant created successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsdisabledexception.  
    MESSAGE 'The key is disabled.' TYPE 'E'.
```

```
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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「[CreateGrant](#)」を参照してください。

CreateKey

次の例は、CreateKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    " iv_description = 'Created by the AWS SDK for SAP ABAP'  
    oo_result = lo_kms->createkey( iv_description = iv_description ).  
    MESSAGE 'KMS key created successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
CATCH /aws1/cx_kmslimitexceededex.  
    MESSAGE 'Limit exceeded for KMS resources.' 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「[CreateKey](#)」を参照してください。

Decrypt

次の例は、Decrypt を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_ciphertext_blob contains the encrypted data  
    oo_result = lo_kms->decrypt(  
        iv_keyid = iv_key_id  
        iv_ciphertextblob = iv_ciphertext_blob  
    ).  
    MESSAGE 'Text decrypted successfully.' TYPE 'I'.  
    CATCH /aws1/cx_kmsdisabledexception.  
        MESSAGE 'The key is disabled.' TYPE 'E'.  
    CATCH /aws1/cx_kmsincorrectkeyex.  
        MESSAGE 'Incorrect key for decryption.' TYPE 'E'.  
    CATCH /aws1/cx_kmsnotfoundexception.  
        MESSAGE 'Key not found.' TYPE 'E'.  
    CATCH /aws1/cx_kmskmsinternalex.  
        MESSAGE 'An internal error occurred.' 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「[復号](#)」を参照してください。

DeleteAlias

次の例は、DeleteAlias を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_alias_name = 'alias/my-key-alias'  
  lo_kms->deletealias( iv_aliasname = iv_alias_name ).  
  MESSAGE 'Alias deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
  MESSAGE 'Alias not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
  MESSAGE 'An internal error occurred.' 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[DeleteAlias](#)」を参照してください。

DescribeKey

次の例は、DescribeKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
  oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).  
  DATA(lo_key) = oo_result->get_keymetadata( ).  
  MESSAGE 'Retrieved key information successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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「[DescribeKey](#)」を参照してください。

DisableKey

次の例は、DisableKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->disablekey( iv_keyid = iv_key_id ).  
    MESSAGE 'KMS key disabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[DisableKey](#)」を参照してください。

EnableKey

次の例は、EnableKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekey( iv_keyid = iv_key_id ).  
    MESSAGE 'KMS key enabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[EnableKey](#)を参照してください。

EnableKeyRotation

次の例は、EnableKeyRotation を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).  
    MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.
```

```

CATCH /aws1/cx_kmsdisabledexception.
  MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsunsupportedopex.
  MESSAGE 'Operation not supported for this key.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' 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[EnableKeyRotation](#)」を参照してください。

Encrypt

次の例は、Encrypt を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_plaintext contains the data to encrypt
  oo_result = lo_kms->encrypt(
    iv_keyid = iv_key_id
    iv_plaintext = iv_plaintext
  ).
  MESSAGE 'Text encrypted successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
  MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' 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「[Encrypt](#)」を参照してください。

GenerateDataKey

次の例は、GenerateDataKey を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_keyspec = 'AES_256'  
    oo_result = lo_kms->generatedatakey(  
        iv_keyid = iv_key_id  
        iv_keyspec = 'AES_256'  
    ).  
    MESSAGE 'Data key generated successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsdisabledexception.  
    MESSAGE 'The key is disabled.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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「[GenerateDataKey](#)」を参照してください。

GetKeyPolicy

次の例は、GetKeyPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    oo_result = lo_kms->getkeypolicy(  
        iv_keyid = iv_key_id  
        iv_policyname = 'default'  
    ).  
    MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.  
    CATCH /aws1/cx_kmsnotfoundexception.  
        MESSAGE 'Key not found.' TYPE 'E'.  
    CATCH /aws1/cx_kmskmsinternalex.  
        MESSAGE 'An internal error occurred.' 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[GetKeyPolicy](#)を参照してください。

ListAliases

次の例は、ListAliases を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_kms->listaliases( ).  
    MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[ListAliases](#)」を参照してください。

ListGrants

次の例は、ListGrants を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).  
    MESSAGE 'Retrieved grants list.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[ListGrants](#)」を参照してください。

ListKeyPolicies

次の例は、ListKeyPolicies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).  
    MESSAGE 'Retrieved key policies list.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' 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[ListKeyPolicies](#)」を参照してください。

ListKeys

次の例は、ListKeys を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_kms->listkeys( ).
MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
CATCH /aws1/cx_kmskmsinternalex.
MESSAGE 'An internal error occurred.' 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[ListKeys](#)」を参照してください。

PutKeyPolicy

次の例は、PutKeyPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_policy = '{"Version":"2012-10-17",          "Statement": [...]}'
  lo_kms->putkeypolicy(
    iv_keyid = iv_key_id
    iv_policyname = 'default'
    iv_policy = iv_policy
  ).
  MESSAGE 'Key policy updated successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmsmalformedplydocex.
  MESSAGE 'Malformed policy document.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' 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[PutKeyPolicy](#)」を参照してください。

ReEncrypt

次の例は、ReEncrypt を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " iv_source_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_destination_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'  
    " iv_ciphertext_blob contains the encrypted data  
    oo_result = lo_kms->reencrypt(  
        iv_sourcekeyid = iv_source_key_id  
        iv_destinationkeyid = iv_destination_key_id  
        iv_ciphertextblob = iv_ciphertext_blob  
    ).  
    MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.  
    CATCH /aws1/cx_kmsdisabledexception.  
        MESSAGE 'The key is disabled.' TYPE 'E'.  
    CATCH /aws1/cx_kmsincorrectkeyex.  
        MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.  
    CATCH /aws1/cx_kmsnotfoundexception.  
        MESSAGE 'Key not found.' TYPE 'E'.  
    CATCH /aws1/cx_kmskmsinternalex.  
        MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [ReEncrypt](#)」を参照してください。

RetireGrant

次の例は、RetireGrant を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " iv_grant_token = 'AQpAM2RhZ...'  
  lo_kms->retiregrant( iv_granttoken = iv_grant_token ).  
  MESSAGE 'Grant retired successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
  MESSAGE 'Grant not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsinvgranttokenex.  
  MESSAGE 'Invalid grant token.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
  MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [RetireGrant](#)」を参照してください。

RevokeGrant

次の例は、RevokeGrant を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
" iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
" iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'
lo_kms->revokegrant(
  iv_keyid = iv_key_id
  iv_grantid = iv_grant_id
).
MESSAGE 'Grant revoked successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
MESSAGE 'Grant or key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidgrantidex.
MESSAGE 'Invalid grant ID.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP [RevokeGrant](#)」を参照してください。

ScheduleKeyDeletion

次の例は、ScheduleKeyDeletion を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
" iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
" iv_pending_window_days = 7
oo_result = lo_kms->schedulekeydeletion(
  iv_keyid = iv_key_id
  iv_pendingwindowdays = iv_pending_window_days
).
MESSAGE 'Key scheduled for deletion.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
```

```
MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
MESSAGE 'An internal error occurred.' 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[ScheduleKeyDeletion](#)を参照してください。

Sign

次の例は、Sign を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
  " iv_message contains the message to sign
  " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
  oo_result = lo_kms->sign(
    iv_keyid = iv_key_id
    iv_message = iv_message
    iv_signingalgorithm = iv_signing_algorithm
  ).
  MESSAGE 'Message signed successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
  MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidkeyusageex.
  MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- APIの詳細については、AWS「SDK for SAP ABAP API リファレンスで[サインイン](#)」を参照してください。

TagResource

次の例は、TagResource を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_tag_key = 'Environment'
  " iv_tag_value = 'Production'
  APPEND NEW /aws1/cl_kmstag(
    iv_tagkey = iv_tag_key
    iv_tagvalue = iv_tag_value
  ) TO lt_tags.

  lo_kms->tagresource(
    iv_keyid = iv_key_id
    it_tags = lt_tags
  ).
  MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmstagexception.
  MESSAGE 'Invalid tag format.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' 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[TagResource](#)」を参照してください。

UpdateAlias

次の例は、UpdateAlias を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  " iv_alias_name = 'alias/my-key-alias'  
  " iv_target_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'  
  lo_kms->updatealias(  
    iv_aliasname = iv_alias_name  
    iv_targetkeyid = iv_target_key_id  
  ).  
  MESSAGE 'Alias updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
  MESSAGE 'Alias or key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
  MESSAGE 'An internal error occurred.' 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[UpdateAlias](#)」を参照してください。

Verify

次の例は、Verify を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
    " iv_message contains the original message
    " iv_signature contains the signature to verify
    " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
    oo_result = lo_kms->verify(
        iv_keyid = iv_key_id
        iv_message = iv_message
        iv_signature = iv_signature
        iv_signingalgorithm = iv_signing_algorithm
    ).
    DATA(lv_valid) = oo_result->get_signaturevalid( ).
    IF lv_valid = abap_true.
        MESSAGE 'Signature is valid.' TYPE 'I'.
    ELSE.
        MESSAGE 'Signature is invalid.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
    CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinvalidsigex.
        MESSAGE 'Invalid signature.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' 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「[Verify](#)」を参照してください。

SDK for SAP ABAP を使用した Lambda の例

次のコード例は、Lambda で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

基本は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

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。

各例には、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。

トピック

- [基本](#)
- [アクション](#)

基本

基本を学ぶ

次のコード例は、以下の操作方法を示しています。

- IAM ロールと Lambda 関数を作成し、ハンドラーコードをアップロードします。
- 1つのパラメータで関数を呼び出して、結果を取得します。
- 関数コードを更新し、環境変数で設定します。
- 新しいパラメータで関数を呼び出して、結果を取得します。返された実行ログを表示します。
- アカウントの関数を一覧表示し、リソースをクリーンアップします。

詳細については、「[コンソールで Lambda 関数を作成する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  "Create an AWS Identity and Access Management (IAM) role that grants AWS
  Lambda permission to write to logs."
  DATA(lv_policy_document) = `{` &&
    ` "Version": "2012-10-17",` &&
    ` "Statement": [` &&
      `{` &&
        ` "Effect": "Allow",` &&
        ` "Action": [` &&
          ` "sts:AssumeRole"` &&
        ` ],` &&
        ` "Principal": {` &&
          ` "Service": [` &&
            ` "lambda.amazonaws.com"` &&
          ` ]` &&
        ` }` &&
      ` }` &&
    ` ]` &&
  `}`.

TRY.
  DATA(lo_create_role_output) = lo_iam->createrole(
    iv_rolename = iv_role_name
    iv_assumerolepolicydocument = lv_policy_document
    iv_description = 'Grant lambda permission to write to logs' ).
  DATA(lv_role_arn) = lo_create_role_output->get_role( )->get_arn( ).
  MESSAGE 'IAM role created.' TYPE 'I'.
  WAIT UP TO 10 SECONDS. " Make sure that the IAM role is ready
  for use. "
  CATCH /aws1/cx_iamentityalrдыexex.
    DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
    lv_role_arn = lo_role->get_role( )->get_arn( ).
  CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.

```

```

    CATCH /aws1/cx_iammalformedplydocex.
      MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
    ENDTRY.

    TRY.
      lo_iam->attachrolepolicy(
        iv_rolename = iv_role_name
        iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
      MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
    CATCH /aws1/cx_iaminvalidinputex.
      MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iamplynnotattachableex.
      MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
      MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    ENDTRY.

    " Create a Lambda function and upload handler code. "
    " Lambda function performs 'increment' action on a number. "
    TRY.
      lo_lmd->createfunction(
        iv_functionname = iv_function_name
        iv_runtime = `python3.9`
        iv_role = lv_role_arn
        iv_handler = iv_handler
        io_code = io_initial_zip_file
        iv_description = 'AWS Lambda code example' ).
      MESSAGE 'Lambda function created.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodestorageexcex.
      MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
      MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
      MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDTRY.

    " Verify the function is in Active state "
    WHILE lo_lmd->getfunction( iv_functionname = iv_function_name )-
>get_configuration( )->ask_state( ) <> 'Active'.

```

```

    IF sy-index = 10.
      EXIT.          " Maximum 10 seconds. "
    ENDIF.
    WAIT UP TO 1 SECONDS.
  ENDWHILE.

  "Invoke the function with a single parameter and get results."
  TRY.
    DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
      `{`  &&
      `  "action": "increment",`  &&
      `  "number": 10`  &&
      `}` ).
    DATA(lo_initial_invoke_output) = lo_lmd->invoke(
      iv_functionname = iv_function_name
      iv_payload = lv_json ).
    ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
    " ov_initial_invoke_payload is returned for testing purposes. "
    DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
    CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
lo_writer_json.
    DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
    MESSAGE 'Lambda function invoked.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
      MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvrequestcontex.
      MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
      MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdunsuppmediatyp00.
      MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
  ENDTRY.

  " Update the function code and configure its Lambda environment with an
environment variable. "
  " Lambda function is updated to perform 'decrement' action also. "
  TRY.
    lo_lmd->updatefunctioncode(
      iv_functionname = iv_function_name
      iv_zipfile = io_updated_zip_file ).

```

```

        WAIT UP TO 10 SECONDS.          " Make sure that the update is
completed. "
        MESSAGE 'Lambda function code updated.' TYPE 'I'.
        CATCH /aws1/cx_lmdcodestorageexcde.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
        CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDMETHOD.

    TRY.
        DATA lt_variables TYPE /aws1/
cl_lmdenvironmentvaria00=>tt_environmentvariables.
        DATA ls_variable LIKE LINE OF lt_variables.
        ls_variable-key = 'LOG_LEVEL'.
        ls_variable-value = NEW /aws1/cl_lmdenvironmentvaria00( iv_value =
'info' ).
        INSERT ls_variable INTO TABLE lt_variables.

        lo_lmd->updatefunctionconfiguration(
            iv_functionname = iv_function_name
            io_environment = NEW /aws1/cl_lmdenvironment( it_variables =
lt_variables ) ).
        WAIT UP TO 10 SECONDS.          " Make sure that the update is
completed. "
        MESSAGE 'Lambda function configuration/settings updated.' TYPE 'I'.
        CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.'
TYPE 'E'.
        CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDMETHOD.

    "Invoke the function with new parameters and get results. Display the
execution log that's returned from the invocation."
    TRY.
        lv_json = /aws1/cl_rt_util=>string_to_xstring(
            `{` ` &&
            ` "action": "decrement", ` ` &&
            ` "number": 10 ` ` &&
            `}` ).

```

```

        DATA(lo_updated_invoke_output) = lo_lmd->invoke(
            iv_functionname = iv_function_name
            iv_payload = lv_json ).
        ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
        " ov_updated_invoke_payload is returned for testing purposes. "
        lo_writer_json = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
        CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
lo_writer_json.
        lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
        CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_lmdinvrequestcontext.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
        CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        CATCH /aws1/cx_lmdunsuppmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
        ENDMETHOD.

        " List the functions for your account. "
        TRY.
            DATA(lo_list_output) = lo_lmd->listfunctions( ).
            DATA(lt_functions) = lo_list_output->get_functions( ).
            MESSAGE 'Retrieved list of Lambda functions.' TYPE 'I'.
            CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        ENDMETHOD.

        " Delete the Lambda function. "
        TRY.
            lo_lmd->deletefunction( iv_functionname = iv_function_name ).
            MESSAGE 'Lambda function deleted.' TYPE 'I'.
            CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
            CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'W'.
        ENDMETHOD.

        " Detach role policy. "
        TRY.

```

```
lo_iam->detachrolepolicy(  
    iv_rolename = iv_role_name  
    iv_policyarn = 'arn:aws:iam::aws:policy/service-role/  
AWSLambdaBasicExecutionRole' ).  
    MESSAGE 'Detached policy from the IAM role.' TYPE 'I'.  
CATCH /aws1/cx_iaminvalidinputex.  
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.  
CATCH /aws1/cx_iamplynottattachableex.  
    MESSAGE 'Service role policies can only be attached to the service-  
linked role for their service.' TYPE 'E'.  
CATCH /aws1/cx_iamunmodableentityex.  
    MESSAGE 'Service that depends on the service-linked role is not  
modifiable.' TYPE 'E'.  
ENDTRY.  
  
" Delete the IAM role. "  
TRY.  
    lo_iam->deleterole( iv_rolename = iv_role_name ).  
    MESSAGE 'IAM role deleted.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.  
CATCH /aws1/cx_iamunmodableentityex.  
    MESSAGE 'Service that depends on the service-linked role is not  
modifiable.' TYPE 'E'.  
ENDTRY.  
  
CATCH /aws1/cx_rt_service_generic INTO lo_exception.  
    DATA(lv_error) = lo_exception->get_longtext( ).  
    MESSAGE lv_error 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。

- [CreateFunction](#)
- [DeleteFunction](#)
- [GetFunction](#)
- [Invoke](#)
- [ListFunctions](#)

- [UpdateFunctionCode](#)
- [UpdateFunctionConfiguration](#)

アクション

CreateFunction

次のコード例は、CreateFunction を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_lmd->createfunction(
    iv_functionname = iv_function_name
    iv_runtime = `python3.9`
    iv_role = iv_role_arn
    iv_handler = iv_handler
    io_code = io_zip_file
    iv_description = 'AWS Lambda code example' ).
  MESSAGE 'Lambda function created.' TYPE 'I'.
CATCH /aws1/cx_lmdcodesigningcfgno00.
  MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdcodestorageexcdex.
  MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
CATCH /aws1/cx_lmdcodeverification00.
  MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidcodesigex.
  MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
  MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
  MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
```

```
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
  CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
  CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
  ENDTRY.
```

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

DeleteFunction

次のコード例は、DeleteFunctionを使用する方法を示しています。

SDK for SAP ABAP

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWSコード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_lmd->deletefunction( iv_functionname = iv_function_name ).
  MESSAGE 'Lambda function deleted.' TYPE 'I'.
  CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
  CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
  CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
  CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
  CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
  ENDTRY.
```

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

GetFunction

次のコード例は、GetFunction を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_lmd->getfunction( iv_functionname = iv_function_name ).  
" oo_result is returned for testing purposes. "  
    MESSAGE 'Lambda function information retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_lmdinvparamvalueex.  
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.  
    CATCH /aws1/cx_lmdserviceexception.  
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'  
TYPE 'E'.  
    CATCH /aws1/cx_lmdtoomanyrequestsex.  
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.  
ENDTRY.
```

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

Invoke

次のコード例は、Invoke を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
    `{` &&
    ` "action": "increment",` &&
    ` "number": 10` &&
    `}` ).
  oo_result = lo_lmd->invoke(
testing purposes. " " oo_result is returned for
    iv_functionname = iv_function_name
    iv_payload = lv_json ).
  MESSAGE 'Lambda function invoked.' TYPE 'I'.
CATCH /aws1/cx_lmdinvparamvalueex.
  MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdinvrequestcontex.
  MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidzipfileex.
  MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
CATCH /aws1/cx_lmdrequesttoolargeex.
  MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
  MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
  MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
  MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
  MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
CATCH /aws1/cx_lmdunsuppedmediatyp00.
  MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
ENDTRY.
```

- APIの詳細については、「AWS SDK for SAP ABAPのAPIリファレンス」の「[呼び出し](#)」を参照してください。

ListFunctions

次のコード例は、ListFunctions を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_lmd->listfunctions( ).      " oo_result is returned for  
testing purposes. "  
    DATA(lt_functions) = oo_result->get_functions( ).  
    MESSAGE 'Retrieved list of Lambda functions.' TYPE 'I'.  
    CATCH /aws1/cx_lmdinvparamvalueex.  
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.  
    CATCH /aws1/cx_lmdserviceexception.  
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'  
TYPE 'E'.  
    CATCH /aws1/cx_lmdtoomanyrequestsex.  
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.  
ENDTRY.
```

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

UpdateFunctionCode

次のコード例は、UpdateFunctionCode を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_lmd->updatefunctioncode(      " oo_result is returned for
testing purposes. "
        iv_functionname = iv_function_name
        iv_zipfile = io_zip_file ).

    MESSAGE 'Lambda function code updated.' TYPE 'I'.
CATCH /aws1/cx_lmdcodesigningcfgno00.
    MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdcodestorageexcdex.
    MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
CATCH /aws1/cx_lmdcodeverification00.
    MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidcodesigex.
    MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.
```

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

UpdateFunctionConfiguration

次のコード例は、UpdateFunctionConfiguration を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_lmd->updatefunctionconfiguration(      " oo_result is returned
for testing purposes. "
        iv_functionname = iv_function_name
        iv_runtime = iv_runtime
        iv_description = 'Updated Lambda function'
        iv_memorysize = iv_memory_size ).

    MESSAGE 'Lambda function configuration/settings updated.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodesigningcfgno00.
    MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodeverification00.
    MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvalidcodesigex.
    MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
    CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.
```

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

SDK for SAP ABAP を使用した Organizations の例

次のコード例は、AWS SDK for SAP ABAP with Organizations を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

AttachPolicy

次の例は、AttachPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_org->attachpolicy(  
        iv_policyid = iv_policy_id  
        iv_targetid = iv_target_id ).  
    MESSAGE 'Policy attached to target.' TYPE 'I'.  
CATCH /aws1/cx_orgaccessdeniedex.  
    MESSAGE 'You do not have permission to attach the policy.' TYPE 'E'.
```

```

CATCH /aws1/cx_orgpolicynotfoundex.
  MESSAGE 'The specified policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_orgtargetnotfoundex.
  MESSAGE 'The specified target does not exist.' TYPE 'E'.
CATCH /aws1/cx_orgduplicateplyatta00.
  MESSAGE 'The policy is already attached to the target.' 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「[AttachPolicy](#)」を参照してください。

CreatePolicy

次の例は、CreatePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  oo_result = lo_org->createpolicy(      " oo_result is returned for testing
purposes. "
    iv_name      = iv_policy_name
    iv_description = iv_policy_description
    iv_content    = iv_policy_content
    iv_type       = iv_policy_type ).
  MESSAGE 'Policy created.' TYPE 'I'.
CATCH /aws1/cx_orgaccessdeniedex.
  MESSAGE 'You do not have permission to create a policy.' TYPE 'E'.
CATCH /aws1/cx_orgduplicatepolicyex.
  MESSAGE 'A policy with this name already exists.' TYPE 'E'.
CATCH /aws1/cx_orgmalformedplydocex.
  MESSAGE 'The policy content is malformed.' 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「[CreatePolicy](#)」を参照してください。

DeletePolicy

次の例は、DeletePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_org->deletepolicy(  
    iv_policyid = iv_policy_id ).  
  MESSAGE 'Policy deleted.' TYPE 'I'.  
  CATCH /aws1/cx_orgaccessdeniedex.  
    MESSAGE 'You do not have permission to delete the policy.' TYPE 'E'.  
  CATCH /aws1/cx_orgpolicynotfoundex.  
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.  
  CATCH /aws1/cx_orgpolicyinuseex.  
    MESSAGE 'The policy is still attached to one or more targets.' 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「[DeletePolicy](#)」を参照してください。

DescribePolicy

次の例は、DescribePolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_org->describepolicy(      " oo_result is returned for testing  
purposes. "  
        iv_policyid = iv_policy_id ).  
    DATA(lo_policy) = oo_result->get_policy( ).  
    MESSAGE 'Retrieved policy details.' TYPE 'I'.  
    CATCH /aws1/cx_orgaccessdeniedex.  
        MESSAGE 'You do not have permission to describe the policy.' TYPE 'E'.  
    CATCH /aws1/cx_orgpolicynotfoundex.  
        MESSAGE 'The specified policy does not exist.' 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「[DescribePolicy](#)」を参照してください。

DetachPolicy

次の例は、DetachPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_org->detachpolicy(  
        iv_policyid = iv_policy_id
```

```

        iv_targetid = iv_target_id ).
    MESSAGE 'Policy detached from target.' TYPE 'I'.
    CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to detach the policy.' TYPE 'E'.
    CATCH /aws1/cx_orgpolicynotfoundex.
        MESSAGE 'The specified policy does not exist.' TYPE 'E'.
    CATCH /aws1/cx_orgtargetnotfoundex.
        MESSAGE 'The specified target does not exist.' TYPE 'E'.
    CATCH /aws1/cx_orgpolicynotattex.
        MESSAGE 'The policy is not attached to the target.' 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[DetachPolicy](#)を参照してください。

ListPolicies

次の例は、ListPolicies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    TRY.
        oo_result = lo_org->listpolicies(      " oo_result is returned for testing
purposes. "
        iv_filter = iv_filter ).
        DATA(lt_policies) = oo_result->get_policies( ).
        MESSAGE 'Retrieved list of policies.' TYPE 'I'.
    CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to list policies.' TYPE 'E'.
    CATCH /aws1/cx_orgawsorgsnotinuseex.
        MESSAGE 'Your account is not a member of an organization.' 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「[ListPolicies](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon Pinpoint の例

次のコード例は、Amazon Pinpoint で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

SendMessage

次の例は、SendMessage を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

E メールメッセージを送信します。

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
```

```

        value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
    ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
    iv_applicationid = iv_app_id
    io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
            io_emailmessage = NEW /aws1/cl_pptemailmessage(
                iv_fromaddress = iv_sender
                io_simpleemail = NEW /aws1/cl_pptsimpleemail(
                    io_subject = NEW /aws1/cl_pptsimpleemailpart(
                        iv_charset = iv_char_set
                        iv_data = iv_subject
                    )
                    io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
                        iv_charset = iv_char_set
                        iv_data = iv_html_message
                    )
                    io_textpart = NEW /aws1/cl_pptsimpleemailpart(
                        iv_charset = iv_char_set
                        iv_data = iv_text_message
                    )
                )
            )
        )
    )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Email message sent successfully.' TYPE 'I'.

```

SMS メッセージを送信します。

```

" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(

```

```

    key = iv_destination_number
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_body = iv_message
        iv_message_type = iv_message_type
        iv_OriginationNumber = iv_OriginationNumber
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'SMS message sent successfully.' TYPE 'I'.

```

既存の E メールテンプレートを使用して E メールメッセージを送信します。

```

" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

```

```

" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
  )
  io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
    io_emailtemplate = NEW /aws1/cl_ppttemplate(
      iv_name = iv_template_name
      iv_version = iv_template_version
    )
  )
)
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Templated email message sent successfully.' TYPE 'I'.

```

既存の SMS テンプレートを使用してテキストメッセージを送信します。

```

" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_message_type = iv_message_type

```

```
        iv_OriginationNumber = iv_OriginationNumber
    )
)
io_TemplateConfiguration = NEW /aws1/cl_ppttemplateconf(
    io_SmsTemplate = NEW /aws1/cl_ppttemplate(
        iv_Name = iv_TemplateName
        iv_Version = iv_TemplateVersion
    )
)
)
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
    IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
    ENDIF.
ENDLOOP.

MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[SendMessages](#)を参照してください。

SDK for SAP ABAP を使用した Amazon Pinpoint SMS および音声 API の例

次のコード例は、Amazon Pinpoint SMS および音声 API で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateConfigurationSet

次の例は、CreateConfigurationSet を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create a new configuration set
  lo_pps->createconfigurationset(
    iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
  ).

  MESSAGE 'Configuration set created successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
  MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_already_exists_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
  MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
  MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_limit_exceeded_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
  MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_too_many_requests_ex.
```

```
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateConfigurationSet](#)」を参照してください。

CreateConfigurationSetEventDestination

次の例は、CreateConfigurationSetEventDestination を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " Create event destination for the configuration set  
    lo_pps->createconfseteventdst(  
        iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-  
set'  
        iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-  
dest'  
        io_eventdestination = io_event_destination  
    ).  
  
    MESSAGE 'Event destination created successfully.' TYPE 'I'.  
  
    CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).  
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_already_exists_ex.  
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).  
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_not_found_ex.  
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).  
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.  
        RAISE EXCEPTION lo_bad_request_ex.  
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).  
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.  
ENDTRY.
```

```

    RAISE EXCEPTION lo_internal_error_ex.
  CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
    MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_limit_exceeded_ex.
  CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateConfigurationSetEventDestination](#)」を参照してください。

DeleteConfigurationSet

次の例は、DeleteConfigurationSet を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Delete the configuration set
  lo_pps->deleteconfigurationset(
    iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
  ).

  MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).

```

```

MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの [DeleteConfigurationSet](#) を参照してください。

DeleteConfigurationSetEventDestination

次の例は、DeleteConfigurationSetEventDestination を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Delete the event destination
  lo_pps->deleteconfseteventdst(
    iv_configurationsetname = iv_configuration_set_name      " e.g., 'my-config-
set'
    iv_eventdestinationname = iv_event_destination_name      " e.g., 'my-event-
dest'
  ).

  MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).

```

```
MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteConfigurationSetEventDestination](#)を参照してください。

GetConfigurationSetEventDestinations

次の例は、GetConfigurationSetEventDestinations を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Get event destinations for the configuration set
  oo_result = lo_pps->getconfseteventdestinations(
    iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
  ).

  " Process the event destinations
  LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
    DATA(lv_dest_name) = lo_event_dest->get_name( ).
    DATA(lv_enabled) = lo_event_dest->get_enabled( ).

    MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
TYPE 'I'.

  " Check for CloudWatch Logs destination
  DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
  IF lo_cloudwatch_dest IS NOT INITIAL.
```

```
DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
MESSAGE | CloudWatch Logs destination: { lv_log_group_arn }| TYPE 'I'.
ENDIF.

" Check for Kinesis Firehose destination
DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
IF lo_firehose_dest IS NOT INITIAL.
    DATA(lv_delivery_stream) = lo_firehose_dest->get_deliverystreamarn( ).
    MESSAGE | Kinesis Firehose destination: { lv_delivery_stream }| TYPE
'I'.
ENDIF.

" Check for SNS destination
DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
IF lo_sns_dest IS NOT INITIAL.
    DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
    MESSAGE | SNS destination: { lv_topic_arn }| TYPE 'I'.
ENDIF.
ENDLOOP.


CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetConfigurationSetEventDestinations](#)を参照してください。

ListConfigurationSets

次の例は、ListConfigurationSets を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " List all configuration sets
  oo_result = lo_pps->listconfigurationsets(
    iv_nexttoken = iv_next_token    " Optional: Token for pagination
    iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
  ).

  " Process the configuration sets
  LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
    DATA(lv_config_set_name) = lo_config_set->get_value( ).
    MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
  ENDLLOOP.

  " Check if there are more results
  DATA(lv_next_token) = oo_result->get_nexttoken( ).
  IF lv_next_token IS NOT INITIAL.
    MESSAGE |More results available. Next token: { lv_next_token }| TYPE 'I'.
  ENDIF.

  CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
  CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ListConfigurationSets](#)を参照してください。

SendVoiceMessage

次の例は、SendVoiceMessage を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
" Create SSML message type object with voice parameters
DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmllmessagetype(
  iv_languagecode = iv_language_code      " e.g., 'en-US'
  iv_voiceid      = iv_voice_id           " e.g., 'Matthew'
  iv_text         = iv_ssml_message       " SSML formatted message text
).

" Create voice message content with the SSML message
DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
  io_ssmlmessage = lo_ssml_message
).

" Send the voice message
DATA(lo_result) = lo_pps->sendvoicemessage(
  iv_Originationphonenum = iv_Origination_number " e.g., '+12065550110'
  iv_callerid            = iv_caller_id           " e.g., '+12065550199'
  iv_destinationphonenum = iv_destination_number " e.g., '+12065550142'
  io_content              = lo_content
).

" Retrieve the message ID from the response
ov_message_id = lo_result->get_messageid( ).

MESSAGE 'Voice message sent successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerror_ex INTO DATA(lo_internal_error_ex).
  MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
```

```
RAISE EXCEPTION lo_internal_error_ex.  
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).  
MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.  
RAISE EXCEPTION lo_too_many_requests_ex.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[SendVoiceMessage](#)」を参照してください。

UpdateConfigurationSetEventDestination

次の例は、UpdateConfigurationSetEventDestination を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " Update the event destination  
    lo_pps->updateconfseteventdst(  
        iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-  
set'  
        iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-  
dest'  
        io_eventdestination = io_event_destination  
    ).  
  
    MESSAGE 'Event destination updated successfully.' TYPE 'I'.  
  
CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
CATCH /aws1/cx_ppsbadrequestsex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
```

```
MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[UpdateConfigurationSetEventDestination](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon Polly の例

次のコード例は、Amazon Polly で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

DeleteLexicon

次の例は、DeleteLexicon を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ply->deletelexicon( iv_name ).  
  MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_plylexiconnotfoundex.  
  MESSAGE 'Lexicon not found.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
  MESSAGE 'Service failure occurred.' 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[DeleteLexicon](#)を参照してください。

DescribeVoices

次の例は、DescribeVoices を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Only pass optional parameters if they have values  
  IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.  
    oo_result = lo_ply->describevoices(  
      iv_engine = iv_engine  
      iv_languagecode = iv_language ).  
  ELSEIF iv_engine IS NOT INITIAL.  
    oo_result = lo_ply->describevoices(  
      iv_engine = iv_engine ).  
  ELSEIF iv_language IS NOT INITIAL.  
    oo_result = lo_ply->describevoices(  
      iv_languagecode = iv_language ).  
  ELSE.  
    oo_result = lo_ply->describevoices( ).  
  ENDIF.  
  MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
```

```
CATCH /aws1/cx_plyinvalidnexttokenex.  
  MESSAGE 'The NextToken is invalid.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
  MESSAGE 'Service failure occurred.' 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「[DescribeVoices](#)」を参照してください。

GetLexicon

次の例は、GetLexicon を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_ply->getlexicon( iv_name ).  
  DATA(lo_lexicon) = oo_result->get_lexicon( ).  
  IF lo_lexicon IS BOUND.  
    DATA(lv_lex_name) = lo_lexicon->get_name( ).  
    MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.  
  ENDIF.  
CATCH /aws1/cx_plylexiconnotfoundex.  
  MESSAGE 'Lexicon not found.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
  MESSAGE 'Service failure occurred.' 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[GetLexicon](#)」を参照してください。

GetSpeechSynthesisTask

次の例は、GetSpeechSynthesisTask を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).  
    DATA(lo_task) = oo_result->get_synthesistask( ).  
    IF lo_task IS BOUND.  
        DATA(lv_status) = lo_task->get_taskstatus( ).  
        MESSAGE |Task status: { lv_status }| TYPE 'I'.  
    ENDIF.  
    CATCH /aws1/cx_plyinvalidtaskidex.  
        MESSAGE 'Invalid task ID.' TYPE 'E'.  
    CATCH /aws1/cx_plyservicefailureex.  
        MESSAGE 'Service failure occurred.' TYPE 'E'.  
    CATCH /aws1/cx_plysynthesistsknotf00.  
        MESSAGE 'Synthesis task not found.' 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[GetSpeechSynthesisTask](#)を参照してください。

ListLexicons

次の例は、ListLexicons を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ply->listlexicons( ).  
    DATA(lt_lexicons) = oo_result->get_lexicons( ).  
    DATA(lv_count) = lines( lt_lexicons ).  
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.  
CATCH /aws1/cx_plyinvalidnexttokenex.  
    MESSAGE 'Invalid NextToken.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
    MESSAGE 'Service failure occurred.' 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[ListLexicons](#)」を参照してください。

ListSpeechSynthesisTasks

次の例は、ListSpeechSynthesisTasks を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    " Only pass optional parameters if they have values  
    IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.  
        oo_result = lo_ply->listspeechsynthesistasks(  
            iv_maxresults = iv_max_results
```

```
        iv_status = iv_status ).
ELSEIF iv_max_results IS NOT INITIAL.
    oo_result = lo_ply->listspeechsynthesistasks(
        iv_maxresults = iv_max_results ).
ELSEIF iv_status IS NOT INITIAL.
    oo_result = lo_ply->listspeechsynthesistasks(
        iv_status = iv_status ).
ELSE.
    oo_result = lo_ply->listspeechsynthesistasks( ).
ENDIF.
DATA(lt_tasks) = oo_result->get_synthesistasks( ).
DATA(lv_count) = lines( lt_tasks ).
MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' 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[ListSpeechSynthesisTasks](#)を参照してください。

PutLexicon

次の例は、PutLexicon を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_ply->putlexicon(
        iv_name = iv_name
        iv_content = iv_content ).
    MESSAGE 'Lexicon created successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidlexiconex.
```

```

    MESSAGE 'Invalid lexicon.' TYPE 'E'.
    CATCH /aws1/cx_plylexiconsizeexc00.
    MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plymaxlexemelength00.
    MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plymaxlexiconsnoexc00.
    MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
    CATCH /aws1/cx_plyunsuppdpplsalpha00.
    MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
    CATCH /aws1/cx_plyunsuppdpplslangu00.
    MESSAGE 'Unsupported PLS 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[PutLexicon](#)を参照してください。

StartSpeechSynthesisTask

次の例は、StartSpeechSynthesisTask を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Only pass optional parameters if they have values
  IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
    oo_result = lo_ply->startspeechsynthesistask(
      iv_engine = iv_engine
      iv_outputformat = iv_audio_format
      iv_outputs3bucketname = iv_s3_bucket
      iv_outputs3keyprefix = iv_s3_key_prefix
      iv_text = iv_text
      iv_voiceid = iv_voice_id

```

```
        iv_languagecode = iv_lang_code ).
ELSEIF iv_lang_code IS NOT INITIAL.
    oo_result = lo_ply->startspeechsynthesistask(
        iv_engine = iv_engine
        iv_outputformat = iv_audio_format
        iv_outputs3bucketname = iv_s3_bucket
        iv_text = iv_text
        iv_voiceid = iv_voice_id
        iv_languagecode = iv_lang_code ).
ELSEIF iv_s3_key_prefix IS NOT INITIAL.
    oo_result = lo_ply->startspeechsynthesistask(
        iv_engine = iv_engine
        iv_outputformat = iv_audio_format
        iv_outputs3bucketname = iv_s3_bucket
        iv_outputs3keyprefix = iv_s3_key_prefix
        iv_text = iv_text
        iv_voiceid = iv_voice_id ).
ELSE.
    oo_result = lo_ply->startspeechsynthesistask(
        iv_engine = iv_engine
        iv_outputformat = iv_audio_format
        iv_outputs3bucketname = iv_s3_bucket
        iv_text = iv_text
        iv_voiceid = iv_voice_id ).
ENDIF.
MESSAGE 'Speech synthesis task started.' TYPE 'I'.
CATCH /aws1/cx_plyinvalids3bucketex.
    MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' 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[StartSpeechSynthesisTask](#)を参照してください。

SynthesizeSpeech

次の例は、SynthesizeSpeech を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Only pass optional language code if it has a value
  IF iv_lang_code IS NOT INITIAL.
    oo_result = lo_ply->synthesizespeech(
      iv_engine = iv_engine
      iv_outputformat = iv_output_fmt
      iv_text = iv_text
      iv_voiceid = iv_voice_id
      iv_languagecode = iv_lang_code ).
  ELSE.
    oo_result = lo_ply->synthesizespeech(
      iv_engine = iv_engine
      iv_outputformat = iv_output_fmt
      iv_text = iv_text
      iv_voiceid = iv_voice_id ).
  ENDIF.
  MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidssmlex.
  MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
  MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
  MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
  MESSAGE 'Text length exceeded maximum.' 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[SynthesizeSpeech](#)を参照してください。

SDK for SAP ABAP を使用した Amazon RDS の例

次のコード例は、Amazon RDS で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateDBParameterGroup

次の例は、CreateDBParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_dbparametergroupfamily = 'mysql8.0'
" iv_description           = 'My custom DB parameter group for MySQL 8.0'
TRY.
    oo_result = lo_rds->createdbparametergroup(
        iv_dbparametergroupname = iv_dbparametergroupname
        iv_dbparametergroupfamily = iv_dbparametergroupfamily
        iv_description           = iv_description ).
    MESSAGE 'DB parameter group created.' TYPE 'I'.
CATCH /aws1/cx_rdsdbparmgralrxfault.
```

```
MESSAGE 'DB parameter group already exists.' TYPE 'I'.
CATCH /aws1/cx_rdsdbprmqrquotaexcd00.
MESSAGE 'DB parameter group quota exceeded.' TYPE 'I'.
ENDTRY.
```

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

DeleteDBParameterGroup

次の例は、DeleteDBParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
  lo_rds->deletedbparametergroup(
    iv_dbparametergroupname = iv_dbparametergroupname ).
  MESSAGE 'DB parameter group deleted.' TYPE 'I'.
CATCH /aws1/cx_rdsdbprmqrnotfndfault.
  MESSAGE 'DB parameter group not found.' TYPE 'I'.
CATCH /aws1/cx_rdsinldbprmqrstatef00.
  MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
ENDTRY.
```

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

DescribeDBEngineVersions

次の例は、DescribeDBEngineVersions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_engine                = 'mysql'
" iv_dbparametergroupfamily = 'mysql8.0' (optional - filters by parameter group
family)
TRY.
  oo_result = lo_rds->describedbengineversions(
    iv_engine                = iv_engine
    iv_dbparametergroupfamily = iv_dbparametergroupfamily ).
  DATA(lv_version_count) = lines( oo_result->get_dbengineversions( ) ).
  MESSAGE |Retrieved { lv_version_count } engine versions.| TYPE 'I'.
ENDTRY.
```

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

DescribeDBParameterGroups

次の例は、DescribeDBParameterGroups を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
  oo_result = lo_rds->describedbparametergroups(
    iv_dbparametergroupname = iv_dbparametergroupname ).
```

```
MESSAGE 'DB parameter group retrieved.' TYPE 'I'.
CATCH /aws1/cx_rdsdbprmgrnotfndfault.
MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

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

DescribeDBParameters

次の例は、DescribeDBParameters を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_source                 = 'user' (optional - filters by parameter source)
TRY.
  oo_result = lo_rds->describedbparameters(
    iv_dbparametergroupname = iv_dbparametergroupname
    iv_source                 = iv_source ).
  DATA(lv_param_count) = lines( oo_result->get_parameters( ) ).
  MESSAGE |Retrieved { lv_param_count } parameters.| TYPE 'I'.
CATCH /aws1/cx_rdsdbprmgrnotfndfault.
  MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

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

DescribeOrderableDBInstanceOptions

次の例は、DescribeOrderableDBInstanceOptions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_engine          = 'mysql'
" iv_engineversion  = '8.0.35'
TRY.
    oo_result = lo_rds->descroderabledbinoptions(
        iv_engine          = iv_engine
        iv_engineversion  = iv_engineversion ).
    DATA(lv_option_count) = lines( oo_result->get_orderabledbinoptions( ) ).
    MESSAGE |Retrieved { lv_option_count } orderable DB instance options.| TYPE
'I'.
ENDTRY.
```

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

ModifyDBParameterGroup

次の例は、ModifyDBParameterGroup を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_dbparametergroupname = 'mydbparametergroup'
" it_parameters - table containing parameter objects with:
"   - parametername = 'max_connections'
"   - parametervalue = '100'
```

```
" - applymethod = 'immediate' or 'pending-reboot'  
TRY.  
    oo_result = lo_rds->modifydbparametergroup(  
        iv_dbparametergroupname = iv_dbparametergroupname  
        it_parameters             = it_parameters ).  
    MESSAGE 'DB parameter group modified.' TYPE 'I'.  
CATCH /aws1/cx_rdsdbprmgrnotfndfault.  
    MESSAGE 'DB parameter group not found.' TYPE 'I'.  
CATCH /aws1/cx_rdsinvdbprmgrstatef00.  
    MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.  
ENDTRY.
```

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

SDK for SAP ABAP を使用した Amazon Redshift の例

次のコード例は、Amazon Redshift で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateCluster

次の例は、CreateCluster を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

クラスターを作成します。

```
TRY.  
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'  
  " Example values: iv_node_type = 'ra3.4xlarge'  
  " Example values: iv_master_username = 'awsuser'  
  " Example values: iv_master_password = 'AwsUser1000'  
  " Example values: iv_publicly_accessible = abap_true  
  " Example values: iv_number_of_nodes = 2  
  oo_result = lo_rsh->createcluster(  
    iv_clusteridentifier = iv_cluster_identifier  
    iv_nodetype = iv_node_type  
    iv_masterusername = iv_master_username  
    iv_masteruserpassword = iv_master_password  
    iv_publiclyaccessible = iv_publicly_accessible  
    iv_numberofnodes = iv_number_of_nodes  
  ).  
  MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.  
  CATCH /aws1/cx_rshclustalrddyefault.  
    MESSAGE 'Cluster already exists.' TYPE 'I'.  
  CATCH /aws1/cx_rshclstquotaexcdfault.  
    MESSAGE 'Cluster quota exceeded.' TYPE 'I'.  
ENDTRY.
```

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

DeleteCluster

次の例は、DeleteCluster を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

クラスターを削除します。

```
TRY.  
  " Example values: iv_cluster_identifer = 'my-redshift-cluster'  
  lo_rsh->deletecluster(  
    iv_clusteridentifier = iv_cluster_identifer  
    iv_skipfinalclustersnapshot = abap_true  
  ).  
  MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_rshclustnotfoundfault.  
  MESSAGE 'Cluster not found.' TYPE 'I'.  
CATCH /aws1/cx_rshinvcluststatefault.  
  MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.  
ENDTRY.
```

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

DescribeClusters

次の例は、DescribeClusters を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

クラスターを記述する

```

TRY.
  " Example values: iv_cluster_identifer = 'my-redshift-cluster' (optional)
  oo_result = lo_rsh->describeclusters(
    iv_clusteridentifer = iv_cluster_identifer
  ).
  lt_clusters = oo_result->get_clusters( ).
  lv_cluster_count = lines( lt_clusters ).
  MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
  MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.

```

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

DescribeStatement

次の例は、DescribeStatement を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
  " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx'
  oo_result = lo_rsd->describestatement(
    iv_id = iv_statement_id
  ).
  lv_status = oo_result->get_status( ).
  MESSAGE |Statement status: { lv_status }| TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
  MESSAGE 'Statement not found.' TYPE 'I'.
CATCH /aws1/cx_rsdinternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.

```

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

ExecuteStatement

次の例は、ExecuteStatement を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
  " Example values: iv_database_name = 'dev'
  " Example values: iv_user_name = 'awsuser'
  " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
  " Example values: it_parameter_list - SQL parameters for parameterized
queries

  " Only pass parameters if the list is not empty
IF it_parameter_list IS NOT INITIAL.
  oo_result = lo_rsd->executestatement(
    iv_clusteridentifier = iv_cluster_identifier
    iv_database = iv_database_name
    iv_dbuser = iv_user_name
    iv_sql = iv_sql
    it_parameters = it_parameter_list
  ).
ELSE.
  oo_result = lo_rsd->executestatement(
    iv_clusteridentifier = iv_cluster_identifier
    iv_database = iv_database_name
    iv_dbuser = iv_user_name
    iv_sql = iv_sql
  ).
```

```
ENDIF.  
  
lv_statement_id = oo_result->get_id( ).  
MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.  
CATCH /aws1/cx_rsdexecutestatementex.  
MESSAGE 'Statement execution error.' TYPE 'I'.  
CATCH /aws1/cx_rsdresourcenotfoundex.  
MESSAGE 'Resource not found.' TYPE 'I'.  
ENDTRY.
```

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

GetStatementResult

次の例は、GetStatementResult を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

ステートメントの結果を確認します。

```
TRY.  
  " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx'  
  " Handle pagination for large result sets  
  
DO.  
  lo_result_page = lo_rsd->getstatementresult(  
    iv_id = iv_statement_id  
    iv_nexttoken = lv_next_token  
  ).  
  
  " Collect records from this page  
  lt_page_records = lo_result_page->get_records( ).  
  APPEND LINES OF lt_page_records TO lt_all_records.
```

```
" Check if there are more pages
lv_next_token = lo_result_page->get_nexttoken( ).
IF lv_next_token IS INITIAL.
    EXIT. " No more pages
ENDIF.
ENDDO.

" For the last call, set oo_result for return value
oo_result = lo_result_page.
lv_record_count = lines( lt_all_records ).
MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found or results not available.' TYPE 'I'.
CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.
```

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

ListDatabases

次の例は、ListDatabases を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
    " Example values: iv_database_name = 'dev'
    " Example values: iv_database_user = 'awsuser'
    oo_result = lo_rsd->listdatabases(
        iv_clusteridentifier = iv_cluster_identifier
        iv_database = iv_database_name
```

```

        iv_dbuser = iv_database_user
    ).
    lt_databases = oo_result->get_databases( ).
    lv_db_count = lines( lt_databases ).
    MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
CATCH /aws1/cx_rsddatabaseconnex.
    MESSAGE 'Database connection error.' TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.

```

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

ModifyCluster

次の例は、ModifyCluster を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

クラスターを変更します。

```

TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'
    lo_rsh->modifycluster(
        iv_clusteridentifier = iv_cluster_identifier
        iv_preferredmaintenancetime = iv_pref_maintenance_wn
    ).
    MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
CATCH /aws1/cx_rshinvalcluststatefault.
    MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.
ENDTRY.

```

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

SDK for SAP ABAP を使用した Amazon Rekognition の例

次のコード例は、Amazon Rekognition で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CompareFaces

次のコード例は、CompareFaces を使用する方法を示しています。

詳細については、「[イメージ内の顔を比較する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
" Create S3 object reference for the source image
DATA(lo_source_s3obj) = NEW /aws1/cl_reks3object(
```

```

        iv_bucket = iv_source_s3_bucket
        iv_name = iv_source_s3_key ).

" Create source image object
DATA(lo_source_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_source_s3obj ).

" Create S3 object reference for the target image
DATA(lo_target_s3obj) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_target_s3_bucket
    iv_name = iv_target_s3_key ).

" Create target image object
DATA(lo_target_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_target_s3obj ).

" Compare faces
oo_result = lo_rek->comparefaces(
    io_sourceimage = lo_source_image
    io_targetimage = lo_target_image
    iv_similaritythreshold = iv_similarity ).

DATA(lt_face_matches) = oo_result->get_facematches( ).
DATA(lt_unmatched_faces) = oo_result->get_unmatchedfaces( ).

" Get counts of matched and unmatched faces
DATA(lv_matched_count) = lines( lt_face_matches ).
DATA(lv_unmatched_count) = lines( lt_unmatched_faces ).

" Output detailed comparison results
DATA(lv_message) = |Face comparison completed: | &&
                  |{ lv_matched_count } matched face(s), | &&
                  |{ lv_unmatched_count } unmatched face(s).|.
MESSAGE lv_message TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
MESSAGE 'Invalid parameter value.' 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[CompareFaces](#)を参照してください。

CreateCollection

次の例は、CreateCollection を使用する方法を説明しています。

詳細については、「[コレクションを作成する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_rek->createcollection(  
    iv_collectionid = iv_collection_id ).  
  MESSAGE 'Collection created successfully.' TYPE 'I'.  
CATCH /aws1/cx_rekresrcalrdyexistsex.  
  MESSAGE 'Collection already exists.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
  MESSAGE 'Invalid parameter value.' 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「[CreateCollection](#)」を参照してください。

DeleteCollection

次の例は、DeleteCollection を使用する方法を説明しています。

詳細については、「[コレクションを削除する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_rek->deletecollection(
    iv_collectionid = iv_collection_id ).
  MESSAGE 'Collection deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
  MESSAGE 'Collection not found.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DeleteCollection](#)」を参照してください。

DeleteFaces

次の例は、DeleteFaces を使用する方法を説明しています。

詳細については、「[コレクションから顔を削除する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_rek->deletefaces(
    iv_collectionid = iv_collection_id
    it_faceids = it_face_ids ).

  DATA(lt_deleted_faces) = oo_result->get_deletedfaces( ).
  DATA(lv_deleted_count) = lines( lt_deleted_faces ).
  DATA(lv_msg6) = |{ lv_deleted_count } face(s) deleted successfully.|.
  MESSAGE lv_msg6 TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
  MESSAGE 'Collection not found.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DeleteFaces](#)」を参照してください。

DescribeCollection

次の例は、DescribeCollection を使用する方法を説明しています。

詳細については、「[コレクションを定義する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_rek->describecollection(  
        iv_collectionid = iv_collection_id ).  
    DATA(lv_face_count) = oo_result->get_facecount( ).  
    DATA(lv_msg) = |Collection described: { lv_face_count } face(s) indexed.|.  
    MESSAGE lv_msg TYPE 'I'.  
CATCH /aws1/cx_rekresourcenotfoundex.  
    MESSAGE 'Collection not found.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' 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「[DescribeCollection](#)」を参照してください。

DetectFaces

次の例は、DetectFaces を使用する方法を説明しています。

詳細については、「[イメージ内の顔を検出する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect faces in the image with all attributes
  DATA(lt_attributes) = VALUE /aws1/cl_rekattributes_w=>tt_attributes( ).
  DATA(lo_attr_wrapper) = NEW /aws1/cl_rekattributes_w( iv_value = 'ALL' ).
  INSERT lo_attr_wrapper INTO TABLE lt_attributes.

  oo_result = lo_rek->detectfaces(
    io_image = lo_image
    it_attributes = lt_attributes ).

  DATA(lt_face_details) = oo_result->get_facedetails( ).
  DATA(lv_detected_count) = lines( lt_face_details ).
  DATA(lv_msg8) = |{ lv_detected_count } face(s) detected in image.|.
  MESSAGE lv_msg8 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DetectFaces](#)」を参照してください。

DetectLabels

次の例は、DetectLabels を使用する方法を説明しています。

詳細については、「[イメージ内のラベルを検出する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect labels in the image
  oo_result = lo_rek->detectlabels(
    io_image = lo_image
    iv_maxlabels = iv_max_labels ).

  DATA(lt_labels) = oo_result->get_labels( ).
  DATA(lv_label_count) = lines( lt_labels ).
  DATA(lv_msg9) = |{ lv_label_count } label(s) detected in image.|.
  MESSAGE lv_msg9 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DetectLabels](#)」を参照してください。

DetectModerationLabels

次の例は、DetectModerationLabels を使用する方法を説明しています。

詳細については、「[不適切なイメージを検出する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect moderation labels
  oo_result = lo_rek->detectmoderationlabels(
    io_image = lo_image ).

  DATA(lt_moderation_labels) = oo_result->get_moderationlabels( ).
  DATA(lv_mod_count) = lines( lt_moderation_labels ).
  DATA(lv_msg10) = |{ lv_mod_count } moderation label(s) detected.|.
  MESSAGE lv_msg10 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DetectModerationLabels](#)を参照してください。

DetectText

次の例は、DetectText を使用する方法を説明しています。

詳細については、「[イメージ内のテキストを検出する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect text in the image
  oo_result = lo_rek->detecttext(
    io_image = lo_image ).

  DATA(lt_text_detections) = oo_result->get_textdetections( ).
  DATA(lv_text_count) = lines( lt_text_detections ).
  DATA(lv_msg11) = |{ lv_text_count } text detection(s) found.|.
  MESSAGE lv_msg11 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[DetectText](#)を参照してください。

IndexFaces

次の例は、IndexFaces を使用する方法を説明しています。

詳細については、「[コレクションに顔を追加する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Index faces in the image
  oo_result = lo_rek->indexfaces(
    iv_collectionid = iv_collection_id
    io_image = lo_image
    iv_externalimageid = iv_external_id
    iv_maxfaces = iv_max_faces ).

  DATA(lt_face_records) = oo_result->get_facerecords( ).
  DATA(lv_indexed_count) = lines( lt_face_records ).
  DATA(lv_msg2) = |{ lv_indexed_count } face(s) indexed successfully.|.
  MESSAGE lv_msg2 TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
  MESSAGE 'Collection not found.' TYPE 'E'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[IndexFaces](#)」を参照してください。

ListCollections

次の例は、ListCollections を使用する方法を説明しています。

コレクションの詳細については、「[コレクションを一覧表示する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_rek->listcollections(  
        iv_maxresults = iv_max_results ).  
  
    DATA(lt_collection_ids) = oo_result->get_collectionids( ).  
    DATA(lv_coll_count) = lines( lt_collection_ids ).  
    DATA(lv_msg7) = |{ lv_coll_count } collection(s) found.|.  
    MESSAGE lv_msg7 TYPE 'I'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' 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[ListCollections](#)」を参照してください。

ListFaces

次の例は、ListFaces を使用する方法を説明しています。

詳細については、「[コレクションに顔を保存する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
  oo_result = lo_rek->listfaces(  
    iv_collectionid = iv_collection_id  
    iv_maxresults = iv_max_results ).
```

```
  DATA(lt_faces) = oo_result->get_faces( ).
```

```
  DATA(lv_face_count2) = lines( lt_faces ).
```

```
  DATA(lv_msg3) = |{ lv_face_count2 } face(s) found in collection.|.
```

```
  MESSAGE lv_msg3 TYPE 'I'.
```

```
CATCH /aws1/cx_rekresourcenotfoundex.
```

```
  MESSAGE 'Collection not found.' TYPE 'E'.
```

```
CATCH /aws1/cx_rekinvalidparameterex.
```

```
  MESSAGE 'Invalid parameter value.' 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[ListFaces](#)を参照してください。

RecognizeCelebrities

次の例は、RecognizeCelebrities を使用する方法を説明しています。

詳細については、「[イメージ内で有名人を認識する](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Create S3 object reference for the image  
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(  
    iv_bucket = iv_s3_bucket  
    iv_name = iv_s3_key ).  
  
  " Create image object  
  DATA(lo_image) = NEW /aws1/cl_rekimage(  
    io_s3object = lo_s3object ).  
  
  " Recognize celebrities  
  oo_result = lo_rek->recognizecelebrities(  
    io_image = lo_image ).  
  
  DATA(lt_celebrity_faces) = oo_result->get_celebrityfaces( ).  
  DATA(lv_celeb_count) = lines( lt_celebrity_faces ).  
  DATA(lv_msg12) = |{ lv_celeb_count } celebrity/celebrities recognized.|.  
  MESSAGE lv_msg12 TYPE 'I'.  
CATCH /aws1/cx_rekinvalids3objectex.  
  MESSAGE 'Invalid S3 object.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
  MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [RecognizeCelebrities](#)」を参照してください。

SearchFaces

次の例は、SearchFaces を使用する方法を説明しています。

詳細については、[顔 \(フェイス ID\) を検索する](#) を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  oo_result = lo_rek->searchfaces(
    iv_collectionid = iv_collection_id
    iv_faceid = iv_face_id
    iv_facematchthreshold = iv_threshold
    iv_maxfaces = iv_max_faces ).

  DATA(lt_face_matches) = oo_result->get_facematches( ).
  DATA(lv_match_count2) = lines( lt_face_matches ).
  DATA(lv_msg5) = |Face search completed: { lv_match_count2 } match(es)
found.|.
  MESSAGE lv_msg5 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection or face not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' 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[SearchFaces](#)を参照してください。

SearchFacesByImage

次の例は、SearchFacesByImage を使用する方法を説明しています。

詳細については、「[顔を検索する \(イメージ\)](#)」を参照してください。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).
```

```
" Create image object
DATA(lo_image) = NEW /aws1/cl_rekimage(
  io_s3object = lo_s3object ).

" Search for matching faces
oo_result = lo_rek->searchfacesbyimage(
  iv_collectionid = iv_collection_id
  io_image = lo_image
  iv_facematchthreshold = iv_threshold
  iv_maxfaces = iv_max_faces ).

DATA(lt_face_matches) = oo_result->get_facematches( ).
DATA(lv_match_count) = lines( lt_face_matches ).
DATA(lv_msg4) = |Face search completed: { lv_match_count } match(es)
found.|.
MESSAGE lv_msg4 TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
  MESSAGE 'Collection not found.' TYPE 'E'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' 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[SearchFacesByImage](#)を参照してください。

SDK for SAP ABAP を使用した Amazon S3 の例

次のコード例は、Amazon S3 で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

基本は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

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。

各例には、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。

トピック

- [基本](#)
- [アクション](#)
- [シナリオ](#)

基本

基本を学ぶ

次のコード例は、以下の操作方法を示しています。

- バケットを作成し、そこにファイルをアップロードします。
- バケットからオブジェクトをダウンロードします。
- バケット内のサブフォルダにオブジェクトをコピーします。
- バケット内のオブジェクトを一覧表示します。
- バケットオブジェクトとバケットを削除します。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
    " determine our region from our session
    DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
    DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
```

```
" When in the us-east-1 region, you must not specify a constraint
" In all other regions, specify the region as the constraint
IF lv_region = 'us-east-1'.
  CLEAR lo_constraint.
ELSE.
  lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
ENDIF.

lo_s3->createbucket(
  iv_bucket = iv_bucket_name
  io_createbucketconfiguration = lo_constraint ).
MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

"Upload an object to an S3 bucket."
TRY.
  "Get contents of file from application server."
  DATA lv_file_content TYPE xstring.
  OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
  READ DATASET iv_key INTO lv_file_content.
  CLOSE DATASET iv_key.

  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
    iv_body = lv_file_content ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
  DATA(lo_result) = lo_s3->getobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).
  DATA(lv_object_data) = lo_result->get_body( ).
  MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
```

```
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
  lo_s3->copyobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    iv_copysource = |{ iv_bucket_name }/{ iv_key }| ).
  MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" List objects in the bucket. "
TRY.
  DATA(lo_list) = lo_s3->listobjects(
    iv_bucket = iv_bucket_name ).
  MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ' ' INTO text.
ENDLOOP.
MESSAGE text TYPE 'I'.

" Delete the objects in a bucket. "
TRY.
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }| ).
  MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
```

```
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDRY.

  " Delete the bucket. "
  TRY.
    lo_s3->deletebucket(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDRY.
```

- 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。
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

アクション

CopyObject

次のコード例は、CopyObject を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_s3->copyobject(
    iv_bucket = iv_dest_bucket
    iv_key = iv_dest_object
    iv_copysource = |{ iv_src_bucket }/{ iv_src_object }| ).
  MESSAGE 'Object copied to another bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' 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「[CopyObject](#)」を参照してください。

CreateBucket

次のコード例は、CreateBucket を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " determine our region from our session
  DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
  DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
  " When in the us-east-1 region, you must not specify a constraint
  " In all other regions, specify the region as the constraint
  IF lv_region = 'us-east-1'.
    CLEAR lo_constraint.
  ELSE.
    lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
  ENDIF.
```

```
lo_s3->createbucket(  
    iv_bucket = iv_bucket_name  
    io_createbucketconfiguration = lo_constraint ).  
MESSAGE 'S3 bucket created.' TYPE 'I'.  
CATCH /aws1/cx_s3_bucketalrddyexists.  
    MESSAGE 'Bucket name already exists.' TYPE 'E'.  
CATCH /aws1/cx_s3_bktalrddyownedbyyou.  
    MESSAGE 'Bucket already exists and is owned by you.' 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「[CreateBucket](#)」を参照してください。

DeleteBucket

次のコード例は、DeleteBucket を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  
    lo_s3->deletebucket(  
        iv_bucket = iv_bucket_name ).  
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.  
    CATCH /aws1/cx_s3_nosuchbucket.  
        MESSAGE 'Bucket does not exist.' 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「[DeleteBucket](#)」を参照してください。

DeleteBucketCors

次の例は、DeleteBucketCors を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_s3->deletebucketcors(  
    iv_bucket = iv_bucket_name ).  
  MESSAGE 'Bucket CORS configuration deleted.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[DeleteBucketCors](#)を参照してください。

DeleteBucketLifecycle

次の例は、DeleteBucketLifecycle を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_s3->deletebucketlifecycle(  
    iv_bucket = iv_bucket_name ).
```

```
MESSAGE 'Bucket lifecycle configuration deleted.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' 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[DeleteBucketLifecycle](#)を参照してください。

DeleteBucketPolicy

次の例は、DeleteBucketPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_s3->deletebucketpolicy(
    iv_bucket = iv_bucket_name ).
MESSAGE 'Bucket policy deleted.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' 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[DeleteBucketPolicy](#)を参照してください。

DeleteObject

次の例は、DeleteObject を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_s3->deleteobject(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key ).  
  MESSAGE 'Object deleted from S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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「[DeleteObject](#)」を参照してください。

DeleteObjects

次の例は、DeleteObjects を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_s3->deleteobjects( " oo_result is returned for  
testing purposes. "  
    iv_bucket = iv_bucket_name  
    io_delete = NEW /aws1/cl_s3_delete( it_objects = it_object_keys ) ).  
  MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
```

```
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[DeleteObjects](#)」を参照してください。

GetBucketAcl

次の例は、GetBucketAcl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_s3->getbucketacl(      " oo_result is returned for testing  
purposes. "  
  iv_bucket = iv_bucket_name ).  
  MESSAGE 'Retrieved bucket ACL.' TYPE 'I'.  
  CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[GetBucketAcl](#)」を参照してください。

GetBucketCors

次の例は、GetBucketCors を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_s3->getbucketcors(          " oo_result is returned for
testing purposes. "
    iv_bucket = iv_bucket_name ).
    MESSAGE 'Retrieved bucket CORS configuration.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' 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[GetBucketCors](#)」を参照してください。

GetBucketLifecycleConfiguration

次の例は、GetBucketLifecycleConfiguration を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_s3->getbucketlifecycleconf(          " oo_result is returned
for testing purposes. "
    iv_bucket = iv_bucket_name ).
    MESSAGE 'Retrieved bucket lifecycle configuration.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
```

```
MESSAGE 'Bucket does not exist.' 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[GetBucketLifecycleConfiguration](#)を参照してください。

GetBucketPolicy

次の例は、GetBucketPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_s3->getbucketpolicy(           " oo_result is returned for  
testing purposes. "  
    iv_bucket = iv_bucket_name ).  
    DATA(lv_policy) = oo_result->get_policy( ).  
    MESSAGE 'Retrieved bucket policy.' TYPE 'I'.  
    CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' 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[GetBucketPolicy](#)を参照してください。

GetObject

次の例は、GetObject を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_s3->getobject(          " oo_result is returned for testing
purposes. "
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key ).
    DATA(lv_object_data) = oo_result->get_body( ).
    MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' 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「[GetObject](#)」を参照してください。

GetObjectAcl

次の例は、GetObjectAcl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_s3->getobjectacl(      " oo_result is returned for testing
purposes. "
```

```
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key ).
    MESSAGE 'Retrieved object ACL.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetObjectAcl](#)を参照してください。

GetObjectLegalHold

次の例は、GetObjectLegalHoldを使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHubには、その他のリソースもあります。用例一覧を検索し、[AWSコード例リポジトリ](#)での設定と実行の方法を確認してください。

```
    TRY.
        oo_result = lo_s3->getobjectlegalhold(           " oo_result is returned for
testing purposes. "
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key ).
        MESSAGE 'Retrieved object legal hold status.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[GetObjectLegalHold](#)を参照してください。

GetObjectLockConfiguration

次の例は、GetObjectLockConfiguration を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_s3->getobjectlockconfiguration(      " oo_result is  
returned for testing purposes. "  
    iv_bucket = iv_bucket_name ).  
    MESSAGE 'Retrieved object lock configuration.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' 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[GetObjectLockConfiguration](#)」を参照してください。

HeadBucket

次の例は、HeadBucket を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_s3->headbucket(      " oo_result is returned for testing  
purposes. "
```

```
        iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket exists and you have access to it.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' 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[HeadBucket](#)」を参照してください。

ListBuckets

次の例は、ListBuckets を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_s3->listbuckets(          " oo_result is returned for testing
purposes. "
    ).
    DATA(lv_bucket_count) = lines( oo_result->get_buckets( ) ).
    MESSAGE |Retrieved { lv_bucket_count } buckets in all regions.| TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list buckets.' 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[ListBuckets](#)」を参照してください。

ListObjectVersions

次の例は、ListObjectVersions を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_s3->listobjectversions(           " oo_result is returned for  
testing purposes. "  
        iv_bucket = iv_bucket_name  
        iv_prefix = iv_prefix ).  
    MESSAGE 'Retrieved object versions.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' 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[ListObjectVersions](#)」を参照してください。

ListObjectsV2

次の例は、ListObjectsV2 を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_s3->listobjectsv2(           " oo_result is returned for  
testing purposes. "  
        iv_bucket = iv_bucket_name ).  
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
```

```
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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「[ListObjectsV2](#)」を参照してください。

PutBucketAcl

次の例は、PutBucketAcl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Example: Grant log delivery access to a bucket  
  " iv_grantwrite = 'uri=http://acs.amazonaws.com/groups/s3/LogDelivery'  
  lo_s3->putbucketacl(  
    iv_bucket = iv_bucket_name  
    iv_grantwrite = iv_grantwrite ).  
  MESSAGE 'Bucket ACL updated.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[PutBucketAcl](#)」を参照してください。

PutBucketCors

次の例は、PutBucketCors を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Example: Allow PUT, POST, DELETE methods from http://www.example.com  
  lo_s3->putbucketcors(  
    iv_bucket = iv_bucket_name  
    io_corsconfiguration = NEW /aws1/cl_s3_corsconfiguration(  
      it_corsrules = it_cors_rules ) ).  
  MESSAGE 'Bucket CORS configuration set.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[PutBucketCors](#)」を参照してください。

PutBucketLifecycleConfiguration

次の例は、PutBucketLifecycleConfiguration を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Example: Expire objects with prefix 'logs/' after 30 days  
  lo_s3->putbucketlifecycleconf(  
    iv_bucket = iv_bucket_name
```

```

        io_lifecycleconfiguration = NEW /aws1/cl_s3_bucketlconf(
            it_rules = it_lifecycle_rule ) ).
    MESSAGE 'Bucket lifecycle configuration set.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' 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「[PutBucketLifecycleConfiguration](#)」を参照してください。

PutBucketPolicy

次の例は、PutBucketPolicy を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " Example policy JSON string
    " iv_policy = '{"Version":"2012-10-17",          "Statement":
[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:user/
user"},"Action":["s3:GetObject"],"Resource":["arn:aws:s3:::bucketname/*"]}]}'
    lo_s3->putbucketpolicy(
        iv_bucket = iv_bucket_name
        iv_policy = iv_policy ).
    MESSAGE 'Bucket policy set.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' 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[PutBucketPolicy](#)」を参照してください。

PutBucketVersioning

次の例は、PutBucketVersioning を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Example: Enable versioning on a bucket  
  " iv_status = 'Enabled'  
  lo_s3->putbucketversioning(  
    iv_bucket = iv_bucket_name  
    io_versioningconfiguration = NEW /aws1/cl_s3_versioningconf(  
      iv_status = iv_status ) ).  
  MESSAGE 'Bucket versioning enabled.' TYPE 'I'.  
  CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' 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[PutBucketVersioning](#)を参照してください。

PutObject

次の例は、PutObject を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Get contents of file from application server."
DATA lv_body TYPE xstring.
OPEN DATASET iv_file_name FOR INPUT IN BINARY MODE.
READ DATASET iv_file_name INTO lv_body.
CLOSE DATASET iv_file_name.

"Upload/put an object to an S3 bucket."
TRY.
  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_file_name
    iv_body = lv_body ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' 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「[PutObject](#)」を参照してください。

PutObjectAcl

次の例は、PutObjectAcl を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Example: Grant read access to an AWS user
  " iv_grantread = 'emailAddress=user@example.com'
  lo_s3->putobjectacl(
    iv_bucket = iv_bucket_name
    iv_key = iv_object_key
```

```
        iv_grantread = iv_grantread ).
    MESSAGE 'Object ACL updated.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' 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「[PutObjectAcl](#)」を参照してください。

PutObjectLegalHold

次の例は、PutObjectLegalHold を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    " Example: Set legal hold status to ON
    " iv_status = 'ON'
    lo_s3->putobjectlegalhold(
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key
        io_legalhold = NEW /aws1/cl_s3_objlocklegalhold(
            iv_status = iv_status ) ).
    MESSAGE 'Object legal hold status set.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' 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「[PutObjectLegalHold](#)」を参照してください。

PutObjectLockConfiguration

次の例は、PutObjectLockConfiguration を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  " Example: Enable object lock with default retention  
  " iv_enabled = 'Enabled'  
  lo_s3->putobjectlockconfiguration(  
    iv_bucket = iv_bucket_name  
    io_objectlockconfiguration = NEW /aws1/cl_s3_objectlockconf(  
      iv_objectlockenabled = iv_enabled ) ).  
  MESSAGE 'Object lock configuration set.' TYPE 'I'.  
  CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' 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「[PutObjectLockConfiguration](#)」を参照してください。

PutObjectRetention

次の例は、PutObjectRetention を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Example: Set retention mode to GOVERNANCE for 30 days  
  " iv_mode = 'GOVERNANCE'  
  " iv_retain_date should be a timestamp in the future  
  lo_s3->putobjectretention(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key  
    io_retention = NEW /aws1/cl_s3_objectlockret(  
      iv_mode = iv_mode  
      iv_retainuntildate = iv_retain_date )  
    iv_bypassgovernanceretention = abap_true ).  
  MESSAGE 'Object retention set.' TYPE 'I'.  
  CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
  CATCH /aws1/cx_s3_nosuchkey.  
    MESSAGE 'Object key does not exist.' 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「[PutObjectRetention](#)」を参照してください。

シナリオ

署名付き URL を作成する

次のコード例は、Amazon S3 の署名付き URL を作成し、オブジェクトをアップロードする方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

署名付きリクエストを作成してS3 オブジェクトを取得します。

```
" iv_bucket_name is the bucket name
" iv_key is the object name like "myfile.txt"

DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

"Upload a nice Hello World file to an S3 bucket."
TRY.
    DATA(lv_contents) = cl_abap_codepage=>convert_to( 'Hello, World' ).
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_contents
        iv_contenttype = 'text/plain' ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" now generate a presigned URL with a 600-second expiration
DATA(lo_presigner) = lo_s3->get_presigner( iv_expires_sec = 600 ).
" the presigner getobject() method has the same signature as
" lo_s3->getobject(), but it doesn't actually make the call.
" to the service. It just prepares a presigned URL for a future call
DATA(lo_presigned_req) = lo_presigner->getobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).

" You can provide this URL to a web page, user, email etc so they
" can retrieve the file. The URL will expire in 10 minutes.
ov_url = lo_presigned_req->get_url( ).
```

SDK for SAP ABAP を使用した SageMaker AI の例

次のコード例は、SageMaker AI で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

CreateEndpoint

次のコード例は、CreateEndpoint を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_production_variants TYPE /aws1/  
cl_sgmproductionvariant=>tt_productionvariantlist.  
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.  
DATA oo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
```

```
"Create a production variant as an ABAP object."
"Identifies a model that you want to host and the resources chosen to deploy for
hosting it."
lo_production_variants = NEW #( iv_variantname = iv_variant_name
                               iv_modelname = iv_model_name
                               iv_initialinstancecount =
iv_initial_instance_count
                               iv_instancetype = iv_instance_type ).

INSERT lo_production_variants INTO TABLE lt_production_variants.

"Create an endpoint configuration."
TRY.
    oo_ep_config_result = lo_sgm->createendpointconfig(
        iv_endpointconfigname = iv_endpoint_config_name
        it_productionvariants = lt_production_variants ).
    MESSAGE 'Endpoint configuration created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Create an endpoint."
TRY.
    oo_result = lo_sgm->createendpoint(      " oo_result is returned for testing
purposes. "
        iv_endpointconfigname = iv_endpoint_config_name
        iv_endpointname = iv_endpoint_name ).
    MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' 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「[CreateEndpoint](#)」を参照してください。

CreateModel

次のコード例は、CreateModel を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.

"Create an ABAP object for the container image based on input variables."
lo_primarycontainer = NEW #( iv_image = iv_container_image
                           iv_modeldataurl = iv_model_data_url ).


"Create an Amazon SageMaker model."
TRY.
    oo_result = lo_sgm->createmodel(      " oo_result is returned for testing
purposes. "
    iv_executionrolearn = iv_execution_role_arn
    iv_modelname = iv_model_name
    io_primarycontainer = lo_primarycontainer ).
    MESSAGE 'Model created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' 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「[CreateModel](#)」を参照してください。

CreateTrainingJob

次のコード例は、CreateTrainingJob を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithm_spec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
```

```
"Create ABAP internal table for hyperparameters based on input variables."
```

```
"These hyperparameters are based on the Amazon SageMaker built-in algorithm,
XGBoost."
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
```

```
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
```

```
INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_eval_metric ).
```

```
INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_scale_pos_weight ).
```

```
INSERT VALUE #( key = 'scale_pos_weight' value = lo_hyperparameters_w ) INTO
TABLE lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_subsample ).
INSERT VALUE #( key = 'subsample' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_objective ).
INSERT VALUE #( key = 'objective' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_num_round ).
INSERT VALUE #( key = 'num_round' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

"Create ABAP objects for training data sources."

```
lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                             iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                             iv_s3uri = iv_trn_data_s3uri ).
```

```
lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).
```

```
lo_trn_channel = NEW #( iv_channelname = 'train'
                        io_datasource = lo_trn_datasource
                        iv_compressiontype = iv_trn_data_compressiontype
                        iv_contenttype = iv_trn_data_contenttype ).
```

```
INSERT lo_trn_channel INTO TABLE lt_input_data_config.
```

"Create ABAP objects for validation data sources."

```
lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                             iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                             iv_s3uri = iv_val_data_s3uri ).
```

```
lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).
```

```
lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).
```

```
INSERT lo_val_channel INTO TABLE lt_input_data_config.
```

"Create an ABAP object for algorithm specification."

```
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
```

```

                                iv_traininginputmode =
iv_training_input_mode ).

"Create an ABAP object for resource configuration."
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                            iv_instancetype = iv_instance_type
                            iv_volumesizeingb = iv_volume_sizeingb ).

"Create an ABAP object for output data configuration."
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

"Create a training job."
TRY.
    oo_result = lo_sgm->createtrainingjob( " oo_result is returned for
testing purposes. "
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
    it_inputdataconfig      = lt_input_data_config
    io_algorithmspecification = lo_algorithm_specification
    io_outputdataconfig     = lo_output_data_config
    io_resourceconfig       = lo_resource_config
    io_stoppingcondition    = lo_stopping_condition ).
    MESSAGE 'Training job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourcenotfound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' 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「[CreateTrainingJob](#)」を参照してください。

CreateTransformJob

次のコード例は、CreateTransformJob を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_transforminput TYPE REF TO /aws1/cl_sgmtransforminput.
DATA lo_transformoutput TYPE REF TO /aws1/cl_sgmtransformoutput.
DATA lo_transformresources TYPE REF TO /aws1/cl_sgmtransformresources.
DATA lo_datasource TYPE REF TO /aws1/cl_sgmtransformdatasrc.
DATA lo_s3datasource TYPE REF TO /aws1/cl_sgmtransforms3datasrc.

"Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data
source."
lo_s3datasource = NEW #( iv_s3uri = iv_tf_data_s3uri
                        iv_s3datatype = iv_tf_data_s3datatype ).

"Create an ABAP object for data source."
lo_datasource = NEW #( io_s3datasource = lo_s3datasource ).

"Create an ABAP object for transform data source."
lo_transforminput = NEW #( io_datasource = lo_datasource
                          iv_contenttype = iv_tf_data_contenttype
                          iv_compressiontype = iv_tf_data_compressiontype ).

"Create an ABAP object for resource configuration."
lo_transformresources = NEW #( iv_instancecount = iv_instance_count
                              iv_instancetype = iv_instance_type ).

"Create an ABAP object for output data configuration."
lo_transformoutput = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create a transform job."
TRY.
    oo_result = lo_sgm->createtransformjob( " oo_result is returned for
testing purposes. "
    iv_modelname = iv_tf_model_name
    iv_transformjobname = iv_tf_job_name
    io_transforminput = lo_transforminput
```

```
        io_transformoutput = lo_transformoutput
        io_transformresources = lo_transformresources ).
    MESSAGE 'Transform job created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceNotFound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceLimitExcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDRY.
```

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

DeleteEndpoint

次のコード例は、DeleteEndpoint を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Delete an endpoint."
TRY.
    lo_sgm->deleteendpoint(
        iv_endpointname = iv_endpoint_name ).
    MESSAGE 'Endpoint configuration deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_endpoint_exception).
        DATA(lv_endpoint_error) = |"{ lo_endpoint_exception->av_err_code }" -
{ lo_endpoint_exception->av_err_msg }|.
        MESSAGE lv_endpoint_error TYPE 'E'.
    ENDRY.

"Delete an endpoint configuration."
TRY.
    lo_sgm->deleteendpointconfig(
```

```
        iv_endpointconfigname = iv_endpoint_config_name ).
    MESSAGE 'Endpoint deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_endpointconfig_exception).
        DATA(lv_endpointconfig_error) = |"{ lo_endpointconfig_exception-
>av_err_code }" - { lo_endpointconfig_exception->av_err_msg }|.
    MESSAGE lv_endpointconfig_error 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「[DeleteEndpoint](#)」を参照してください。

DeleteModel

次のコード例は、DeleteModel を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_sgm->deletemodel(
        iv_modelname = iv_model_name ).
    MESSAGE 'Model deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[DeleteModel](#)」を参照してください。

DescribeTrainingJob

次のコード例は、DescribeTrainingJob を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sgm->describetrainingjob(      " oo_result is returned for  
testing purposes. "  
    iv_trainingjobname = iv_training_job_name ).  
    MESSAGE 'Retrieved description of training job.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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「[DescribeTrainingJob](#)」を参照してください。

ListAlgorithms

次のコード例は、ListAlgorithms を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
oo_result = lo_sgm->listalgorithms(           " oo_result is returned for
testing purposes. "
    iv_namecontains = iv_name_contains ).
MESSAGE 'Retrieved list of algorithms.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
MESSAGE lv_error 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「[ListAlgorithms](#)」を参照してください。

ListModels

次のコード例は、ListModels を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_sgm->listmodels(           " oo_result is returned for
testing purposes. "
    iv_namecontains = iv_name_contains ).
MESSAGE 'Retrieved list of models.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
MESSAGE lv_error 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「[ListModels](#)」を参照してください。

ListNotebookInstances

次のコード例は、ListNotebookInstances を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sgm->listnotebookinstances(      " oo_result is returned  
for testing purposes. "  
    iv_namecontains = iv_name_contains ).  
    MESSAGE 'Retrieved list of notebook instances.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error 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「[ListNotebookInstances](#)」を参照してください。

ListTrainingJobs

次のコード例は、ListTrainingJobs を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
```

```
        oo_result = lo_sgm->listtrainingjobs(      " oo_result is returned for
testing purposes. "
        iv_namecontains = iv_name_contains
        iv_maxresults = iv_max_results ).
    MESSAGE 'Retrieved list of training jobs.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error 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「[ListTrainingJobs](#)」を参照してください。

シナリオ

モデルとエンドポイントの使用を開始する

次のコード例は、以下の操作方法を示しています。

- トレーニングジョブを開始し、SageMaker AI モデルを作成します。
- エンドポイント設定を作成する
- エンドポイントを作成して、リソースをクリーンアップします。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
```

```
DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithmsspec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lv_model_data_url TYPE /aws1/sgmurl.

lv_model_data_url = iv_s3_output_path && iv_training_job_name && '/output/
model.tar.gz'.

"Create ABAP internal table for hyperparameters based on input variables."
"These hyperparameters are based on Amazon SageMaker built-in algorithm -
XGBoost"
lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_eval_metric ).
INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_scale_pos_weight ).
INSERT VALUE #( key = 'scale_pos_weight' value = lo_hyperparameters_w ) INTO
TABLE lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_subsample ).
INSERT VALUE #( key = 'subsample' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_objective ).
INSERT VALUE #( key = 'objective' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_num_round ).
INSERT VALUE #( key = 'num_round' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

"Create ABAP internal table for data based on input variables."

"Training data."

```
lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                             iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                             iv_s3uri = iv_trn_data_s3uri ).
```

```
lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).
```

```
lo_trn_channel = NEW #( iv_channelname = 'train'
                        io_datasource = lo_trn_datasource
                        iv_compressiontype = iv_trn_data_compressiontype
                        iv_contenttype = iv_trn_data_contenttype ).
```

```
INSERT lo_trn_channel INTO TABLE lt_input_data_config.
```

"Validation data."

```
lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                             iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                             iv_s3uri = iv_val_data_s3uri ).
```

```
lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).
```

```
lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).
```

```
INSERT lo_val_channel INTO TABLE lt_input_data_config.
```

"Create an ABAP object for algorithm specification based on input variables."

```
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                     iv_traininginputmode =
iv_training_input_mode ).
```

"Create an ABAP object for resource configuration."

```
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                             iv_instancetype = iv_instance_type
                             iv_volumesizeingb = iv_volume_sizeingb ).
```

```
"Create an ABAP object for output data configuration."
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

TRY.
  lo_sgm->createtrainingjob(
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
    it_inputdataconfig      = lt_input_data_config
    io_algorithmspecification = lo_algorithm_specification
    io_outputdataconfig     = lo_output_data_config
    io_resourceconfig       = lo_resource_config
    io_stoppingcondition    = lo_stopping_condition ).
  MESSAGE 'Training job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
  MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Wait for training job to be completed."
lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
WHILE lo_training_result->get_trainingjobstatus( ) <> 'Completed'.
  IF sy-index = 30.
    EXIT.          "Maximum 900 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
ENDWHILE.

"Create ABAP object for the container image based on input variables."
lo_primarycontainer = NEW #( iv_image = iv_training_image
                             iv_modeldataurl = lv_model_data_url ).

"Create an Amazon SageMaker model."
TRY.
```

```

    lo_sgm->createmodel(
        iv_executionrolelearn = iv_role_arn
        iv_modelname = iv_model_name
        io_primarycontainer = lo_primarycontainer ).
    MESSAGE 'Model created.' TYPE 'I'.
CATCH /aws1/cx_sgmresource-limit-excd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Create an endpoint production variant."
lo_production_variants = NEW #( iv_variantname = iv_ep_variant_name
                                iv_modelname = iv_model_name
                                iv_initialinstancecount =
iv_ep_initial_instance_count
                                iv_instancetype = iv_ep_instance_type ).
INSERT lo_production_variants INTO TABLE lt_production_variants.

TRY.
    "Create an endpoint configuration."
    lo_ep_config_result = lo_sgm->createendpointconfig(
        iv_endpointconfigname = iv_ep_cfg_name
        it_productionvariants = lt_production_variants ).
    MESSAGE 'Endpoint configuration created.' TYPE 'I'.

    "Create an endpoint."
    oo_ep_output = lo_sgm->createendpoint(           " oo_ep_output is returned for
testing purposes. "
        iv_endpointconfigname = iv_ep_cfg_name
        iv_endpointname = iv_ep_name ).
    MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresource-limit-excd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Wait for endpoint creation to be completed."
DATA(lo_endpoint_result) = lo_sgm->describeendpoint( iv_endpointname =
iv_ep_name ).
WHILE lo_endpoint_result->get_endpointstatus( ) <> 'InService'.
    IF sy-index = 30.
        EXIT.           "Maximum 900 seconds."
    ENDIF.
    WAIT UP TO 30 SECONDS.
    lo_endpoint_result = lo_sgm->describeendpoint( iv_endpointname = iv_ep_name ).
ENDWHILE.

```

```
TRY.
  "Delete an endpoint."
  lo_sgm->deleteendpoint(
    iv_endpointname = iv_ep_name ).
  MESSAGE 'Endpoint deleted' TYPE 'I'.

  "Delete an endpoint configuration."
  lo_sgm->deleteendpointconfig(
    iv_endpointconfigname = iv_ep_cfg_name ).
  MESSAGE 'Endpoint configuration deleted.' TYPE 'I'.

  "Delete model."
  lo_sgm->deletemodel(
    iv_modelname = iv_model_name ).
  MESSAGE 'Model deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_endpointconfig_exception).
  DATA(lv_endpointconfig_error) = |"{ lo_endpointconfig_exception-
>av_err_code }" - { lo_endpointconfig_exception->av_err_msg }|.
  MESSAGE lv_endpointconfig_error 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。
 - [CreateEndpoint](#)
 - [CreateEndpointConfig](#)
 - [CreateModel](#)
 - [CreateTrainingJob](#)
 - [DeleteEndpoint](#)
 - [DeleteEndpointConfig](#)
 - [DeleteModel](#)
 - [DescribeEndpoint](#)
 - [DescribeTrainingJob](#)

SDK for SAP ABAP を使用した Secrets Manager の例

次のコード例は、Secrets Manager で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

BatchGetSecretValue

次の例は、BatchGetSecretValue を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " iv_filter_name = 'mySecret'
  DATA(lo_result) = lo_smr->batchgetsecretvalue(
    it_filters = VALUE /aws1/cl_smrfilter=>tt_filterslisttype(
      (
        NEW /aws1/cl_smrfilter(
          iv_key = 'name'
          it_values = VALUE /aws1/
cl_smrfiltvalsstringlist_w=>tt_filtvaluesstringlist(
          ( NEW /aws1/cl_smrfiltvalsstringlist_w( iv_value = iv_filter_name ) )
```

```

        )
    )
)
).
ot_secret_values = lo_result->get_secretvalues( ).
MESSAGE 'Secrets retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_smrresourcenotfoundex.
    MESSAGE 'One or more requested secrets were not found.' TYPE 'E'.
CATCH /aws1/cx_smrdecryptionfailure.
    MESSAGE 'Failed to decrypt one or more secrets.' TYPE 'E'.
CATCH /aws1/cx_smrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'E'.
CATCH /aws1/cx_smrinvalidrequestex.
    MESSAGE 'Invalid request.' 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[BatchGetSecretValue](#)を参照してください。

GetSecretValue

次の例は、GetSecretValue を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    " iv_secret_name = 'MySecretName'
    DATA(lo_result) = lo_smr->getsecretvalue( iv_secretid = iv_secret_name ).
    ov_secret_value = lo_result->get_secretstring( ).
    MESSAGE 'Secret value retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_smrresourcenotfoundex.
    MESSAGE 'The requested secret was not found.' TYPE 'E'.
CATCH /aws1/cx_smrdecryptionfailure.

```

```
MESSAGE 'Failed to decrypt the secret.' TYPE 'E'.
CATCH /aws1/cx_smrinvalidparameterex.
MESSAGE 'Invalid parameter provided.' TYPE 'E'.
CATCH /aws1/cx_smrinvalidrequestex.
MESSAGE 'Invalid request.' 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[GetSecretValue](#)を参照してください。

SDK for SAP ABAP を使用した Amazon SES の例

次のコード例は、Amazon SES で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateReceiptFilter

次の例は、CreateReceiptFilter を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" iv_allow = abap_true means 'Allow', abap_false means 'Block'
DATA(lv_policy) = COND /aws1/sesreceiptfilterpolicy(
  WHEN iv_allow = abap_true THEN 'Allow'
  ELSE 'Block'
).

DATA(lo_ip_filter) = NEW /aws1/cl_sesreceiptipfilter(
  iv_policy = lv_policy
  iv_cidr = iv_ip_address_or_range
).

DATA(lo_filter) = NEW /aws1/cl_sesreceiptfilter(
  iv_name = iv_filter_name
  io_ipfilter = lo_ip_filter
).


TRY.
  lo_ses->createreceiptfilter( io_filter = lo_filter ).
  MESSAGE 'Receipt filter created successfully' TYPE 'I'.
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
  DATA(lv_error) = |Filter already exists: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateReceiptFilter](#)」を参照してください。

CreateReceiptRule

次の例は、CreateReceiptRule を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create S3 action for copying emails to S3
DATA(lo_s3_action) = NEW /aws1/cl_sess3action(
  iv_bucketname = iv_bucket_name
  iv_objectkeyprefix = iv_prefix
).

" Create receipt action with S3 action
DATA(lo_action) = NEW /aws1/cl_sesreceiptaction(
  io_s3action = lo_s3_action
).

" Create list of actions
DATA lt_actions TYPE /aws1/cl_sesreceiptaction=>tt_receiptactionslist.
APPEND lo_action TO lt_actions.

" Create receipt rule
DATA(lo_rule) = NEW /aws1/cl_sesrecepitrule(
  iv_name = iv_rule_name
  iv_enabled = abap_true
  it_recipients = it_recipients
  it_actions = lt_actions
).

TRY.
  lo_ses->createrecepitrule(
    iv_rulesetname = iv_rule_set_name
    io_rule = lo_rule
  ).
  MESSAGE 'Receipt rule created successfully' TYPE 'I'.
CATCH /aws1/cx_sesinvalids3confex INTO DATA(lo_ex1).
  DATA(lv_error) = |Invalid S3 configuration: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
```

```
lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateReceiptRule](#)」を参照してください。

CreateReceiptRuleSet

次の例は、CreateReceiptRuleSet を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_ses->createrecepitruleset( iv_rulesetname = iv_rule_set_name ).
  MESSAGE 'Receipt rule set created successfully' TYPE 'I'.
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
  DATA(lv_error) = |Rule set already exists: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateReceiptRuleSet](#)」を参照してください。

CreateTemplate

次の例は、CreateTemplate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(  
  iv_templatename = iv_name  
  iv_subjectpart = iv_subject  
  iv_textpart = iv_text  
  iv_htmlpart = iv_html  
) .  
  
TRY .  
  lo_ses->createtemplate( io_template = lo_template ).  
  MESSAGE 'Template created successfully' TYPE 'I'.  
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).  
  DATA(lv_error) = |Template already exists: { lo_ex1->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex1.  
CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).  
  lv_error = |Invalid template: { lo_ex2->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex2.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).  
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex_generic.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateTemplate](#)」を参照してください。

DeleteIdentity

次の例は、DeleteIdentity を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ses->deleteidentity( iv_identity = iv_identity ).  
    MESSAGE 'Identity deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteIdentity](#)を参照してください。

DeleteReceiptFilter

次の例は、DeleteReceiptFilter を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ses->deletereciptfilter( iv_filtername = iv_filter_name ).  
    MESSAGE 'Receipt filter deleted successfully' TYPE 'I'.  
ENDTRY.
```

```
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteReceiptFilter](#)」を参照してください。

DeleteReceiptRule

次の例は、DeleteReceiptRule を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_ses->deletereciptrule(
    iv_rulesetname = iv_rule_set_name
    iv_rulename = iv_rule_name
  ).
  MESSAGE 'Receipt rule deleted successfully' TYPE 'I'.
CATCH /aws1/cx_sesrulesetdoesnotexist INTO DATA(lo_ex1).
  DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteReceiptRule](#)」を参照してください。

DeleteReceiptRuleSet

次の例は、DeleteReceiptRuleSet を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ses->deletereciptruleset( iv_rulesetname = iv_rule_set_name ).  
  MESSAGE 'Receipt rule set deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_sescannotdeleteex INTO DATA(lo_ex1).  
  DATA(lv_error) = |Cannot delete rule set: { lo_ex1->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex1.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).  
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex_generic.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteReceiptRuleSet](#)を参照してください。

DeleteTemplate

次の例は、DeleteTemplate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ses->deletetemplate( iv_templatename = iv_template_name ).  
  MESSAGE 'Template deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteTemplate](#)」を参照してください。

DescribeReceiptRuleSet

次の例は、DescribeReceiptRuleSet を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_ses->describerecepitruleset(  
    iv_rulesetname = iv_rule_set_name  
  ).  
  MESSAGE 'Receipt rule set described successfully' TYPE 'I'.  
CATCH /aws1/cx_sesrulesetdoesnotexist INTO DATA(lo_ex1).  
  DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex1.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).  
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex_generic.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeReceiptRuleSet](#)」を参照してください。

GetIdentityVerificationAttributes

次の例は、GetIdentityVerificationAttributes を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lt_identities TYPE /aws1/cl_sesidentitylist_w=>tt_identitylist.
APPEND NEW /aws1/cl_sesidentitylist_w( iv_value = iv_identity ) TO
lt_identities.

TRY.
  DATA(lo_result) = lo_ses->getidentityverificationattrs(
    it_identities = lt_identities
  ).

  DATA(lt_attrs) = lo_result->get_verificationattributes( ).
  IF lt_attrs IS NOT INITIAL.
    LOOP AT lt_attrs ASSIGNING FIELD-SYMBOL(<ls_attr>).
      ov_status = <ls_attr>-value->get_verificationstatus( ).
      EXIT.
    ENDLLOOP.
  ELSE.
    ov_status = 'NotFound'.
  ENDIF.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetIdentityVerificationAttributes](#)」を参照してください。

GetTemplate

次の例は、GetTemplate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    DATA(lo_result) = lo_ses->gettemplate( iv_templatename = iv_template_name ).  
    oo_template = lo_result->get_template( ).  
    MESSAGE 'Template retrieved successfully' TYPE 'I'.  
CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).  
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex1.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).  
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex_generic.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetTemplate](#)」を参照してください。

ListIdentities

次の例は、ListIdentities を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_ses->listidentities(  
    iv_identitytype = iv_identity_type  
    iv_maxitems = iv_max_items  
  ).  
  ot_identities = lo_result->get_identities( ).  
  MESSAGE 'Identities retrieved successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ListIdentities](#)を参照してください。

ListReceiptFilters

次の例は、ListReceiptFilters を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_ses->listreceiptfilters( ).  
  ot_filters = lo_result->get_filters( ).
```

```
MESSAGE 'Receipt filters retrieved successfully' TYPE 'I'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListReceiptFilters](#)」を参照してください。

ListTemplates

次の例は、ListTemplates を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.
DATA(lo_result) = lo_ses->listtemplates( iv_maxitems = iv_max_items ).
ot_templates = lo_result->get_templatesmetadata( ).
MESSAGE 'Templates retrieved successfully' TYPE 'I'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListTemplates](#)」を参照してください。

SendEmail

次の例は、SendEmail を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
" Create message object
DATA(lo_subject) = NEW /aws1/cl_sescontent( iv_data = iv_subject ).
DATA(lo_text_body) = NEW /aws1/cl_sescontent( iv_data = iv_text ).
DATA(lo_html_body) = NEW /aws1/cl_sescontent( iv_data = iv_html ).
DATA(lo_body) = NEW /aws1/cl_sesbody(
  io_text = lo_text_body
  io_html = lo_html_body
).
DATA(lo_message) = NEW /aws1/cl_sesmessage(
  io_subject = lo_subject
  io_body = lo_body
).

TRY.
  " Send email
  DATA(lo_result) = lo_ses->sendemail(
    iv_source = iv_source
    io_destination = io_destination
    io_message = lo_message
    it_replytoaddresses = it_reply_tos
  ).
  ov_msg_id = lo_result->get_messageid( ).
  MESSAGE 'Email sent successfully' TYPE 'I'.
CATCH /aws1/cx_sesacctsendingpause00 INTO DATA(lo_ex1).
  DATA(lv_error) = |Account sending paused: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_sesmessagerejected INTO DATA(lo_ex2).
  lv_error = |Message rejected: { lo_ex2->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex2.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
```

```
    RAISE EXCEPTION lo_ex_generic.  
  ENDRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[SendEmail](#)」を参照してください。

SendTemplatedEmail

次の例は、SendTemplatedEmail を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  " Send templated email  
  DATA(lo_result) = lo_ses->sendtemplatedemail(  
    iv_source = iv_source  
    io_destination = io_destination  
    iv_template = iv_template_name  
    iv_templatedata = iv_template_data  
    it_replytoaddresses = it_reply_tos  
  ).  
  ov_msg_id = lo_result->get_messageid( ).  
  MESSAGE 'Templated email sent successfully' TYPE 'I'.  
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).  
  DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex1.  
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).  
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex_generic.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[SendTemplatedEmail](#)を参照してください。

UpdateTemplate

次の例は、UpdateTemplate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
  lo_ses->updatetemplate( io_template = lo_template ).
  MESSAGE 'Template updated successfully' TYPE 'I'.
CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
  DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
  lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex2.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[UpdateTemplate](#)」を参照してください。

VerifyDomainIdentity

次の例は、VerifyDomainIdentity を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    DATA(lo_result) = lo_ses->verifydomainidentity( iv_domain =  
iv_domain_name ).  
    ov_token = lo_result->get_verificationtoken( ).  
    MESSAGE 'Domain verification initiated' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[VerifyDomainIdentity](#)」を参照してください。

VerifyEmailIdentity

次の例は、VerifyEmailIdentity を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_ses->verifyemailidentity( iv_emailaddress = iv_email_address ).  
    MESSAGE 'Email verification initiated' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[VerifyEmailIdentity](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon SES API v2 の例

次のコード例は、Amazon SES API v2 で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateContact

次の例は、CreateContact を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  lo_se2->createcontact(  
    iv_contactlistname = iv_contact_list_name  
    iv_emailaddress = iv_email_address ).  
  MESSAGE 'Contact created successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2alreadyexistsex.  
  MESSAGE 'Contact already exists.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex.  
  MESSAGE 'Bad request.' TYPE 'E'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Contact list not found.' 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「[CreateContact](#)」を参照してください。

CreateContactList

次の例は、CreateContactList を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  lo_se2->createcontactlist(  
    iv_contactlistname = iv_contact_list_name ).  
  MESSAGE 'Contact list created successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2alreadyexistsex.  
  MESSAGE 'Contact list already exists.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request - contact list limit may be reached.' TYPE 'I'.  
  " Re-raise the exception so the caller can handle it  
  RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).  
  MESSAGE 'Limit exceeded - contact list limit reached.' TYPE 'I'.  
  " Re-raise the exception so the caller can handle it  
  RAISE EXCEPTION lo_limit_exceeded.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateContactList](#)」を参照してください。

CreateEmailIdentity

次の例は、CreateEmailIdentity を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_se2->createemailidentity(
    iv_emailidentity = iv_email_identity ).
  MESSAGE 'Email identity created successfully.' TYPE 'I'.
CATCH /aws1/cx_se2alreadyexistsex.
  MESSAGE 'Email identity already exists.' TYPE 'I'.
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
  MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
  MESSAGE lo_limit_exceeded TYPE 'I' DISPLAY LIKE 'E'.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [CreateEmailIdentity](#) を参照してください。

CreateEmailTemplate

次の例は、CreateEmailTemplate を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_template_content) = NEW /aws1/cl_se2emailtmplcontent(
    iv_subject = iv_subject
    iv_html = iv_html
    iv_text = iv_text ).

  lo_se2->createemailtemplate(
    iv_templatename = iv_template_name
    io_templatecontent = lo_template_content ).
  MESSAGE 'Email template created successfully.' TYPE 'I'.
CATCH /aws1/cx_se2alreadyexistsex.
  MESSAGE 'Email template already exists.' TYPE 'I'.
CATCH /aws1/cx_se2badrequestex.
  MESSAGE 'Bad request.' TYPE 'E'.
```

```
CATCH /aws1/cx_se2limitexceededex.  
  MESSAGE 'Limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP [CreateEmailTemplate](#)」を参照してください。

DeleteContactList

次の例は、DeleteContactList を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  lo_se2->deletecontactlist(  
    iv_contactlistname = iv_contact_list_name ).  
  MESSAGE 'Contact list deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Contact list not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteContactList](#)」を参照してください。

DeleteEmailIdentity

次の例は、DeleteEmailIdentity を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  lo_se2->deleteemailidentity(  
    iv_emailidentity = iv_email_identity ).  
  MESSAGE 'Email identity deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Email identity not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteEmailIdentity](#)」を参照してください。

DeleteEmailTemplate

次の例は、DeleteEmailTemplate を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_se2->deleteemailtemplate(  
    iv_templatename = iv_template_name ).  
  MESSAGE 'Email template deleted successfully.' TYPE 'I'.  
ENDTRY.
```

```
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Email template not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteEmailTemplate](#)を参照してください。

ListContacts

次の例は、ListContacts を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_se2->listcontacts(  
    iv_contactlistname = iv_contact_list_name ).  
  DATA(lv_count) = lines( oo_result->get_contacts( ) ).  
  MESSAGE |Retrieved { lv_count } contacts from list.| TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).  
  MESSAGE 'Contact list not found.' TYPE 'I'.  
  RAISE EXCEPTION lo_not_found.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListContacts](#)を参照してください。

SendEmail

次の例は、SendEmail を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

メッセージを送信します。

```
TRY.
  " Create destination with recipient address
  DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
  APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
TO lt_to_addresses.
  DATA(lo_destination) = NEW /aws1/cl_se2destination(
    it_toaddresses = lt_to_addresses ).

  " Create message content
  DATA(lo_subject) = NEW /aws1/cl_se2content( iv_data = iv_subject ).
  DATA(lo_text_body) = NEW /aws1/cl_se2content( iv_data = iv_text_body ).
  DATA(lo_html_body) = NEW /aws1/cl_se2content( iv_data = iv_html_body ).
  DATA(lo_body) = NEW /aws1/cl_se2body(
    io_text = lo_text_body
    io_html = lo_html_body ).
  DATA(lo_message) = NEW /aws1/cl_se2message(
    io_subject = lo_subject
    io_body = lo_body ).

  DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
    io_simple = lo_message ).

  " Send the email
  lo_se2->sendemail(
    iv_fromemailaddress = iv_from_email_address
    io_destination = lo_destination
    io_content = lo_content ).
  MESSAGE 'Email sent successfully.' TYPE 'I'.
```

```

CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
  MESSAGE 'Account suspended.' TYPE 'I'.
  RAISE EXCEPTION lo_account_suspended.
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
  MESSAGE 'Bad request.' TYPE 'I'.
  RAISE EXCEPTION lo_bad_request.
CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
  MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
  RAISE EXCEPTION lo_message_rejected.
ENDTRY.

```

テンプレートを使用してメッセージを送信します。

```

TRY.
  " Create destination with recipient address
  DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
  APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
TO lt_to_addresses.
  DATA(lo_destination) = NEW /aws1/cl_se2destination(
    it_toaddresses = lt_to_addresses ).

  " Create template reference
  DATA(lo_template) = NEW /aws1/cl_se2template(
    iv_templatename = iv_template_name
    iv_templatedata = iv_template_data ).

  DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
    io_template = lo_template ).

  " Create list management options
  DATA(lo_list_mgmt) = NEW /aws1/cl_se2listmanagementopts(
    iv_contactlistname = iv_contact_list_name ).

  " Send the email using template
  lo_se2->sendemail(
    iv_fromemailaddress = iv_from_email_address
    io_destination = lo_destination
    io_content = lo_content
    io_listmanagementoptions = lo_list_mgmt ).
  MESSAGE 'Email sent using template successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).

```

```
MESSAGE 'Account suspended.' TYPE 'I'.
RAISE EXCEPTION lo_account_suspended.
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
MESSAGE 'Bad request.' TYPE 'I'.
RAISE EXCEPTION lo_bad_request.
CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
RAISE EXCEPTION lo_message_rejected.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[SendEmail](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon SNS の例

次のコード例は、Amazon SNS で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

CreateTopic

次のコード例は、CreateTopic を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sns->createtopic( iv_name = iv_topic_name ). " oo_result is  
returned for testing purposes. "  
    MESSAGE 'SNS topic created' TYPE 'I'.  
    CATCH /aws1/cx_snstopiclimitexcdex.  
        MESSAGE 'Unable to create more topics. You have reached the maximum number  
of topics allowed.' 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「[CreateTopic](#)」を参照してください。

DeleteTopic

次のコード例は、DeleteTopic を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_sns->deletetopic( iv_topicarn = iv_topic_arn ).  
    MESSAGE 'SNS topic deleted.' TYPE 'I'.  
    CATCH /aws1/cx_snsnotfoundexception.  
        MESSAGE 'Topic does not exist.' 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「[DeleteTopic](#)」を参照してください。

GetTopicAttributes

次のコード例は、GetTopicAttributes を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sns->gettopicattributes( iv_topicarn = iv_topic_arn ). "  
oo_result is returned for testing purposes. "  
    DATA(lt_attributes) = oo_result->get_attributes( ).  
    MESSAGE 'Retrieved attributes/properties of a topic.' TYPE 'I'.  
    CATCH /aws1/cx_snsnotfoundexception.  
        MESSAGE 'Topic does not exist.' 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「[GetTopicAttributes](#)」を参照してください。

ListSubscriptions

次のコード例は、ListSubscriptions を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_sns->listsubscriptions( ).           " oo_result is
returned for testing purposes. "
    DATA(lt_subscriptions) = oo_result->get_subscriptions( ).
    MESSAGE 'Retrieved list of subscribers.' TYPE 'I'.
    CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list subscribers.' 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「[ListSubscriptions](#)」を参照してください。

ListTopics

次のコード例は、ListTopics を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_sns->listtopics( ).                 " oo_result is returned for
testing purposes. "
    DATA(lt_topics) = oo_result->get_topics( ).
    MESSAGE 'Retrieved list of topics.' TYPE 'I'.
    CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list topics.' 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「[ListTopics](#)」を参照してください。

Publish

次のコード例は、Publish を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sns->publish(           " oo_result is returned for  
testing purposes. "  
        iv_topicarn = iv_topic_arn  
        iv_message = iv_message ).  
    MESSAGE 'Message published to SNS topic.' TYPE 'I'.  
CATCH /aws1/cx_snsnotfoundexception.  
    MESSAGE 'Topic does not exist.' TYPE 'E'.  
ENDTRY.
```

属性を含むメッセージをトピックに発行します。

```
TRY.  
    oo_result = lo_sns->publish(           " oo_result is returned for  
testing purposes. "  
        iv_topicarn = iv_topic_arn  
        iv_message = iv_message  
        it_messageattributes = it_msg_attrs ).  
    MESSAGE 'Message with attributes published to SNS topic.' TYPE 'I'.  
CATCH /aws1/cx_snsnotfoundexception.  
    MESSAGE 'Topic does not exist.' TYPE 'E'.  
ENDTRY.
```

マルチフォーマットメッセージをトピックに発行します。

```
" Build JSON message structure for multi-format message
```

```
DATA(lv_json_message) = |\{ "default": "{ iv_default_message }", "sms":
"{ iv_sms_message }", "email": "{ iv_email_message }" \}|.

TRY.
    oo_result = lo_sns->publish(
testing purposes. " " oo_result is returned for
    iv_topicarn = iv_topic_arn
    iv_message = lv_json_message
    iv_subject = iv_subject
    iv_messagestructure = 'json' ).
    MESSAGE 'Multi-format message published to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' 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「[Publish](#)」を参照してください。

SetSubscriptionAttributes

次の例は、SetSubscriptionAttributes を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_sns->setsubscriptionattributes(
    iv_subscriptionarn = iv_subscription_arn
    iv_attributename = 'FilterPolicy'
    iv_attributevalue = iv_filter_policy ).
    MESSAGE 'Added filter policy to subscription.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Subscription does not exist.' 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「[SetSubscriptionAttributes](#)」を参照してください。

SetTopicAttributes

次の例は、SetTopicAttributes を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_sns->settopicattributes(  
    iv_topicarn = iv_topic_arn  
    iv_attributename = iv_attribute_name  
    iv_attributevalue = iv_attribute_value ).  
  MESSAGE 'Set/updated SNS topic attributes.' TYPE 'I'.  
  CATCH /aws1/cx_snsnotfoundexception.  
    MESSAGE 'Topic does not exist.' 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「[SetTopicAttributes](#)」を参照してください。

Subscribe

次のコード例は、Subscribe を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

E メールアドレスをトピックにサブスクライブします。

```
TRY.
    oo_result = lo_sns->subscribe(
        iv_topicarn = iv_topic_arn
        iv_protocol = 'email'
        iv_endpoint = iv_email_address
        iv_returnsubscriptionarn = abap_true ).
    MESSAGE 'Email address subscribed to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
CATCH /aws1/cx_snssubscriptionlmt00.
    MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' 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「[Subscribe](#)」を参照してください。

Unsubscribe

次のコード例は、Unsubscribe を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_sns->unsubscribe( iv_subscriptionarn = iv_subscription_arn ).
    MESSAGE 'Subscription deleted.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Subscription does not exist.' TYPE 'E'.
CATCH /aws1/cx_snsinvalidparameterex.
```

```
MESSAGE 'Subscription with "PendingConfirmation" status cannot be deleted/
unsubscribed. Confirm subscription before performing unsubscribe operation.' 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「[Unsubscribe](#)」を参照してください。

シナリオ

FIFO トピックを作成して発行する

次のコード例は、FIFO Amazon SNS トピックを作成、発行する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

FIFO トピックを作成し、そのトピックに Amazon SQS FIFO キューをサブスクライブして、Amazon SNS トピックにメッセージを発行します。

```
" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsm_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsm_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsm_w( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

TRY.
  DATA(lo_create_result) = lo_sns->createtopic(
    iv_name = iv_topic_name
    it_attributes = lt_tpc_attributes ).
  DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
```

```

        ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
        MESSAGE 'FIFO topic created' TYPE 'I'.
        CATCH /aws1/cx_snsstopiclimitexcdex.
        MESSAGE 'Unable to create more topics. You have reached the maximum number
of topics allowed.' TYPE 'E'.
        ENDRTRY.

" Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
topic. "
" Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
an SNS FIFO topic. "
        TRY.
            DATA(lo_subscribe_result) = lo_sns->subscribe(
                iv_topicarn = lv_topic_arn
                iv_protocol = 'sqs'
                iv_endpoint = iv_queue_arn ).
            DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
            ov_subscription_arn = lv_subscription_arn.
ov_subscription_arn is returned for testing purposes. "
            MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
            CATCH /aws1/cx_snsnotfoundexception.
            MESSAGE 'Topic does not exist.' TYPE 'E'.
            CATCH /aws1/cx_snssubscriptionlmt00.
            MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
            ENDRTRY.

" Publish message to SNS topic. "
        TRY.
            DATA lt_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>tt_messageattributemap.
            DATA ls_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>ts_messageattributemap_maprow.
            ls_msg_attributes-key = 'Importance'.
            ls_msg_attributes-value = NEW /aws1/cl_snsmessageattrvalue( iv_datatype =
'String'
                                                                    iv_stringvalue =
'High' ).
            INSERT ls_msg_attributes INTO TABLE lt_msg_attributes.

            DATA(lo_result) = lo_sns->publish(
                iv_topicarn = lv_topic_arn

```

```

        iv_message = 'The price of your mobile plan has been increased from $19
to $23'
        iv_subject = 'Changes to mobile plan'
        iv_messagegroupid = 'Update-2'
        iv_messagededuplicationid = 'Update-2.1'
        it_messageattributes = lt_msg_attributes ).
    ov_message_id = lo_result->get_messageid( ).
ov_message_id is returned for testing purposes. "
    MESSAGE 'Message was published to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' 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。
 - [CreateTopic](#)
 - [Publish](#)
 - [Subscribe](#)

SMS テキストメッセージを発行する

次のコードサンプルは、Amazon SNS を使用して SMS メッセージを発行する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

" iv_phone_number = '+12065550101' - Phone number in E.164 format
TRY.
    oo_result = lo_sns->publish(
testing purposes. "
        iv_phonenumber = iv_phone_number
        iv_message = iv_message ).
    MESSAGE 'Message published to phone number.' TYPE 'I'.

```

```
CATCH /aws1/cx_snsnotfoundexception.  
    MESSAGE 'Phone number does not exist.' 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「[Publish](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon SQS の例

次のコード例は、Amazon SQS で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

CreateQueue

次のコード例は、CreateQueue を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

Amazon SQS 標準キューを作成します。

```

TRY.
    oo_result = lo_sqs->createqueue( iv_queue_name = iv_queue_name ).      "
oo_result is returned for testing purposes. "
    MESSAGE 'SQS queue created.' TYPE 'I'.
    CATCH /aws1/cx_sqsqueuedeletedrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
    CATCH /aws1/cx_sqsqueueexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
ENDTRY.

```

メッセージが到着するまで待機する Amazon SQS キューを作成します。

```

TRY.
    DATA lt_attributes TYPE /aws1/cl_sqsqueueattrmap_w=>tt_queueattributemap.
    DATA ls_attribute TYPE /aws1/
cl_sqsqueueattrmap_w=>ts_queueattributemap_maprow.
    ls_attribute-key = 'ReceiveMessageWaitTimeSeconds'.      " Time in
seconds for long polling, such as how long the call waits for a message to arrive
in the queue before returning. "
    ls_attribute-value = NEW /aws1/cl_sqsqueueattrmap_w( iv_value =
iv_wait_time ).
    INSERT ls_attribute INTO TABLE lt_attributes.
    oo_result = lo_sqs->createqueue(      " oo_result is returned
for testing purposes. "
        iv_queue_name = iv_queue_name
        it_attributes = lt_attributes ).
    MESSAGE 'SQS queue created.' TYPE 'I'.
    CATCH /aws1/cx_sqsqueuedeletedrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
    CATCH /aws1/cx_sqsqueueexists.
        MESSAGE 'A queue with this name already exists.' 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「[CreateQueue](#)」を参照してください。

DeleteMessage

次の例は、DeleteMessage を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_sqs->deletemessage(  
        iv_queueurl = iv_queue_url  
        iv_receipthandle = iv_receipt_handle ).  
    MESSAGE 'Message deleted from SQS queue.' TYPE 'I'.  
CATCH /aws1/cx_sqsinvalididformat.  
    MESSAGE 'The specified receipt handle is not valid.' TYPE 'E'.  
CATCH /aws1/cx_sqsreceipthandleisinv.  
    MESSAGE 'The specified receipt handle is not valid for the current version.'  
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[DeleteMessage](#)」を参照してください。

DeleteMessageBatch

次の例は、DeleteMessageBatch を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_sqs->deletemessagebatch(      " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        it_entries = it_entries ).
    MESSAGE 'Messages deleted from SQS queue.' TYPE 'I'.
CATCH /aws1/cx_sqsbtcentsnotdist00.
    MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
CATCH /aws1/cx_sqsemtypbatchrequest.
    MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
CATCH /aws1/cx_sqsinvbatchentryid.
    MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
CATCH /aws1/cx_sqstoomanyentriesin00.
    MESSAGE 'The batch request contains more entries than allowed.' 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[DeleteMessageBatch](#)」を参照してください。

DeleteQueue

次の例は、DeleteQueue を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_sqs->deletequeue( iv_queueurl = iv_queue_url ).
    MESSAGE 'SQS queue deleted' TYPE 'I'.
ENDTRY.
```

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

GetQueueUrl

次のコード例は、GetQueueUrl を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sqs->getqueueurl( iv_queuename = iv_queue_name ).      "  
oo_result is returned for testing purposes. "  
    MESSAGE 'Queue URL retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_sqsqueuedoesnotexist.  
        MESSAGE 'The requested queue does not exist.' 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「[GetQueueUrl](#)」を参照してください。

ListQueues

次のコード例は、ListQueues を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_sqs->listqueues( ).      " oo_result is returned for  
testing purposes. "  
    MESSAGE 'Retrieved list of queues.' TYPE 'I'.  
ENDTRY.
```

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

ReceiveMessage

次のコード例は、ReceiveMessage を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

Amazon SQS キューからメッセージを受信します。

```
TRY.  
    oo_result = lo_sqs->receivemessage( iv_queueurl = iv_queue_url ).      "  
oo_result is returned for testing purposes. "  
    DATA(lt_messages) = oo_result->get_messages( ).  
    MESSAGE 'Message received from SQS queue.' TYPE 'I'.  
    CATCH /aws1/cx_sqsoverlimit.  
        MESSAGE 'Maximum number of in-flight messages reached.' TYPE 'E'.  
ENDTRY.
```

ロングポーリングサポートを使用して Amazon SQS キューからメッセージを受信します。

```
TRY.  
    oo_result = lo_sqs->receivemessage(      " oo_result is returned for  
testing purposes. "  
        iv_queueurl = iv_queue_url
```

```

        iv_waitemptyseconds = iv_wait_time ).    " Time in seconds for long
polling, such as how long the call waits for a message to arrive in the queue
before returning. " ).
    DATA(lt_messages) = oo_result->get_messages( ).
    MESSAGE 'Message received from SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsoverlimit.
        MESSAGE 'Maximum number of in-flight messages reached.' 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「[ReceiveMessage](#)」を参照してください。

SendMessage

次のコード例は、SendMessage を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

    TRY.
        oo_result = lo_sqs->sendmessage(          " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        iv_messagebody = iv_message ).
        MESSAGE 'Message sent to SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsinvalidmsgconts.
        MESSAGE 'Message contains non-valid characters.' TYPE 'E'.
    CATCH /aws1/cx_sqsunsupportedop.
        MESSAGE 'Operation not supported.' 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「[SendMessage](#)」を参照してください。

SendMessageBatch

次の例は、SendMessageBatch を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    oo_result = lo_sqs->sendmessagebatch(           " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        it_entries = it_messages ).
    MESSAGE 'Messages sent to SQS queue.' TYPE 'I'.
CATCH /aws1/cx_sqsbcentidsnotdist00.
    MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
CATCH /aws1/cx_sqsbbatchreqtoolong.
    MESSAGE 'The length of all the messages put together is more than the
limit.' TYPE 'E'.
CATCH /aws1/cx_sqsemptybatchrequest.
    MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
CATCH /aws1/cx_sqsinvbatchentryid.
    MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
CATCH /aws1/cx_sqstoomanyentriesin00.
    MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
CATCH /aws1/cx_sqsunsupportedop.
    MESSAGE 'Operation not supported.' 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[SendMessageBatch](#)を参照してください。

シナリオ

FIFO トピックを作成して発行する

次のコード例は、FIFO Amazon SNS トピックを作成、発行する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

FIFO トピックを作成し、そのトピックに Amazon SQS FIFO キューをサブスクライブして、Amazon SNS トピックにメッセージを発行します。

```
" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsmmap_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmmap_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmmap_w( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

TRY.
  DATA(lo_create_result) = lo_sns->createtopic(
    iv_name = iv_topic_name
    it_attributes = lt_tpc_attributes ).
  DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
  ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
  MESSAGE 'FIFO topic created' TYPE 'I'.
CATCH /aws1/cx_snstopiclimitexcdex.
  MESSAGE 'Unable to create more topics. You have reached the maximum number
of topics allowed.' TYPE 'E'.
ENDTRY.

" Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
topic. "
```

```

    " Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
    an SNS FIFO topic. "
    TRY.
        DATA(lo_subscribe_result) = lo_sns->subscribe(
            iv_topicarn = lv_topic_arn
            iv_protocol = 'sqs'
            iv_endpoint = iv_queue_arn ).
        DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
        ov_subscription_arn = lv_subscription_arn.
    "
    ov_subscription_arn is returned for testing purposes. "
    MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    CATCH /aws1/cx_snssubscriptionlmt00.
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
    ENDTRY.

    " Publish message to SNS topic. "
    TRY.
        DATA lt_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>tt_messageattributemap.
        DATA ls_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>ts_messageattributemap_maprow.
        ls_msg_attributes-key = 'Importance'.
        ls_msg_attributes-value = NEW /aws1/cl_snsmessageattrvalue( iv_datatype =
'String'
                                                                    iv_stringvalue =
'String' ).
        INSERT ls_msg_attributes INTO TABLE lt_msg_attributes.

        DATA(lo_result) = lo_sns->publish(
            iv_topicarn = lv_topic_arn
            iv_message = 'The price of your mobile plan has been increased from $19
to $23'
            iv_subject = 'Changes to mobile plan'
            iv_messagegroupid = 'Update-2'
            iv_messagededuplicationid = 'Update-2.1'
            it_messageattributes = lt_msg_attributes ).
        ov_message_id = lo_result->get_messageid( ).
    "
    ov_message_id is returned for testing purposes. "
    MESSAGE 'Message was published to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' 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。
 - [CreateTopic](#)
 - [Publish](#)
 - [Subscribe](#)

SDK for SAP ABAP を使用した Step Functions の例

次のコード例は、Step Functions で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateActivity

次の例は、CreateActivity を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_sfn->createactivity(  
    iv_name = iv_name  
  ).  
  ov_activity_arn = lo_result->get_activityarn( ).  
  MESSAGE 'Activity created successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnactivityalrddyex.  
  MESSAGE 'Activity already exists.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidname.  
  MESSAGE 'Invalid activity name.' TYPE 'E'.  
CATCH /aws1/cx_sfnactivitylimitexcd.  
  MESSAGE 'Activity limit exceeded.' 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「[CreateActivity](#)」を参照してください。

CreateStateMachine

次の例は、CreateStateMachine を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_sfn->createstatemachine(  
    iv_name = iv_name  
    iv_definition = iv_definition  
    iv_rolearn = iv_role_arn  
  ).  
  ov_state_machine_arn = lo_result->get_statemachearn( ).  
  MESSAGE 'State machine created successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnstatemachearlex.  
  MESSAGE 'State machine already exists.' TYPE 'E'.  
CATCH /aws1/cx_sfninvaliddefinition.
```

```
MESSAGE 'Invalid state machine definition.' TYPE 'E'.
CATCH /aws1/cx_sfninvalidname.
MESSAGE 'Invalid state machine name.' TYPE 'E'.
CATCH /aws1/cx_sfninvalidarn.
MESSAGE 'Invalid role ARN.' 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「[CreateStateMachine](#)」を参照してください。

DeleteActivity

次の例は、DeleteActivity を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_sfn->deleteactivity(
    iv_activityarn = iv_activity_arn
  ).
  MESSAGE 'Activity deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_sfninvalidarn.
  MESSAGE 'Invalid activity ARN.' 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[DeleteActivity](#)」を参照してください。

DeleteStateMachine

次の例は、DeleteStateMachine を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_sfn->deletestatemachine(  
    iv_statemachearn = iv_state_machine_arn  
  ).  
  MESSAGE 'State machine deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid state machine ARN.' TYPE 'E'.  
CATCH /aws1/cx_sfnvalidationex.  
  MESSAGE 'Validation error occurred.' 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[DeleteStateMachine](#)を参照してください。

DescribeExecution

次の例は、DescribeExecution を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  oo_result = lo_sfn->describeexecution(  
    iv_executionarn = iv_execution_arn  
  ).  
  MESSAGE 'Execution described successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_sfnextdoesnotexist.  
  MESSAGE 'Execution does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid execution ARN.' 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「[DescribeExecution](#)」を参照してください。

DescribeStateMachine

次の例は、DescribeStateMachine を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  oo_result = lo_sfn->describestatemachine(  
    iv_statemachearn = iv_state_machine_arn  
  ).  
  MESSAGE 'State machine described successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnstatemachinedoes00.  
  MESSAGE 'State machine does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid state machine ARN.' 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「[DescribeStateMachine](#)」を参照してください。

GetActivityTask

次の例は、GetActivityTask を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
    oo_result = lo_sfn->getactivitytask(  
        iv_activityarn = iv_activity_arn  
    ).  
    MESSAGE 'Activity task retrieved successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnactivitydoesnotex.  
    MESSAGE 'Activity does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
    MESSAGE 'Invalid activity ARN.' TYPE 'E'.  
CATCH /aws1/cx_sfnactivityworkerlm00.  
    MESSAGE 'Activity worker limit exceeded.' 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[GetActivityTask](#)を参照してください。

ListActivities

次の例は、ListActivities を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    DATA(lo_result) = lo_sfn->listactivities( ).
```

```
DATA(lt_activities) = lo_result->get_activities( ).
LOOP AT lt_activities INTO DATA(lo_activity).
  IF lo_activity->get_name( ) = iv_name.
    ov_activity_arn = lo_activity->get_activityarn( ).
    EXIT.
  ENDIF.
ENDLOOP.
MESSAGE 'Activities listed successfully.' TYPE 'I'.
CATCH /aws1/cx_sfninvalidtoken.
  MESSAGE 'Invalid pagination token.' 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[ListActivities](#)を参照してください。

ListStateMachines

次の例は、ListStateMachines を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_result) = lo_sfn->liststatemachines( ).
  DATA(lt_state_machines) = lo_result->get_statemachines( ).
  LOOP AT lt_state_machines INTO DATA(lo_state_machine).
    IF lo_state_machine->get_name( ) = iv_name.
      ov_state_machine_arn = lo_state_machine->get_statemachinearn( ).
      EXIT.
    ENDIF.
  ENDLOOP.
  MESSAGE 'State machines listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid pagination token.' 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[ListStateMachines](#)」を参照してください。

SendTaskSuccess

次の例は、SendTaskSuccess を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    lo_sfn->sendtasksuccess(  
        iv_tasktoken = iv_task_token  
        iv_output = iv_task_response  
    ).  
    MESSAGE 'Task success sent successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfninvalidtoken.  
    MESSAGE 'Invalid task token.' TYPE 'E'.  
CATCH /aws1/cx_sfntaskdoesnotexist.  
    MESSAGE 'Task does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidoutput.  
    MESSAGE 'Invalid task output.' TYPE 'E'.  
CATCH /aws1/cx_sfntasktimedout.  
    MESSAGE 'Task timed out.' 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[SendTaskSuccess](#)」を参照してください。

StartExecution

次の例は、StartExecution を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_sfn->startexecution(  
    iv_statemachinearn = iv_state_machine_arn  
    iv_input = iv_input  
  ).  
  ov_execution_arn = lo_result->get_executionarn( ).  
  MESSAGE 'Execution started successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnstatemachinedoes00.  
  MESSAGE 'State machine does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid state machine ARN.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidexecinput.  
  MESSAGE 'Invalid execution input.' TYPE 'E'.  
CATCH /aws1/cx_sfnexeclimitexceeded.  
  MESSAGE 'Execution limit exceeded.' 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[StartExecution](#)」を参照してください。

SDK for SAP ABAP を使用した Systems Manager の例

次のコード例は、Systems Manager で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック


- [アクション](#)

アクション

CreateDocument

次の例は、CreateDocument を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ssm->createdocument(  
    iv_name = iv_name  
    iv_content = iv_content  
    iv_documenttype = 'Command' ).  
  MESSAGE 'Document created.' TYPE 'I'.  
CATCH /aws1/cx_ssmdocalreadyexists.  
  MESSAGE 'Document already exists.' TYPE 'I'.  
CATCH /aws1/cx_ssminvaliddoccontent.  
  MESSAGE 'Invalid document content.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateDocument](#)」を参照してください。

CreateMaintenanceWindow

次の例は、CreateMaintenanceWindow を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ssm->createmaintenancewindow(  
        iv_name = iv_name  
        iv_schedule = iv_schedule  
        iv_duration = iv_duration  
        iv_cutoff = iv_cutoff  
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).  
    MESSAGE 'Maintenance window created.' TYPE 'I'.  
CATCH /aws1/cx_ssmresrclimitexcdex.  
    MESSAGE 'Resource limit exceeded.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateMaintenanceWindow](#)」を参照してください。

CreateOpsItem

次の例は、CreateOpsItem を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_ssm->createopsitem(  
        iv_title = iv_title
```

```
        iv_source = iv_source
        iv_category = iv_category
        iv_severity = iv_severity
        iv_description = iv_description ).
    MESSAGE 'OpsItem created.' TYPE 'I'.
CATCH /aws1/cx_ssmopsitemlimitexcdex.
    MESSAGE 'You have exceeded your open OpsItem limit.' TYPE 'I'.
CATCH /aws1/cx_ssmopsitemalrddyexex.
    MESSAGE 'OpsItem already exists.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateOpsItem](#)」を参照してください。

DeleteDocument

次の例は、DeleteDocument を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_ssm->deletedocument( iv_name = iv_name ).
    MESSAGE 'Document deleted.' TYPE 'I'.
CATCH /aws1/cx_ssminvaliddocument.
    MESSAGE 'Invalid document.' TYPE 'I'.
CATCH /aws1/cx_ssmassocinstances.
    MESSAGE 'Document has associated instances.' TYPE 'I'.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DeleteDocument](#)」を参照してください。

DeleteMaintenanceWindow

次の例は、DeleteMaintenanceWindow を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
lo_ssm->deletemaintenancewindow( iv_windowid = iv_window_id ).  
MESSAGE 'Maintenance window deleted.' TYPE 'I'.  
CATCH /aws1/cx_ssminternalservererr.  
MESSAGE 'Internal server error occurred.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[DeleteMaintenanceWindow](#)」を参照してください。

DeleteOpsItem

次の例は、DeleteOpsItem を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

TRY.

```
lo_ssm->deleteopsitem( iv_opsitemid = iv_ops_item_id ).  
MESSAGE 'OpsItem deleted.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsiteminvparamex.
```

```
MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DeleteOpsItem](#)」を参照してください。

DescribeDocument

次の例は、DescribeDocument を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
TRY.  
  DATA(lo_result) = lo_ssm->describedocument( iv_name = iv_name ).  
  DATA(lo_document) = lo_result->get_document( ).  
  IF lo_document IS BOUND.  
    rv_status = lo_document->get_status( ).  
    MESSAGE |Document status: { rv_status }| TYPE 'I'.  
  ENDIF.  
CATCH /aws1/cx_ssminvaliddocument.  
  MESSAGE 'Invalid document.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeDocument](#)」を参照してください。

DescribeOpsItems

次の例は、DescribeOpsItems を使用する方法を説明しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Create filter for OpsItem ID
  DATA(lt_filters) = VALUE /aws1/cl_ssmopsitemfilter=>tt_opsitemfilters(
    ( NEW /aws1/cl_ssmopsitemfilter(
      iv_key = 'OpsItemId'
      it_values = VALUE /aws1/
cl_ssmopsitemfiltvals_w=>tt_opsitemfiltervalues(
      ( NEW /aws1/cl_ssmopsitemfiltvals_w( iv_value = iv_ops_item_id ) )
    )
    iv_operator = 'Equal'
  ) )
  ).

  " Use paginator to get all results
  DATA(lo_paginator) = lo_ssm->get_paginator( ).
  DATA(lo_iterator) = lo_paginator->describeopsitems(
    it_opsitemfilters = lt_filters ).

  rv_found = abap_false.

  WHILE lo_iterator->has_next( ).
    DATA(lo_result) = CAST /aws1/cl_ssmdescropsitemsrsp( lo_iterator-
>get_next( ) ).
    LOOP AT lo_result->get_opsitemsummaries( ) INTO DATA(lo_item).
      DATA(lv_title) = lo_item->get_title( ).
      DATA(lv_status) = lo_item->get_status( ).
      MESSAGE |The OpsItem title is { lv_title } and the status is
{ lv_status }| TYPE 'I'.
      rv_found = abap_true.
    ENDLOOP.
  ENDWHILE.
  CATCH /aws1/cx_ssminternalservererr.
    MESSAGE 'Internal server error occurred.' TYPE 'I'.
  ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[DescribeOpsItems](#)」を参照してください。

ListCommandInvocations

次の例は、ListCommandInvocations を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  " Use paginator to get all results
  DATA(lo_paginator) = lo_ssm->get_paginator( ).
  DATA(lo_iterator) = lo_paginator->listcommandinvocations(
    iv_instanceid = iv_instance_id ).

  DATA lv_count TYPE i VALUE 0.

  WHILE lo_iterator->has_next( ).
    DATA(lo_result) = CAST /aws1/cl_ssmlistcmdinvcsresult( lo_iterator-
>get_next( ) ).
    LOOP AT lo_result->get_commandinvocations( ) INTO DATA(lo_invocation).
      lv_count = lv_count + 1.
      DATA(lv_requested_datetime) = lo_invocation->get_requesteddatetime( ).
      MESSAGE |Command invocation requested at: { lv_requested_datetime }|
TYPE 'I'.
    ENDLLOOP.
  ENDWHILE.

  MESSAGE |{ lv_count } command invocation(s) found for instance
{ iv_instance_id }.| TYPE 'I'.
  CATCH /aws1/cx_ssminvalidinstanceid.
    MESSAGE 'Invalid instance ID.' TYPE 'I'.
  CATCH /aws1/cx_ssminvalidcommandid.
```

```
MESSAGE 'Invalid command ID.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[ListCommandInvocations](#)」を参照してください。

SendCommand

次の例は、SendCommand を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  DATA(lo_result) = lo_ssm->sendcommand(  
    it_instanceids = it_instance_ids  
    iv_documentname = iv_document_name  
    iv_timeoutseconds = 3600 ).  
  DATA(lo_command) = lo_result->get_command( ).  
  IF lo_command IS BOUND.  
    rv_command_id = lo_command->get_commandid( ).  
    MESSAGE 'Command sent successfully.' TYPE 'I'.  
  ENDIF.  
  CATCH /aws1/cx_ssminvaliddocument.  
    MESSAGE 'Invalid document.' TYPE 'I'.  
  CATCH /aws1/cx_ssminvalidinstanceid.  
    MESSAGE 'Invalid instance ID.' TYPE 'I'.  
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの「[SendCommand](#)」を参照してください。

UpdateMaintenanceWindow

次の例は、UpdateMaintenanceWindow を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ssm->updatemaintenancewindow(  
    iv_windowid = iv_window_id  
    iv_name = iv_name  
    iv_enabled = iv_enabled  
    iv_schedule = iv_schedule  
    iv_duration = iv_duration  
    iv_cutoff = iv_cutoff  
    iv_allowunassociatedtargets = iv_allow_unassociated_targets ).  
  MESSAGE 'Maintenance window updated.' TYPE 'I'.  
CATCH /aws1/cx_ssmdoesnotexistex.  
  MESSAGE 'Maintenance window does not exist.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[UpdateMaintenanceWindow](#)」を参照してください。

UpdateOpsItem

次の例は、UpdateOpsItem を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_ssm->updateopsitem(  
    iv_opsitemid = iv_ops_item_id  
    iv_title = iv_title  
    iv_description = iv_description  
    iv_status = iv_status ).  
  MESSAGE 'OpsItem updated.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsitemnotfoundex.  
  MESSAGE 'OpsItem not found.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsiteminvparamex.  
  MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[UpdateOpsItem](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon Textract の例

次のコード例は、Amazon Textract で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

AnalyzeDocument

次のコード例は、AnalyzeDocument を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Detects text and additional elements, such as forms or tables,"
"in a local image file or from in-memory byte data."
"The image must be in PNG or JPG format."

"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name      = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_document) = NEW /aws1/cl_texdocument( io_s3object = lo_s3object ).

"Analyze document stored in Amazon S3."
TRY.
  oo_result = lo_tex->analyzedocument(          "oo_result is returned for testing
purposes."
  io_document      = lo_document
```

```
it_featuretypes = lt_featuretypes ).
LOOP AT oo_result->get_blocks( ) INTO DATA(lo_block).
  IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, '.
    MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
  ENDIF.
ENDLOOP.
MESSAGE 'Analyze document completed.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texhlquotaexceededex.
  MESSAGE 'Human loop quota exceeded.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.


CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' 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「[AnalyzeDocument](#)」を参照してください。

DetectDocumentText

次のコード例は、DetectDocumentText を使用する方法を示しています。

SDK for SAP ABAP

 Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Detects text in the input document."
"Amazon Textract can detect lines of text and the words that make up a line of
text."
"The input document must be in one of the following image formats: JPEG, PNG,
PDF, or TIFF."

"Create an ABAP object for the Amazon S3 object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
      iv_name   = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_document) = NEW /aws1/cl_texdocument( io_s3object = lo_s3object ).
"Analyze document stored in Amazon S3."
TRY.
    oo_result = lo_tex->detectdocumenttext( io_document = lo_document ).
"oo_result is returned for testing purposes."
    LOOP AT oo_result->get_blocks( ) INTO DATA(lo_block).
        IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, ' .
            MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
        ENDIF.
    ENDLOOP.
    DATA(lo_metadata) = oo_result->get_documentmetadata( ).
    MESSAGE 'The number of pages in the document is ' && lo_metadata->
ask_pages( ) TYPE 'I'.
    MESSAGE 'Detect document text completed.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
    MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
    MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
    MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
    MESSAGE 'Internal server error.' TYPE 'E'.
```

```
CATCH /aws1/cx_texinvalidparameterex.  
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.  
CATCH /aws1/cx_texinvalids3objectex.  
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.  
CATCH /aws1/cx_texprovthruputexcdex.  
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.  
CATCH /aws1/cx_texthrottlingex.  
  MESSAGE 'The request processing exceeded the limit' TYPE 'E'.  
CATCH /aws1/cx_texunsupporteddocex.  
  MESSAGE 'The document is not supported.' 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「[DetectDocumentText](#)」を参照してください。

GetDocumentAnalysis

次のコード例は、GetDocumentAnalysis を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Gets the results for an Amazon Textract"  
"asynchronous operation that analyzes text in a document."  
TRY.  
  oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).  
"oo_result is returned for testing purposes."  
  WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.  
    IF sy-index = 10.  
      EXIT.          "Maximum 300 seconds."  
    ENDIF.  
    WAIT UP TO 30 SECONDS.  
    oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).  
  ENDWHILE.
```

```
DATA(lt_blocks) = oo_result->get_blocks( ).
LOOP AT lt_blocks INTO DATA(lo_block).
  IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, '.
    MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
  ENDIF.
ENDLOOP.
MESSAGE 'Document analysis retrieved.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidjobidex.
  MESSAGE 'Job ID is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' 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「[GetDocumentAnalysis](#)」を参照してください。

StartDocumentAnalysis

次のコード例は、StartDocumentAnalysis を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

"Starts the asynchronous analysis of an input document for relationships
"between detected items such as key-value pairs, tables, and selection
elements."

"Create ABAP objects for feature type."

"Add TABLES to return information about the tables."

"Add FORMS to return detected form data."

"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

```
DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).
```

"Create an ABAP object for the Amazon S3 object."

```
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name = iv_s3object ).
```

"Create an ABAP object for the document."

```
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
```

"Start async document analysis."

TRY.

```
oo_result = lo_tex->startdocumentanalysis( "oo_result is returned for
testing purposes."
```

```
io_documentlocation = lo_documentlocation
```

```
it_featuretypes = lt_featuretypes ).
```

```
DATA(lv_jobid) = oo_result->get_jobid( ).
```

```
MESSAGE 'Document analysis started.' TYPE 'I'.
```

```
CATCH /aws1/cx_texaccessdeniedex.
```

```
MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
```

```
CATCH /aws1/cx_texbaddocumentex.
```

```
MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
```

```
CATCH /aws1/cx_texdocumenttoolargeex.
```

```
MESSAGE 'The document is too large.' TYPE 'E'.
```

```
CATCH /aws1/cx_texidempotentprmmis00.
```

```
MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
```

```
CATCH /aws1/cx_texinternalservererr.
```

```
MESSAGE 'Internal server error.' TYPE 'E'.
```

```
CATCH /aws1/cx_texinvalidkmskeyex.
```

```
MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
```

```
CATCH /aws1/cx_texinvalidparameterex.
```

```
MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
```

```
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texlimitexceeddex.
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' 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「[StartDocumentAnalysis](#)」を参照してください。

StartDocumentTextDetection

次のコード例は、StartDocumentTextDetection を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Starts the asynchronous detection of text in a document."
"Amazon Textract can detect lines of text and the words that make up a line of
text."

"Create an ABAP object for the Amazon S3 object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name   = iv_s3object ).
"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
"Start document analysis."
TRY.
```

```
oo_result = lo_tex->startdocumenttextdetection( io_documentlocation =
lo_documentlocation ).
DATA(lv_jobid) = oo_result->get_jobid( ).           "oo_result is returned
for testing purposes."
MESSAGE 'Document analysis started.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texidempotentprmmis00.
MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texlimitexceeddex.
MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
MESSAGE 'The document is not supported.' 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「[StartDocumentTextDetection](#)」を参照してください。

シナリオ

ドキュメント分析を開始する

次のコード例は、以下の操作方法を示しています。

- 非同期分析を開始します。

- ドキュメント分析を取得します。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

"Start document analysis."
TRY.
  DATA(lo_start_result) = lo_tex->startdocumentanalysis(
    io_documentlocation = lo_documentlocation
    it_featuretypes = lt_featuretypes ).
  MESSAGE 'Document analysis started.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texidempotentprmmis00.
```

```

    MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
  CATCH /aws1/cx_texinternalservererr.
    MESSAGE 'Internal server error.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidkmskeyex.
    MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidparameterex.
    MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
  CATCH /aws1/cx_texinvalids3objectex.
    MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
  CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
  CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
  CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

"Get job ID from the output."
DATA(lv_jobid) = lo_start_result->get_jobid( ).

"Wait for job to complete."
oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).      " oo_result
is returned for testing purposes. "
WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.
  IF sy-index = 10.
    EXIT.                  "Maximum 300 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).
ENDWHILE.

DATA(lt_blocks) = oo_result->get_blocks( ).
LOOP AT lt_blocks INTO DATA(lo_block).
  IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, '.
    MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
  ENDIF.
ENDLOOP.

```

- 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。

- [GetDocumentAnalysis](#)
- [StartDocumentAnalysis](#)

SDK for SAP ABAP を使用した Amazon Transcribe の例

次のコード例は、Amazon Transcribe で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)

アクション

CreateVocabulary

次の例は、CreateVocabulary を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  IF it_phrases IS NOT INITIAL.  
    oo_result = lo_tnb->createvocabulary(  
      iv_vocabularyname = iv_vocabulary_name  
      iv_languagecode = iv_language_code
```

```
        it_phrases = it_phrases ).
ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        iv_vocabularyfileuri = iv_vocab_file_uri ).
ENDIF.
MESSAGE 'Custom vocabulary created.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
    MESSAGE lo_conflict_ex TYPE 'I'.
    RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの「[CreateVocabulary](#)」を参照してください。

DeleteTranscriptionJob

次の例は、DeleteTranscriptionJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
    lo_tnb->deletetranscriptionjob( iv_job_name ).
    MESSAGE 'Transcription job deleted.' TYPE 'I'.
```

```
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteTranscriptionJob](#)」を参照してください。

DeleteVocabulary

次の例は、DeleteVocabulary を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  lo_tnb->deletevocabulary( iv_vocabulary_name ).
  MESSAGE 'Vocabulary deleted.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex TYPE 'I'.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[DeleteVocabulary](#)」を参照してください。

GetTranscriptionJob

次の例は、GetTranscriptionJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_tnb->gettranscriptionjob( iv_job_name ).  
    DATA(lo_job) = oo_result->get_transcriptionjob( ).  
    MESSAGE 'Retrieved transcription job details.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
    MESSAGE lo_internal_ex TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetTranscriptionJob](#)」を参照してください。

GetVocabulary

次の例は、GetVocabulary を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    oo_result = lo_tnb->getvocabulary( iv_vocabulary_name ).  
    MESSAGE 'Retrieved vocabulary details.' TYPE 'I'.  
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
    CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
    MESSAGE lo_internal_ex TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[GetVocabulary](#)」を参照してください。

ListTranscriptionJobs

次の例は、ListTranscriptionJobs を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
    IF iv_job_filter IS NOT INITIAL.
```

```

        oo_result = lo_tnb->listtranscriptionjobs( iv_jobnamecontains =
iv_job_filter ).
    ELSE.
        oo_result = lo_tnb->listtranscriptionjobs( ).
    ENDIF.
    MESSAGE 'Retrieved transcription jobs list.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.

```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[ListTranscriptionJobs](#)を参照してください。

ListVocabularies

次の例は、ListVocabularies を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

TRY.
    IF iv_vocab_filter IS NOT INITIAL.
        oo_result = lo_tnb->listvocabularies( iv_namecontains = iv_vocab_filter ).
    ELSE.
        oo_result = lo_tnb->listvocabularies( ).
    ENDIF.
    MESSAGE 'Retrieved vocabularies list.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).

```

```
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[ListVocabularies](#)」を参照してください。

StartTranscriptionJob

次の例は、StartTranscriptionJob を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
  DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
  IF iv_vocabulary_name IS NOT INITIAL.
    lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
  ENDIF.

  oo_result = lo_tnb->starttranscriptionjob(
    iv_transcriptionjobname = iv_job_name
    io_media = lo_media
    iv_mediaformat = iv_media_format
    iv_languagecode = iv_language_code
    io_settings = lo_settings ).

  MESSAGE 'Transcription job started.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
```

```
RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
MESSAGE lo_conflict_ex TYPE 'I'.
RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- APIの詳細については、AWS SDK for SAP ABAP API リファレンスの[StartTranscriptionJob](#)を参照してください。

UpdateVocabulary

次の例は、UpdateVocabulary を使用する方法を説明しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.
  IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      it_phrases = it_phrases ).
  ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      iv_vocabularyfileuri = iv_vocab_file_uri ).
  ENDIF.
MESSAGE 'Vocabulary updated.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
MESSAGE lo_bad_request_ex TYPE 'I'.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
```

```
MESSAGE lo_limit_ex TYPE 'I'.
RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
MESSAGE lo_not_found_ex TYPE 'I'.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
MESSAGE lo_conflict_ex TYPE 'I'.
RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- API の詳細については、AWS SDK for SAP ABAP API リファレンスの[UpdateVocabulary](#)」を参照してください。

SDK for SAP ABAP を使用した Amazon Translate の例

次のコード例は、Amazon Translate で AWS SDK for SAP ABAP を使用してアクションを実行し、一般的なシナリオを実装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。

各例には完全なソースコードへのリンクが含まれており、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。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

DescribeTextTranslationJob

次のコード例は、DescribeTextTranslationJob を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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。

ListTextTranslationJobs

次のコード例は、ListTextTranslationJobs を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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_xl8textxl8translationjobfilt.

"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。

StartTextTranslationJob

次のコード例は、StartTextTranslationJob を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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_dataaccessrolelearn = iv_dataaccessrolelearn
    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。

StopTextTranslationJob

次のコード例は、StopTextTranslationJob を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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。

TranslateText

次のコード例は、TranslateText を使用する方法を示しています。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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_xl8textsizegmtexcdex.
    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_xl8unsuppdedlanguage00.
    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。

シナリオ

翻訳ジョブを開始する

次のコード例は、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。

SDK for SAP ABAP

Note

GitHub には、その他のリソースもあります。用例一覧を検索し、[AWS コード例リポジトリ](#)での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_dataaccessrolelearn = iv_dataaccessrolelearn
  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_xl8unsuppdedlanguage00.
  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 for SAP ABAP ナレッジ MCP サーバーの使用

AWS SDK for SAP ABAP ナレッジ MCP サーバーは、AI コーディングアシスタントが AWS 統合用の正確な ABAP コードを生成できるようにする専門知識リソースです。AI 対応 IDE をこのサーバーに接続することで、自然言語で構築する内容を記述し、AWS SDK for SAP ABAP を適切に呼び出す構文的に正しい ABAP コードを受け取ることができます。

MCP サーバーは、AI アシスタントを外部のナレッジソースに接続するためのオープンスタンダードである [Model Context Protocol \(MCP\)](#) を使用します。サーバーは AWS SDK for SAP ABAP リリースと同期して毎日更新されるため、IDE は常に最新の正確な SDK 情報にアクセスできます。

AWS SDK for SAP ABAP ナレッジ MCP サーバーは追加料金なしで利用できます。AWS SDK for SAP ABAP を使用して SAP アプリケーションで消費する AWS リソースとサービスに対してのみ料金が発生します。

トピック

- [利点](#)
- [MCP サーバーのセットアップ](#)
- [MCP サーバーでできること](#)
- [重要な考慮事項](#)

利点

AWS SDK for SAP ABAP ナレッジ MCP サーバーは、ABAP 開発者に次の利点を提供します。

- 正確なコード生成 – AI コーディングアシスタントは、AWS SDK for SAP ABAP 固有のパターン、メソッド署名、データ型、例外処理に関する深い知識を得ます。これにより、一般的なエラーがなくなり、コンパイルエラーとデバッグ時間が大幅に短縮されます。
- 即時 SDK 検出 – MCP サーバーは、200 以上の AWS サービスをカバーする完全な AWS SDK for SAP ABAP ナレッジベースを公開します。ドキュメントを手動で検索しなくても、利用可能なサービス、オペレーション、データ型を検出できます。
- 常に最新 – ナレッジベースは SDK リリースと同期して毎日更新されます。新しいサービスと更新されたメソッド署名は、IDE ですぐに使用できます。
- ゼロインストール – セットアップでは、IDE の MCP 設定に URL を貼り付けるだけで済みます。ローカルソフトウェアのインストール、依存関係管理、IT 承認プロセスは必要ありません。

MCP サーバーのセットアップ

IDE を AWS SDK for SAP ABAP ナレッジ MCP サーバーに接続するには、IDE の MCP 設定に次の設定を追加します。設定形式とファイルの場所は IDE によって異なる場合があります。詳細については、IDE の MCP ドキュメントを参照してください。

次の例は、Kiro の設定を示しています。ワークスペース MCP 設定ファイル `.kiro/settings/mcp.json` またはユーザーレベルの設定ファイル に追加します `~/.kiro/settings/mcp.json`。

```
{
  "mcpServers": {
    "abap-sdk-knowledge": {
      "url": "https://sdk-for-sap-abap-knowledge-mcp.global.api.aws/mcp",
      "disabled": false,
      "autoApprove": ["*"]
    }
  }
}
```

`autoApprove` この設定により、IDE は各リクエストの承認を求めることなく MCP サーバーのツールを使用できます。MCP サーバーは読み取り専用であり、データを変更したりコードを実行したりしないため、安全です。

設定を保存すると、IDE は MCP サーバーに自動的に接続されます。ほとんどの IDEs では再起動は必要ありません。「で利用できるサービスは何ですか？」など、AWS SDK for SAP ABAP について AI アシスタントに質問することで、接続を確認できます。 AWS AWS SDK for SAP ABAP

IDE が MCP サーバーに接続しない場合は、IDE がモデルコンテキストプロトコルをサポートし、設定ファイルの構文が IDE の要件と一致していることを確認します。

Note

MCP サーバーエンドポイントは HTTPS を使用し、認証を必要としません。IDE は、AWS SDK for SAP ABAP の使用に関して AI アシスタントに質問する場合にのみサーバーと通信します。独自のビジネスロジックと SAP アプリケーションコードは、開発環境内に残ります。

MCP サーバーでできること

接続すると、AI コーディングアシスタントは MCP サーバーが提供する以下の機能を使用できます。

- サービス検出 – AWS サービスメタデータやファクトリクラス名など、AWS SDK for SAP ABAP で利用可能なすべてのサービスを一覧表示して検索します。
- オペレーションの詳細 – SDK オペレーションの ABAP メソッド署名、入力パラメータ、戻り値タイプ、例外タイプを取得します。
- データ型定義 – フィールド名やタイプなど、SDK データ型の ABAP 構造定義を検索します。
- コード例 – 各サービスの一般的なユースケースをカバーする、特定のオペレーションの実用的な ABAP コード例を取得します。
- セッションとクライアントの作成 – SDK セッションの作成、サービスクライアントの初期化、正しい例外処理によるオペレーションの呼び出しを行うための定型 ABAP コードを生成します。

たとえば、次のようなリクエストを AI アシスタントに求めることができます。

- 「SQS キューからメッセージを読み取る ABAP コードを記述します。」
- 「テキスト生成のために Amazon Bedrock モデルを呼び出す ABAP コードを生成します。」
- 「ABAP から Amazon S3 にファイルをアップロードする方法を示します」

AI アシスタントは、MCP サーバーの知識を使用して、適切なセッション作成、クライアントの初期化、例外処理パターンなど、AWS SDK for SAP ABAP 規則に従う構文的に正しい ABAP コードを生成します。

重要な考慮事項

AWS SDK for SAP ABAP ナレッジ MCP サーバーを使用する場合は、次の点に注意してください。

- MCP サーバーは SDK ナレッジへの読み取り専用アクセスを提供します。ABAP コードを実行したり、SAP システムに接続したりすることはありません。
- MCP サーバーは AWS SDK for SAP ABAP のみを対象としています。AWS SDK for SAP ABAP の使用に関連しない一般的な ABAP プログラミング支援は提供されません。
- MCP サーバーは、カスタムまたはサードパーティーの ABAP ライブラリをサポートしていません。

- MCP サーバーは、最新バージョンの SDK に基づいてドキュメントを配信します。これは、SAP システムにインストールされているバージョンと一致しない場合があります。
- 生成されたコード例は出発点です。本番環境で使用する前に、生成されたすべてのコードを確認してテストします。
- 生成されたコードの品質と形式は、IDE の AI アシスタント機能によって異なります。

このページで説明されている AWS SDK for SAP ABAP ナレッジ MCP サーバーとは別に、には [AWS ナレッジ MCP サーバー](#) AWS も用意されています。このサーバーは、すべての AWS サービスにわたって幅広い AWS サービス情報、アーキテクチャガイダンス、トラブルシューティングを提供します。2 つのサーバーは相互に補完し合い、一緒に使用すると特に効果的です。

のセキュリティ AWS SDK for SAP ABAP

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、セキュリティを最も重視する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 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説明しています。

- クラウドのセキュリティ – AWS は、AWS のサービス で実行されるインフラストラクチャを保護する責任を担います AWS クラウド。は、安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、[AWS コンプライアンスプログラム](#)コンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。が適用されるコンプライアンスプログラムの詳細については AWS SDK for SAP ABAP、「[コンプライアンスプログラムAWS のサービス による対象範囲内](#)」および「[コンプライアンスプログラム](#)」を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する によって決まり AWS のサービス ます。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このセクションでは、次のトピックについて説明します。

トピック

- [での SAP システム認証 AWS](#)
- [IAM セキュリティのベストプラクティス](#)
- [SAP 認証](#)
- [セキュアなオペレーション](#)
- [SSF 暗号化でのシークレットアクセスキー認証の使用](#)
- [IAM Roles Anywhere での証明書の使用](#)
- [クロスアカウントアクセスのソースプロファイルの使用](#)
- [SAP 認証情報ストアの使用](#)

での SAP システム認証 AWS

SAP システムが SAP ユーザー AWS に代わって を呼び出す前に、SAP システムは に対して自身を認証する必要があります AWS。 は、 の SDK プロファイル設定で選択されている次の 3 つの基本認証方法 AWS SDK for SAP ABAP をサポートしています IMG。

AWS SDK for SAP ABAP - BTP エディションは、SAP 認証情報ストアを使用する [the section called “シークレットアクセスキー認証”](#) メソッドでのみ認証できます。

クロスアカウントアクセスシナリオの場合、SDK for SAP ABAP はソースプロファイルもサポートしています。これにより、いずれかの基本認証方法を使用して、アカウント間で複数の IAM ロールの前提条件を連鎖できます。詳細については、「[the section called “クロスアカウントアクセスのソースプロファイル認証”](#)」を参照してください。

トピック

- [Amazon EC2 インスタンスメタデータの認証](#)
- [シークレットアクセスキー認証](#)
- [IAM Roles Anywhere を使用した証明書ベースの認証](#)
- [クロスアカウントアクセスのソースプロファイル認証](#)
- [次のステップ](#)

Amazon EC2 インスタンスメタデータの認証

Amazon EC2 で実行されている SAP システムは、Amazon EC2 インスタンスのメタデータから有効期間が短く、自動的に更新される認証情報を取得できます。詳細については、「[Amazon EC2 インスタンスメタデータの認証情報の使用](#)」を参照してください。

SDK for SAP ABAP を使用する場合は、この認証方法を強くお勧めします。有効にするには、Basis 管理者がアウトバウンド HTTP 通信を有効にする必要があります。必要な Basis の設定はこれだけです。

Note

この認証方法は、SAP システムが Amazon EC2 で実行されている場合にのみ適用されません。オンプレミスや他のクラウド環境でホストされている SAP システムは、この方法では認証できません。

シークレットアクセスキー認証

この方法では、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 AWS で SAP システムを認証します。SAP システムは IAM ユーザー AWS を使用してにログインします。詳細については、「[IAM ユーザーのアクセスキーの管理](#)」を参照してください。

Basis 管理者は、AWS IAM 管理者からアクセスキー ID とシークレットアクセスキーを受け取ります。SAP システムは、アクセスキー ID とシークレットアクセスキーを保存するように設定する必要があります。

- 保護、保存、転送 (SSF)
 - SSF 機能を使用して AWS SDK for SAP ABAP を認証します。詳細については、「[デジタル署名と暗号化](#)」を参照してください。
 - SSF02 レポートを使用して SSF の envelope と develop の機能をテストすることもできます。詳細については、「[SSF インストールのテスト](#)」を参照してください。
 - SDK for SAP ABAP 用に SSF を設定する手順は、/AWS1/IMG トランザクションで説明されています。[技術的前提条件] に進み、オンプレミスシステムの [追加設定] を選択します。詳細な設定手順については、「[SSF 暗号化によるシークレットアクセスキー認証の使用](#)」を参照してください。
- SAP 認証情報ストア
 - SAP 認証情報ストアを使用して、AWS SDK for SAP ABAP - BTP エディションを認証します。詳細については、「[SAP 認証情報ストアとは](#)」を参照してください。
 - 設定手順については、「[SAP 認証情報ストアの使用](#)」を参照してください。

IAM Roles Anywhere を使用した証明書ベースの認証

Roles Anywhere での認証には、認証機関 (CA) によって発行された X.509 AWS Identity and Access Management 証明書を使用できます。証明書は STRUST で設定する必要があります。CA を IAM Roles Anywhere にトラストアンカーとして登録する必要があります。また、IAM Roles Anywhere が引き受けるロールとポリシーを指定するために、プロファイルを作成する必要があります。詳細については、「[AWS Identity and Access Management Roles Anywhere での信頼アンカーとプロファイルの作成](#)」を参照してください。

SDK for SAP ABAP で IAM Roles Anywhere を使用する方法的詳細な手順については、「[IAM Roles Anywhere で証明書を使用する](#)」を参照してください。

Note

証明書の失効は、インポートされた証明書失効リストを使用する場合にのみサポートされません。詳細については、「[失効](#)」を参照してください。

クロスアカウントアクセスのソースプロファイル認証

ソースプロファイルは、AWS アカウント間で複数の IAM ロールの前提条件を連鎖できる高度な機能です。この方法では、1つのプロファイルがロールを引き受け、AWS CLI の `source_profile` パラメータと同様に別のロールを引き受けます。

ソースプロファイルは、3つの基本認証方法 (インスタンスメタデータ、シークレットアクセスキー、または証明書ベース) のいずれかで動作します。チェーン内の最初のプロファイルは、これらの基本メソッドのいずれかを使用する必要があります。チェーン内の後続のプロファイルは、前のプロファイルの認証情報を使用して次のロールを引き受けます。

これは、ターゲットリソースに到達するために複数の AWS アカウントをトラバースする必要があるクロスアカウントアクセスシナリオに役立ちます。詳細な設定手順については、「[クロスアカウントアクセスのソースプロファイルの使用](#)」を参照してください。

次のステップ

で SAP システムを認証すると AWS、SDK for SAP ABAP は を自動的に実行 `sts:assumeRole` し、SAP ユーザーのビジネス機能に適した IAM ロールを引き受けます。

IAM セキュリティのベストプラクティス

IAM 管理者は、次の3つの主要分野を担当します。

- SAP システムが Amazon EC2 メタデータまたはシークレットキー認証情報を使用して自身を認証できることを確認します。
- SAP システムが `sts:assumeRole` を使用して自身を昇格させるのに必要な権限を持っていることを確認します。
- 論理 IAM ロールごとに、ビジネス機能の実行に必要な権限 (Amazon S3、DynamoDB、またはその他のサービスに必要な権限など) を持つ SAP ユーザー用の IAM ロールを作成します。これらは、SAP ユーザーが引き受けるロールです。

詳細については、「SAP Lens: AWS Well-Architected フレームワーク」の「[セキュリティ](#)」の章を参照してください。

トピック

- [Amazon EC2 インスタンスプロファイルのベストプラクティス](#)
- [SAP ユーザーの IAM ロール](#)
- [ソースプロファイルのセキュリティに関する考慮事項](#)

Amazon EC2 インスタンスプロファイルのベストプラクティス

SAP システムが稼働する Amazon EC2 インスタンスには、そのインスタンスプロファイルに基づく一連の認証があります。一般的に、インスタンスプロファイルに必要なのは `sts:assumeRole` を呼び出す権限だけで、SAP システムが必要に応じてビジネス固有の IAM ロールを引き受けることができます。このように他のロールに昇格することで、ABAP プログラムは、その業務に必要な最小特権をユーザーに与えるロールを引き受けることができるようになります。例えば、インスタンスプロファイルには次のステートメントが含まれる場合があります。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::012345678912:role/finance-cfo",
        "arn:aws:iam::012345678912:role/finance-auditor",
        "arn:aws:iam::012345678912:role/finance-reporting"
      ]
    }
  ]
}
```

前の例では、SAP システムが CFO、監査者、または REPORTING ユーザーの IAM ロールを引き受けることを許可します。AWS SDK は SAP のユーザーの PFCG ロールに基づいて、ユーザーの正しい IAM ロールを選択します。

Amazon EC2 インスタンスプロファイルは他の機能にも使用できます。

- [AWS SAP HANA 用 Backint エージェント](#)
- [オーバーレイ IP アドレスルーティングによる SAP on AWS の高可用性](#)

これらのソリューションでは、バックアップやフェイルオーバーに固有のロールに対する `sts:assumeRole` 権限が必要な場合もあれば、インスタンスプロファイルに直接権限を割り当てる必要がある場合もあります。

SAP ユーザーの IAM ロール

ABAP プログラムには、ユーザーのジョブを実行するためのアクセス許可が必要です。DynamoDB テーブルを読み取る、Amazon S3 の PDF オブジェクトで Amazon Textract を呼び出す、AWS Lambda 関数を実行します。すべての AWS SDKs で同じセキュリティモデルが使用されます。別の AWS SDK で使用されていた既存の IAM ロールを使用できます。

SAP ビジネスアナリストは、必要な論理ロールごとに IAM ロールの `arn:aws:` を IAM 管理者に依頼します。例えば、財務シナリオでは、ビジネスアナリストが以下の論理 IAM ロールを定義する場合があります。

- CFO
- AUDITOR
- REPORTING

IAM 管理者は論理 IAM ロールごとに IAM ロールを定義します。

CFO

- `arn:aws:iam::0123456789:role/finance-cfo`
- Amazon S3 バケットへの読み取りおよび書き込みアクセス許可
- DynamoDB データベースへの読み取りおよび書き込みアクセス許可

AUDITOR

- `arn:aws:iam::0123456789:role/finance-auditor`

- Amazon S3 バケットへの読み取りアクセス許可
- DynamoDB データベースへの読み取りアクセス許可

REPORTING

- `arn:aws:iam::0123456789:role/finance-reporting`
- DynamoDB データベースへの読み取りアクセス許可
- Amazon S3 バケットのアクセス許可なし

ビジネスアナリストは IAM ロールをマッピングテーブルに入力し、論理 IAM ロールを物理 IAM ロールにマッピングします。

SAP ユーザーの IAM ロールでは、信頼できるプリンシパルに `sts:assumeRole` アクションを許可する必要があります。信頼できるプリンシパルは、AWSでの SAP システムの認証方法によって異なる場合があります。詳細については、「[プリンシパルの指定](#)」を参照してください。

以下に、最も一般的な SAP シナリオの例をいくつか示します。

- Amazon EC2 上で実行され、インスタンスプロファイルが割り当てられている SAP システム — ここでは、Amazon EC2 インスタンスプロファイルが IAM ロールにアタッチされます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/SapInstanceProfile"
      }
    }
  ]
}
```

- Amazon EC2 上で実行され、インスタンスプロファイルがない SAP システム — ここでは、Amazon EC2 が SAP ユーザーのロールを引き継ぎます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "Service": [ "ec2.amazonaws.com" ]
      }
    }
  ]
}
```

- オンプレミスで稼働する SAP システム — オンプレミスで稼働する SAP システムは、シークレットアクセスキーを使用してのみ認証できます。詳細については、「[AWSでの SAP システム認証](#)」を参照してください。

この場合、SAP ユーザーが引き受けるすべての IAM ロールには、その SAP ユーザーを信頼する信頼関係が必要です。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/SAP_SYSTEM_S4H"
      }
    }
  ]
}
```

ソースプロファイルのセキュリティに関する考慮事項

ソースプロファイルを使用する場合:

IAM ロールの管理

重要: ソースプロファイルチェーンの IAM ロールは、不正アクセスと特権のエスカレーションを防ぐために厳密に管理する必要があります。

- 最小特権の原則を適用する - 各ロールの特定の目的に必要な最小限のアクセス許可のみを付与します
- ロールのアクセス許可を定期的に監査する - ロールポリシーを四半期ごと、または要件が変更されたときに確認および更新する
- ロール使用状況のモニタリング - を使用して AssumeRole API コールを追跡し、異常なパターンを特定します。
- 信頼関係の制限 - どのプリンシパルが各ロールを引き受けることができるかを、絶対にアクセスを必要とするプリンシパルのみに制限します。
- 信頼ポリシーで条件を使用する - 必要に応じて、ソース IP、MFA 要件、時間ベースの制限などの条件を追加します。
- ロールの目的を文書化する - 各ロールの目的とするユースケースと必要なアクセス許可を明確に文書化します。

認可とアクセスコントロール

- チェーン内のすべての中間プロファイルに適切な信頼ポリシーが設定されていることを確認します。
- ユーザーは、中間プロファイルを含むチェーン内のすべてのプロファイルに対する /AWS1/SESS 認可が必要です
- 各 IAM ロールは、チェーン内の以前のロールを明示的に信頼する必要があります

技術的な保護

- SDK は、過剰な STS API コールを防ぐために、最大チェーン深度 5 プロファイルを適用します。
- 循環参照は自動的に検出され、防止されます
- ベースプロファイル認証方法が検証され、標準メソッド (INST、SSF、または RLA) が使用されていることを確認します。

ソースプロファイルの設定の詳細については、[「クロスアカウントアクセスでのソースプロファイルの使用」](#)を参照してください。

SAP 認証

SDK の設定に必要な認可は、SDK エディションによって異なります。

トピック

- [設定の権限](#)
- [エンドユーザー向けの SAP 認証](#)

設定の権限

詳細については、次のタブを参照してください。

SDK for SAP ABAP

SDK for SAP ABAP を設定するには、以下の権限が必要です。

- S_TCODE
 - TCD = /AWS1/IMG
- S_TABU_DIS
 - ACTVT = 02, 03
- DICBERCLS

次の権限グループから選択します。

- /AWS1/CFG- AWS SDK for SAP ABAP v1 - 設定
- /AWS1/MOD- AWS SDK for SAP ABAP v1 - ランタイム
- /AWS1/PFL- AWS SDK for SAP ABAP v1 - SDK プロファイル
- /AWS1/RES- AWS SDK for SAP ABAP v1 - 論理リソース
- /AWS1/TRC- AWS SDK for SAP ABAP v1 - トレース

SDK for SAP ABAP - BTP edition

SDK for SAP ABAP - BTP Edition が 設定にアクセスできるようにするには、次のステップを使用します。

1. ビジネスロールテンプレートを使用して新しいSAP_BR_BPC_EXPERTビジネスロールを作成します。このテンプレートは、Custom Business Configuration アプリケーションへのアクセスを提供します。
2. 一般的なロールの詳細で、アクセスカテゴリに移動し、読み取り、書き込み、値のヘルプに無制限を選択します。
3. Business Catalog タブに移動し、/AWS1/RTBTP_BCATビジネスカタログを割り当てて SDK 設定へのアクセスを提供します。
4. Business Users タブに移動し、ビジネスユーザーを割り当てて SDK 設定へのアクセスを許可します。

エンドユーザー向けの SAP 認証

前提条件: SDK プロファイルの定義

SAP セキュリティ管理者がロールを定義する前に、ビジネスアナリストは SDK for SAP ABAP または /AWS1/IMG AWS SDK for SAP ABAP - BTP エディションのカスタムビジネス設定アプリケーションのトランザクションで SDK プロファイルを定義します。通常、SDK プロファイルにはビジネス機能 (ZFINANCE、ZBILLING、ZMFG、ZPAYROLL など) に従って名前が付けられます。ビジネスアナリストは SDK プロファイルごとに、CFO、AUDITOR、REPORTING などの短い名前で論理 IAM ロールを定義します。これらは IAM セキュリティ管理者によって実際の IAM ロールにマッピングされます。

PFCG またはビジネスロールを定義する

Note

PFCG ロールは、SAP BTP、ABAP 環境ではビジネスロールと呼ばれます。

次に、SAP セキュリティ管理者は SDK プロファイルへのアクセス/AWS1/SESSを許可する認可オブジェクトを追加します。

認証オブジェクト /AWS1/SESS

- フィールド /AWS1/PROF = ZFINANCE

また、ユーザーを職務に応じて各 SDK プロファイルの論理 IAM ロールにマッピングする必要があります。例えば、レポートへのアクセス権を持つ財務監査人には、AUDITOR という論理 IAM ロールの権限が与えられる場合があります。

認証オブジェクト /AWS1/LR0L

- フィールド /AWS1/PROF = ZFINANCE
- フィールド /AWS1/LR0L = AUDITOR

一方、読み取り/書き込み権限を持つ CFO には、CFO の論理的な役割を許可する PFCG ロールが与えられる場合があります。

認証オブジェクト /AWS1/LR0L

- フィールド /AWS1/PROF = ZFINANCE
- フィールド /AWS1/LR0L = CFO

一般に、ユーザーは SDK プロファイルごとに 1 つの論理 IAM ロールに対してのみ認証される必要があります。ユーザーが複数の IAM ロールに対して承認されている場合 (たとえば、CFO が AUDITOR CFO と論理 IAM ロールの両方に対して承認されている場合)、AWS SDK は優先度の高い (シーケンス番号の低い) ロールが有効になるようにすることで、関連付けを解除します。

すべてのセキュリティシナリオと同様に、ユーザーには職務を遂行するための最小特権を与える必要があります。PFCG ロールを管理する簡単な方法は、許可する SDK プロファイルと論理ロールに従って単一の PFCG ロールに名前を付けることです。例えば、ロール Z_AWS_PFL_ZFINANCE_CFO はプロファイル ZFINANCE と論理 IAM ロール CFO へのアクセスを許可します。その後、これらの単一ロールを、職務を定義する複合ロールに割り当てることができます。各企業には役割管理に関する独自の戦略があるため、自社に合った PFCG 戦略を定義することをお勧めします。

セキュアなオペレーション

保管中のデータの暗号化

AWS シークレットアクセスキーは SDK の認証に使用されます。これらは、SAP によって SSF または認証情報ストア機能を使用して暗号化されます。

転送中のデータの暗号化

へのすべての呼び出し AWS のサービス は HTTPS で暗号化されます。SAP ICM は HTTPS 接続を管理します。AWS 証明書は STRUST で信頼されている必要があります。

API の使用

ABAP ユーザが `sts:assumeRole` を使用するロールを引き受けると、セッション名には `USERID-SID-MANDT` というタイトルが付けられます。

- `USERID` は `SY-UNAME` 変数から取得した ABAP ユーザです。
- `SID` は `SY-SYSID` 変数から取得した ABAP システム ID です。
- `MANDT` は `SY-MANDT` 変数から取得した ABAP クライアントです。

セッション名は CloudTrail にユーザー名として表示されます。これにより、ABAP ユーザーからの API コールを、呼び出しを開始したシステム、クライアント、ユーザーまでさかのぼって追跡できるようになります。詳細については、「[What is AWS CloudTrail?](#)」を参照してください。

SSF 暗号化でのシークレットアクセスキー認証の使用

オンプレミス SAP システム (または他のクラウドで実行されているシステム) は、シークレットアクセスキー認証 AWS を使用して で認証できます AWS Identity and Access Management。SAP の [Secure Store and Forward Mechanism](#) (SSF) は、IAM ユーザーの AWS 認証情報 (アクセスキー ID とシークレットアクセスキー) を暗号化して安全に保存するために使用されます。SAP システムは IAM ユーザー AWS を使用して にログインします。詳細については、「[IAM ユーザーのアクセスキーの管理](#)」を参照してください。

前提条件

設定を開始する前に、次の前提条件を満たす必要があります。

- SAP ユーザーの IAM ロールは IAM 管理者が作成する必要があります。ロールには、必要な を呼び出すためのアクセス許可が必要です AWS のサービス。詳細については、「[IAM 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。
- `/AWS1/IMG` トランザクションを実行する許可を作成します。詳細については、「[設定の権限](#)」を参照してください。

手順

SSF で暗号化された認証情報ストレージを設定するには、次の手順に従います。

Steps

- [ステップ 1 – 認証情報ストレージ用の SSF アプリケーションを定義する](#)
- [ステップ 2 – SSF アプリケーションの暗号化パラメータを設定する](#)
- [ステップ 3 – SSF アプリケーションの PSE を作成する](#)
- [ステップ 4 – SSF アプリケーションを に割り当てる AWS SDK for SAP ABAP](#)
- [ステップ 5 – SSF で暗号化された認証情報を使用するように SDK プロファイルを設定する](#)

ステップ 1 – 認証情報ストレージ用の SSF アプリケーションを定義する

1. トランザクションコードを実行して SSF アプリケーションSE16を定義します。
2. SSFAPPLIC テーブル名を入力し、[新規エントリ] を選択します。
3. 以下の詳細を入力します。
 - APPLIC: ZAWS1 (SSF アプリケーションの名前)。
 - DESCRIPT: SSF Encryption for the AWS SDK for SAP ABAP (説明)。
 - 残りのフィールドのSelected(X)オプションを選択します。
4. [保存] を選択します。

ステップ 2 – SSF アプリケーションの暗号化パラメータを設定する

1. トランザクションコードを実行して/n/AWS1/IMG、実装ガイド (IMG) を起動します AWS SDK for SAP ABAP。
2. IMG ノードAWS SDK for SAP ABAP 設定 > 技術的前提条件 > オンプレミスシステムの追加設定を展開します。
3. SSF パラメータの設定 IMG アクティビティを実行します。
4. [新しいエントリ] を選択し、前のステップで作成した SSF アプリケーションを選択します。[保存] を選択します。
5. ハッシュアルゴリズムを SHA256 (またはそれ以上) に変更し、暗号化アルゴリズムを AES256-CBC に変更します。その他の設定はデフォルトのままにして、[保存] を選択します。

これらの暗号化設定は、AWS 認証情報を安全に暗号化するために使用されます。

ステップ 3 – SSF アプリケーションの PSE を作成する

1. /n/AWS1/IMG トランザクションを実行し、AWS SDK for SAP ABAP 設定 > 技術的前提条件 > オンプレミスシステムの追加設定を選択します。
2. IMG アクティビティを実行します。SSF アプリケーションの PSE を作成します。これにより、に移動しますSTRUST transaction。[Edit] (編集) を選択します。
3. [the section called “ステップ 1”](#) で作成した SSF アプリケーションを右クリックし、[作成] を選択します。その他の設定はすべてデフォルトのままにして、[続行] を選択します。アルゴリズムとして DSA RSAではなく を選択していることを確認してください。

ステップ 4 – SSF アプリケーションを に割り当てる AWS SDK for SAP ABAP

1. /n/AWS1/IMG トランザクションを実行し、AWS SDK for SAP ABAP 設定 > 技術的前提条件 > オンプレミスシステムの追加設定を選択します。
2. IMG アクティビティを実行します。SSF アプリケーションを に割り当てます AWS SDK for SAP ABAP。
3. 新しいエントリを選択し、 で作成された SSF アプリケーションを入力します[the section called “ステップ 1”](#)。[保存] を選択します。

ステップ 5 – SSF で暗号化された認証情報を使用するように SDK プロファイルを設定する

1. /n/AWS1/IMG トランザクションを実行し、AWS SDK for SAP ABAP 設定 > アプリケーション設定を選択します。
2. IMG アクティビティ SDK プロファイルを実行します。
3. [新規エントリ] を選択します。プロファイル名と説明を入力します。[保存] を選択します。
4. 作成したエントリを強調表示し、[認証と設定] ツリーブランチをクリックします。
5. 新しいエントリを選択し、次の詳細を入力します。
 - SID: SAP システムのシステム ID。
 - クライアント: SAP システムのクライアント。
 - シナリオ ID: Basis 管理者が作成したDEFAULTシナリオを選択します。
 - AWS Region: 呼び出し先の AWS リージョン。

- 認証方法: ドロップダウン Credentials from SSF Storage から選択し、保存を選択します。認証情報の設定を選択し、IAM ユーザーのアクセスキー ID とシークレットアクセスキーを入力します。
 - IAM ロールを無効にする: これをデフォルトのままにします。つまり、オフにします。
 - [保存] を選択します。
6. [IAM ロールマッピング] ツリーブランチをクリックします。[新規エントリ] を選択します。IAM 管理者から提供された論理 IAM ロールと IAM ロール ARN AWS のシーケンス番号、名前を入力します。[保存] を選択します。

詳細については、「[アプリケーションの設定](#)」を参照してください。

IAM Roles Anywhere での証明書の使用

SAP システムは、AWS Identity and Access Management Roles Anywhere で証明書ベースの認証 AWS を使用してで認証できます。STRUST で証明書を設定し、SDK プロファイルを /AWS1/IMG で設定する必要があります。

前提条件

証明用のセットアップを開始する前に、次の前提条件を満たす必要があります。

- 認証局 (CA) が発行する X.509 証明書は、以下の要件を満たしている必要があります。
 - 署名証明書は v3 証明書でなければなりません。
 - チェーンの証明書の数が 5 を超えてはなりません。
 - 証明書は RSA または ECDSA アルゴリズムをサポートしている必要があります。
- CA を IAM Roles Anywhere にトラストアンカーとして登録し、IAM Roles Anywhere のロール/ポリシーを指定するプロファイルを作成します。詳細については、[AWS Identity and Access Management 「Roles Anywhere での信頼アンカーとプロファイルの作成」](#)を参照してください。
- SAP ユーザーの IAM ロールは IAM 管理者が作成する必要があります。ロールには、必要な を呼び出すためのアクセス許可が必要です AWS のサービス。詳細については、「[IAM 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。
- /AWS1/IMG トランザクションを実行する許可を作成します。詳細については、「[設定の権限](#)」を参照してください。

手順

以下の手順に従って、証明書ベースの認証を設定します。

Steps

- [ステップ 1 – SAP の Secure Store and Forward \(SSF\) を使用して SSF アプリケーションを定義する](#)
- [ステップ 2 – SSF パラメータを設定する](#)
- [ステップ 3 – PSE と証明書のリクエストを作成する](#)
- [ステップ 4 – 証明書レスポンスを関連する PSE にインポートする](#)
- [ステップ 5 – IAM Roles Anywhere を使用するように SDK プロファイルを設定する](#)

ステップ 1 – SAP の Secure Store and Forward (SSF) を使用して SSF アプリケーションを定義する

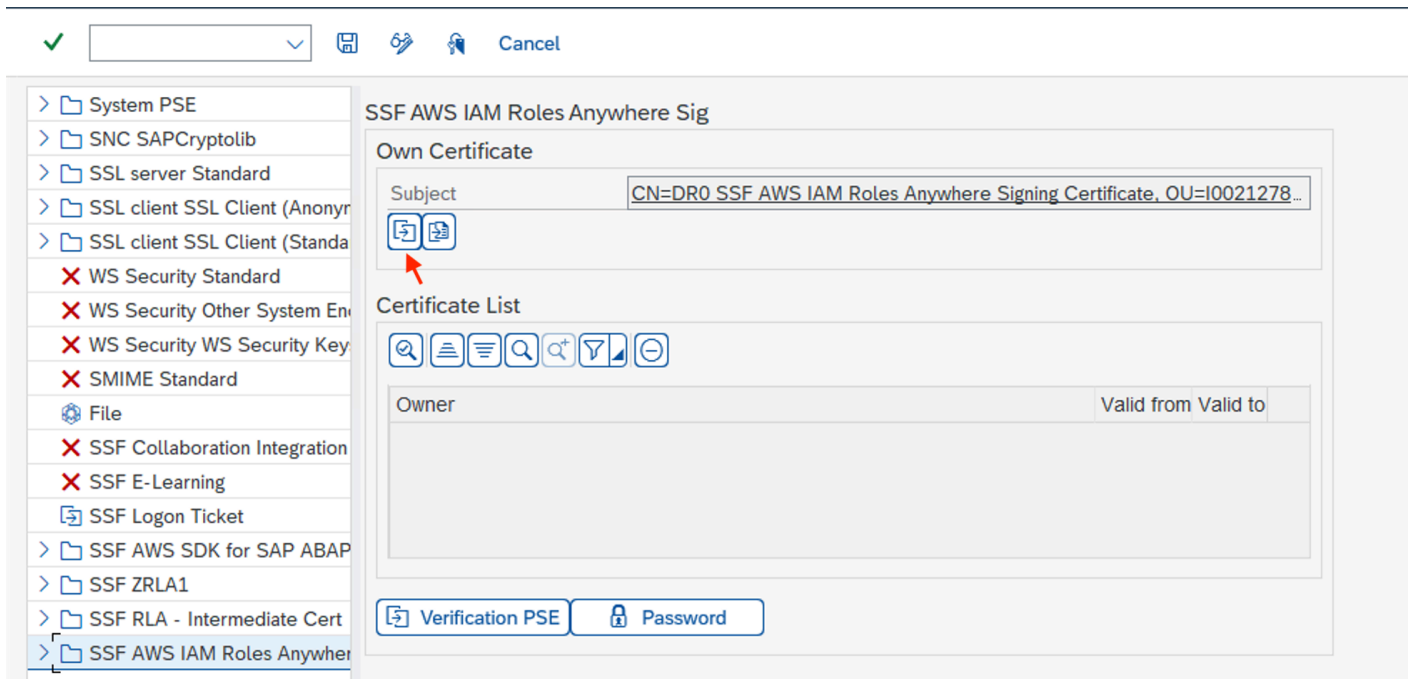
1. トランザクションコード SE16 を実行して SSF アプリケーションを定義します。
2. SSFAPPLIC テーブル名を入力し、[新規エントリ] を選択します。
3. APPLIC フィールドに SSF アプリケーションの名前を入力し、DESCRIPT フィールドに説明を入力して、残りのフィールドで Selected (X) オプションを選択します。

ステップ 2 – SSF パラメータを設定する

1. を実行して AWS SDK for SAP ABAP 実装ガイド (IMG) を起動/n/AWS1/IMGします。
2. [AWS SDK for SAP ABAP の設定] > [技術的前提条件] > [オンプレミスシステムの追加の設定] を選択します。
3. [SSF パラメータの設定] IMG アクティビティを実行します。
4. [新しいエントリ] を選択し、前のステップで作成した SSF アプリケーションを選択します。[保存] を選択します。
5. ハッシュアルゴリズムを SHA256 に変更し、暗号化アルゴリズムを AES256-CBC に変更します。その他の設定はデフォルトのままにして、[保存] を選択します。

ステップ 3 – PSE と証明書のリクエストを作成する

1. /n/AWS1/IMG トランザクションを実行し、[AWS SDK for SAP ABAP の設定] > [技術的前提条件] > [オンプレミスシステムの追加の設定] を選択します。
2. Create PSE for SSF Application IMG アクティビティを実行します。
3. STRUST トランザクションの [編集] を選択します。
4. [the section called “ステップ 1”](#) で作成した SSF アプリケーションを右クリックし、[作成] を選択します。その他の設定はすべてデフォルトのままにして、[続行] を選択します。
5. [証明書リクエストを作成] を選択します。次の画像を参照してください。デフォルトのオプションを保持して、[続行] を選択します。生成された証明書リクエストをコピーまたはエクスポートして、CA に提供します。CA はリクエストを検証し、署名付きのパブリックキー証明書で応答します。



署名プロセスは、CA と、CA が使用するテクノロジーによって異なります。例については、[「Private Certificate Authority によるプライベートエンドエンティティ証明書の発行」](#) を参照してください。AWS

ステップ 4 – 証明書レスポンスを関連する PSE にインポートする

1. /n/AWS1/IMG トランザクションを実行し、[AWS SDK for SAP ABAP の設定] > [技術的前提条件] > [オンプレミスシステムの追加の設定] を選択します。

2. Create PSE for SSF Application IMG アクティビティを実行します。
3. STRUST トランザクションの [編集] を選択します。
4. SSF アプリケーションを選択し、件名の下のパセクションにある [証明書レスポンスのインポート] を選択します。証明書レスポンスをコピーしてテキストボックスに貼り付けるか、ファイルシステムからファイルをインポートします。[続行] > [保存] を選択します。
5. 証明書の詳細は、件名を 2 回選択すると表示できます。情報は証明書セクションに表示されません。

ステップ 5 – IAM Roles Anywhere を使用するように SDK プロファイルを設定する

1. /n/AWS1/IMG トランザクションを実行し、AWS SDK for SAP ABAP 設定 > アプリケーション設定を選択します。
2. 新しい SDK プロファイルを作成し、名前を付けます。
3. 認証方法として [IAM Roles Anywhere] を選択します。
 - 左側のペインで、[認証と設定] を選択します。
 - 新しいエントリを作成し、SAP システムの情報と AWS リージョンを入力します。
 - 認証方法として [IAM Roles Anywhere] を選択し、[保存] を選択します。
 - [詳細を入力] を選択し、ポップアップウィンドウで、[the section called “ステップ 1”](#) で作成した SSF アプリケーションを選択します。「[the section called “前提条件”](#)」で作成した [トラストアンカー ARN] と [プロファイル ARN] を入力します。次の画像を参照してください。[Continue] (続行) をクリックします。

4. 左側のペインで [IAM ロールマッピング] を選択します。名前を入力し、IAM 管理者から提供された IAM ロールの ARN を指定します。

詳細については、「[アプリケーションの設定](#)」を参照してください。

クロスアカウントアクセスのソースプロファイルの使用

ソースプロファイルを使用すると、SAP システムは IAM ロールの前提条件を連鎖させることで、複数のアカウントの AWS リソースにアクセスできます。1 つのプロファイルはロールを引き受け、CLI AWS の `source_profile` パラメータと同様に別のロールを引き受けます。これは、ターゲットリソースに到達するために複数のアカウントをトラバースする必要があるクロス AWS アカウントアクセスシナリオに役立ちます。

例: SAP システムはアカウント A (111111111111) で実行され、アカウント C (333333333333) の Amazon S3 バケットにアクセスする必要があります。3 つのプロファイルを設定します。

1. DEV_BASE Amazon EC2 インスタンスメタデータから基本認証情報を取得し、アカウント A でロール P を引き受ける
2. SHARED_SERVICES は DEV_BASE 認証情報を使用してアカウント B のロール Q を引き受けます (222222222222)
3. PROD_S3_ACCESS は SHARED_SERVICES 認証情報を使用してアカウント C でロール R を引き受けます

アプリケーションがを使用する場合 PROD_S3_ACCESS、SDK は自動的にチェーンを実行します。インスタンスメタデータから認証情報を取得する → ロール P を引き受ける → ロール Q を引き受ける → ロール R を引き受けます。

前提条件

ソースプロファイルを設定する前に、次の前提条件を満たす必要があります。

- チェーンの各ステップの IAM ロールは、IAM 管理者が作成する必要があります。各ロールには以下が必要です。
 - 必要な を呼び出すアクセス許可 AWS のサービス
 - チェーン内の以前のロールが引き受けることを許可するように設定された信頼関係

詳細については、「[IAM 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。

- /AWS1/IMG トランザクションを実行する許可を作成します。詳細については、「[設定の権限](#)」を参照してください。
- ユーザーは、中間プロファイルを含む、チェーン内のすべてのプロファイルに対する/AWS1/SESS認可を持っている必要があります。

手順

ソースプロファイルを設定するには、次の手順に従います。

Steps

- [ステップ 1 – ベースプロファイルを設定する](#)
- [ステップ 2 – 連鎖プロファイルを設定する](#)

ステップ 1 – ベースプロファイルを設定する

ベースプロファイルはチェーン内の最初のプロファイルであり、標準の認証方法を使用する必要があります。

1. /n/AWS1/IMG トランザクションを実行して、AWS SDK for SAP ABAP 実装ガイド (IMG) を起動します。
2. AWS SDK for SAP ABAP 設定 > アプリケーション設定 > SDK プロファイルを選択します。
3. 新しいエントリを選択し、プロファイル名と説明を入力して、ベースプロファイルとして使用する新しいプロファイルを作成します。[保存] を選択します。

Note

標準認証方法 (INST、SSF、または RLA) で既に設定されている既存のプロファイルを使用している場合は、このセクションの残りのステップをスキップして、[直接進むことができます](#) [the section called “ステップ 2”](#)。

4. 作成したプロファイルを選択し、認証と設定 > 新規エントリを選択し、次の詳細を入力します。
 - SID: SAP システムのシステム ID
 - クライアント: SAP システムのクライアント
 - シナリオ ID: Basis 管理者が作成したDEFAULTシナリオを選択します。
 - AWS リージョン: 呼び出し先の AWS リージョン

- 認証方法: 次のいずれかを選択します。
 - Amazon EC2 で実行されている SAP システムのメタデータを介したインスタンスロール
 - オンプレミスまたはその他のクラウドシステムの SSF ストレージからの認証情報
 - 証明書ベースの認証用の IAM Roles Anywhere

[保存] を選択します。

5. IAM ロールマッピング > 新しいエントリを選択し、次のように入力します。

- シーケンス番号: 1
- 論理 IAM ロール: わかりやすい名前 (例: DEV_BASE_ROLE)
- IAM ロール ARN: 最初のアカウントの IAM ロールの ARN (例: `arn:aws:iam::111111111111:role/DevBaseRole`)

[保存] を選択します。

ステップ 2 – 連鎖プロファイルを設定する

チェーン内の各中間プロファイルと最終プロファイルを設定します。

SHARED_SERVICESプロファイル (からのチェーン**DEV_BASE**):

1. `/n/AWS1/IMG` トランザクションを実行します。
2. AWS SDK for SAP ABAP 設定 > アプリケーション設定 > SDK プロファイルを選択します。
3. [新規エントリ] を選択します。プロファイル名 (例: SHARED_SERVICES) と説明を入力します。[保存] を選択します。
4. 作成したプロファイルを選択し、認証と設定 > 新規エントリを選択し、次の詳細を入力します。
 - SID: SAP システムのシステム ID
 - クライアント: SAP システムのクライアント
 - シナリオ ID: Basis 管理者が作成したDEFAULTシナリオを選択します。
 - AWS Region: 呼び出し先の AWS リージョン
 - 認証方法: ドロップダウンからソースプロファイルを選択する
 - ソースプロファイル ID: ベースプロファイルのプロファイル ID を入力します (例: DEV_BASE)

[保存] を選択します。

~~5. IAM ロールマッピング > 新しいエントリを選択し、次のように入力します。~~

- シーケンス番号: 1
- 論理 IAM ロール: わかりやすい名前 (例: SHARED_ROLE)
- IAM ロール ARN: `arn:aws:iam::222222222222:role/SharedServicesRole`

[保存] を選択します。

PROD_S3_ACCESS プロファイル (からのチェーン **SHARED_SERVICES**):

と同じ手順を繰り返しますが **SHARED_SERVICES**、以下を実行します。

- を名前 **PROD_S3_ACCESS** として使用する
- ソースプロファイル ID を に設定する **SHARED_SERVICES**
- IAM ロールマッピング `arn:aws:iam::333333333333:role/ProdS3AccessRole` で **PROD_S3_ROLE** と を使用する

IAM ロール管理、信頼ポリシー設定、認可要件などのセキュリティのベストプラクティスについては、[「IAM 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。

SAP 認証情報ストアの使用

SAP 認証情報ストアは、シークレットアクセスキー認証用の認証情報を安全に保存するために SAP Business Technology Platform で使用されます AWS。サービスを使用するには、サブスクリプションが必要です。

次の手順では、SDK プロファイルが既に設定されていることを前提としています。詳細については、[「 の設定 AWS SDK for SAP ABAP 」](#) を参照してください。

設定を開始する前に、前提条件を満たしていることを確認してください。詳細については、[「SAP 認証情報ストア」](#) を参照してください。

トピック

- [設定手順](#)
- [SDK での SAP 認証情報ストアの使用](#)

設定手順

Steps

- [ステップ 1: 認証の設定を構成する](#)
- [ステップ 2: サービスキーを作成する](#)
- [ステップ 3: サービスキーを .p12形式に変換する](#)
- [ステップ 4: SAP BTP、ABAP 環境に接続する](#)

ステップ 1: 認証の設定を構成する

認証用の Credential Store 設定を構成するには、次の手順を実行します。

1. SAP 認証情報ストアインスタンスの設定タブに移動します。
2. 設定の編集を選択します。
 - デフォルトの認証タイプとして相互 TLS を選択します。
 - ペイロード暗号化ステータスが無効を選択します。ペイロードは HTTPS で転送中に暗号化されます。ただし、現在ペイロードを二重暗号化することはできません。
3. [保存] を選択します。

ステップ 2: サービスキーを作成する

次の手順を使用して、Credential Store のサービスキーを作成します。

1. SAP 認証情報ストアアプリケーションの左側のペインで、サービスキーに移動します。
2. サービスキーの作成を選択します。
3. サービスキーの名前を入力し、作成を選択します。

サービスキーは、選択した認証タイプに基づいて作成されます。サービスキーをダウンロードし、後で使用するために安全に保管します。

ステップ 3: サービスキーを .p12形式に変換する

通信システムのアウトバウンドユーザーを作成するには、.p12形式のクライアント証明書が必要です。次の手順を使用して、Credential Store Service キーで提供されている .p12証明書の詳細から証明書を生成します。

1. SAP トラストセンターサービスから SAP クラウドルート CA 証明書 (SAP で必要) をダウンロードします。 <https://support.sap.com/en/offerings-programs/support-services/trust-center-services.html>

2. SAP クラウドルート CA 証明書を任意のテキストファイル形式で開きます。ファイルの最後に Enter キーを押し、サービスキーの証明書フィールドから証明書をコピーペーストします。新しい行文字\nを実際の新しい行 (Enter) に置き換え、証明書全体を .cer ファイル形式で保存します。
3. サービスキーのキーフィールドからキーをコピーします。このプライベートキーは機密データとして扱う必要があります。テキストファイルに貼り付け、新しい行文字\nを実際の新しい行 (Enter) に置き換えます。プライベートキーをテキストファイルに保存します。
4. 前のステップで生成された証明書とプライベートキーを使用して、次のコマンドを実行して .p12 証明書を生成します。

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in  
<certificate.cer>
```

コマンドでは、エクスポートパスワードの検証が必要でした。さらに使用するためにパスワードを保持します。

プライベートキーに保存された .key テキストファイルを削除します。

ステップ 4: SAP BTP、ABAP 環境に接続する

SAP 認証情報ストアに接続するように SAP BTP、ABAP 環境を設定します。

トピック

- [通信システム](#)
- [コミュニケーションの配置](#)

通信システム

次のステップを使用して、SAP BTP、ABAP 環境から SAP 認証情報ストアへの通信を有効にする通信システムを作成します。

1. ABAP 環境システムの Fiori 起動パッドを開きます。
2. 通信システムタイルを選択してアプリケーションを開きます。
3. [新規] を選択します。
4. 通信システムの名前と ID を入力し、作成を選択します。たとえば、システム に名前を付けることができます ZSAP_CREDSTORE。
5. その他の必要な情報を入力します。

- ホスト名: サービスキー URL からホスト名をコピーします。たとえば、URL が の場合 `https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`、ホスト名は `credstore.mesh.cf.us10.hana.ondemand.com`。
 - アウトバウンド通信のユーザー: を選択して新しいユーザー+を追加します。
 - a. 認証メカニズムとして SSL クライアント証明書を選択します。
 - b. 新しい証明書のアップロードを選択します。
 - 前のステップで生成された .p12 証明書を参照します。
 - 説明を入力します。
 - .p12 証明書の生成に使用されたエクスポートパスワードを入力します。
 - アップロードを選択します。
 - c. Create を選択してアウトバウンドユーザーを作成します。
6. [保存] を選択します。
 7. 前のステップでダウンロードしたサービスキーを削除します。

コミュニケーションの配置

以下のステップを使用して、アウトバウンド通信の通信シナリオを提供する通信配置を作成します。

1. ABAP 環境システムの Fiori 起動パッドを開きます。
2. 通信配置タイルを選択してアプリケーションを開きます。
3. [新規] を選択します。
4. 通信シナリオ を選択し /AWS1/CRED_COMM_SCENARIO、通信配置の名前を入力します。例えば、Z_AWS_SDK_TO_SAP_CREDSTORE。
5. [作成] を選択します。
6. コミュニケーションシステムフィールドで、前のステップで作成したコミュニケーションシステムを参照します。その他の情報は、システムの選択後に自動入力されます。
7. [保存] を選択します。
8. 接続の確認を選択して接続をテストします。

この設定が完了すると、ABAP 環境は通信配置を使用して、アウトバウンドサービス (HTTP) 経由で SAP 認証情報ストアサービスを使用できます。

SDK での SAP 認証情報ストアの使用

Steps

- [ステップ 1: 名前空間と認証情報を作成する \(複数可\)](#)
- [ステップ 2: カスタムビジネス設定アプリケーションを設定する](#)

ステップ 1: 名前空間と認証情報を作成する (複数可)

SAP ヘルプを使用して SAP 認証情報ストアに名前空間と認証情報を作成する – [認証情報を作成、編集、削除します](#)。

キータイプの認証情報を作成するには、次の詳細を入力します。

- 名前空間 – 名前空間の名前を入力し、関連する認証情報をグループ化します。
- 名前 – キーの名前を入力します。が推奨されます。ここでaws-0123456789012-usernameは次のようになります。
 - 0123456789012 は、認証情報がアクセスを許可する AWS アカウント ID です。
 - username は、認証情報が属する IAM ユーザー名です。
- 値 – base-64 でエンコードされたシークレットアクセスキーを入力します。次のコマンドを使用して、base-64 でシークレットアクセスキーをエンコードします。

```
xargs echo -n | base64 # just press enter, do not enter arguments on the command line
MySecretAccessKey
Ctrl-D
```

コマンドは標準入力からシークレットアクセスキーを読み取り、末尾の改行なしで base64 に渡します。base-64 でエンコードされたシークレットアクセスキーを画面に出力します。SAP 認証情報ストアに値をコピーした後、ターミナルをクリアまたは閉じます。

- ユーザー名 – アクセスキー ID を入力します。
- [作成] を選択します。

1 つの認証情報を持つ新しい名前空間が作成され、この名前空間内で認証情報を追加、削除、または変更できます。

最小権限の原則に従って、名前空間に保存されている認証情報へのアクセスを管理します。

ステップ 2: カスタムビジネス設定アプリケーションを設定する

次の手順を使用して、SDK による認証に使用する認証情報を定義するようにカスタムビジネス設定アプリケーションを設定します。

1. ABAP 環境システムの Fiori 起動パッドを開きます。
2. カスタムビジネス設定タイルを参照してアプリケーションを開きます。
3. SDK プロファイルのビジネス設定を開きます。
4. SAP 認証情報ストアの認証設定を構成する必要がある SDK プロファイルを選択します。
5. 選択したプロファイルの認証と設定タブで編集を選択し、次の詳細を入力します。
 - 認証方法 – SAP 認証情報ストアから認証情報を選択します。
 - 名前空間 – SAP 認証情報ストアで作成された名前空間を入力します。詳細については、「[the section called “ステップ 1: 名前空間と認証情報を作成する \(複数可\)”](#)」を参照してください。
 - キー名 – 作成したキー認証情報の名前を入力します。詳細については、「[the section called “ステップ 1: 名前空間と認証情報を作成する \(複数可\)”](#)」を参照してください。
 - 通信配置 – 作成した通信配置の名前を入力します。詳細については、「[the section called “コミュニケーションの配置”](#)」を参照してください。
6. 適用を選択して AWS SDK プロファイル画面に移動します。
7. Select Transport を選択して、値のヘルプを使用してトランスポートを選択します。
8. [保存] を選択します。

トラブルシューティング AWS SDK for SAP ABAP

このセクションでは、発生する可能性のあるエラーシナリオのトラブルシューティング手順について説明します。

トピック

- [インポートの失敗](#)
- [未指定の場所の制約](#)
- [SSL エラー](#)
- [プロファイル設定](#)
- [IAM 認可](#)
- [必要なアクションを実行するための承認](#)
- [アクティブシナリオ](#)
- [コード内の特殊文字](#)
- [接続](#)

インポートの失敗

問題 — クラス 'CL_SYSTEM_UUID' にはインターフェイス 'IF_SYSTEM_UUID_RFC4122_STATIC' が含まれていません

原因 — SAP NOTE 0002619546 がシステム上にありません。

解決策 — [SAP Note 0002619546](#) がシステムに適用されていることを確認してください。

未指定の場所の制約

問題 — 未指定の場所の制約は、このリクエストが送信された region 固有のエンドポイントと互換性がありません。

原因 – Amazon S3 バケットに io_createbucketconfiguration パラメータの AWS リージョンがありません。

解決策 — us-east-1 を除く任意のリージョンでバケットを作成する場合は、createbucket() で io_createbucketconfiguration パラメータを使用して Amazon S3 バケットのリージョンを指定します。us-east-1 の制約を指定する必要はありません。

次に、正しく設定された `io_createbucketconfiguration` パラメータの例を示します。

```
createbucket(  
  iv_bucket = 'amzn-s3-demo-bucket'  
  io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )  
).
```

SSL エラー

問題 — SSL サーバー証明書のホスト名の不一致または `docs.aws.amazon.com:443` との SSL ハンドシェイクが失敗しました: `SSSLERR_NO_SSL_RESPONSE`

原因 — DEFAULT プロファイルで `icm/HTTPS/client_sni_enabled` パラメータが `TRUE` に設定されていません。

解決策 — 以下の手順に従って、特定の問題やその他の SSL 関連の問題のトラブルシューティングを行います。

1. SAPGUI を開き、コマンドバーに移動します。
2. トランザクション `RZ10` を実行します。
3. [プロファイル] に移動し、DEFAULT プロファイルを選択します。バージョンは自動的に入力されます。
4. [プロファイルの編集] セクションで [延長メンテナンス] を選択し、[変更] を選択します。
5. `icm/HTTPS/client_sni_enabled` パラメータを検索します。

- パラメータが存在する場合は、[パラメータ値] を編集して `TRUE` に設定します。
- パラメータが存在しない場合は、次のステップに従ってパラメータを作成します。

1. [パラメータ] を選択します。

Note

編集 (鉛筆アイコン) ではなく、作成用にパラメータを選択していることを確認してください。

2. [パラメータ名] フィールドに `icm/HTTPS/client_sni_enabled` と入力します。
3. [パラメータ値] フィールドに `TRUE` と入力します。

4. [保存] を選択します。
6. これらの変更を DEFAULT プロファイルに保存し、終了します。

プロファイル設定

問題 — <sid>:<client> のシナリオ DEFAULT を持つプロファイル <profile_name> の下に設定が見つかりませんでした

原因 — <profile_name> が正しくないか、設定されていません。

解決策 — 次の手順を実行してプロファイルを設定します。

1. SAPGUI を開いてトランザクション /n/AWS1/IMG を実行します。
2. [アプリケーション設定] > [SDK プロファイル] に移動します。
 - プロファイルが設定されている場合は、プロファイル名が正しいことを確認します。
 - プロファイルが設定されていない場合は、手順に従ってプロファイルを設定します。
3. [新規エントリ] を選択します。
 - a. プロファイルの [名前] と [説明] に入力します。
 - b. [保存] を選択します。
4. 前のステップで作成したエントリを選択し、[認証と設定] を選択します。
5. [新規エントリ] を選択し、次の詳細を入力して [保存] を選択します。
 - SID
 - クライアント
 - シナリオ ID
 - AWS リージョン
 - 認証方法
 - で実行されている SAP システムのメタデータからインスタンスロールを選択します AWS。
 - オンプレミスまたは他のクラウドで実行されている SAP システムの場合は、[SSF ストレージからの認証情報] を選択します。
6. [IAM ロールマッピング] > [新規エントリ] の順に選択し、次の詳細を入力して [保存] を選択します。

- シーケンス番号
- 論理 IAM ロール
- IAM ロール ARN

IAM 認可

問題 — ロール <iam_role_arn> を引き受けられないか、ユーザー <user_arn> にリソース <iam_role_arn> で sts:AssumeRole を実行する権限がありません

原因 – このエラーが表示される理由として考えられるものは、次のとおりです。

- 指定された IAM ロール ARN が正しくない
- IAM ユーザーに IAM ロールへのアクセス許可がない
- 引き受け元の IAM ロールと引き受け先の IAM ロールまたは IAM ユーザーとの間に信頼関係がない

解決策 — 次の手順を使用して、IAM ロール ARN が正しいことを確認します。

1. SAPGUI を開いてトランザクション /n/AWS1/IMG を実行します。
2. [アプリケーション設定] > [SDK プロファイル] に移動し、IAM ロールで設定したプロファイルを選択します。
3. [IAM ロールマッピング] を選択し、IAM ロール ARN を確認または修正します。
 - IAM ロール ARN が正しい場合は、IAM ロールが正しく設定されていることを確認してください。詳細については、「[IAM ロールのトラブルシューティング](#)」を参照してください。

必要なアクションを実行するための承認

問題 — ユーザー <user_arn> にはリソース <resource_arn> で <action> を実行する権限がありません

原因 — ユーザーにはアクションを実行する権限がありません。

解決策 — user_arn に resource_arn で指定された action を実行するために必要な権限を設定する必要があります。詳細については、「[IAM リソースにアクセスするために必要な許可](#)」を参照してください。

アクティブシナリオ

問題 — アクティブなシナリオが設定されていません

原因 — アクティブなシナリオの設定に失敗しました。

解決策 — アクティブなシナリオを設定するには、「[ランタイム設定](#)」を参照してください。

コード内の特殊文字

警告 — 0x00A0 という文字を ABAP ワードの一部にすることはできません。

Note

この警告の前には、さまざまなエラーメッセージが表示されることがあります。

原因 — 異なるソースからコードをコピーして貼り付けると、コードに特殊文字が挿入される可能性があります。

解決策 — ABAP ソースコードエディタにコードを貼り付けると、次のポップアップが表示されます。

改行なしのスペース文字が検出されました。スペースに変換しますか？

この質問に対する答えとして [はい] を選択します。また、コードボックスのコピーボタンを使用する代わりに、コピーするコードを選択することをお勧めします。

接続

問題 — SCLNT_HTTP (411): tla.region.amazonaws.com:443 への直接接続に失敗しました:
NIECONN_REFUSED(-10)

原因 — SAP システムがインターネットに接続できず、tla.region.amazonaws.com のポート 443 への TCP/IP 接続を確立できません。

解決策 — SAP システムは、直接またはプロキシサーバーを介して、HTTPS ポート 443 上の AWS エンドポイントへの接続を確立する必要があります。インターネット接続を確立/検証する方法には次のようなものがあります。

- NAT またはインターネットゲートウェイを介したインターネットへの直接アウトバウンド接続
- プロキシサーバー経由の接続

詳細については、「[プロキシサーバー経由の接続](#)」を参照してください。

その他のトピック

このセクションでは、次のトピックについて説明します。

トピック

- [AWS SDK for SAP ABAP リリース](#)
- [SAP ライセンス](#)

AWS SDK for SAP ABAP リリース

AWS SDK for SAP ABAP はトランスポートで配信され、AWS SDK for SAP ABAP - BTP エディションはアドオンとして配信されます。トランスポートとアドオンをインポートするメカニズムは異なりますが、技術的な機能は同じです。詳細については、「[セットアップ](#)」を参照してください。

トピック

- [リリース戦略](#)
- [ベストプラクティス](#)
- [SDK for SAP ABAP のパッチ適用](#)
- [追加モジュールのインストール](#)
- [SDK for SAP ABAP のアンインストール](#)

リリース戦略

のバージョン 1 AWS SDK for SAP ABAP は頻繁に更新されます。新しいパッチは、のリリースと更新に基づいて毎週または毎日リリースされます AWS のサービス。のパッチには、SDK のパッチレベルを更新するバグ修正やその他の変更を含める AWS のサービス ことができます。詳細については、[AWS SDKs](#)」を参照してください。

ベストプラクティス

すべての SAP システム (開発、QA、本番稼働環境) で SDK for SAP ABAP のパッチレベルを同じに保つことを推奨しています。

SDK にパッチを適用するときは、サンドボックスに最新バージョンをインポートしてください。その後、通常の変更管理手順に従って、開発、QA、および本番稼働システムにインポートできます。

SDK for SAP ABAP のパッチ適用

SDK for SAP ABAP の各リリースは、すべてのバグ修正、機能、アップデートを含む一連の累積トランスポートとして提供されます。パッチとインストールトランスポートの違いはありません。最新のトランスポートをインポートして、SDK for SAP ABAP にパッチを適用する必要があります。

core ランタイムモジュールと API モジュールの依存関係により、個々の API モジュールに個別にパッチを適用することはできません。パッチを適用するときは、特定のcoreモジュールが使用されなくなった場合でも、モジュールとインストールされている他のすべての API モジュールを更新する必要があります。

以下のシナリオでは、パッチ適用プロセスについて説明します。

• シナリオ 1: SDK バージョンの定期的な更新

最新の SDK バージョンにパッチを適用する場合:

- すべてのトランスポート (コア API モジュールとインストールされている API モジュール) を同時にインポートする
- 特定のインポートシーケンスは必要ありません
- 例えば、SDK のインストール時に core、ec2、および lmd トランスポートをインポートした場合、パッチを適用する際には、core、ec2、および lmd の最新のトランスポートをインポートする必要があります。

• シナリオ 2: 新しい API モジュールを追加する

現在インストールされているモジュールとは異なる SDK リリースから新しい API モジュール (Amazon Textract などtex) をインポートする場合:

- 既存のモジュールはすべて、新しいモジュールの SDK リリースバージョンと一致するようにパッチを適用する必要があります。
- 同時にインポートする (特定のシーケンスなし):
 - 既存のすべてのモジュール (core およびインストールされている API モジュール) の最新トランスポート
 - 新しい API モジュールのトランスポート tex

• シナリオ 3: 既存の API モジュールを更新する

既存の API モジュール (Amazon Translate などx18) を最新の SDK バージョンに更新する場合、相互依存関係のため、個々のモジュールの更新はサポートされていません。「」で説明されている

プロセスに従って、coreモジュールとインストールされているすべての API モジュールを一括に更新して、SDK 全体にパッチを適用する必要があります [Regular SDK Version Update \(p. 590\)](#)。

追加モジュールのインストール

既存の core およびモジュールと同じパッチレベルで新しいモジュールのトランスポートをインポートして、SAP システムに追加の API モジュールをインストールします。より新しいバージョンのモジュールをインポートする場合は、「[the section called “SDK for SAP ABAP のパッチ適用”](#)」に記載されているガイドラインに従ってください。これにより、すべての SDK モジュールでパッチレベルに互換性があることが保証されます。

SDK for SAP ABAP のアンインストール

SDK for SAP ABAP をアンインストールするには、<https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip> から削除トランスポートキットをダウンロードする必要があります。

```
curl "https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip" -o "uninstall-abapsdk-LATEST.zip"
```

署名ファイルは <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.sig> からダウンロードできます。ファイルを検証するには、「[SDK for SAP ABAP の検証](#)」を参照してください。

SAP システムにインストールされている SDK モジュールごとに、対応する削除トランスポートを上記の ZIP ファイルからインポートする必要があります。SDK 全体をアンインストールしなくても、1つのモジュールを削除できます。削除したいモジュールの削除トランスポートのみをインポートすることで削除できます。SDK 全体とそのすべてのモジュールをアンインストールする場合は、コア削除トランスポートを最後にインポートする必要があります。

開発システム、QA システム、または本番稼働システムでアンインストールを試みる前に、サンドボックスでアンインストールをテストすることをお勧めします。

考慮事項

SDK をアンインストールする前に、以下の考慮事項を確認してください。

- からの SDK 設定は失われます。IMG はインストール時に再構成する必要があります。
- SDK に依存する Z プログラムがある場合、SDK を削除すると構文エラーが生成されます。

- SDK 認可リファレンスを含む PFCG またはビジネスロールは、SDK の削除後に無効な認可を持ちます。SDK をアンインストールする前に、PFCG ロールから SDK 認可リファレンスを削除してください。

Note

AWS SDK for SAP ABAP - 開発者プレビュー中に BTP エディションをアンインストールすることはできません。

SAP ライセンス

SAP ソフトウェアの使用には SAP の規約が適用されます。ソフトウェア配布条件や間接ライセンス条件を含む SAP ライセンス条件を遵守する責任はお客様にあります。提供されている情報はいずれも法的助言ではなく、ライセンスの遵守を目的として当てにするべきものではありません。お使用の SAP ソフトウェアのライセンスや権限に関してご質問がございましたら、御社内の法務部門、SAP、または SAP 再販業者にお問い合わせください。

質問: SDK for SAP ABAP を使用することで SAP ライセンスに影響しますか？

応答: 独自の AWS SDK for SAP ABAP ABAP コード AWS のサービスでを使用できます。これは、SAP システムと AWS のサービスとの間の統合シナリオで使用されます。SAP システムのデータをサードパーティ (SAP 以外) のシステムに送信したり、そのシステムによって作成したりするシナリオは、いずれも間接ライセンスに影響する可能性があります。SAP では、ユーザーベースの計算や結果ベースの計算など、間接アクセスを定義するためのアプローチが複数あります。間接アクセスを定義する方法は、SAP との契約によって異なります。SAP との契約書に記載されているガイダンスを把握しておく必要があり、詳細については SAP またはその再販業者と話し合うことができます。

2018 年に SAP は、「SAP インストールベースのお客様向け間接アクセスガイド」と「デジタル時代の SAP ERP 価格設定 - 間接/デジタルアクセスに対応」という 2 つの文書をリリースしました。これらの文書は SAP の Web サイトに掲載されており、間接ライセンス手法の例となっています。ただし、これらの文書にはお客様と SAP との特定の契約は反映されていません。

AWS SDK for SAP ABAP デベロッパーガイドのドキュメント履歴

次の表に、のドキュメントリリースを示します AWS SDK for SAP ABAP。

変更	説明	日付
新しいコンテンツ	サービスごとのプロキシサーバーの上書き を追加しました。	2026 年 5 月 15 日
新しいコンテンツ	AWS SDK for SAP ABAP ナレッジ MCP サーバーの使用 を追加しました。	2026 年 4 月 24 日
新しいコンテンツ	クロスアカウントアクセスのソースプロファイルの使用 を追加しました。	2026 年 3 月 4 日
新しいコンテンツ	SSF 暗号化によるシークレットアクセスキー認証の使用 を追加しました。	2025 年 1 月 12 日
新しいコンテンツ	SDK for SAP ABAP - BTP エディションの開発者レビュー。	2024 年 5 月 31 日
新しいコンテンツ	「 IAM Roles Anywhere での証明書の使用 」が追加されました。	2023 年 12 月 1 日
新しいコンテンツ	「 SDK による製品の構築 」が追加されました。	2023 年 12 月 1 日
新しいコンテンツ	「 再試行動作 」が追加されました。	2023 年 12 月 1 日

新しいコンテンツ	「 SAP ライセンス 」が追加されました。	2023 年 9 月 22 日
パブリックリリース	AWS SDK for SAP ABAP デベロッパーガイドの初回再起動。	2023 年 6 月 30 日
新しいコンテンツ	AWS SDK for SAP ABAP 機能 を追加しました。	2023 年 5 月 30 日
新しいコンテンツ	「 AWS SDK for SAP ABAP のトラブルシューティング 」を追加しました。	2023 年 2 月 17 日
開発者プレビュー	デベロッパーガイドのデベロ AWS SDK for SAP ABAP ツ パープレビュー。	2022 年 11 月 17 日

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。