



開発者ガイド

# AWS SDK の SAP ABAP



# AWS SDK の SAP ABAP: 開発者ガイド

Copyright © 2024 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
使用開始 .....	3
ステップ 1: AWS アカウントを準備する .....	3
IAM SAPユーザーの ロール .....	3
認証 .....	4
手順 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
のインストール SDK - BTPエディション .....	27
- BTP Edition SAP ABAP SDKに をインストールする .....	27
モジュール .....	28
- SAP ABAP BTPエディションSDK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構成 .....	33
SDK プロファイル .....	33

論理リソースリゾルバー .....	34
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 .....	37
ゲートウェイエンドポイント .....	37
カスタムインターフェイスのエンドポイント .....	38
複数のリージョンでエンドポイントにアクセスする .....	39
サービスプロバイダーの設定 .....	40
更新、トレース、テレメトリー .....	40
SAP システムの更新 .....	41
トレース .....	41
Telemetry .....	42
SDK の使用 .....	44
データの表現 .....	44
データ型 .....	44
AWS データ型 .....	46
プログラムの例 .....	47
前提条件 .....	48
コード .....	48
コードセクション .....	49
概念 .....	51
API クラス .....	52
追加のオブジェクト .....	52
構造クラス .....	53
配列 .....	55
マップ .....	56
高レベルの関数 .....	56
機能 .....	1
プログラムによる設定 .....	57
ウェイター .....	57
ページネーター .....	59

再試行動作 .....	60
製品の構築 .....	60
製品 ID を設定する .....	61
制限事項 .....	61
コードの例 .....	63
Amazon Bedrock ランタイム .....	63
Anthropic Claude .....	64
Stable Diffusion .....	67
CloudWatch .....	69
アクション .....	70
シナリオ .....	74
DynamoDB .....	77
基礎 .....	77
アクション .....	70
Amazon EC2 .....	91
アクション .....	70
Kinesis .....	107
基礎 .....	77
アクション .....	70
Lambda .....	117
アクション .....	70
シナリオ .....	74
Amazon S3 .....	132
基礎 .....	77
アクション .....	70
SageMaker .....	140
アクション .....	70
シナリオ .....	74
Amazon SNS .....	159
アクション .....	70
シナリオ .....	74
Amazon SQS .....	168
アクション .....	70
シナリオ .....	74
Amazon Textract .....	176
アクション .....	70

シナリオ .....	74
Amazon Translate .....	186
アクション .....	70
シナリオ .....	74
セキュリティ .....	196
システム認証 .....	196
メタデータ認証 .....	197
シークレットアクセスキー認証 .....	197
IAM Roles Anywhere を使用した証明書ベースの認証 .....	198
次のステップ .....	198
IAM セキュリティのベストプラクティス .....	199
Amazon EC2 インスタンスプロファイルのベストプラクティス .....	199
SAP ユーザーの IAM ロール .....	200
SAP 認証 .....	203
設定の権限 .....	203
エンドユーザー向けの SAP 認証 .....	204
セキュアなオペレーション .....	205
保管中のデータの暗号化 .....	205
転送中のデータの暗号化 .....	206
API の使用 .....	2
証明書の使用 .....	206
前提条件 .....	206
手順 .....	207
認証情報ストア .....	21
設定ステップ .....	210
SDK での SAP 認証情報ストアの使用 .....	212
トラブルシューティング .....	216
インポートの失敗 .....	216
未指定の場所の制約 .....	216
SSL エラー .....	217
プロファイル設定 .....	218
IAM 認証 .....	219
アクションの承認 .....	219
アクティブシナリオ .....	36
特殊文字 .....	220
接続 .....	220

---

その他のトピック .....	222
リリース .....	222
リリース戦略 .....	222
ベストプラクティス .....	199
SDK for SAP ABAP のパッチ適用 .....	223
追加モジュールのインストール .....	223
SDK for SAP ABAP のアンインストール .....	223
SAP ライセンス .....	224
ドキュメント履歴 .....	226
.....	CCXXVII

# とは AWS SDK for SAP ABAP

AWS SDK for SAP ABAP は、 が提供する のサービスへのインターフェイスを ABAP AWS 言語で提供します。を使用すると SDK、 、 ABAP レポート BADIs、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、 、 Amazon DynamoDB などのその他の ABAP アーティファクトを に実装できます Amazon Translate。SAP NetWeaver 7.4 以降および SAP Business Technology Platform 環境で、 ABAP ベースのシステム用に を開発することもできます。詳細については、 [「 - SAP ABAP BTP エディション AWS SDK 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](#)
- [追加リソース](#)

## の機能 AWS SDK for SAP ABAP

AWS SDK for SAP ABAP は、 SAP ベロッパー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。例えば、すべての `true` および `false` 文字列 AWS のサービス を使用して XML および JSON 構造体のブールデータを表しますが、 SDK はこれらを ABAP ネイティブ 'X' および ' ' 1 文字の値 SAP ABAP 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など、可能な限りネイティブ ABAP コンストラクトを使用します。そのため、 ABAP プログラマーは基盤となる JSON と XML シリアル化、または API プロトコルのワイヤ形式について心配する必要はありません。

## 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とその基盤となる依存関係のメンテナンスとサポートについては、 [「 AWS SDKs および ツール リファレンス ガイド」](#) の以下を参照してください。

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



## API リファレンス

の完全なリストについては 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。

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

for - BTP Edition デベロッパープレビュー SAP ABAP SDKの完全なモジュールリストを確認するには TLAs、 「 for [AWS SDK- BTP Edition SAP ABAP - 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。で消費した AWS リソースとサービスに対してのみ料金が発生しますSDK。

## 追加リソース

このガイドに加えて、 SDKの SAP では以下のオンラインリソースを使用できますABAP。

- [SAP AWS ドキュメントの](#)
- [AWS 開発者ブログ](#)
- [AWS デベロッパーフォーラム](#)
- [AWS SDK コードサンプルライブラリ](#)
- [@awsdevelopers](#)(Twitter)

# の開始方法 AWS SDK for SAP ABAP

このセクションでは、 の使用を開始する方法について説明しますSDK。これには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ます。を初めて使用する場合は AWS SDK、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。

## ステップ

- [ステップ 1: AWS アカウントを準備する](#)
- [手順 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 の の使用を開始するにはSAPABAP、アクティブな が必要です AWS アカウント 。SAP システムがオンプレミス、SAPビジネステクノロジープラットフォーム (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、 内の サービスを呼び出し AWS ます AWS アカウント。

## トピック

- [IAM SAPユーザーの ロール](#)
- [認証](#)

## IAM SAPユーザーの 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で 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。後で使用するために、IAMロールの Amazon リソースネーム (ARN) を書き留めます。

- ユースケースEC2として Amazon 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られるようにするには、ロールに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。

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

この例では、の AWS アカウント プリンシパルがロールを引き受け"**111122223333**"ることができることを示しています。これは、に適した広範なアクセス許可です proof-of-concept。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。
  - SSFオンプレミスSAPシステムから暗号化された認証情報
  - BTP、ABAP環境の SAP Credential Store SAP サービスからの認証情報
- 特定のロール — SAPシステムが Amazon にありEC2、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 にありEC2、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。

### ロケーション

- [AWS クラウド上](#)
- [オンプレミスSAPBTP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、次のアクセス許可を持つインスタンスプロファイルがあることを確認します。

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

前のステップで書き留めARNを追加します。

このアクセス許可により、SAPシステムはABAPユーザーに代わってSapDemoTranslateロールを引き受けることができます。

## オンプレミスSAPBTPまたはその他のクラウド

SAP システムがオンプレミス、オンプレミス、SAPBTPまたは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 ユーザーの作成 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SapDemoSIDとして を使用します。SIDはシステムのSAP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 とは保持します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、オンプレミス、SAPBTPまたは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 Credential Store SAP 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](#)

## 手順 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

SAP システム内のSAPABAPトランスポートSDKに をインポートします。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、 [「 のインストールSDK」を参照してくださいSAPABAP](#)。

### SDK for SAP ABAP - BTP edition

Deploy Product SAP ABAP アプリケーションBTPを使用して - Edition SDKに をインストールします。詳細については、 [「 - SAP ABAP BTPエディション SDK 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

を設定する前にSDK、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認証」](#)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の 実装ガイドを開きますSAPABAP。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/n/AWS1/IMG にと入力し、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
    - OPT-IN: enh テレメトリ : 空白のままにします。
  - [Save] 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 を使用して暗号化SAPSSFされ、SAPの などの設定済みの暗号化ライブラリが必要です CommonCryptoLib。

の SSFSDKを設定する手順はSAPABAP、/AWS1/IMGトランザクションで説明されています。

### Note

上記の前提条件は、SAPシステムが Amazon で実行されている場合には適用されません EC2。SAP Amazon で実行されている システムは、Amazon EC2インスタンスメタデータから有効期間がEC2短く、認証情報を自動的にローテーションします。

## 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 (コマンドバー/n/AWS1/IMGに /AWS1/IMGと入力し、Enter を選択) を実行して、[の実装ガイドを開きます AWS SDK](#)。

- 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
    - [Save] 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DEFAULTシナリオを見つけ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 システムで が実行されている場合は 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 で実行されている場合は、メタデータ経由でインスタンスロールを選択しますEC2。
      - 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 ロールの無効化は空白のままにします。
- [Save] 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 : と入力しますTESTUSER。
- IAM 「ロールARN」と入力します。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、「[認証情報ストアの使用 SAP](#)」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DEFAULTシナリオを見つけ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 システムで が実行されている場合は AWS、実行中の AWS リージョンを入力します。
- 認証方法: 認証情報ストア からSAP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 - BTP Edition と SAPCredential Store SAP ABAP との通信を確立するために作成された通信配置の名前を入力します。
- IAM ロールの無効化は空白のままにします。
- 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ル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ルール : と入力しますTESTUSER。
- IAM 「ルールARN」と入力します。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 デフォルトでは、ユーザーは AWS 機能を使用する権限がありません。ユーザーは、認証を使用して明示的にSAP認証されている必要があります。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。
  - [認証データを変更] 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。
  - 以下の認証オブジェクトを追加します。
    - /AWS1/LROL
    - /AWS1/SESS
  - 認証ツリーに次のように入力します。
    - にアクセスするためのプロファイルAWSAPIs : DEMO

- 論理IAMロール: TESTUSER
- [Save] 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。

### PFCG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、DEMOSDKプロファイルで設定された設定で関数を使用する AWS SDK権限が付与されます。承認されたユーザーは、そのプロファイルのTESTUSER論理IAMロールにマッピングされた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。
  - 機能をテスト AWS SDK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。
  - [Save] 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。
- 「一般的なロールの詳細」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 に設定します。
- 「制限を維持」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 「セッションを選択」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**、 の追加 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、 の追加 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、新しく作成されたビジネスロールを、SDK機能进行测试するSAPビジネスユーザーに割り当てます。[Save] 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には、DEMOSDKプロファイルで設定された設定で AWS SDK関数を使用する権限が付与されます。承認されたユーザーは、そのプロファイルのTESTUSER論理IAMロールにマッピングされた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。
  - [Save] 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_xl8unsuppdedlanguage00 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] を選択します。
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_xl8unsuppdedlanguage00 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.
```

を使用するABAPコードの記述方法の詳細についてはSDK、[「 の使用 AWS SDK for SAP ABAP 」](#)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 ロールが SDK プロファイルの使用を許可し、それを論理 IAM ロールにマッピングするが、SAP システムが IAM ロールを引き受けるように IAM アクセス許可が設定されていない場合、出力は次のようになります。

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

この場合、で定義されている IAM ロール、ユーザー、またはその両方に対する IAM アクセス許可と信頼設定を確認します [the section called “ステップ 1: AWS アカウントを準備する”](#)。

### 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 ロールが SDKプロファイルの使用を許可し、それを論理IAMロールにマッピングするが、SAPシステムがIAMロールを引き受けるようにIAMアクセス許可が設定されていない場合、出力は次のようになります。

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

この場合、で定義されているIAMロール、ユーザー、またはその両方に対するIAMアクセス許可と信頼設定を確認します [the section called “ステップ 1: AWS アカウントを準備する”](#)。

# 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](#)
- [for - BTP Edition SAP ABAP のインストール AWS SDK](#)

## 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。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>

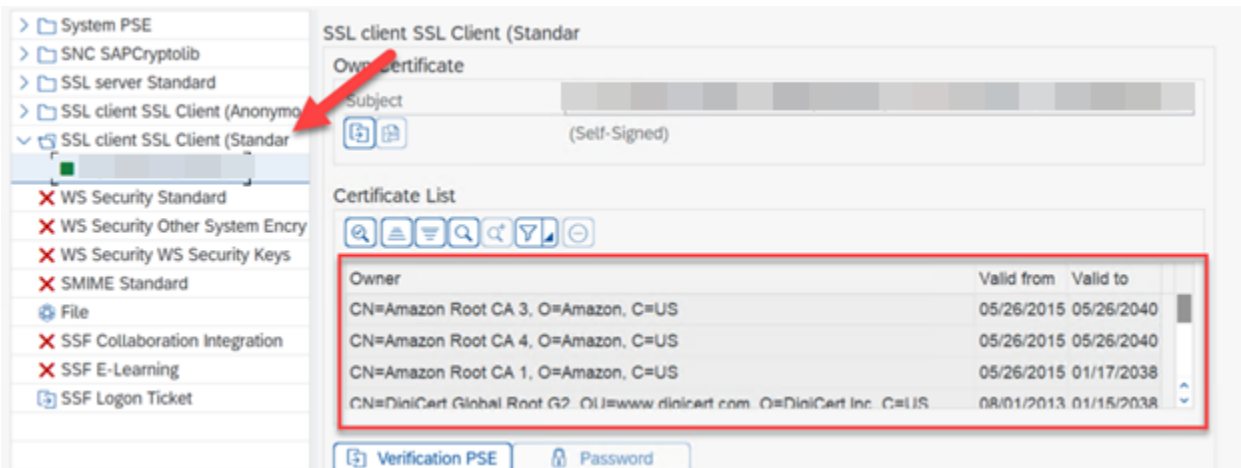
## アウトバウンド接続

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だけに使用されます。HTTP トラフィックはホスト内にとどまります。

メタデータにより、の SAP システムは AWS、SAP Secure Store にシークレットキーを保存することなく、自身を安全に認証できます。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 ランドスケープポータル – BTP エディション](#)
- [SAP 認証情報ストア – BTP エディション](#)

### SAP ランドスケープポータル – 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、[「ランドスケープポータル」](#)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。

デベロッパーのプレビューでは、AWS SDK for SAP ABAP - BTP エディションを認証するための唯一のサポートされているメカニズムは、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認証情報ストア。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 認証情報ストアサービスは、ABAP BTP システムの外部にある 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-----  
MIICIjANBgkqhkiG9w0BAQEFAAACg8AMIICCgKCAgEAmS3oN3wKBh4HJ0Ga0tye
```

```
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
051sTdEJJpfLEWJJZZNK3v9fGWkyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
BWZFW+Ux00kILz+8vCFSXMXZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuisJC21MPkz0
cHOR31MNsMLzm+d/gVklT31R/JwAcFCkXTwvR8/V0WNGZZXdVUbefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLj pz8dH
Zg25o79U9EJ1n+1pqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIUURLPKLVQUw
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、AWS Support 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 フォルダには core サブフォルダが含まれています。core サブフォルダには、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とも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、および 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 ( のモジュールを含む Amazon Translate) トランスポート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は次のとおりです。

- 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 AWS Support。
- 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は次のとおりです。

- 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 AWS Support。
- 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 AWS Support 前にお問い合わせください。

## for - BTP Edition SAP ABAP のインストール AWS SDK

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。をインストールするにはSDK、[AWS SDK- BTP Edition デベロッパープレビュー のSAPABAP 参加フォーム](#)に を入力します。

SDK for - BTP Edition SAP ABAP 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「ランドスケープポータル」](#)と[SAP「認証情報ストア」](#)を参照してください。

### トピック

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

## - BTP Edition SAP ABAP SDKに をインストールする

1. SAP ランドスケープポータルインスタンスに移動し、製品のデプロイ 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 AWS Support ない場合は、[お問い合わせ](#)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 Edition のバージョンを選択します。
4. 使用可能なシステム で、 をインストールするすべての のチェックボックスをオンにします SDK。SIDs
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 で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

for - BTP Edition の AWS SDKデベロッパープレビューにはSAPABAP、以下のモジュールが含まれています。

- [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 \[sgm\]](#)
- [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 データ API \[rsd\]](#)

## - SAP ABAP BTPエディションSDKのパッチ適用

for - BTP Edition SAP ABAP SDK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SDKシステムに をインストールすると、SDKは選択した新しいバージョン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SDKで 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 の .INIファイルで定義されている設定と直接似ていますSDKs。

ランタイムSDK設定を除く設定は、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。本番稼働環境では、トランスポート設定は推奨されません。

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

### の設定 AWS SDK for SAP ABAP

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

- SAP ABAP BTPエディション AWS SDKの設定

- Edition SDKの SAP ABAP を設定するには、次のステップに従いますBTP。

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、設定のタイプに基づいてSDK設定をグループ化できます。設定をグループ化するには、次のステップに従います。

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	
<b>Application Configuration</b>		
SDK Profile	Maintain AWS SDK Profile	>
Logical Resource Resolver	Maintain Logical Resource Resolution	>
<b>Global Settings</b>		
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構成](#)
- [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](https://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。
- [Save] 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
- [Save] 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構成

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。例えば、トランザクション ZVA01、ZVA02、ZVA03が請求書関連のトランザクションを拡張し AWS のサービス、Amazon S3、などの で実行 Amazon SageMakerする場合 AWS Lambda、 という SDK プロファイルZINVOICEを作成できます。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トランスアセンテーションと 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。
- 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。

#### 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、必要な設定を取得する 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 バケットなどの 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](#)
- [オプトイン: 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、サポート目的でテレメトリ情報を AWS 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、AWS Support 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 エンドポイントに HTTPS 呼び出しを行うことで AWS のサービスを使用します。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](#)
- [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] を選択します。
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。

## 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、インターネットがなくても VPC に高性能アクセスを提供するゲートウェイエンドポイントを提供しています。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 – [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?](#)」を参照してください。

次の例では、AWS STS と Amazon Translate のインターフェイスエンドポイントが作成されます。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。その後、AWS セッションクラス (AWS STS を使用して IAM ロールを引き継ぐ) のファクトリーメソッドに渡され、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。

## 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 プロファイルに定義されているデフォルトの AWS リージョン 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( ).  
ENDLOOP.  
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SDK全体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、お客様に代わって自分の 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 は、サービスプロバイダーの設定SAPABAPをサポートします。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SDKに影響します。サービスプロバイダーの設定は、- SAP ABAP BTPエディションSDK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. 新規エントリを選択し、ビジネス要件に基づいて設定を調整します。
  - メタデータを無効にする – EC2 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、がすべてのクライアントのEC2インスタンスメタデータSDKにアクセスできないようにします。プログラムは、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SDKが発生します。
4. [Save] 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 AWS Support が発生した場合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 Support 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 AWS Support。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 AWS Support 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 AWS Support、は特定の API コールが行われた理由 AWS Support 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 バージョンに限定されています。

# 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](#)
- [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データ型の標準セットがあります。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 " "
文字列	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	STRING	ABAP は DECFLOAT をサポートしていますが、NaN、無限大、負の無限大などの値を表すことはできません。AWSSDK はこれらを内部的に文字列として表現し、実行時に DECFLOAT16 に変換します。NaN、無限大、ま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
ダブル	STRING	

AWS データ型	ABAP データ型	コメント
		対応してこれら进行处理できません。
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	
タイムスタンプ	TZNTSTMP	TZNTSTMP 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、以下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 で AWS のサービスをサポートするために、次のアプローチが統合されました。

- 特定の AWS データ型は ABAP では表現できません。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 データはオンライン上では JSON または XML として表され、値はオプションです。例えば、JSON で AWS のサービスが返す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 ロールにマップ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 アクセス権限を付与します。
    - 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/LR0L 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 システムが SDK プロファイルに定義されている方法を使用して AWS に対して自身を認証できる。
- 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プされた 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( ).
  ENDLOOP.
  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。AUTHORIZATION-CHECK への呼び出しは、認証オブジェクト /AWS1/SESS が満たされていることを確認するために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 ロールにマップ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。例えば、ABAP プログラムが Amazon S3、AWS Lambda、および 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 オブジェクトが作成されます。

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採用しており、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 ネイティブの CONVERT TIME STAMP コマンドで日付に変換できます。GET\_SIZE( ) は INT4 を返し、簡単な計算やフォーマット操作に役立ちます。

```
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、または認証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、ショートダンプ、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 [AWS SDK for SAP ABAP - モジュールリスト](#)」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 構造を渡しているためです。これは、引数のフラットセットでは表現できません。

クラスに 1 つの属性しか含まれていない場合でも、引RETURNING数は常にクラスです。

## 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ず。

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す。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 ハッシュテーブルでは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 を正確にミラーリングし、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。ファクトリークラスには、[API ドキュメント](#) とともに、残りの API とともに文書化されている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](#)
- [ウェイター](#)
- [ページネーター](#)
- [再試行動作](#)

## プログラムによる設定

AWS SDK for SAP ABAP には `/n/AWS1/IMG IMG` トランスアセンションを使用し、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。

## ウェイター

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。例えば、 の `CREATETABLE( )` API はすぐにテーブルステータス



で 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。ウェイターを使用すると、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。

ウェイターは、サービスオペレーションを使用して、AWS リソースが意図した状態に達するか、リソースが意図した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 = |example-bucket| ).

" Wait until the newly created bucket becomes available
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
    iv_bucket = |example-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越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ー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.
ENDTRY.
```

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

## 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「[再試行動作](#)」を参照してください。AWS SDKs

## SDK による製品の構築

AWS のサービスを使用する製品または ABAP アドオンは、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。

製品 ID をセッションに追加すると、各サービス呼び出しで AWS 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。このテレメトリにより、お客様の顧客に対して SDK の問題が発生した場合に、AWS Support は呼び出しを行っている製品を特定できます。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。

### 制限事項

AWS SDK for SAP ABAP には、すべての 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プロトコルバインディングに依存するモジュールは機能ioteventsしません。MQTTは HTTP ベースのプロトコルではなく、現在ではサポートされていません AWS SDK for SAP ABAP。
- HTTP/2 ストリーミング機能に依存するモジュールはまだサポートされていません。イベントストリームと連携するサービスの特定の操作はまだサポートされておらず、Amazon Kinesis Video Streams などのサービスのメディアストリーミング操作は機能しません。

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 エディションでは、デベロッパープレビュー中に次の制限があります。

- 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 SAPABAPコード例の

このトピックのコード例は、SAPABAPでの AWS SDK 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し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。

## サービス

- [SDK に を使用する Amazon Bedrock ランタイムの例 SAP ABAP](#)
- [CloudWatch SDKで を使用する の例 SAP ABAP](#)
- [の を使用する DynamoDB SDK の例 SAP ABAP](#)
- [SDK に を使用する Amazon EC2の例 SAP ABAP](#)
- [SDK で を使用する Kinesis の例 SAP ABAP](#)
- [SDK に を使用する Lambda の例 SAP ABAP](#)
- [で を使用する Amazon S3 SDK の例 SAP ABAP](#)
- [SageMaker SDKで を使用する の例 SAP ABAP](#)
- [SDK で を使用する Amazon SNSの例 SAP ABAP](#)
- [SDK に を使用する Amazon SQSの例 SAP ABAP](#)
- [SDK に を使用する Amazon Textract の例 SAP ABAP](#)
- [SDK の を使用した Amazon Translate の例 SAP ABAP](#)

## SDK に を使用する Amazon Bedrock ランタイムの例 SAP ABAP

次のコード例は、Amazon Bedrock ランタイムSAPABAPで のを使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

## トピック

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

## Anthropic Claude

### InvokeModel

次のコード例は、Invoke Model 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API。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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可能性のある /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て、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て、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 詳細については、[InvokeModel](#)「」のAWS SDKSAPABAPAPI「」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 を呼び出してイメージを生成する方法を示しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

Stable Diffusion を使用してイメージを作成します。

```
"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て、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 詳細については、[InvokeModel](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## CloudWatch SDKで を使用する の例 SAP ABAP

次のコード例は、SAPABAPでの を使用してアクションを実行し、AWS SDK一般的なシナリオを実装する方法を示しています CloudWatch。

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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

トピック

- [アクション](#)
- [シナリオ](#)

## アクション

### DeleteAlarms

次の例は、DeleteAlarms を使用する方法を説明しています。

SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteAlarms](#)「」の AWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeAlarms](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DisableAlarmActions](#) 「」のAWS SDKSAPABAPAPI 「」を参照してください。

## EnableAlarmActions

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

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[EnableAlarmActions](#) 「」のAWS SDKSAPABAPAPI 「」を参照してください。

## ListMetrics

次のコード例は、ListMetrics を使用する方法を示しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListMetrics](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[PutMetricAlarm](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## シナリオ

### アラームの使用を開始

次のコードサンプルは、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 の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
CREATE OBJECT lo_alarmname EXPORTING 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'.
  ENDMETHOD.

  "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'.
  ENDMETHOD.

  "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'.
  ENDMETHOD.
```

- API 詳細については、AWS SDK 「」の以下のトピックを参照してくださいSAPABAPAPI。
  - [DeleteAlarms](#)
  - [DescribeAlarms](#)
  - [DisableAlarmActions](#)
  - [PutMetricAlarm](#)

# の使用する DynamoDB SDK の例 SAP ABAP

次のコード例は、DynamoDB SAPABAPで のを使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

## トピック

- [基礎](#)
- [アクション](#)

## 基礎

### 基本を学ぶ

次のコードサンプルは、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する。
- 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ファイルからテーブルに映画データを書き込みます。
- 特定の年にリリースされた映画を照会する。
- 何年もの間にリリースされた映画をスキャンする。
- テーブルからムービーを削除し、テーブルを削除します。

## SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 ZCL_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'.
    ENDRY.

    " 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'.
    ENDRY.

    " Query item from table.
    TRY.
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(

```

```

        ( 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_attributevaluelist = 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 詳細については、AWS SDK 「」 の以下のトピックを参照してくださいSAPABAPI。
  - [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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[CreateTable](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteItem](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteTable](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeTable](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

**GetItem**

次の例は、GetItem を使用する方法を説明しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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(lo_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 詳細については、[GetItem](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

### Note

については、「」 を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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( ).
    ENDLLOOP.
    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 詳細については、[ListTables](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## PutItem

次の例は、PutItem を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[PutItem](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_attributevaluelist(
  ( 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_attributevaluelist = 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.
  ENDLOOP.
  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 詳細については、「の[クエリ](#) AWS SDKSAPABAPI」を参照してください。



## Scan

次の例は、Scan を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_attributevaluelist = 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.
  ENDLOOP.
  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 詳細については、「[のスキャンAWS SDKSAPABAPAPI](#)」を参照してください。

## UpdateItem

次の例は、UpdateItem を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[UpdateItem](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## SDK に を使用する Amazon EC2の例 SAP ABAP

次のコード例は、Amazon SAPABAPで のを使用してアクションを実行し、AWS SDK一般的なシナリオを実装する方法を示していますEC2。

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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

## トピック

- [アクション](#)

## アクション

### AllocateAddress

次の例は、AllocateAddress を使用する方法を説明しています。

SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[AllocateAddress](#) 「」のAWS SDKSAPABAPAPI 「」を参照してください。

### AssociateAddress

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

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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(
        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'.
ENDTRY.
```

- API 詳細については、[AssociateAddress](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.
  ENDRY.

```

- API 詳細については、[CreateKeyPair](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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(
      " oo_result is
returned for testing purposes. "
      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'.
  ENDRY.

```

- API 詳細については、[CreateSecurityGroup](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteKeyPair](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.  
ENDTRY.
```

```

    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 詳細については、[DeleteSecurityGroup](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeAddresses](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeAvailabilityZones](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

**DescribeInstances**

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

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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( ).
  ENDLLOOP.
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 詳細については、[DescribeInstances](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeKeyPairs](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## DescribeRegions

次の例は、DescribeRegions を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeRegions](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## DescribeSecurityGroups

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.  
  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 詳細については、[DescribeSecurityGroups](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[MonitorInstances](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## RebootInstances

次の例は、RebootInstances を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[RebootInstances](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ReleaseAddress](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## RunInstances

次の例は、RunInstances を使用する方法を説明しています。

## SDK の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_tagsspecifications TYPE /aws1/
cl_ec2tagsspecification=>tt_tagsspecificationlist.
DATA ls_tagsspecifications LIKE LINE OF lt_tagsspecifications.
ls_tagsspecifications = NEW /aws1/cl_ec2tagsspecification(
  iv_resourcetype = 'instance'
  it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
    (NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
  )
).
APPEND ls_tagsspecifications TO lt_tagsspecifications.

TRY.
  " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
  oo_result = lo_ec2->runinstances(                                " oo_result is
returned for testing purposes. "
  iv_imageid = iv_ami_id
  iv_instancetype = 't2.micro'
  iv_maxcount = 1
  iv_mincount = 1
  it_tagsspecifications = lt_tagsspecifications
  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 詳細については、[RunInstances](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[StartInstances](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.
  ENDIF.

```

```
" 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.
```

- API 詳細については、[StopInstances](#) 「」のAWS SDKSAPABAPAPI 「」を参照してください。

## SDK で を使用する Kinesis の例 SAP ABAP

次のコード例は、Kinesis SAPABAPで のを使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

トピック

- [基礎](#)
- [アクション](#)

## 基礎

### 基本を学ぶ

次のコードサンプルは、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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、AWS SDK 「」の以下のトピックを参照してくださいSAPABAPAPI。
  - [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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[CreateStream](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

**DeleteStream**

次の例は、DeleteStream を使用する方法を説明しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_knslimitexceeddex .  
    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 詳細については、[DeleteStream](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_knslimitexceeddex .  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
CATCH /aws1/cx_knsresourceinuseex .  
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.  
ENDTRY.
```



- API 詳細については、[DescribeStream](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_knsprovthruputexcex .
```

```
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 詳細については、[GetRecords](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_knslimitexceedex .
    MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[ListStreams](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_knskms throttlingex .
    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 詳細については、[PutRecord](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    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'.
ENDTRY.
```

- API 詳細については、[RegisterStreamConsumer](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## SDK に を使用する Lambda の例 SAP ABAP

次のコード例は、Lambda SAPABAPで の を使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

トピック

- [アクション](#)
- [シナリオ](#)

## アクション

### CreateFunction

次の例は、CreateFunction を使用する方法を説明しています。

SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_lmdcodestorageexclex.
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 詳細については、[CreateFunction](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteFunction](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[GetFunction](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## Invoke

次の例は、Invoke を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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(
    " oo_result is returned for
testing purposes. "
    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_lmdunsuppmediatyp00.
  MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
ENDTRY.

```

- API 詳細については、[「で を呼び出す AWS SDK SAP ABAP API」](#) を参照してください。

## ListFunctions

次の例は、ListFunctions を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListFunctions](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[UpdateFunctionCode](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## UpdateFunctionConfiguration

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.  
ENDTRY.
```

```
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 詳細については、[UpdateFunctionConfiguration](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## シナリオ

### 関数の使用を開始します

次のコードサンプルは、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つのパラメーター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。
- 新しいパラメーター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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'
  ).
  MESSAGE 'IAM role created.' TYPE 'I'.
  WAIT UP TO 10 SECONDS.          " Make sure that the IAM role is ready
for use. "
  CATCH /aws1/cx_iamentityalrddyex.
    MESSAGE 'IAM role already exists.' TYPE 'E'.
  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_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.

" 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 = lo_create_role_output->get_role( )->get_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_lmdcodestorageexcdex.
  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_lmdunsuppedmediatyp00.
    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_lmdcodestorageexclex.
        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.

```

```

    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'.
    ENDTRY.

    "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_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'.
  ENDRTRY.

" 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'.
  ENDRTRY.

" 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 'E'.
  ENDRTRY.

" 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 'E'.
    CATCH /aws1/cx_iamplynotattachableex.
        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'.
    ENDRY.

" 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 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
        MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    ENDRY.

    CATCH /aws1/cx_rt_service_generic INTO lo_exception.
        DATA(lv_error) = lo_exception->get_longtext( ).
        MESSAGE lv_error TYPE 'E'.
    ENDRY.
```

- API 詳細については、AWS SDK 「」 の以下のトピックを参照してくださいSAPABAPAPI。
  - [CreateFunction](#)
  - [DeleteFunction](#)
  - [GetFunction](#)
  - [Invoke](#)
  - [ListFunctions](#)
  - [UpdateFunctionCode](#)
  - [UpdateFunctionConfiguration](#)

# で使用する Amazon S3 SDK の例 SAP ABAP

次のコード例は、Amazon S3 SAPABAPでの を使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

## トピック

- [基礎](#)
- [アクション](#)

## 基礎

### 基本を学ぶ

次のコードサンプルは、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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
  ).
  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'.
    ENDTRY.

" 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'.
    ENDTRY.
```

- API 詳細については、AWS SDK 「」の以下のトピックを参照してくださいSAPABAPAPI。
  - [CopyObject](#)
  - [CreateBucket](#)
  - [DeleteBucket](#)
  - [DeleteObjects](#)
  - [GetObject](#)
  - [ListObjectsV2](#)
  - [PutObject](#)

## アクション

### CopyObject

次の例は、CopyObject を使用する方法を説明しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[CopyObject](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
TRY.  
  lo_s3->createbucket(  
    iv_bucket = iv_bucket_name
```

```
    ).  
    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 詳細については、[CreateBucket](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## DeleteBucket

次の例は、DeleteBucket を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteBucket](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteObject](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

**GetObject**

次のコード例は、GetObject を使用する方法を示しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[GetObject](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、「」の「」の [ListObjectsV2](#)AWS SDKSAPABAPAPI「」を参照してください。

## PutObject

次のコード例は、PutObject を使用する方法を示しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[PutObject](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## SageMaker SDKで を使用する の例 SAP ABAP

次のコード例は、SAPABAPでの を使用してアクションを実行し、AWS SDK一般的なシナリオを実装する方法を示しています SageMaker。

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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
CREATE OBJECT lo_production_variants
  EXPORTING
    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 詳細については、[CreateEndpoint](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## CreateModel

次の例は、CreateModel を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
CREATE OBJECT lo_primarycontainer
  EXPORTING
    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_sgmresourcecelimitexcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[CreateModel](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## CreateTrainingJob

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_max_depth.  
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE  
lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_eta.  
INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w ) INTO TABLE  
lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_eval_metric.  
INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w ) INTO TABLE  
lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_scale_pos_weight.  
INSERT VALUE #( key = 'scale_pos_weight' value = lo_hyperparameters_w ) INTO  
TABLE lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_subsample.  
INSERT VALUE #( key = 'subsample' value = lo_hyperparameters_w ) INTO TABLE  
lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_objective.  
INSERT VALUE #( key = 'objective' value = lo_hyperparameters_w ) INTO TABLE  
lt_hyperparameters.
```

```
CREATE OBJECT lo_hyperparameters_w EXPORTING 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."

```
CREATE OBJECT lo_trn_s3datasource
```

```
EXPORTING
  iv_s3datatype           = iv_trn_data_s3datatype
  iv_s3datadistributiontype = iv_trn_data_s3datadistribution
  iv_s3uri                = iv_trn_data_s3uri.

CREATE OBJECT lo_trn_datasource
EXPORTING
  io_s3datasource = lo_trn_s3datasource.

CREATE OBJECT lo_trn_channel
EXPORTING
  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."
CREATE OBJECT lo_val_s3datasource
EXPORTING
  iv_s3datatype           = iv_val_data_s3datatype
  iv_s3datadistributiontype = iv_val_data_s3datadistribution
  iv_s3uri                = iv_val_data_s3uri.

CREATE OBJECT lo_val_datasource
EXPORTING
  io_s3datasource = lo_val_s3datasource.

CREATE OBJECT lo_val_channel
EXPORTING
  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."
CREATE OBJECT lo_algorithm_specification
EXPORTING
  iv_trainingimage      = iv_training_image
  iv_traininginputmode = iv_training_input_mode.
```



```
"Create an ABAP object for resource configuration."
CREATE OBJECT lo_resource_config
  EXPORTING
    iv_instancecount = iv_instance_count
    iv_instancetype  = iv_instance_type
    iv_volumesizeingb = iv_volume_sizeingb.

"Create an ABAP object for output data configuration."
CREATE OBJECT lo_output_data_config
  EXPORTING
    iv_s3outputpath = iv_s3_output_path.

"Create an ABAP object for stopping condition."
CREATE OBJECT lo_stopping_condition
  EXPORTING
    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_sgmresourceLimitExcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[CreateTrainingJob](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."

```
CREATE OBJECT lo_s3datasource  
EXPORTING  
    iv_s3uri      = iv_tf_data_s3uri  
    iv_s3datatype = iv_tf_data_s3datatype.
```

"Create an ABAP object for data source."

```
CREATE OBJECT lo_datasource  
EXPORTING  
    io_s3datasource = lo_s3datasource.
```

"Create an ABAP object for transform data source."

```
CREATE OBJECT lo_transforminput  
EXPORTING  
    io_datasource      = lo_datasource  
    iv_contenttype     = iv_tf_data_contenttype  
    iv_compressiontype = iv_tf_data_compressiontype.
```

"Create an ABAP object for resource configuration."

```
CREATE OBJECT lo_transformresources  
EXPORTING  
    iv_instancecount = iv_instance_count  
    iv_instancetype  = iv_instance_type.
```

```
"Create an ABAP object for output data configuration."
CREATE OBJECT lo_transformoutput
  EXPORTING
    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'.
ENDTRY.
```

- API 詳細については、[CreateTransformJob](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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'.
ENDTRY.

"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 詳細については、[DeleteEndpoint](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## DeleteModel

次の例は、DeleteModel を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteModel](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## DescribeTrainingJob

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

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DescribeTrainingJob](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## ListAlgorithms

次の例は、ListAlgorithms を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListAlgorithms](#) 「」のAWS SDKSAPABAPAPI 「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListModels](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## ListNotebookInstances

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

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListNotebookInstances](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## ListTrainingJobs

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListTrainingJobs](#) 「」のAWS SDKSAPABAPAPI 「」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

次のコードサンプルは、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 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 の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_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.
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"
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_max_depth.
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_eta.

```

```
INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_eval_metric.
INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_scale_pos_weight.
INSERT VALUE #( key = 'scale_pos_weight' value = lo_hyperparameters_w ) INTO
TABLE lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_subsample.
INSERT VALUE #( key = 'subsample' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_objective.
INSERT VALUE #( key = 'objective' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

CREATE OBJECT lo_hyperparameters_w EXPORTING 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."
CREATE OBJECT lo_trn_s3datasource
EXPORTING
    iv_s3datatype           = iv_trn_data_s3datatype
    iv_s3datadistributiontype = iv_trn_data_s3datadistribution
    iv_s3uri                 = iv_trn_data_s3uri.

CREATE OBJECT lo_trn_datasource EXPORTING io_s3datasource = lo_trn_s3datasource.

CREATE OBJECT lo_trn_channel
EXPORTING
    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."
CREATE OBJECT lo_val_s3datasource
EXPORTING
```

```
    iv_s3datatype           = iv_val_data_s3datatype
    iv_s3datadistributiontype = iv_val_data_s3datadistribution
    iv_s3uri                 = iv_val_data_s3uri.

CREATE OBJECT lo_val_datasource EXPORTING io_s3datasource = lo_val_s3datasource.

CREATE OBJECT lo_val_channel
  EXPORTING
    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."
CREATE OBJECT lo_algorithm_specification
  EXPORTING
    iv_trainingimage     = iv_training_image
    iv_traininginputmode = iv_training_input_mode.

"Create an ABAP object for resource configuration."
CREATE OBJECT lo_resource_config
  EXPORTING
    iv_instancecount    = iv_instance_count
    iv_instancetype     = iv_instance_type
    iv_volumesizeingb   = iv_volume_sizeingb.

"Create an ABAP object for output data configuration."
CREATE OBJECT lo_output_data_config EXPORTING iv_s3outputpath =
iv_s3_output_path.

"Create an ABAP object for stopping condition."
CREATE OBJECT lo_stopping_condition EXPORTING 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."
    CREATE OBJECT lo_primarycontainer
    EXPORTING
        iv_image      = iv_training_image
        iv_modeldataurl = lv_model_data_url.

    "Create an Amazon SageMaker model."
    TRY.
        lo_sgm->createmodel(
            iv_executionrolearn = iv_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.

    "Create an endpoint production variant."
    CREATE OBJECT lo_production_variants
    EXPORTING
```

```
    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_sgmresourcecelimitexcd.
    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 詳細については、AWS SDK「」の以下のトピックを参照してくださいSAPABAPI。
  - [CreateEndpoint](#)
  - [CreateEndpointConfig](#)
  - [CreateModel](#)
  - [CreateTrainingJob](#)
  - [DeleteEndpoint](#)
  - [DeleteEndpointConfig](#)
  - [DeleteModel](#)
  - [DescribeEndpoint](#)
  - [DescribeTrainingJob](#)

## SDK で を使用する Amazon SNS の例 SAP ABAP

次のコード例は、Amazon SAPABAPで のを使用してアクションを実行し、AWS SDK一般的なシナリオを実装する方法を示していますSNS。

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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

トピック

- [アクション](#)
- [シナリオ](#)

## アクション

### CreateTopic

次の例は、CreateTopic を使用する方法を説明しています。

SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[CreateTopic](#) 「」の AWS SDKSAPABAPAPI 「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteTopic](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[GetTopicAttributes](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListSubscriptions](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## ListTopics

次の例は、ListTopics を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListTopics](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## Publish

次の例は、Publish を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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.
```

- API 詳細については、「の[公開](#)AWS SDKSAPABAPAPI」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[SetTopicAttributes](#) 「」のAWS SDKSAPABAPAPI 「」を参照してください。

## Subscribe

次の例は、Subscribe を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_snssubscriptionlmte00.
    MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、「[のサブスクリプト AWS SDKSAPABAPAPI](#)」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 の SAP ABAP

#### Note

については、「[」](#)を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、「[「のサブスクリプション解除」AWS SDK](#)」を参照してください  
SAPABAPAPI。

## 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 トピックを作成して発行する方法を示しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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、Amazon SQSFIFO キューをトピックに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 = '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 詳細については、AWS SDK 「」の以下のトピックを参照してくださいSAPABAPI。
  - [CreateTopic](#)
  - [Publish](#)
  - [Subscribe](#)

## SDK に を使用する Amazon SQSの例 SAP ABAP

次のコード例は、Amazon SAPABAPで の を使用してアクションを実行し、AWS SDK一般的なシナリオを実装する方法を示していますSQS。

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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

トピック

- [アクション](#)
- [シナリオ](#)

## アクション

### CreateQueue

次の例は、CreateQueue を使用する方法を説明しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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のを待機する 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 詳細については、[CreateQueue](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[DeleteQueue](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## GetQueueUrl

次の例は、GetQueueUrl を使用する方法を説明しています。

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_queue_name = iv_queue_name ).      "  
oo_result is returned for testing purposes. "  
    MESSAGE 'Queue URL retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_sqsqueue_does_not_exist.  
        MESSAGE 'The requested queue does not exist.' TYPE 'E'.  
ENDTRY.
```

- API 詳細については、[GetQueueUrl](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[ListQueues](#)「」のAWS SDKSAPABAPAPI「」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 の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_waittimeseconds = 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 詳細については、[ReceiveMessage](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## SendMessage

次の例は、SendMessage を使用する方法を説明しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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(
testing purposes. " " oo_result is returned for
    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 詳細については、[SendMessage](#)「」のAWS SDKSAPABAPAPI「」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、FIFOAmazon SNSトピックを作成して発行する方法を示しています。

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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、Amazon SQSFIFOキューをトピックに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_snstopicattrsmw=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmw=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmw( 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_snssubscriptionlmte00.

```

```

        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
        ENDRY.

        " 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'.
        ENDRY.

```

- API 詳細については、AWS SDK 「」 の以下のトピックを参照してくださいSAPABAPAPI。
  - [CreateTopic](#)
  - [Publish](#)
  - [Subscribe](#)

## SDK に を使用する Amazon Textract の例 SAP ABAP

次のコード例は、Amazon Textract SAPABAPで の を使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_textthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
    CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
  ENDTRY.

```

- API 詳細については、[AnalyzeDocument](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_texpvthruputexcdex.
    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 詳細については、[DetectDocumentText](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## GetDocumentAnalysis

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

## SDK の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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_textthrottlingex.
MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[GetDocumentAnalysis](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## StartDocumentAnalysis

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[StartDocumentAnalysis](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## StartDocumentTextDetection

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

## SDK の SAP ABAP

 Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 詳細については、[StartDocumentTextDetection](#)「」のAWS SDKSAPABAPAPI「」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

次のコードサンプルは、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 の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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 Textextract 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_texlimitexceededex.
  MESSAGE 'An Amazon Textextract 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 詳細については、AWS SDK 「」 の以下のトピックを参照してくださいSAPABAPAPI。
  - [GetDocumentAnalysis](#)
  - [StartDocumentAnalysis](#)

## SDK の を使用した Amazon Translate の例 SAP ABAP

次のコード例は、Amazon Translate SAPABAPで の を使用してアクションを実行し、AWS SDK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います。このリンクには、コンテキスト内でコードをセットアップして実行する方法の手順が記載されています。

### トピック

- [アクション](#)
- [シナリオ](#)

## アクション

### DescribeTextTranslationJob

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

SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
    EXPORTING
        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 詳細については、[DescribeTextTranslationJob](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## ListTextTranslationJobs

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

SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
CREATE OBJECT lo_filter
  EXPORTING
    iv_jobname = iv_jobname.

TRY.
  oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned for
testing purposes."
  EXPORTING
    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 詳細については、[ListTextTranslationJobs](#) 「」 のAWS SDKSAPABAPAPI 「」 を参照してください。

## StartTextTranslationJob

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

### SDK の SAP ABAP

#### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
```

```
CREATE OBJECT lo_inputdataconfig  
EXPORTING  
    iv_s3uri      = iv_input_data_s3uri  
    iv_contenttype = iv_input_data_contenttype.
```

```
"Create an ABAP object for the output data config."
```

```
CREATE OBJECT lo_outputdataconfig  
EXPORTING  
    iv_s3uri = iv_output_data_s3uri.
```

```
"Create an internal table for target languages."
```

```
CREATE OBJECT lo_targetlanguagecodes  
EXPORTING  
    iv_value = iv_targetlanguagecode.  
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.
```

```
TRY.
  oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
  EXPORTING
    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.
```

- API 詳細については、[StartTextTranslationJob](#) 「」 のAWS SDKSAPABAPAPI 「」 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 の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."  
    EXPORTING  
        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 詳細については、[StopTextTranslationJob](#) 「」の AWS SDKSAPABAPAPI 「」を参照してください。

**TranslateText**

次の例は、TranslateText を使用する方法を説明しています。

## SDK の SAP ABAP

**Note**

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
    EXPORTING
        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 詳細については、[TranslateText](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

## シナリオ

### 翻訳ジョブを開始する

次のコードサンプルは、以下の操作方法を示しています。

- 非同期バッチ翻訳ジョブを開始します。

- 非同期ジョブが完了するまで待ちます。
- 非同期ジョブを記述します。

## SDK の SAP ABAP

### Note

については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)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."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri      = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

"Create an ABAP object for the output data config."
CREATE OBJECT lo_outputdataconfig
  EXPORTING
    iv_s3uri = iv_output_data_s3uri.

"Create an internal table for target languages."
CREATE OBJECT lo_targetlanguagecodes
  EXPORTING
    iv_value = iv_targetlanguagecode.
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
  DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
    EXPORTING
      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."
        EXPORTING
            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 詳細については、AWS SDK 「」の以下のトピックを参照してくださいSAPABAPAPI。
  - [DescribeTextTranslationJob](#)
  - [StartTextTranslationJob](#)

# 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 とユーザーの間で共有される責任です。[責任共有モデル](#)ではこれを、クラウドのセキュリティ、およびクラウド内で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。また、は、安全に使用できるサービス AWS も提供します。コンプライアンス[AWS プログラム](#)コンプライアンスプログラムの一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。に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 による対象範囲内の](#)」、「[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 AWS のサービス、使用する によって決まります。また、お客様は、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

- [SAP でのシステム認証 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](#)
- [IAM Roles Anywhere での証明書の使用](#)
- [SAP 認証情報ストアの使用](#)

## SAP でのシステム認証 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。では、 のSDKプロファイル設定で選択された次の 3 つの認証方法 AWS SDK for SAP ABAP をサポートしていますIMG。

AWS SDK for SAP ABAP - BTP Edition は、Credential Store SAP 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。

## トピック

- [Amazon EC2 インスタンスメタデータ認証](#)
- [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](#)
- [次のステップ](#)

## Amazon EC2 インスタンスメタデータ認証

SAP Amazon で実行されている システムは、Amazon EC2 インスタンスメタデータから有効期間が短く、自動的にローテーションされる認証情報を取得EC2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。

に を使用する場合は、この認証方法を強くお勧めSDKしますSAPABAP。を有効にするには、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 で実行されている場合にのみ適用されます EC2。SAP オンプレミスまたは他のクラウド環境でホストされている 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、でSAPシステムを認証します AWS。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、IAM管理者から AWS 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。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SAPシステムを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 SAP を認証 AWS SDKしますABAP。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 envelopeと SSFのdevelope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。
- の SSFSDKを設定する手順はSAPABAP、/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。
- SAP 認証情報ストア
  - SAP 認証情報ストアを使用して - SAP ABAP BTPエディションの認証 AWS SDKを行います。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

認証機関 (CA) によって発行された X.509 証明書は、Roles Anywhere AWS Identity and Access Management での認証に使用できません。証明書は STRUST で設定する必要があります。CA は信頼アンカーとして IAM Roles Anywhere に登録する必要があります。プロファイルを作成して Roles Anywhere IAM が引き受けるロールとポリシーを指定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。

で IAM Roles Anywhere を使用方法の詳細な手順についてはSDKSAPABAP、「Roles [Anywhere での証明書の使用IAM](#)」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。

## 次のステップ

でSAPシステムを認証した後 AWS、 は SDKSAPABAP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](#)

## 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。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::0123456789:role/finance-cfo",
```

```
        "arn:aws:iam::0123456789:role/finance-auditor",  
        "arn:aws:iam::0123456789:role/finance-reporting"  
    ]  
}  
]  
}
```

上記の例では、SAP システムが CFO、AUDITOR、または REPORTING ユーザーの IAM ロールを引き受けることができます。AWSSDK は、SAP でのユーザーの PFCG ロールに基づいて、ユーザーに適した適切な 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。

- [SAP HANA 向け AWS 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、DynamoDB テーブルの読み取り、Amazon S3 内の PDF オブジェクトに対する Amazon Textract の呼び出し、AWS Lambda 関数の実行など、ユーザーのジョブを実行するための権限が必要です。すべての AWS SDK で同じセキュリティモデルが使用されています。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。

```
{
  "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。

```
{
  "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。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/SAP_SYSTEM_S4H"
      }
    }
  ]
}
```

# 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 エディションが 設定にアクセスできるようにするには、次のステップに従います。

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」で「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許可します。
4. ビジネスユーザータブに移動し、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 AWS SDK /AWS1/IMG 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を許可する認証オブジェクト /AWS1/SESS 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 (例えば、CFO が 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 AUDITOR 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、関係を破ります。

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 ユーザーが `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 ユーザーです。
- `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。

セッション名はユーザー名 `CloudTrail` 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。

## IAM Roles Anywhere での証明書の使用

SAP システムは、AWS Identity and Access Management Roles Anywhere での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 でのトラスト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 [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。

### ステップ

- [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。

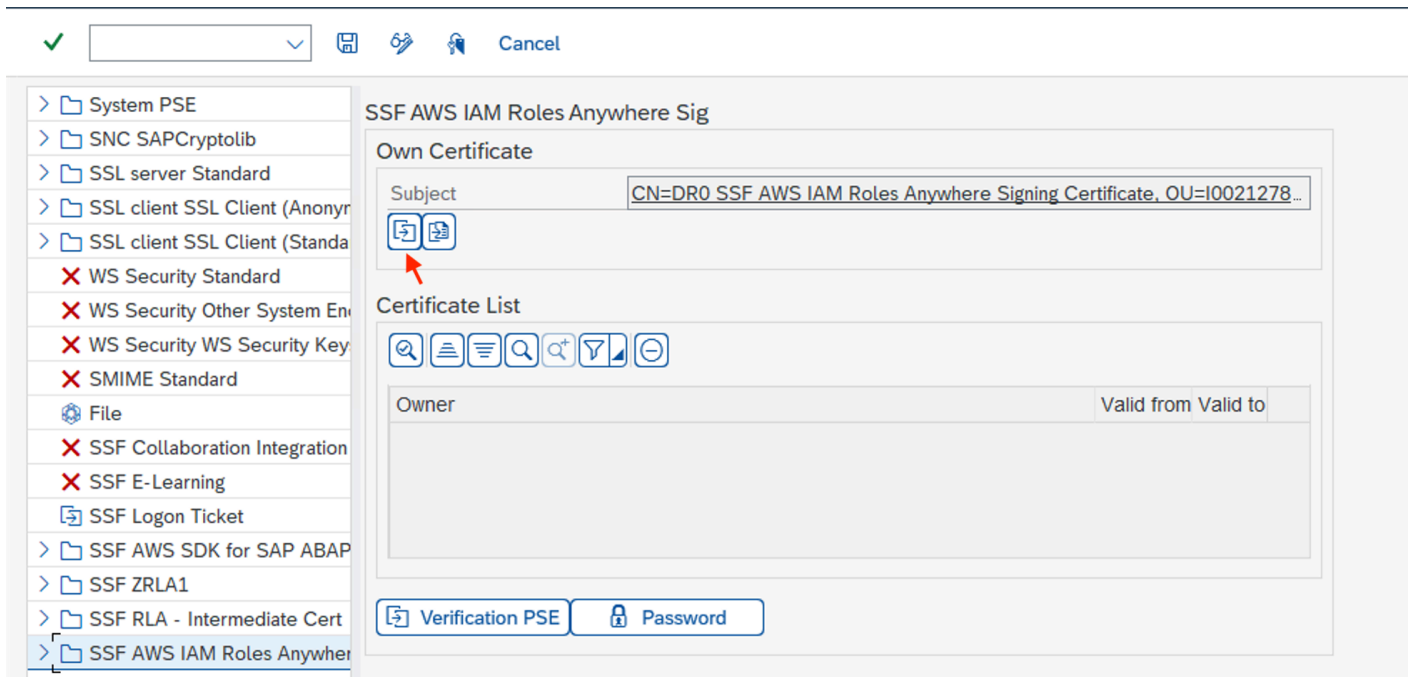
### ステップ 2 – SSF パラメータを設定する

1. を実行して /n/AWS1/IMG、AWS SDK for SAP ABAP 実装ガイド (IMG) を起動します。
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。[Save] 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、件名の下に PSE 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 の設定] > [技術的前提条件] > [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。



Select Signing Certificate issued by your certificate authority (CA) from SSF

Certificate (SSF Application)

Enter your IAM Roles Anywhere details

Trust Anchor ARN

Profile ARN

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。

## SAP 認証情報ストアの使用

SAP Credential Store は、への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

- [設定ステップ](#)
- [SDK での SAP 認証情報ストアの使用](#)

## 設定ステップ

ステップ

- [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 の設定を行うには、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で無効を選択します。ペイロードは HTTPS で転送中に暗号化されます。ただし、ペイロードは現在、二重暗号化できません。
3. [Save] を選択します。

## ステップ 2: サービスキーを作成する

認証情報ストアのサービスキーを作成するには、次のステップに従います。

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、 の作成を選択します。

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つ。

## 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。認証情報ストアサービスキーで提供される.p12証明書の詳細から証明書を生成するには、次のステップを実行します。

1. SAP Trust Center Services から SAP Cloud Root CA [証明書 \(SAP で必要\)](#) をダウンロードします。

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/貼り付けします。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、新しい行文字を\n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。パスワードは保持して、後で使用します。

プライベートキーに保存されている .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

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するには、次のステップを実行します。

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、 の作成を選択します。例えば、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。例えば、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. 作成 を選択してアウトバウンドユーザーを作成します。

## 6. [Save] 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 です。
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的に入力されます。
7. [Save] 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

### ステップ

- [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](#)。

次の詳細を入力して、キー タイプの認証情報を作成します。

- 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。を推奨aws-0123456789012-usernameします。ここで、
  - 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。シークレットアクセスキーを base-64 でエンコードするには、次のコマンドを使用します。

```
xargs echo -n | base64 # just press enter, do not enter arguments on the command line
MySecretAccessKey
Ctrl-D
```

コマンドは、シークレットアクセスキーを標準入力から読み取り、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。

認証情報が 1 つある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。

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するには、最小特権の原則に従います。

## 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

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するには、次のステップに従います。

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 – 作成したサービスキーの名前を入力します。詳細については、「[the section called “ステップ 2: サービスキーを作成する”](#)」を参照してください。
  - コミュニケーションの配置 – 作成したコミュニケーションの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. トランスポートの選択 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. [Save] を選択します。

# AWS SDK for SAP ABAP のトラブルシューティング

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 認証](#)
- [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 バケットの `io_createbucketconfiguration` パラメータに 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 = 'test-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. [Save] (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> が DEFAULT のシナリオで、プロファイル <profile\_name> の下に構成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. [Save] (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
    - AWS で実行されている SAP システムの場合は、[メタデータを介したインスタンスロール] を選択します。
    - 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 認証

問題 — <iam\_role\_arn> ロール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、HTTPS ポート 443 で直接またはプロキシサーバー経由で 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 モジュールには依存関係があるため、インストールした core モジュールやその他すべてのモジュールには、それらのモジュールをもう使用しなくてもパッチを適用する必要があります。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。

## 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 認証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証を持ちます。SDK をアンインストールする前に、PFCG ロールから SDK 認証リファレンスを削除してください。

### Note

AWS SDK for SAP ABAP - BTP エディションは、デベロッパープレビュー中にアンインストールできません。

## 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？

回答: AWS SDK for SAP ABAP では、お客様独自の ABAP コードで AWS のサービスを使用できます。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
<a href="#">新しいコンテンツ</a>	SDK for SAP ABAP - BTP エディションのデベロッパープレビュー。	2024 年 5 月 31 日
<a href="#">新しいコンテンツ</a>	「 <a href="#">IAM Roles Anywhere での証明書の使用</a> 」が追加されました。	2023 年 12 月 1 日
<a href="#">新しいコンテンツ</a>	「 <a href="#">SDK による製品の構築</a> 」が追加されました。	2023 年 12 月 1 日
<a href="#">新しいコンテンツ</a>	「 <a href="#">再試行動作</a> 」が追加されました。	2023 年 12 月 1 日
<a href="#">新しいコンテンツ</a>	「 <a href="#">SAP ライセンス</a> 」が追加されました。	2023 年 9 月 22 日
<a href="#">パブリックリリース</a>	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
<a href="#">新しいコンテンツ</a>	<a href="#">AWS SDK for SAP ABAP 機能</a> を追加しました。	2023 年 5 月 30 日
<a href="#">新しいコンテンツ</a>	「 <a href="#">AWS SDK for SAP ABAP のトラブルシューティング</a> 」を追加しました。	2023 年 2 月 17 日
<a href="#">開発者プレビュー</a>	デベロッパーガイドの AWS SDK for SAP ABAP デベロッパープレビュー。	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。