



Developer Guide

AWS SDK for SAP ABAP



AWS SDK for SAP ABAP: Developer Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS SDK for SAP ABAP?	1
Features of AWS SDK for SAP ABAP	1
Maintenance	1
API reference	2
Pricing	2
Resources	2
Getting started	3
Sign up for an AWS account	3
Step 1: Prepare your AWS account	3
IAM role for SAP users	4
Authentication	5
Step 2: Install the SDK	6
Step 3: Configure the SDK	6
Step 4: Functional setup	8
Step 5: Authorize SAP Users	10
Step 6: Write the code	12
Step 7: Run the application	15
Setting up	17
SAP prerequisites	17
SDK for SAP ABAP	17
SDK for SAP ABAP - BTP edition	21
Installing AWS SDK for SAP ABAP	22
Download the SDK	22
Verify the file	22
AWS SDK Transports	23
Installing SDK - BTP edition	26
Install SDK for SAP ABAP - BTP edition	27
Modules	27
Patching SDK for SAP ABAP - BTP edition	28
Configuring	29
Global settings	30
Technical settings	31
Configure scenarios	31
Application configuration	32

SDK profile	32
Logical resource resolver	33
Example	34
Runtime settings	34
Log and trace	35
OPT-IN: enhanced telemetry	35
Active scenario	35
Advanced connectivity scenarios	35
Connection through a proxy server	36
Connection through a packet inspecting firewall	37
Gateway endpoints	37
Custom interface endpoints	37
Advanced routing	38
Per-service proxy server override	39
Accessing endpoints in multiple Regions	40
Service provider settings	41
Refresh, trace, and telemetry	42
SAP system refresh	42
Trace	43
Telemetry	44
Using the SDK	45
Data representation	45
Data types	46
AWS data types	47
Example program	48
Prerequisites	49
Code	49
Code sections	50
Concepts	52
API classes	53
Additional objects	53
Structure classes	53
Arrays	55
Maps	56
Higher level functions	57
Features	1

Programmatic configuration	57
Waiters	58
Paginators	59
Retry behavior	60
Presigners	61
Cross-account IAM role chaining	62
Building products	62
Setting a product ID	62
Customize HTTP requests to AWS	63
Implement an enhancement	63
Filter the enhancement	63
Code the enhancement	63
Limitations	64
Code examples	65
ACM	67
Actions	67
API Gateway	75
Actions	67
Application Recovery Controller	83
Actions	67
Aurora	88
Actions	67
Auto Scaling	93
Actions	67
Amazon Bedrock Runtime	103
Anthropic Claude	103
Stable Diffusion	106
Amazon Bedrock Agents Runtime	109
Actions	67
CloudFront	110
Actions	67
CloudWatch	113
Actions	67
Scenarios	121
CloudWatch Logs	123
Actions	67

Amazon Cognito Identity Provider	125
Actions	67
Amazon Comprehend	131
Actions	67
AWS Config	144
Actions	67
AWS Control Tower	146
Actions	67
Firehose	158
Actions	67
DynamoDB	160
Basics	160
Actions	67
Amazon EC2	174
Actions	67
Amazon ECR	196
Actions	67
Amazon EMR	203
Actions	67
EventBridge Scheduler	208
Actions	67
AWS Glue	213
Actions	67
HealthImaging	226
Actions	67
HealthLake	243
Actions	67
IAM	256
Actions	67
AWS IoT	281
Actions	67
AWS IoT data	292
Actions	67
AWS IoT SiteWise	294
Get started	294
Actions	67

Amazon Keyspaces	302
Actions	67
Kinesis	310
Basics	160
Actions	67
AWS KMS	320
Get started	294
Actions	67
Lambda	340
Basics	160
Actions	67
Organizations	354
Actions	67
Amazon Pinpoint	359
Actions	67
Amazon Pinpoint SMS and Voice API	364
Actions	67
Amazon Polly	373
Actions	67
Amazon RDS	382
Actions	67
Amazon Redshift	387
Actions	67
Amazon Rekognition	394
Actions	67
Amazon S3	409
Basics	160
Actions	67
Scenarios	121
Amazon S3 Control	434
Actions	67
SageMaker AI	442
Actions	67
Scenarios	121
Secrets Manager	459
Actions	67

Amazon SES	462
Actions	67
Amazon SES API v2	478
Actions	67
Amazon SNS	488
Actions	67
Scenarios	121
Amazon SQS	499
Actions	67
Scenarios	121
Step Functions	509
Actions	67
Systems Manager	517
Actions	67
Amazon Textract	526
Actions	67
Scenarios	121
Amazon Transcribe	537
Actions	67
Amazon Translate	545
Actions	67
Scenarios	121
MCP Server	553
Benefits	553
Setting up the MCP server	554
What the MCP server can do	555
Important considerations	555
Security	557
System authentication	558
Metadata authentication	558
Secret access key authentication	559
Certificate-based authentication using IAM Roles Anywhere	559
Source profile authentication for cross-account access	560
Next step	560
Best practices for IAM Security	560
Best practice for Amazon EC2 instance profile	561

IAM roles for SAP users	562
Source Profile Security Considerations	565
SAP authorizations	566
Authorizations for configuration	566
SAP authorizations for end users	567
Secure operations	568
Encryption Of Data At Rest	568
Encryption Of Data In Transit	568
API Usage	2
Secret Key (SSF)	569
Prerequisites	569
Procedure	569
IAM Roles Anywhere	572
Prerequisites	572
Procedure	572
Source Profile	62
Prerequisites	576
Procedure	576
Credential Store	21
Configuration steps	579
Using SAP Credential Store with the SDK	580
Troubleshoot	585
Import failure	585
Unspecified location constraint	585
SSL error	586
Profile configuration	587
IAM authorization	588
Authorization for actions	588
Active scenario	35
Special characters	589
Connectivity	589
Additional topics	590
Releases	590
Release strategy	590
Best practices	560
Patching SDK for SAP ABAP	591

Installing an additional module	592
Uninstalling SDK for SAP ABAP	592
SAP licensing	593
Document history	594

What is AWS SDK for SAP ABAP?

AWS SDK for SAP ABAP provides an interface to the services offered by AWS in the ABAP language. Using the SDK, you can implement ABAP BADIs, reports, transactions, OData services, and other ABAP artifacts on AWS services, such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Translate, and more. You can also develop for ABAP-based systems, starting from SAP NetWeaver 7.4 and in an SAP Business Technology Platform environment. For more information, see [Installing AWS SDK for SAP ABAP - BTP edition](#).

Topics

- [Features of AWS SDK for SAP ABAP](#)
- [Maintenance and support for SDK major versions](#)
- [API reference](#)
- [Pricing](#)
- [Additional resources](#)

Features of AWS SDK for SAP ABAP

AWS SDK for SAP ABAP has been designed to feel familiar and natural to SAP developers. For example, while all AWS services use the `true` and `false` strings to represent Boolean data in XML and JSON structures, SDK for SAP ABAP converts these to ABAP-native `'X'` and `' '` single-character values. SDK for SAP ABAP uses native ABAP constructs as much as possible, including in data types and timestamp formats. As a result, the ABAP programmer does not need to be concerned about the underlying JSON and XML serialization or with the wire format of the API protocol.

Maintenance and support for SDK major versions

For information about maintenance and support for SDK major versions and their underlying dependencies, see the following in the [AWS SDKs and Tools Reference Guide](#):

- [AWS SDKs and tools maintenance policy](#)
- [AWS SDKs and tools and version support matrix](#)

API reference

To see a complete list of AWS SDK for SAP ABAP APIs, see [AWS SDK for SAP ABAP - API Reference Guide](#).

To see a complete module list of AWS SDK for SAP ABAP TLAs, see [AWS SDK for SAP ABAP - Module List](#).

To see a complete module list of SDK for SAP ABAP - BTP edition developer preview TLAs, see [AWS SDK for SAP ABAP - BTP edition - Module List](#).

Pricing

AWS SDK for SAP ABAP is available to you at no additional cost. You only pay for AWS resources and services that you consume with the SDK.

Additional resources

In addition to this guide, the following online resources are available for SDK for SAP ABAP.

- [AWS SDK for SAP ABAP Knowledge MCP Server](#) – Connect your AI-enabled IDE to generate accurate ABAP code for AWS integrations.
- [AWS SDK Code Example Library](#)
- [SAP on AWS documentation](#)
- [AWS developer blog](#)
- [AWS developer forums](#)
- [@awsdevelopers](#)(Twitter)

Getting started with AWS SDK for SAP ABAP

This section describes how to get started with the SDK. It includes information about installing the SDK, performing basic configuration, and creating a Hello World code example that translates a phrase from one language to another. If you are new to AWS SDK, we recommend performing these steps in a sandbox environment.

Steps

- [Sign up for an AWS account](#)
- [Step 1: Prepare your AWS account](#)
- [Step 2: Install the SDK](#)
- [Step 3: Configure the SDK](#)
- [Step 4: Functional setup](#)
- [Step 5: Authorize SAP Users](#)
- [Step 6: Write the code](#)
- [Step 7: Run the application](#)

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see [Getting started with an AWS account](#) in the *AWS Account Management Reference Guide*.

Step 1: Prepare your AWS account

To get started with SDK for SAP ABAP, you must have an active AWS account . You need an AWS account even if your SAP system is hosted on-premises, on SAP Business Technology Platform (BTP) or with another cloud provider.

If your SAP system is running on AWS Cloud, then you will be making calls to AWS services in your AWS account.

Topics

- [IAM role for SAP users](#)
- [Authentication](#)

IAM role for SAP users

- Create an IAM role with the instructions provided in the *AWS Identity and Access Management User Guide*. For more information, see [Creating a role to delegate permissions to an AWS service](#). Note the Amazon Resource Name (ARN) of the IAM role for later use.
- Select Amazon EC2 as the use case.
- Use SapDemoTranslate as the name of the role.
- Attach TranslateReadOnly profile to the role.
- The role must have the following entities to enable the SAP system to assume the role. Replace **"111122223333"** with your AWS account number.

JSON

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

This example shows that any principal from the AWS account **"111122223333"** can assume the role. This is a broad permission that is suitable for proof-of-concept. You can use a narrower principal for production, such as the following examples.

- A specific user – when the SAP system is using either one of the following:
 - SSF-encrypted credentials from an on-premises SAP system
 - Credentials from SAP Credential Store service on SAP BTP, ABAP environment

- A specific role – when the SAP system is on Amazon EC2 and there is an instance profile.
- Amazon EC2 – when the SAP system is on Amazon EC2 and there is *no* instance profile.

For more information, see [Best practices for IAM Security](#).

Authentication

Authentication depends on where your SAP system is hosted.

Locations

- [On AWS Cloud](#)
- [On-premises, SAP BTP or other cloud](#)

On AWS Cloud

Ensure that the EC2 instance on which your SAP system is running has an instance profile with the following permissions.

JSON

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

Add the ARN that you noted in the previous step.

This permission enables your SAP system to assume the SapDemoTranslate role on behalf of the ABAP user.

On-premises, SAP BTP or other cloud

If your SAP system is located on-premises, on SAP BTP or on other cloud, use the following steps to establish a connection for authentication using secret access key.

1. Create an IAM user. For more information, see [Creating IAM users \(console\)](#).
2. Use SapDemoSID as the name of the IAM user. SID is the system ID of your SAP system.
3. Assign SapDemoTranslate role to this user.

Retain the `access_key` and `secret_access_key`. You must configure these credentials in your SAP system.

Note

If your SAP system is located on-premises, on SAP BTP or on other cloud, you can authenticate using one of the following options.

- [Secret access key authentication](#) using SSF or SAP Credential Store
- [Using certificates with IAM Roles Anywhere](#)

Step 2: Install the SDK

See the following tabs for installation instructions.

SDK for SAP ABAP

Import SDK for SAP ABAP transports in your SAP system. You can import the transports into any client. For more information, see [Installing SDK for SAP ABAP](#).

SDK for SAP ABAP - BTP edition

Install SDK for SAP ABAP - BTP edition using the **Deploy Product** application. For more information, see [Installing SDK for SAP ABAP - BTP edition](#).

Step 3: Configure the SDK

Before configuring the SDK, ensure that you have the required authorizations. For more information, see [SAP authorizations](#).

See the following tabs for configuration instructions.

SDK for SAP ABAP

Run the `/AWS1/IMG` transaction to open the Implementation Guide for SDK for SAP ABAP. To run this transaction, enter `/n/AWS1/IMG` in the command bar of your SAP system, and then choose **Enter**.

Complete the following configurations.

- Go to **Technical Prerequisites**.
 - Review the recommended [Parameters](#) and [HTTPS connectivity](#).
- Go to **Global Settings** → **Configure Scenarios**.
 - Change the settings, according to the recommendations in [Global settings](#).
- Go to **Global Settings** → **Technical Settings**.
 - Change the settings, according to the recommendations in [Global settings](#).
- Go to **Runtime Settings** → **Log And Trace**.
 - Select **New Entries**.
 - **Trace level:** No Trace.
 - **Maximum Dump Lines:** 100.
 - **OPT-IN: enh telemetry:** Keep this blank.
 - Select **Save**.
- Go to **Runtime Settings** → **Active Scenario**.
 - Under **New Scenario**, select DEFAULT.
 - Select **Commit Scenario Change**.
 - Accept the prompt.

Prerequisites for On-Premises Systems

If your SAP system is running on-premises or in another cloud, then the credentials must be stored in your SAP database. The credentials are encrypted using SAP SSF and require a configured cryptographic library, such as SAP's CommonCryptoLib.

The steps for configuring SSF for SDK for SAP ABAP are described in the `/AWS1/IMG` transaction.

Note

The preceding prerequisite does not apply if your SAP system is running on Amazon EC2. SAP systems running on Amazon EC2 retrieve short-lived, automatically rotating credentials from the Amazon EC2 instance metadata.

SDK for SAP ABAP - BTP edition

Open your ABAP environment in a web browser, and navigate to the Custom Business Configurations application.

Complete the following configurations.

- Go to **Configure Scenarios**.
 - Change the settings, according to the recommendations in [Global settings](#).
- Go to **Technical Settings**.
 - Change the settings, according to the recommendations in [Global settings](#).

Step 4: Functional setup

See the following tabs for setup instructions.

SDK for SAP ABAP

Run transaction `/AWS1/IMG` (enter `/n/AWS1/IMG` in the command bar, and choose **Enter**) to open the implementation guide for AWS SDK.

- Go to **Application Configuration** → **SDK Profile**.
 - Select **New Entries**.
 - **Profile**: DEMO.
 - **Description**: Demo profile.
 - Select **Save**.
- Highlight the entry that you created and click on the **Authentication And Settings** tree branch.
 - Select **New Entries**.
 - **SID**: The system ID of the SAP system that you are currently in.

- **Client:** The client of the SAP system that you are currently in.
- **Scenario ID:** The dropdown list where you'll find the DEFAULT scenario created by your Basis administrator.
- **AWS Region:** enter the AWS Region that you want to make calls to. If your SAP system is running in AWS, enter the AWS Region that it is running in.
- Authentication Method:
 - Select **Instance Role via Metadata** if your SAP system is running on Amazon EC2.
 - Select **Credentials from SSF Storage** if your SAP system is running on-premises or in another cloud.
 - Select **Set Credentials**.
 - Enter the Access Key ID and Secret Access Key that you created in the previous step.
- Keep **Disable IAM roles** blank.
- Select **Save**.
- Click on the **IAM Role Mapping** tree branch.
 - Select **New Entries**.
 - Enter **Sequence number:** 010.
 - Enter **Logical IAM role:** TESTUSER.
 - Enter **IAM Role ARN:** enter the arn:aws: of the IAM role containing the TranslateReadOnly policy created in the previous step.

SDK for SAP ABAP - BTP edition

Set up authentication using SAP Credential Store. For more information, see [Using SAP Credential Store](#).

Open your ABAP environment in a web browser, and navigate to the Custom Business Configurations application.

- Go to **SDK Profile**.
 - Select **Edit** to create a new profile.
 - **Profile:** DEMO.
 - **Description:** Demo profile.
- Select the right arrow key next to the created entry to navigate to **Authentication and Settings** tab.

Select **New Entries**.

- **SID**: The system ID of the SAP system that you are currently in.
- **Client**: The client of the SAP system that you are currently in.
- **Scenario ID**: The dropdown list where you'll find the DEFAULT scenario created by your Basis administrator.
- **AWS Region**: enter the AWS Region that you want to make calls to. If your SAP system is running in AWS, enter the AWS Region that it is running in.
- Authentication Method: Select **Credentials from SAP Credential Store**.
- Enter the **Namespace** and **Key name** of the credentials stored in SAP Credentials Store.
- Enter the name of the **Communication Arrangement** created to establish communication between SDK for SAP ABAP - BTP edition and SAP Credential Store.
- Keep **Disable IAM roles** blank.
- Right-click on the right arrow key next to the created entry to navigate to **IAM Role Mapping** tab.

Select **New Entries**.

- Enter **Sequence number**: 010.
- Enter **Logical IAM role**: TESTUSER.
- Enter **IAM Role ARN**: enter the arn:aws: of the IAM role containing the `TranslateReadOnly` policy created in the previous step.

Step 5: Authorize SAP Users

SAP users are not authorized to use AWS functionality by default. The users must be explicitly authorized using SAP authorizations. See the following tabs for more details.

SDK for SAP ABAP

Create a PFCG role

- Go to transaction PFCG
- Enter the role name `ZAWS_SDK_DEMO_TESTUSER` and select **Create Single Role**.
 - **Description**: Role for demo AWS SDK functionality.
 - Go to the **Authorizations** tab.

- Select **Change Authorization Data** and accept the informational pop-up.
- At the *Choose Template* pop-up, select **Do not select templates**.
- Select **Add Manually** from the toolbar.
- Add the following authorization objects:
 - /AWS1/LROL
 - /AWS1/SESS
- In the authorization tree, enter:
 - Profile for accessing AWS APIs: DEMO
 - Logical IAM Role: TESTUSER
- Select **Save**.
- Select **Generate**.
- Select **Back**.
- Select **Save** to save the role.

Assign the PFCG role to SAP users

Any user who has the ZAWS_SDK_DEMO_TESTUSER role assigned will be authorized to use AWS SDK functions with the settings configured in DEMO SDK profile. The authorized user will also assume the IAM role mapped to the TESTUSER logical IAM role in that profile.

- Run transaction SU01.
 - Enter the user ID of an SAP user who will be testing AWS SDK functionality.
 - Select **Change**.
 - Go to the **Roles** tab and assign ZAWS_SDK_DEMO_TESTUSER role to the user.
 - Select **Save**.

SDK for SAP ABAP - BTP edition

Create a Business role

- Open your ABAP environment in a web browser, and navigate to the **Maintain Business Roles** application.
- **Select Create from Template**, and enter the following details.

- **Template** – Choose **/AWS1/RT_BTP_ENDUSER**.
 - **New Business Role ID** – Enter an ID.
 - **New Business Role Description** – Enter a description.
 - Select **OK** to see the page for the business role.
 - Under **General Role Details** tab, go to **Access Categories**, and set the **Write, Read, Value Help** field as **Restricted**.
 - Select **Maintain Restrictions**, and expand **Assigned Restriction Types** from the left navigation pane. Update the following field in the **Restrictions and Values** section.
 - Under **Choose SDK Session**, select the pencil icon next to **SDK Profile**, and navigate to the **Ranges** tab. Enter **DEMO**, and select **Add**.
 - Under **Choose Logical IAM Role**, select the pencil icon next to **Logical IAM Role**, and navigate to the **Ranges** tab. Enter **TESTUSER**, and select **Add**.
- Select the pencil icon next to **SDK Profile**, and navigate to the **Ranges** tab. Enter **DEMO**, and select **Add**
- Navigate back to the Business Role template, and open the **Business Users** tab. Select **Add** to assign the newly created Business Role to an SAP business user who will test the SDK functionality. Select **Save**.

Any business user assigned to the created Business Role will be authorized to use AWS SDK functions with the settings configured in DEMO SDK profile. The authorized user will also assume the IAM role mapped to the TESTUSER logical IAM role in that profile.

Step 6: Write the code

See the following tabs for more details.

SDK for SAP ABAP

1. Open transaction SE38.
 - Enter ZDEMO_TRANSLATE_HELLO_WORLD as the program name.
 - Select **Create**.
 - Enter **AWS SDK Hello World In Any Language** as the title.
 - Type: choose **Executable Program**.

- Status: choose **Test Program**.
- Select **Save**.
- Save the program as a **Local Object**.

Add the following code.

```

*&-----*
*& 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. Right-click on the package where the ABAP class needs to be created, then select **New > ABAP class**.
2. Enter **ZCL_DEMO_XL8_SIMPLE** for Class name, and add a Class description. Select **Next**.
3. Create or choose a transport request. Select **Finish**.

Add the following code.

```
CLASS zcl_demo_xl8_simple DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

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

CLASS zcl_demo_xl8_simple IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

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

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

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

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

For details on how to write ABAP code that uses the SDK, see [Using AWS SDK for SAP ABAP](#).

Step 7: Run the application

See the following tabs for more details.

SDK for SAP ABAP

Run the application in SE38. If successful, the following will be your output.

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

If you are missing authorizations, configuration, or Basis prerequisites, you might get an error message. See the following example.

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

If your SAP role authorizes you to use an SDK profile and map it to a logical IAM role while your IAM permissions are not configured for the SAP system to assume the IAM role, the following will be your output.

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

In this case, review your IAM permissions and trust configuration on the IAM roles, users, or both defined in [the section called “Step 1: Prepare your AWS account”](#).

SDK for SAP ABAP - BTP edition

Run the application on **Eclipse > Run As > ABAP Application (Console)**. If successful, the following will be your output.

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

If you are missing authorizations, configuration, or Basis prerequisites, you might get an error message. See the following example.

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

If your SAP role authorizes you to use an SDK profile and map it to a logical IAM role while your IAM permissions are not configured for the SAP system to assume the IAM role, the following will be your output.

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

In this case, review your IAM permissions and trust configuration on the IAM roles, users, or both defined in [the section called “Step 1: Prepare your AWS account”](#).

Setting up

This section provides information about how to set up your development environment to use AWS SDK for SAP ABAP.

Topics

- [SAP prerequisites](#)
- [Installing AWS SDK for SAP ABAP](#)
- [Installing AWS SDK for SAP ABAP - BTP edition](#)

SAP prerequisites

The following prerequisites for installing the SDK are applicable when your SAP systems are hosted on AWS.

Topics

- [Prerequisites for AWS SDK for SAP ABAP](#)
- [Prerequisites for AWS SDK for SAP ABAP - BTP edition](#)

Prerequisites for AWS SDK for SAP ABAP

The following are the prerequisites for AWS SDK for SAP ABAP.

Topics


- [Basis release](#)
- [Kernel release](#)
- [Parameters](#)
- [Notes](#)
- [Outbound connectivity](#)
- [HTTPS connectivity](#)
- [Access to Amazon EC2 instance metadata](#)

Basis release

SDK for SAP ABAP is compatible with SAP NetWeaver 7.4 and higher. SDK for SAP ABAP doesn't touch any SAP application tables. It is completely agnostic about the applications, such as SAP Enterprise Resource Planning and SAP Landscape Transformation Replication Server.

The minimum supported SP-Level for SAP_BASIS 740 is SP 0008. For more information, see [SAP Note 1856171 - Supporting form fields of the same name in CL_HTTP_ENTITY](#) (requires SAP portal access). Based on your business requirements, you can choose a higher SP-Level, as shown in the following image.

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
PL_BASIS	740	0006	SAPK-74006INSAPBASIS	Basis Plus for

There is no minimum SP-Level requirement for SAP_BASIS 750 and higher releases.

Kernel release

SDK for SAP ABAP and tools that use the Internet Communication Manager (ICM) for HTTP connectivity, rely on the SAP kernel for its cryptographic, HTTP, XML, and JSON capabilities. We recommend using the latest kernel release that is compatible with your SAP NetWeaver platform. The minimum requirement is kernel release **741**. For more information, see [SAP Note 2083594 - SAP Kernel Versions and SAP Kernel Patch Levels](#) (requires SAP portal access).

If you are using kernel release 741 or 742, the following patch levels are required:

- 741 patchno 212
- 742 patchno 111

Parameters

Your SAP system must support Server Name Indication (SNI) as described in the following SAP Notes (requires SAP portal access).

- [SAP Note 2124480 - ICM/Web Dispatcher: TLS Extension Server Name Indication \(SNI\) as client](#)
- [SAP Note 2582368 - SapSSL update for client-side sending of TLS extension SNI by saphttp, sapkprotp, sldreg](#)

Configure the following parameter in the DEFAULT.PFL file.

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

Notes

Apply the following SAP Note to your system.

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

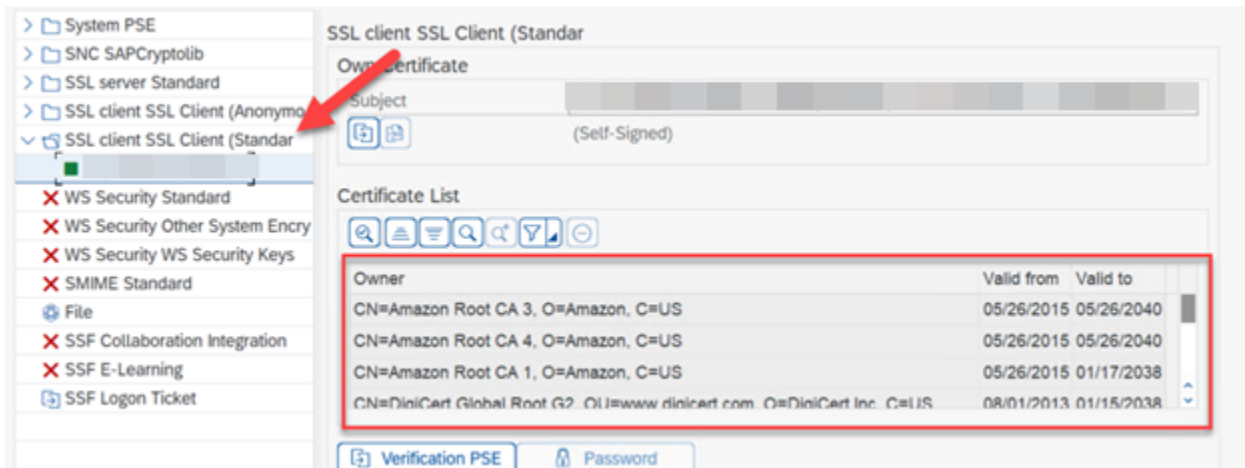
Outbound connectivity

SDK for SAP ABAP is an HTTPS client. The SAP system sends HTTPS messages outbound. There is no requirement of an inbound connectivity.

HTTPS connectivity

All AWS API calls are made with encrypted HTTPS channels. The SAP system must be set up to trust AWS certificates to establish an outbound HTTPS connection.

1. Go to <https://www.amazontrust.com/repository/>.
2. Under **Root CAs**, download all the certificates using the *PEM* link.
3. Import these certificates in STRUST of your SSL Client (Standard) PSE on each of your SAP systems, as shown in the following image.



Access to Amazon EC2 instance metadata

ABAP system makes unencrypted HTTP connections to localhost (<http://169.254.169.254>) to enable Amazon EC2 instance metadata. The HTTP channel is used only to retrieve AWS credentials from the local server. The HTTP traffic stays within the host.

The metadata allows an SAP system in AWS to securely authenticate itself without storing a secret key in the SAP Secure Store. This feature is applicable only to SAP systems hosted on Amazon EC2.

Configure the DEFAULT.PFL file with the following parameter to enable your SAP system to make an unencrypted outbound HTTP connection.

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

Use the following parameter to enable the outbound HTTP connection without enabling the inbound connection.

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

Verify that your SAP system is configured for outbound HTTP connections with the following steps:

1. Run **SMICM** transaction.
2. Go to **Active Services**.
3. Verify that you see a **green check mark** in the HTTP row, under *Active* column, as shown in the following image.

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

Prerequisites for AWS SDK for SAP ABAP - BTP edition

The following are the only prerequisites for AWS SDK for SAP ABAP - BTP edition.

Topics

- [SAP Landscape Portal – BTP edition](#)
- [SAP Credential Store – BTP edition](#)

SAP Landscape Portal – BTP edition

This prerequisite is only applicable for AWS SDK for SAP ABAP - BTP edition.

SAP Landscape Portal is the only supported mechanism to install add-ons in an SAP BTP environment. Ensure that you are subscribed to use this service. For more information, see [Landscape Portal](#).

SAP Credential Store – BTP edition

This prerequisite is only applicable for AWS SDK for SAP ABAP - BTP edition.

In the developer preview, secret access key authentication is the only supported mechanism for authenticating AWS SDK for SAP ABAP - BTP edition. The SDK reads the credentials from the Credential Store, and stores the secret access key securely.

You must meet the following prerequisites.

- Subscription to Credential Store.
- Credential Store assigned as an entitlement to your BTP sub-account. See [Initial setup](#) for more details.
- A service instance with standard plan for Credential Store. See [Create a service instance](#) for more details.

For more information, see [Using SAP Credential Store](#).

The SAP Credential Store service runs in SAP BTP outside of the ABAP BTP system. See [SAP Credential Store](#) for more details.

Installing AWS SDK for SAP ABAP

Topics

- [Download SDK for SAP ABAP](#)
- [Verify SDK for SAP ABAP file – optional](#)
- [AWS SDK Transports](#)

Download SDK for SAP ABAP

Download the SDK from <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip>.

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

When the download is complete, we recommend that you unzip the downloaded file into a directory, such as `/tmp/awssdk`.

Verify SDK for SAP ABAP file – *optional*

This optional step of validating the signature of your SDK file helps you confirm that your SDK has not been tampered with. Use the following steps to verify your SDK file.

1. Download the SDK SIGNATURE file with the following command.

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

2. Copy the following public key, and save it to a file named `abapsdk-signing-key.pem`.

```
-----BEGIN PUBLIC KEY-----  
MIICIjANBgkqhkiG9w0BAQEFAAACg8AMIICCGKCAgEAmS3oN3wKBh4HJ0Ga0tye  
15RR5909nuw0Jx0vEDCT709wUrxS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv  
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
```

```
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuiscJ21MPkz0
cHOR31MNsMLzm+d/gVklT31R/JwAcFCkXTwvR8/V0WNGZZXdVUbefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLj pz8dH
Zg25o79U9EJ1n+1pqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIURLPKLVQUw
otVRzNDgLOA7eA667NimegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. Verify the downloaded SDK ZIP file with the following command. The command requires `openssl` that is part of many Linux distributions.

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

4. Verify that the output of the preceding command is `Verified OK`.
5. If the output is `Verification Failure`, repeat the preceding steps. If you continue to receive a failed output, don't install the SDK and contact Support.

AWS SDK Transports

Topics

- [Contents](#)
- [Importing](#)
- [Namespace](#)

Contents

The installation of SDK for SAP ABAP is completed through ABAP Transports. You must import these transports into your development or sandbox environment.

Each SDK for SAP ABAP release completely replaces the previous one. There is no need to apply incremental transports. The transports are bundled in a ZIP file. The following is the structure of the ZIP file.

```
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
.
.
.
```

The `transports` folder contains a `core` subfolder. The `core` subfolder contains the core runtime transports and a subfolder for each module, named by the three letter abbreviation of the module. For a complete module list of the TLAs, see [AWS SDK for SAP ABAP - Module List](#).

AWS SDK transports are workbench requests. Depending on the configuration of your TMS routes, the SDK may not automatically forward to your quality assurance and production queues after importing into the previous system. You must manually add them to the queue of each system.

When your project is ready for the next phase, AWS SDK can be imported along with separate transports containing your own Z code with business functionality. If you are using a change control system, such as SAP Change Request Management (ChaRM), consult your ChaRM administrator for correct handling of third-party transports.

Importing

Topics

- [Key pointers](#)
- [Time to import](#)

AWS SDK transports are client-independent. The core transport is mandatory and contains the SDK runtime code, the API for AWS Security Token Service, and the API for Amazon Simple Storage Service. The remaining SDK modules are each delivered in a separate transport. To keep the size of the SDK small in your system, each SDK module is optional. You can install additional modules later, if required for your business logic.

For example, if you want to use the APIs for Amazon S3 and Amazon Translate, import the core transport (containing core runtime, Amazon S3, and AWS STS modules) and the x18 transport (containing the module for Amazon Translate) transports.

To see a complete list of SDK for SAP ABAP APIs, see [SDK for SAP ABAP - API Reference Guide](#).

The following are key pointers when importing AWS SDK transports.

- Each transport is delivered as Knnnnnn .AWS and Rnnnnnn .AWS
 - Knnnnnn .AWS must be copied to /usr/sap/trans/cofiles
 - Rnnnnnn .AWS must be copied to /usr/sap/trans/data.
- When importing transports, you must select the **Ignore Invalid Component Version** found at *Import Transport Request > Options > Import Options*.
- All desired transports can be imported simultaneously.
- If importing the transports separately, the core transport must be imported first.
- The release level of all the transports must be identical.

Time to import

AWS SDK transports may take many minutes to import. The transports are successful if STMS shows a green (RC=0) or yellow (RC=4) light.

- A red light (RC=8) indicates that the import had a syntax error.
 - Select **Request** → **Display** → **Logs to examine the import error**.
 - During the import, if an error is thrown because of a missing interface IF_SYSTEM_UUID_RFC4122_STATIC, then ensure that SAP Note 2619546 is applied to the system. For more information, see [Notes](#).
 - If the cause of the error is unknown, contact Support.
- A red lightning bolt (RC=12) indicates that the transport files have not been correctly loaded into /usr/sap/trans or do not have the necessary permissions.

Key pointers

The following are key pointers when importing AWS SDK transports.

- Each transport is delivered as Knnnnnn .AWS and Rnnnnnn .AWS
 - Knnnnnn .AWS must be copied to /usr/sap/trans/cofiles
 - Rnnnnnn .AWS must be copied to /usr/sap/trans/data.

- When importing transports, you must select the **Ignore Invalid Component Version** found at *Import Transport Request > Options > Import Options*.
- All desired transports can be imported simultaneously.
- If importing the transports separately, the core transport must be imported first.
- The release level of all the transports must be identical.

Time to import

AWS SDK transports may take many minutes to import. The transports are successful if STMS shows a green (RC=0) or yellow (RC=4) light.

- A red light (RC=8) indicates the import had a syntax error.
 - Select **Request** → **Display** → **Logs to examine the import error**.
 - During the import, if an error is thrown due to a missing interface `IF_SYSTEM_UUID_RFC4122_STATIC`, then ensure that SAP Note 2619546 is applied to the system. For more information, see [Notes](#).
 - If the cause of the error is unknown, contact Support.
- A red lightning bolt (RC=12) indicates the transport files have not been correctly loaded into `/usr/sap/trans` or do not have the necessary permissions.

Namespace

SDK for SAP ABAP uses the `/AWS1/` namespace and does not modify SAP objects or any other objects in your system with the following exception.

- AWS auth objects are in an **Auth Object Class**. Auth Object Classes are limited to four characters and do not support namespaces. SDK for SAP ABAP uses Auth Object Class is YAW1. If you already have an auth object class YAW1 in transaction SU21, contact Support before installation.

Installing AWS SDK for SAP ABAP - BTP edition

The BTP edition is in developer preview, and can be installed by joining the preview. To install the SDK, fill the participation form at [AWS SDK for SAP ABAP - BTP edition developer preview](#).

Before installing SDK for SAP ABAP - BTP edition, ensure that you are meeting the required prerequisites. For more information, see [SAP Landscape Portal](#) and [SAP Credential Store](#).

Topics

- [Install SDK for SAP ABAP - BTP edition](#)
- [Modules](#)
- [Patching SDK for SAP ABAP - BTP edition](#)

Install SDK for SAP ABAP - BTP edition

1. Go to your SAP Landscape Portal instance, and launch the **Deploy Product** fiori application.
2. In **Products**, select **/AWS1/SDK_OMNI** under **Partner Products**.

Contact Support if you do not see /AWS1/SDK_OMNI after being accepted in the developer preview.

3. In **Target Version**, choose the version of SDK for SAP ABAP - BTP edition you want to install on your system.
4. In **Available Systems**, check the checkboxes for all the SIDs on which you want to install the SDK.
5. Select **Deploy**, enter the scheduling details, and select **Schedule**. You can monitor the progress in **Product Version Deployment Status**.

The installation may take 30-45 minutes, and includes system downtime. For more details, see [Deploy Product](#).

Modules

The following modules are included in the developer preview of AWS SDK for SAP ABAP - BTP edition.

- [Amazon API Gateway \[agw\]](#)
- [Amazon Athena \[ath\]](#)
- [Amazon Bedrock Runtime \[bdr\]](#)
- [Amazon Comprehend \[cpd\]](#)
- [Amazon EventBridge \[evb\]](#)
- [Amazon Forecast \[fcs\]](#)
- [Amazon Kinesis \[kns\]](#)

- [Amazon Data Firehose \[frh\]](#)
- [Amazon SageMaker AI \[sgm\]](#)
- [Amazon Simple Notification Service \[sns\]](#)
- [Amazon Simple Queue Service \[sqs\]](#)
- [Amazon Simple Storage Service \[s3\]](#)
- [AWS Systems Manager \[ssm\]](#)
- [Amazon Textract \[tex\]](#)
- [Amazon Transcribe \[tnb\]](#)
- [Amazon Translate \[x18\]](#)
- [AWS CloudTrail \[tr1\]](#)
- [AWS IoT \[iot\]](#)
- [AWS KMS \[kms\]](#)
- [AWS Lambda \[lmd\]](#)
- [AWS Secrets Manager \[smr\]](#)
- [AWS Security Token Service \[sts\]](#)
- [AWS Transfer Family \[trn\]](#)
- [IAM Roles Anywhere \[r1a\]](#)
- [Amazon Redshift Data API \[rsd\]](#)

Patching SDK for SAP ABAP - BTP edition

The patching process for SDK for SAP ABAP - BTP edition is similar to the installation process. If you install the SDK on a system that has an already installed older version, then the SDK is patched to your choice of new version.

Configuring AWS SDK for SAP ABAP

Before using AWS SDK for SAP ABAP, you must configure the SDK with technical and functional settings that are necessary for the SDK operations. Some settings are transportable and some are runtime settings. Many of the settings are directly analogous to the settings defined in .INI files for other SDKs.

The SDK configurations, except for Runtime settings, must be completed in your development environment. You can transport configurations to QA and production following the usual transport and change control rules. Transportable configuration is not recommended for production environments.

If you do not have permissions to configure AWS SDK, see [SAP authorizations](#).

Configuring AWS SDK for SAP ABAP

To run the configuration transaction, enter /n/AWS1/IMG in the SAPGUI command bar.

Configuring AWS SDK for SAP ABAP - BTP edition

Use the following steps to configure SDK for SAP ABAP - BTP edition.

1. Open your ABAP environment in a web browser.
2. Navigate to Custom Business Configurations application.

To create a customizing request using the Export Customizing Transports application, see [Working in the Export Customizing Transports App - Create Request](#).

In the Custom Business Configuration application, you can group configurations based on the type of SDK settings. Use the following steps to group configurations.

1. Open your ABAP environment in a web browser, and navigate to the Custom Business Configurations application.
2. Select **Settings** > **Group**, and choose **Configuration Group** from the drop-down list. Select **OK**.
3. The configurations are now available in a hierarchical structure as displayed in the image. To save the view, see [Views \(Variant Management\) - Components](#).

Custom Business Configurations (4)

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

This section covers the following topics.

Topics

- [Global settings](#)
- [Application configuration](#)
- [Runtime settings](#)
- [Advanced connectivity scenarios](#)
- [Service provider settings](#)
- [Refresh, trace, and telemetry topics for AWS SDK for SAP ABAP](#)

Global settings

Use /n/AWS1/IMG IMG transaction for AWS SDK for SAP ABAP, and Custom Business Configuration application for AWS SDK for SAP ABAP - BTP edition to configure the global settings. This topic uses IMG and Custom Business Configuration interchangeably.

This section covers the following topics.

Topics

- [Technical settings](#)
- [Configure scenarios](#)

Technical settings

The Global settings of /AWS1/IMG transaction affect the behavior of the entire SDK. These settings are generally configured by a Basis administrator. You can set these values to the following recommended settings.

- Select **New Entries**.
 - **S3 regionalization**: Access us-east-1 buckets by using s3.amazonaws.com.
 - **STS regionalization**: Access STS by using global endpoint.
 - **Disable EC2 metadata**: Keep this field blank. This field is read-only in the BTP edition, and is set to 'Yes' by default.
 - **Metadata Endpt Mode**: Use IPv4 metadata endpoint. This field is read-only in the BTP edition, and is auto-updated.
 - **Metadata Endpt URL**: Keep this field blank. This field is read-only in the BTP edition.
- Select **Save**.

Configure scenarios

Scenarios enable AWS SDK to more efficiently switch settings during a multi-Region disaster testing or disaster recovery testing scenario. You may not need this feature, and instead need only to configure the following DEFAULT scenario.

- Select **New Entries**.
 - Scenario ID:DEFAULT
 - Scenario Description: Default Scenario
- Select **Save**.

If you have a multi-Region disaster recovery setup or other unique cases that require a quick change of settings, then you can configure multiple scenarios.

- **DEFAULT** - Standard operation.
- **DR** - Special configuration if a disaster requires moving the entire system to another Region.
- **DR_TEST** - Special configuration for simulating a disaster, for example, in a temporary clone of production.

Application configuration

Configuring SDK for SAP ABAP is similar to configuring other ABAP-based applications. It is organized into different *profiles* to group the settings of various scenarios. An ABAP SDK profile defines the settings required for a specific application scenario. For example, if transactions ZVA01, ZVA02, and ZVA03 are invoice-related transactions enhanced and runs on AWS services, such as Amazon S3, AWS Lambda, and Amazon SageMaker AI, then an SDK profile called ZINVOICE can be made. This profile can group the technical settings, SAP authorizations, and IAM role mappings for the invoice-related functionality.

Use `/n/AWS1/IMG` transaction for AWS SDK for SAP ABAP, and Custom Business Configuration application for AWS SDK for SAP ABAP - BTP edition to configure the global settings. This topic uses IMG and Custom Business Configuration interchangeably.

Topics

- [SDK profile](#)
- [Logical resource resolver](#)
- [Example](#)

SDK profile

An ABAP SDK profile defines the following for each SID and client.

Note

The client is always 100 in SAP BTP, ABAP environment.

- The default AWS Region for all API calls. For example, if your SAP systems are running in the us-east-1 Region, it is likely that your other AWS resources are also in the same Region, and this should be your default Region. Your ABAP code can override the default Region.

- Authentication method
 - For SAP systems running on Amazon EC2, we strongly recommend choosing instance role metadata to benefit from the short-lived, automatically rotating credentials.
 - For SAP systems running on-premises or in another cloud, you must choose credentials from SSF storage.
 - For ABAP systems running on SAP BTP, you must choose credentials from SAP Credential Store. For more information, see [Using SAP Credential Store for authentication](#).
 - For cross-account role chaining scenarios, choose Source Profile and specify a source profile ID. This enables automatic resolution and execution of role assumption chains. For more information, see [Using Source Profile for Cross-Account Access](#).
- A mapping of logical IAM roles to IAM roles.
 - This mapping is sorted in the order of descending priority.
 - An IAM role of highest priority for which a user is authorized in a PFCG role will automatically be selected for the user.
- An optional mapping of services to custom endpoints. This configuration is discussed in [Advanced connectivity scenarios](#)

Note

PFCG roles are called Business Roles in SAP BTP, ABAP environment.

When an ABAP program wants to connect to an AWS service, it will specify an ABAP SDK profile that pulls the necessary settings. An AUTHORIZATION-CHECK will be performed to confirm that the user has permissions to access the SDK profile. Your SAP Security Administrator can define a PFCG role granting you access to the appropriate users.

Logical resource resolver

Logical resource resolver enables you with a standard place to store resource names. It ships with SDK for SAP ABAP. Its action is similar to the way that FILE transaction maps logical file names to physical file names.

A logical resource defines the concept of an AWS resource, such as the Amazon S3 bucket that holds our invoices. This logical resource, for example, can be named ZINVOICES_OUTBOUND

and it can map to a different physical bucket name, depending on whether the SAP system is development, QA, or production.

SDK for SAP ABAP is set up such that a QA system resolves logical resources to the QA physical resources, even after a system refresh from production. The resource mappings for ALL systems is defined in your development SAP system and transported forward. This approach is different from the usual setup in SAP systems where the mapping is handled as master data and set in each system. The advantage of logical resource resolver offered by SDK for SAP ABAP is that the chances of a mistaken transport after system refreshes are almost none.

Example

There are four separate Amazon S3 buckets - one each for development, production, and QA, as well as a second QA bucket for regression testing.

When the SDK resolves a logical resource like ZINVOICE_OUTBOUND to a physical resource, it checks SY-SYD and SY-MANDT to ask *Which SID and client am I running in?*, and automatically selects the correct physical resource.

If the mapping of a resource in production needs to change, you must change the mapping in the IMG of the development system and transport it forward. This ensures that reassigning AWS resources to an SAP system is subject to change control as with any other transport.

Note

As the SDK configuration is client-dependent, reassignment of resources is transported in a customizing request, and the transport must be imported into each client.

Runtime settings

This section covers the following topics.

Note

These settings are not transportable and are local to each SAP system.

Topics

- [Log and trace](#)
- [OPT-IN: enhanced telemetry](#)
- [Active scenario](#)

Log and trace

You can activate a trace for debugging purposes. It is recommended to keep the trace level at **No Trace**, unless diagnosing a technical issue. For more information, see [secure operation](#).

These settings are not applicable to SDK for SAP ABAP - BTP edition.

OPT-IN: enhanced telemetry

All SDKs send telemetry information to AWS for support purposes. You can opt in for enhanced telemetry. This is particularly useful when you contact Support to identify the source of a particular API call. For more information, see [Trace](#) and [Telemetry](#).

These settings are not applicable to SDK for SAP ABAP - BTP edition.

Active scenario

Activate your DEFAULT scenario in this transaction. This activation is required only once for each system and should not be changed unless the system is undergoing a multi-Region disaster recovery. In a multi-Region setup, you can use this setting to switch your SAP system to a disaster recovery environment or disaster recovery test scenarios.

Advanced connectivity scenarios

AWS SDK for SAP ABAP consumes AWS services by making HTTPS calls to AWS endpoints. In general, AWS endpoints are accessible over the internet. An SAP system must be able to reach out to the internet to establish these outbound connections. SDK for SAP ABAP never requires an inbound connection from the internet into the SAP system.

The following scenarios offer different ways to establish the outbound connection.

Scenarios

- [Connection through a proxy server](#)
- [Connection through a packet inspecting firewall](#)

- [Gateway endpoints](#)
- [Custom interface endpoints](#)
- [Advanced routing](#)
- [Per-service proxy server override](#)
- [Accessing endpoints in multiple Regions](#)

Connection through a proxy server

To establish a connection through a proxy server, use the following steps.

1. In the SDK, go to Transaction **SICF**.
2. Choose **Execute**.
3. In the menu, choose **Client > Proxy server**.
4. Set **Proxy setting** as **Active**.
5. In the field for **No Proxy for the Following Addresses**, list any exceptions separated by semicolons.
6. In the **HTTP Protocol** and **HTTPs Protocol** fields, specify the connection details for your proxy server.

The SDK is unaware of the proxy server, and does not require any settings to use the SAP system's proxy server configuration.

Note

If you use [Amazon EC2 instance metadata authentication](#), then the SAP system cannot use the proxy server to access the local instance metadata at `http://169.254.169.254`. You must include `169.254.169.254` in the field for *No Proxy for the Following Addresses*.

Note

You can override the proxy server behavior on a per-service basis in the **Advanced Routing** section of `/AWS1/IMG`. For more information, see [the section called "Per-service proxy server override"](#).

Connection through a packet inspecting firewall

You can configure a packet inspecting firewall for outbound connection. These firewalls decrypt the SSL traffic, and then re-encrypt it before passing it on to the endpoint. This configuration usually requires the firewall to issue its own certificates to the SAP system that is consuming an AWS service. You must install your firewall's CA certificate in STRUST. For more information, see [HTTPS connectivity](#).

Gateway endpoints

Some AWS services offer gateway endpoints to provide a VPC with high-performance access without internet. These endpoints are transparent to SDK for SAP ABAP, and do not require any configuration.

For more information, see [Gateway endpoints](#).

Custom interface endpoints

If you need to override the default endpoint resolution with a custom endpoint, you can use an interface endpoint to provide your VPC with high-performance access without internet. For more information, see [Configure an interface endpoint](#).

When not using private DNS, these endpoints have their own DNS addresses, and an ABAP program must explicitly override the usual endpoint resolution logic. For more information, see AWS re:Post – [Why can't I resolve service domain names for an interface VPC endpoint?](#)

In the following example, an interface endpoint is created for AWS STS and Amazon Translate. The SAP system is not using private DNS, and calls the services with custom endpoint. The logical resources defined in `/AWS1/IMG` represent the physical interface endpoint addresses, such as `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`. This avoids hard coding the DNS in code.

In the following code, the logical resources in `/AWS1/IMG` are first resolved to physical endpoint names. They are then provided to the factory methods of AWS session class (that uses AWS STS to assume an IAM role) and translate API class.

```
" 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...
```

As shown in the example, any method calls on `go_xl8` go to the endpoint `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`. It is also possible to define the routing custom endpoint in the IMG configuration instead of in code, as shown in the next section.

Advanced routing

In the previous section we showed how a custom endpoint can be specified in the `iv_custom_endpoint` argument of the factory methods for the SDK modules. As the number of ABAP programs using the SDK increases, this can become difficult to manage. It is possible to configure a mapping from an AWS service to a custom endpoint in the SDK Profile. For each SID, client, and scenario, the service three-letter abbreviation (TLA) can be mapped to an endpoint URL:

TLA	Custom Endpoint URL
BDR	https://vpce-23456789abcdef012-3c4d5e6f.bedrock-runtime.us-east-1.vpce.amazonaws.com
LMD	https://vpce-123456789abcdef01-2b3c4d5e.lambda.us-east-1.vpce.amazonaws.com
S3	https://vpce-0123456789abcdef0-1a2b3c4d.s3.us-east-1.vpce.amazonaws.com

With this configuration, you do not need to specify `iv_custom_endpoint` in the factory method calls. The custom endpoint is selected automatically from the configuration table. The configuration is specific to the SDK Profile so you can create multiple profiles with different routing to suit your needs. As with other SDK Profile configuration, the routing is SID and client specific so separate routing can be defined for different systems.

Per-service proxy server override

By default, the SDK uses the proxy server settings configured in Transaction SICF (see [the section called "Connection through a proxy server"](#)). The proxy settings in SICF apply globally to all outbound HTTP connections from the SAP system. In some environments, you may need finer control over which AWS services use the proxy server and which connect directly.

In the **Advanced Routing** section of `/AWS1/IMG`, you can configure a **Use Proxy Server** setting for each service. This setting controls whether the SDK routes requests for that service through the proxy server defined in SICF, regardless of the global proxy activation or filter settings.

The following values are available:

- **Default** – Use whatever proxy behavior is configured in SICF. If the proxy is active and the endpoint is not excluded by the filter, the proxy is used. This is the default behavior.
- **Always** – Always route requests for this service through the proxy server defined in SICF, regardless of the global activation or filter settings.

- **Never** – Never route requests for this service through the proxy server, regardless of the global activation or filter settings. Use this when a service endpoint is reachable directly without a proxy, for example when using a VPC endpoint.

For example, you might configure Amazon S3 and AWS STS to never use the proxy (because they are accessed through VPC gateway endpoints), while Amazon SNS always uses the proxy because it is only reachable through the internet.

TLA	Use Proxy Server
EC2	Default
S3	Never
SNS	Always
STS	Never

This configuration is defined per SDK Profile, SID, client, and scenario. You can create multiple profiles with different proxy routing to suit different environments or use cases.

Note

The proxy server host and port are always defined in the **HTTPS Protocol** tab of the proxy settings in Transaction SICF. The per-service override only controls whether the proxy is used, not which proxy server is used.

Accessing endpoints in multiple Regions

AWS endpoint is automatically determined from your default AWS Region that is defined in the SDK profile. You can also specify a region programmatically, overriding the default region. This can be overridden in the factory `CREATE()` method, or later with the SDK's configuration object. For more information, see [Programmatic configuration](#).

In the following example, the factory `CREATE()` method is used to set the region and list the Amazon SQS queues in both `us-east-1` and `us-west-2` Regions.

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

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

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

Service provider settings

Basis administrators sometimes need to control certain features of the SDK across the entire system, from client 000. This is a common scenario for hosting and service providers that operate systems in their own AWS account on behalf of their customers. AWS SDK for SAP ABAP supports Service Provider settings. These settings are configured in client 000, and affect the SDK across all clients. Service Provider settings are not supported in SDK for SAP ABAP - BTP edition.

Service Provider settings are configured in transaction /AWS1/IMG, and must be configured in client 000. Service Provider settings in other clients are ignored. The settings in client 000 take effect across all clients, and supercede other IMG settings in case of conflict.

Use the following steps to configure the Service Provider settings in client 000.

1. Expand the **Service Provider Settings** branch in transaction /AWS1/IMG.
2. Choose **Service Provider Guardrails**
3. Select **New Entries**, and adjust the settings based on your business requirements.

- *Disable EC2 Metadata* – prevents the SDK from accessing EC2 instance metadata in all clients, even if an SDK Profile is configured to authenticate using EC2 instance metadata. The SDK raises an exception if an ABAP program attempts to access instance metadata using the SDK.

4. Select **Save**.

Refresh, trace, and telemetry topics for AWS SDK for SAP ABAP

This section covers the following topics.

Topics

- [SAP system refresh](#)
- [Trace](#)
- [Telemetry](#)

SAP system refresh

After a system refresh, the primary challenge for a Basis administrator is to ensure that the separate systems are not accessing each other's resources. For example, you may want to ensure that your QA SAP system is not accessing the resources, such as an S3 bucket, of your Production landscape.

SDK for SAP ABAP provides a safety-conscious approach of *logical resources* to this challenge. A business analyst can take the following steps.

1. Define a logical resource, such as ZINVOICE_OUTBOUND.
2. Map all systems and clients in the development system.
3. Transport the configuration of ALL systems forward until the production landscape.

Basis steps after a refresh

1. Check the authentication

- If the system is using Secret Access Key authentication, the SSF-encrypted credentials will be invalid because they are stored in master data. The credentials must be re-entered, which may require regenerating a new Secret Access Key in <https://console.aws.amazon.com/iam/>.
- If the system is authenticating with EC2 instance metadata, no steps are required.

Check the trace settings

- In /AWS1/IMG, ensure that the trace settings are what you want. These settings are not transportable.

Trace

Trace output is controlled in the **IMG runtime settings**.

The trace levels that you can use are:

- **No Trace**
- **Trace API calls**
- **Trace API calls and payload**

This option contains unencrypted payload information.

- **Trace API calls, payload, and internal XML transformation**

This option contains unencrypted payload information.

If API trace is activated, traces are written to DIR_WORK in `aws1_trace-YYYY-MM-DD.log` file.

If payload trace is additionally activated, additional files with the title `aws1_payload_*` are created for each call and payload component. The payload trace length can be limited with the length limit applying to each individual payload trace fail.

Payload traces are primarily intended to collect information to be provided to Support in the event of a serialization error. We recommend that you choose **No Trace** unless you're attempting to diagnose an SDK error.

Note

Payload traces can contain unencrypted business information. We recommend turning these traces on only for a request by AWS Support to help you troubleshoot. You can turn these traces off after resolution. Traces are not automatically deleted, and need to be removed by the system administrator when no longer needed.

These settings are not applicable to SDK for SAP ABAP - BTP edition.

Telemetry

SDKs send telemetry information to Support. SDK for SAP ABAP collects the following information:

- OS release and patch level
- SAP_BASIS release and patch level
- SAP Kernel release and patch level

You can opt in to send the following information to Support.

- SAP SID and instance name (host_sid_nn)
- SAP Client (SY-MANDT)
- Transaction code (SY-TCODE) and report (SY-REPID)

The additional information enables Support to help you better. Support can detect why a certain API call was made and can further find the relevant transaction in a SAP system.

Telemetry is limited to the SDK and API versions for SDK for SAP ABAP - BTP edition.

Using AWS SDK for SAP ABAP

SDK for SAP ABAP has two major components.

- SDK Runtime (package /AWS1/RT) – a set of objects that underpin the security, authentication, tracing, configuration, data conversion, and other cross-API functions. The API modules for Amazon S3, AWS STS, IAM Roles Anywhere, and Secrets Manager are mandatory.
- APIs (package /AWS1/API and its sub-packages) – a sub-package for each API where the objects of each API are completely independent of each other, ensuring that a change in one API does not break another API. To see a complete list of AWS SDK for SAP ABAP APIs, see [AWS SDK for SAP ABAP - API Reference Guide](#).

This section covers the following topics.

Topics

- [Representation of data in ABAP](#)
- [Amazon S3 example program](#)
- [SDK for SAP ABAP concepts](#)
- [AWS SDK for SAP ABAP features](#)
- [Building products with SDK](#)
- [Customize HTTP requests to AWS](#)
- [Limitations](#)

Representation of data in ABAP

This section covers the following topics.

Topics

- [Data types](#)
- [AWS data types](#)

Data types

AWS services have a standard set of data types that must be mapped to ABAP data types. See the following table for more details.

AWS data type	ABAP data type	Comments
boolean	C	Single character "X" and " "
String	STRING	
Byte	INT2	INT2 has a larger range than 0-255. Most AWS services will truncate overflows but this behavior is not formally defined.
Short	INT2	
Integer	INT4	
Long	DEC19	INT8 is not available until ABAP 750. DEC19 is used for compatibility and consistency across all supported ABAP platforms.
Blob	XSTRING	Represents binary data
Float	STRING	While ABAP supports DECFLOATs, it cannot represent values such as NaN, Infinity and -Infinity. AWS SDK represents these internally as STRINGs, and converts them to DECFLOAT16 at runtime. If NaN, Infinity or +Infinity are represented, the developer may process these
Double	STRING	

AWS data type	ABAP data type	Comments
		in response to a special set of exceptions or mappings.
bigInteger	STRING	These values represent infinite-length numbers that cannot be represented in ABAP, and STRINGS are used instead of bigInteger.
bigDecimal	STRING	
Timestamp	TZNTSTMPS	TZNTSTMPS enables processing with native ABAP timestamp functions.

AWS services also return the following aggregate data types.

AWS data type	ABAP data type	Comments
Structure	Class	
Union	Class	A union is the same as a structure, except that a union will never have more than one field set. All other fields will be set to <i>No Value</i> .
Array	STANDARD TABLE	
Hash	HASHED TABLE	The hashed table will only have two columns: a KEY (string) and a VALUE (class).

AWS data types

The following approaches have been integrated to support AWS services in ABAP.

- Certain AWS data types cannot be represented in ABAP. For examples, the float data type in ABAP does not support the NaN, Infinity, or -Infinity values. Therefore, the float data type is represented as STRING and is translated to DECFLOAT16 at runtime.
- AWS data is represented on the wire as JSON or XML, and the values are optional. For example, see the following examples returned by an AWS service in JSON.

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

If Ana doesn't have a middle name, the service returns the following output.

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

ABAP does not distinguish between *a string of length 0* and *a string that has no value*. Other languages might assign a NULL value to the string or wrap the string in a construct (such as, Java's `Optional<>` wrapper). These are not supported in ABAP. Therefore, SDK for SAP ABAP facilitates the distinction in values by providing variants of the *getter* method.

Amazon S3 example program

This section walks you through a simple example program to list the contents of an Amazon S3 bucket by calling `ListObjectsV2`.

Topics

- [Prerequisites](#)
- [Code](#)
- [Code sections](#)

Prerequisites

You must meet the following prerequisites to run this example program.

- You have an Amazon S3 bucket. In this tutorial, the bucket is named `demo-invoices.customer.com`.
- Transaction `/AWS1/IMG`:
 - Has a defined SDK profile named `DEMO_S3`.
 - In the SDK profile, the logical IAM role `TESTUSER` must map to an IAM role, such as `arn:aws:iam::111122223333:role/SapDemoFinance` which grants `s3:ListBucket` permission to list the contents of your Amazon S3 bucket.
 - Has a logical resource named `DEMO_BUCKET` that is mapped to your Amazon S3 bucket with the SID and client of your SAP system.
- Your user has a PFCG role that:
 - Authorizes the user to access `DEMO_S3` SDK profile via auth object - `/AWS1/SESS`.
 - Authorizes the user for logical IAM role `TESTUSER` access via auth object - `/AWS1/LROL`.
- Your SAP system can authenticate itself to AWS using the method defined in the SDK profile.
- Your Amazon EC2 instance profile grants your SAP system the rights to `sts:assumeRole` in the IAM role `arn:aws:iam::111122223333:role/SapDemoFinance` mapped in the SDK profile.

Code

The following code block demonstrates what your code would look like.

```
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.

```

Code sections

The following is a review of the code in sections.

```

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

```

The user can't specify a physical bucket name. They specify a logical bucket and the system administrators (specifically the business analyst) in coordination with the AWS administrator map logical buckets to physical buckets in /AWS1/IMG. In most business scenarios, the user doesn't have a chance to choose the logical bucket — the logical resource ID is hard coded in the code or configured in a custom configuration table.

```

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

```

This line establishes a security session and declares that this ABAP program expects to use the DEMO_S3 SDK profile. This call is the connection to the SDK configuration and pulls in the default Region, authentication settings, and the desired IAM Role. A call to AUTHORIZATION-CHECK is automatically made to ensure that authorization object /AWS1/SESS is satisfied. Additionally, AUTHORIZATION-CHECK calls will be made to determine the most powerful (lower sequence number) logical IAM role the user is authorized for, based on the authorization object /AWS1/LROL.

The SDK will assume that the IAM role is mapped to the logical IAM role for the SID and client. Then, the session object activates tracing based on the trace settings in the IMG.

If the user is not authorized for the requested SDK profile or for any available logical IAM role, an exception will be raised.

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

This line resolves the logical resource to a physical bucket name. If the logical resource cannot be resolved because the configuration has no mapping for this SID/client combination, an exception will be raised.

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

This line creates an API object for Amazon S3 using the `create()` method of `/aws1/cl_s3_factory`. The returned object is of type `/aws1/if_s3` which is the interface for an Amazon S3 API. A separate API object must be created for each service. For example, if an ABAP program is consuming Amazon S3, AWS Lambda, and DynamoDB, then it creates API objects from `/aws1/cl_s3_factory`, `/aws1/cl_lmd_factory`, and `/aws1/cl_dyn_factory`.

There are some optional parameters to the constructor which enable you to specify the Region if you want to override the default Region configured IMG. In this way, there can be two instances of `/aws1/if_s3`, one for `us-east-1` and one for `us-west-2`, if you want to copy objects from a bucket in one Region to a bucket in another Region. Similarly, you can create two different security session objects and use them to create two separate instances of `/aws1/cl_s3`, if you need a report to read from a finance-related bucket and write objects to a logistics-related bucket.

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

This line is a call to `ListObjectsv2`. It requires simple input arguments and returns a single object. These objects may represent deep JSON and XML data, de-serialized into an ABAP object-oriented construct. It can be quite complicated in some cases. Now, you only need to process the output to list the contents of the bucket.

```
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.
```

The data is accessed using a GET...() style method that hides the internal representation of the data. GET_CONTENTS() returns an ABAP table and each row itself contains an object representing a single Amazon S3 entry. In most cases, AWS SDK takes this object-oriented approach and all data is represented as objects and tables. The LastModified field is represented as a timestamp that can be converted to a date with the ABAP-native CONVERT TIME STAMP command. the GET_SIZE() returns an INT4 for easy math and formatting operations.

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

All errors – connection, 4xx client, 5xx server, or any ABAP error, such as authorization or configuration errors, are represented as exceptions. You can tackle each exception separately. You have the choice of whether an exception should be handled as an informational error, a retry, warning, short dump, or any other kind of handling.

SDK for SAP ABAP concepts

This section covers the basic concepts of AWS SDK for SAP ABAP.

Topics

- [API classes](#)
- [Additional objects](#)
- [Structure classes](#)
- [Arrays](#)
- [Maps](#)
- [Higher level functions](#)

API classes

Each AWS service is assigned a three-letter acronym or TLA. The service is represented by an interface in the `/AWS1/IF_<TLA>` format. We will call this the service interface. The API class is in the `/AWS1/API_<TLA>` package. The service interface consists of one method for each AWS operation (we will call these methods Operation Methods). To see a complete module list of AWS SDK for SAP ABAP TLAs, see [AWS SDK for SAP ABAP - Module List](#).

Each operation method has some IMPORTING arguments and at the most one RETURNING argument. Often, these arguments will be objects with complicated constructors and a long set of GET...() methods. In many cases, the objects will contain nested objects, recursive references, tables of objects, tables of tables, and so forth. This is because AWS services are passing deep XML and JSON structures, which cannot be represented by a flat set of arguments.

The RETURNING argument is always a class, even if the class contains only a single attribute.

Additional objects

In addition to containing the primary API class, each API package contains various related repository and data dictionary objects.

- A class for each structure-type object.
- A class for any primitive data type which appears in a table. For example, if a service returns a table of strings, the ABAP API will represent it as a table of objects, where each object is a wrapper class that encapsulates a string. This is so that the wrapper class can hide the details of representing a null string that cannot be represented natively in ABAP.
- An exception class for any specific errors defined by the service.
- Data elements for each primitive data type. Each data type has its own data element in order to be self-documenting.
- Additional objects for internal processing, such as XSLT transforms for serializing and de-serializing XML and JSON payloads.

Structure classes

Most AWS data, sent and received by the service, is represented by AWS SDK as classes. These classes represent structures of data and hide the internal details of the storage. In particular, the classes hide the way the SDK represents *this field has no value*.

For each field in a structure class, there are three methods.

GET_field()

The GET_field() method

- Returns the value of the field, or
- If the field has no value, it returns a default value, which you can set as an optional parameter.

For example, consider the following code that prints the location constraint of a bucket.

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

If the bucket has no location constraint at all (as in the case of us-east-1), then GET_LOCATIONCONSTRAINT() will return the empty string. You can override this behavior and specify the desired value if the field has no value at all.

```
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' ).
```

Now the program will write Bucket Location: assuming us-east-1 if getbucketlocation()'s result does not return a location.

It is possible to ask the GET() method to return a specific result if the requested value is completely missing, see the following code example.

```
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' ).
```

In this case, if there is no location constraint, GET_LOCATIONCONSTRAINT() will return NopeNopeNope.

HAS_field()

HAS_field() method is a way to find out if the field has a value or not. See the following example.

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

If a certain field is known to always have a value, there will be no HAS_field() method.

ASK_field()

The ASK_field() method returns the value of the field or raises an exception if it has no value. This is a convenient way to process a number of fields, and bail out of the logic and take a different approach if any of the fields have no value.

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

Note that /AWS1/CX_RT_VALUE_MISSING is a static exception and you will get a warning if you choose not to catch it.

Best practices

In general, you can use the GET_field() method as it treats a null string as an empty string and is the most ABAP-like of the three options. However, it does not let you easily distinguish between situations where the field has a blank value and where the field has no value. If your business logic depends on distinguishing missing data versus blank data, then the HAS or ASK methods let you handle these cases.

Arrays

Arrays are represented as ABAP standard tables of objects.

A JSON array can contain null values, such as the following array: ['cat' , 'dog' , null , 'horse']. This is referred to as a sparse array. It is represented in ABAP as an internal table of object references, and the null value is represented in the table as a true ABAP null value. When iterating through a sparse table, you must check for null values to avoid accessing a null object and getting a CX_SY_REF_IS_INITIAL exception. In practice, sparse arrays are rare in AWS services.

To initialize an array of objects, it is convenient to use the new ABAP 7.40 constructs. Consider this launch of an Amazon EC2 instance with several security groups assigned:

```
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_tagsspecifications = make_tag_spec( 'instance' )  
)
```

Maps

JSON maps are represented in ABAP as Hashed Tables where each table row has only two components.

- KEY – a string which is the UNIQUE KEY of the table.
- VALUE – an object containing the value.

A map is one of the very few cases where AWS SDK uses a true structure, rather than a class. This is necessary because ABAP hashed tables cannot have an object reference as the key field, and AWS map keys are always non-null strings.

Higher level functions

The [API classes](#) described in the preceding section precisely mirror the AWS service APIs and represent those APIs as familiar ABAP classes. In some cases, the SDK also includes higher level functions that build on top of the API classes to simplify certain operations. The higher level functions are included for programmer convenience and do not replace the lower-level API classes.

If the SDK includes higher level functions for a module, they are included in the same transport and can be accessed through a factory class called `/AWS1/CL_TLA_L2_FACTORY`. The factory class includes methods to create various higher level clients for the module that are documented along with the rest of the API with the [API documentation](#).

AWS SDK for SAP ABAP features

AWS SDK for SAP ABAP provides the following features.

Topics

- [Programmatic configuration](#)
- [Waiters](#)
- [Paginators](#)
- [Retry behavior](#)
- [Presigners](#)
- [Cross-account IAM role chaining](#)

Programmatic configuration

Use `/n/AWS1/IMG IMG` transaction for AWS SDK for SAP ABAP, and Custom Business Configuration application for AWS SDK for SAP ABAP - BTP edition for programmatic configuration.

To begin programmatic configuration, begin by retrieving a configuration object with the `get_config()` command.

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

Each configuration object implements `/AWS1/IF_RT_CONFIG` interface that includes GETters and SETters corresponding to the IMG. For example, the default region can be overridden. See the following example command.

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

Some configuration objects have no IMG representation and can only be set programmatically, such as maximum retry attempts. See the following example command.

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

The configuration object of AWS services can also include service specific methods that are not represented in `/aws1/if_rt_config`. For example, Amazon S3 can address a bucket named `foobucket` using either `foobucket.s3.region.amazonaws.com` virtual endpoint or `s3.region.amazonaws.com/foobucket` path style. You can enforce the use of path style with the following example command.

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

For more information about service configurations, see [AWS SDK for SAP ABAP – API Reference Guide](#).

Waiters

When working with asynchronous AWS APIs, you need to wait for a certain resource to become available before taking further actions. For example, the `CREATETABLE()` API of Amazon DynamoDB responds right away with table status `CREATING`. You can initiate read or write operations only after the status of the table has changed to `ACTIVE`. Waiters give you the ability to confirm that AWS resources are in a particular state before performing actions on them.

Waiters use service operations to poll the status of AWS resources until the resource reaches the intended state or until it is determined that the resource doesn't reach the desired state. It can be time-consuming and error-prone to write the code to poll AWS resources continually. Waiters help in simplifying this complexity by taking the responsibility of performing polls on your behalf.

See the following Amazon S3 example using waiter.

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

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

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

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

- In this example, Amazon S3 client is used to create a bucket. The `get_waiter()` command is implemented to specify when the `bucketexists`.
- You must specify the `iv_max_wait_time` parameter for each waiter. It represents the total amount of time a waiter must wait before completion. In the preceding example, a waiter can run for 200 seconds.
- You may need to provide additional inputs for required parameters. In the preceding example, Amazon S3 bucket name is required for `iv_bucket` parameter.
- `/AWS1/CX_RT_WAITER_FAILURE` exception indicates that the waiter exceeded the maximum time specified in `iv_max_wait_time` parameter.
- `/AWS1/CX_RT_WAITER_TIMEOUT` exception indicates that the waiter has stopped due to not reaching the desired state.

Paginatons

Some AWS service operations offer paged responses. They are paginated to return a fixed amount of data with each response. You need to make subsequent requests with a token or a marker to retrieve the entire set of results. For instance, the `ListObjectsV2` Amazon S3 operation return up to 1,000 objects at a time. You must make subsequent requests with the appropriate token to get the next page of results.

Pagination is the process of sending successive requests to pick up where a previous request left off. Paginators are iterators of results provided by SDK for SAP ABAP. You can use paginated APIs with ease, and without understanding the underlying mechanism of API using pagination tokens.

Working with paginators

You can create paginators with the `get_paginator()` method that returns a paginator object. The paginator object calls the operation being paginated. The paginator object accepts required parameters to be provided to the underlying API. This process returns an iterator object that can be used to iterate over paginated results, using the `has_next()` and `get_next()` methods.

- `has_next()` returns a boolean value indicating if there are more responses or pages available for the called operation.
- `get_next()` returns the operation response.

The following example list all objects in an S3 bucket retrieved by using paginator.

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

Retry behavior

SDK for SAP ABAP enables you to configure the maximum number of retries for requests to AWS services that fail due to throttling or transient errors. The number of retries allowed at the service client level, that is the number of times the SDK retries the operation before failing and raising an exception is specified by the `AV_MAX_ATTEMPTS` attribute in the service configuration object. When a service client object is created, the SDK configures `AV_MAX_ATTEMPTS` attribute to a default value of 3. The service configuration object may be used to programmatically set the maximum retry attempt to a desired value. See the following example for more details.

```
" 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

Although the configuration object ABAP SDK allows *retry mode* to be set with the `/AWS1/IF_RT_CONFIG~SET_RETRY_MODE()` method, the SDK only supports the standard retry mode. For more information, see [Retry behavior](#) in AWS SDKs and Tools Reference Guide.

Presigners

You can use presigned URLs to grant time-limited access to some AWS services. A presigned URL can be entered in a browser or used by a program to perform the service operation. You can use the presigned URL multiple times, up to the expiration date and time. For more information, see [Working with presigned URLs](#) SDK for SAP ABAP clients for services that support presigners will have a special method called `GET_PRESIGNER()` to create a presigner for that service. Then call a method of the presigner, which correspond to the methods of the API client, except they return a presigned URL instead of actually performing the operation.

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

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

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

Cross-account IAM role chaining

Cross-account IAM role chaining support enables seamless access to resources across multiple AWS accounts through source profile configuration. This feature allows you to configure multiple role assumptions, where one profile assumes a role that then assumes another role, enabling complex cross-account access patterns.

For more information, see [Using Source Profile for Cross-Account Access](#).

Building products with SDK

A product or ABAP add-on that consumes AWS services can enhance and extend the capabilities of the SDK. You can build such products to use with the SDK.

Topics

- [Setting a product ID](#)

Setting a product ID

It is recommended that you set a product ID when establishing a session inside a product or add-on. See the following example for more details.

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

The product ID must only contain letters, numbers, and underscores with no spaces or special characters. You can match it to the product's technical name or any other identifier. If you develop multiple products or add-ons, the product ID must be unique for each product. For example, the product IDs for Invoice Analyzer, Tax Calculator, and Pricing Engine products can be INVOICE_ANALYZER, TAX_CALCULATOR, and PRICING_ENGINE.

Adding a product ID to the session enhances the telemetry that is sent to AWS with each service call. The product ID and the namespace of the object making the call is included in the telemetry. With this telemetry, Support can identify the product that is making the call in case of your customer facing issues with the SDK. It can help clarify that the call is actually being made by the product, and not your customer's code.

Customize HTTP requests to AWS

The AWS SDK for SAP ABAP handles the process of creating an HTTP request, sending a payload, and receiving a response. You can customize the behavior or content of the HTTP request to meet your own IT requirements. The SDK defines enhancement spot `/AWS1/RT_EHN_HTTP_CLIENT` as a central place to enhance the HTTP communication. The enhancement spot supports adding HTTP headers to the request made to AWS.

Implement an enhancement

SAP provides the following instructions for implementing an enhancement spot:

- [Classic ABAP](#)
- [BTP ABAP](#)

Filter the enhancement

The enhancement spot supports multiple implementations that can be active simultaneously. You can filter the execution of the BAdi based on the following attributes, if you need to ensure that your enhancement runs only on calls to a specific AWS service or API action:

- **TLA** - The three-letter abbreviation of the service, in uppercase letters.
- **OPERATION** - The API action name. For example, the operation to get an object from an S3 bucket is [GetObject](#). The action name is case-sensitive and might not exactly match the ABAP method name.

Code the enhancement

The enhancement provides the following method.

MODIFY_REQ_HEADERS

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

You can append and modify headers in the `CT_HEADERS` internal table. We do not recommend modifying headers, as this alters data that the AWS service uses. Any headers that you add are

ignored by the AWS service, but can be processed by your IT infrastructure, such as proxy servers or other middleware.

The enhancement spot is called before the calculation of authentication and telemetry headers, so these can't be modified by the enhancement.

The following is an example implementation.

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

Limitations

AWS SDK for SAP ABAP includes SDK modules for all AWS services. Some of these modules may have limitations, as described here.

- Modules that rely on MQTT protocol bindings, such as `iotevents`, will not work. MQTT is not an HTTP-based protocol and is currently not supported by AWS SDK for SAP ABAP.
- Operations that return event streams are supported, but will wait until the entire stream is received before returning control to ABAP. Examples are Amazon Bedrock Agents Runtime `InvokeAgent` and AWS Lambda `InvokeWithResponseStream`
- Operations that receive event streams are not supported due to limitations in the underlying ABAP platform. Examples are Amazon Q Business Chat and Amazon Lex `StartConversation`

AWS SDK for SAP ABAP has the following feature limitations.

- The following Amazon S3 features are not yet supported.
 - Multi-Region access points
 - Amazon S3 client-side encryption

AWS SDK for SAP ABAP - BTP edition has the following limitations during the developer preview.

- Some modules may not be available.
- It cannot be uninstalled.
- It is updated less frequently.

SDK for SAP ABAP code examples

The code examples in this topic show you how to use the AWS SDK for SAP ABAP with AWS.

Basics are code examples that show you how to perform the essential operations within a service.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Some services contain additional example categories that show how to leverage libraries or functions specific to the service.

Services

- [ACM examples using SDK for SAP ABAP](#)
- [API Gateway examples using SDK for SAP ABAP](#)
- [Application Recovery Controller examples using SDK for SAP ABAP](#)
- [Aurora examples using SDK for SAP ABAP](#)
- [Auto Scaling examples using SDK for SAP ABAP](#)
- [Amazon Bedrock Runtime examples using SDK for SAP ABAP](#)
- [Amazon Bedrock Agents Runtime examples using SDK for SAP ABAP](#)
- [CloudFront examples using SDK for SAP ABAP](#)
- [CloudWatch examples using SDK for SAP ABAP](#)
- [CloudWatch Logs examples using SDK for SAP ABAP](#)
- [Amazon Cognito Identity Provider examples using SDK for SAP ABAP](#)
- [Amazon Comprehend examples using SDK for SAP ABAP](#)
- [AWS Config examples using SDK for SAP ABAP](#)
- [AWS Control Tower examples using SDK for SAP ABAP](#)
- [Firehose examples using SDK for SAP ABAP](#)
- [DynamoDB examples using SDK for SAP ABAP](#)
- [Amazon EC2 examples using SDK for SAP ABAP](#)

- [Amazon ECR examples using SDK for SAP ABAP](#)
- [Amazon EMR examples using SDK for SAP ABAP](#)
- [EventBridge Scheduler examples using SDK for SAP ABAP](#)
- [AWS Glue examples using SDK for SAP ABAP](#)
- [HealthImaging examples using SDK for SAP ABAP](#)
- [HealthLake examples using SDK for SAP ABAP](#)
- [IAM examples using SDK for SAP ABAP](#)
- [AWS IoT examples using SDK for SAP ABAP](#)
- [AWS IoT data examples using SDK for SAP ABAP](#)
- [AWS IoT SiteWise examples using SDK for SAP ABAP](#)
- [Amazon Keyspaces examples using SDK for SAP ABAP](#)
- [Kinesis examples using SDK for SAP ABAP](#)
- [AWS KMS examples using SDK for SAP ABAP](#)
- [Lambda examples using SDK for SAP ABAP](#)
- [Organizations examples using SDK for SAP ABAP](#)
- [Amazon Pinpoint examples using SDK for SAP ABAP](#)
- [Amazon Pinpoint SMS and Voice API examples using SDK for SAP ABAP](#)
- [Amazon Polly examples using SDK for SAP ABAP](#)
- [Amazon RDS examples using SDK for SAP ABAP](#)
- [Amazon Redshift examples using SDK for SAP ABAP](#)
- [Amazon Rekognition examples using SDK for SAP ABAP](#)
- [Amazon S3 examples using SDK for SAP ABAP](#)
- [Amazon S3 Control examples using SDK for SAP ABAP](#)
- [SageMaker AI examples using SDK for SAP ABAP](#)
- [Secrets Manager examples using SDK for SAP ABAP](#)
- [Amazon SES examples using SDK for SAP ABAP](#)
- [Amazon SES API v2 examples using SDK for SAP ABAP](#)
- [Amazon SNS examples using SDK for SAP ABAP](#)
- [Amazon SQS examples using SDK for SAP ABAP](#)
- [Step Functions examples using SDK for SAP ABAP](#)

- [Systems Manager examples using SDK for SAP ABAP](#)
- [Amazon Textract examples using SDK for SAP ABAP](#)
- [Amazon Transcribe examples using SDK for SAP ABAP](#)
- [Amazon Translate examples using SDK for SAP ABAP](#)

ACM examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with ACM.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AddTagsToCertificate

The following code example shows how to use AddTagsToCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/  
certificate-id'  
    lo_acm->addtagstocertificate(  
        iv_certificatearn = iv_certificate_arn
```

```
        it_tags = it_tags
    ).
    MESSAGE 'Tags added to certificate successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidarnex.
        MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
    CATCH /aws1/cx_acmresourcenotfoundex.
        MESSAGE 'Certificate not found.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidtagex.
        MESSAGE 'Invalid tag provided.' TYPE 'I'.
    CATCH /aws1/cx_acmtoomanytagsex.
        MESSAGE 'Too many tags for certificate.' TYPE 'I'.
    ENDRY.
```

- For API details, see [AddTagsToCertificate](#) in *AWS SDK for SAP ABAP API reference*.

DeleteCertificate

The following code example shows how to use DeleteCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [DeleteCertificate](#) in *AWS SDK for SAP ABAP API reference*.

DescribeCertificate

The following code example shows how to use DescribeCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [DescribeCertificate](#) in *AWS SDK for SAP ABAP API reference*.

GetCertificate

The following code example shows how to use GetCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

- For API details, see [GetCertificate](#) in *AWS SDK for SAP ABAP API reference*.

ImportCertificate

The following code example shows how to use ImportCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Only pass certificate chain if it's provided (it's optional)
  IF iv_certificate_chain IS NOT INITIAL.
    DATA(lo_result) = lo_acm->importcertificate(
      iv_certificate = iv_certificate
      iv_privatekey = iv_private_key
      iv_certificatechain = iv_certificate_chain
    ).
  ELSE.
    lo_result = lo_acm->importcertificate(
      iv_certificate = iv_certificate
      iv_privatekey = iv_private_key

```

```
    ).
  ENDIF.
  ov_certificate_arn = lo_result->get_certificatearn( ).
  MESSAGE 'Certificate imported successfully.' TYPE 'I'.
  CATCH /aws1/cx_acminvalidparameterex.
  MESSAGE 'Invalid parameter provided.' TYPE 'I'.
  CATCH /aws1/cx_acmlimitexceededex.
  MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
  ENDRY.
```

- For API details, see [ImportCertificate](#) in *AWS SDK for SAP ABAP API reference*.

ListCertificates

The following code example shows how to use ListCertificates.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
  TRY.
    oo_result = lo_acm->listcertificates(
      iv_maxitems = iv_max_items
      it_certificatestatuses = it_statuses
      io_includes = io_includes
    ).
    MESSAGE 'Certificates listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_acminvalidargsex.
  MESSAGE 'Invalid arguments provided.' TYPE 'I'.
  CATCH /aws1/cx_acmvalidationex.
  MESSAGE 'Validation error occurred.' TYPE 'I'.
  ENDRY.
```

- For API details, see [ListCertificates](#) in *AWS SDK for SAP ABAP API reference*.

ListTagsForCertificate

The following code example shows how to use ListTagsForCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [ListTagsForCertificate](#) in *AWS SDK for SAP ABAP API reference*.

RemoveTagsFromCertificate

The following code example shows how to use RemoveTagsFromCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

- For API details, see [RemoveTagsFromCertificate](#) in *AWS SDK for SAP ABAP API reference*.

RequestCertificate

The following code example shows how to use RequestCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_domain_name = 'example.com'
  " iv_validation_method = 'DNS' or 'EMAIL'
  DATA(lo_result) = lo_acm->requestcertificate(
    iv_domainname = iv_domain_name
    it_subjectalternativenames = COND #( WHEN it_alternate_domains IS NOT
INITIAL
                                     THEN it_alternate_domains )
    iv_validationmethod = iv_validation_method
  ).

```

```

    ov_certificate_arn = lo_result->get_certificatearn( ).
    MESSAGE 'Certificate requested successfully.' TYPE 'I'.
  CATCH /aws1/cx_acminvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'I'.
  CATCH /aws1/cx_acmlimitexceededex.
    MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
  CATCH /aws1/cx_acminvdomvationoptsex.
    MESSAGE 'Invalid domain validation options.' TYPE 'I'.
  ENDTRY.

```

- For API details, see [RequestCertificate](#) in *AWS SDK for SAP ABAP API reference*.

ResendValidationEmail

The following code example shows how to use ResendValidationEmail.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

```
MESSAGE 'Certificate is not in a valid state.' TYPE 'I'.
CATCH /aws1/cx_acminvdomvationoptsex.
MESSAGE 'Invalid domain validation options.' TYPE 'I'.
ENDTRY.
```

- For API details, see [ResendValidationEmail](#) in *AWS SDK for SAP ABAP API reference*.

API Gateway examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with API Gateway.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateDeployment

The following code example shows how to use CreateDeployment.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_agw->createdeployment(
```

```

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

```

- For API details, see [CreateDeployment](#) in *AWS SDK for SAP ABAP API reference*.

CreateResource

The following code example shows how to use CreateResource.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

```
MESSAGE lo_not_found->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_not_found.
CATCH /aws1/cx_agwtoomanyrequestsex INTO DATA(lo_too_many).
MESSAGE lo_too_many->get_text( ) TYPE 'I'.
RAISE EXCEPTION lo_too_many.
ENDTRY.
```

- For API details, see [CreateResource](#) in *AWS SDK for SAP ABAP API reference*.

CreateRestApi

The following code example shows how to use CreateRestApi.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [CreateRestApi](#) in *AWS SDK for SAP ABAP API reference*.

DeleteRestApi

The following code example shows how to use DeleteRestApi.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [DeleteRestApi](#) in *AWS SDK for SAP ABAP API reference*.

GetResources

The following code example shows how to use GetResources.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [GetResources](#) in *AWS SDK for SAP ABAP API reference*.

GetRestApis

The following code example shows how to use GetRestApis.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

```

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

```

- For API details, see [GetRestApis](#) in *AWS SDK for SAP ABAP API reference*.

PutIntegration

The following code example shows how to use PutIntegration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

- For API details, see [PutIntegration](#) in *AWS SDK for SAP ABAP API reference*.

PutIntegrationResponse

The following code example shows how to use PutIntegrationResponse.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [PutIntegrationResponse](#) in *AWS SDK for SAP ABAP API reference*.

PutMethod

The following code example shows how to use PutMethod.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [PutMethod](#) in *AWS SDK for SAP ABAP API reference*.

PutMethodResponse

The following code example shows how to use PutMethodResponse.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [PutMethodResponse](#) in *AWS SDK for SAP ABAP API reference*.

Application Recovery Controller examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Application Recovery Controller.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

GetRoutingControlState

The following code example shows how to use GetRoutingControlState.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

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

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

```

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

- For API details, see [GetRoutingControlState](#) in *AWS SDK for SAP ABAP API reference*.

UpdateRoutingControlState

The following code example shows how to use UpdateRoutingControlState.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

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

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

```

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

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

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

" If successful, return the result
RETURN.

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

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

ENDTRY.
ENDLOOP.

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

```

- For API details, see [UpdateRoutingControlState](#) in *AWS SDK for SAP ABAP API reference*.

Aurora examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Aurora.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateDBClusterParameterGroup

The following code example shows how to use CreateDBClusterParameterGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    DATA(lo_output) = lo_rds->createdbclusterparamgroup(  
        iv_dbclusterparamgroupname = iv_param_group_name  
        iv_dbparametergroupfamily = iv_param_group_family  
        iv_description = iv_description  
    ).  
    oo_result = lo_output->get_dbclusterparametergroup( ).  
CATCH /aws1/cx_rdsdbparmgralrexfault.  
    " Re-raise exception - parameter group already exists  
    RAISE EXCEPTION TYPE /aws1/cx_rdsdbparmgralrexfault.  
CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
```

```
" Re-raise exception - quota exceeded
RAISE EXCEPTION TYPE /aws1/cx_rdsdbprimgrquotaexcd00.
ENDTRY.
```

- For API details, see [CreateDBClusterParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

DeleteDBClusterParameterGroup

The following code example shows how to use `DeleteDBClusterParameterGroup`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_rds->deletedbclusterparamgroup(
    iv_dbclusterparamgroupname = iv_param_group_name
  ).
CATCH /aws1/cx_rdsdbprimgrnotfndfault.
  " Re-raise exception - parameter group not found
  RAISE EXCEPTION TYPE /aws1/cx_rdsdbprimgrnotfndfault.
CATCH /aws1/cx_rdsinldbprimgrstatef00.
  " Re-raise exception - invalid state
  RAISE EXCEPTION TYPE /aws1/cx_rdsinldbprimgrstatef00.
ENDTRY.
```

- For API details, see [DeleteDBClusterParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBClusterParameterGroups

The following code example shows how to use `DescribeDBClusterParameterGroups`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [DescribeDBClusterParameterGroups](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBClusterParameters

The following code example shows how to use DescribeDBClusterParameters.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA lv_marker TYPE /aws1/rdsstring VALUE ''.  
  DATA lt_all_parameters TYPE /aws1/cl_rdsparameter=>tt_parameterslist.
```

```

DO.
  DATA(lo_output) = lo_rds->describedbclusterparameters(
    iv_dbclusterparamgroupname = iv_param_group_name
    iv_source = iv_source
    iv_marker = lv_marker
  ).

  LOOP AT lo_output->get_parameters( ) INTO DATA(lo_param).
    IF iv_name_prefix IS INITIAL OR
       lo_param->get_parametername( ) CP |{ iv_name_prefix }*|.
      APPEND lo_param TO lt_all_parameters.
    ENDIF.
  ENDLOOP.

  lv_marker = lo_output->get_marker( ).
  IF lv_marker IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_parameters = lt_all_parameters.
CATCH /aws1/cx_rdsdbprimgrnotfndfault.
  " Re-raise exception - parameter group not found
  RAISE EXCEPTION TYPE /aws1/cx_rdsdbprimgrnotfndfault.
ENDTRY.

```

- For API details, see [DescribeDBClusterParameters](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBEngineVersions

The following code example shows how to use DescribeDBEngineVersions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_engine           = 'mysql'
```

```

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

```

- For API details, see [DescribeDBEngineVersions](#) in *AWS SDK for SAP ABAP API reference*.

DescribeOrderableDBInstanceOptions

The following code example shows how to use `DescribeOrderableDBInstanceOptions`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" iv_engine          = 'mysql'
" iv_engineversion = '8.0.35'
TRY.
  oo_result = lo_rds->descrorderabledbinstoptions(
    iv_engine          = iv_engine
    iv_engineversion = iv_engineversion ).
  DATA(lv_option_count) = lines( oo_result->get_orderabledbinstoptions( ) ).
  MESSAGE |Retrieved { lv_option_count } orderable DB instance options.| TYPE
'I'.
ENDTRY.

```

- For API details, see [DescribeOrderableDBInstanceOptions](#) in *AWS SDK for SAP ABAP API reference*.

ModifyDBClusterParameterGroup

The following code example shows how to use `ModifyDBClusterParameterGroup`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [ModifyDBClusterParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

Auto Scaling examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Auto Scaling.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateAutoScalingGroup

The following code example shows how to use CreateAutoScalingGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

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

TRY.
  " Create launch template specification
  lo_launch_template = NEW /aws1/cl_aslaunchtemplatespec(
    iv_launchtemplatename = iv_launch_template_name
    iv_version = '$Default' ).

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

```

ELSE.
  " EC2-Classic or default VPC - use availability zones
  ao_asc->createautoscalinggroup(
    iv_autoscalinggroupname = iv_group_name
    it_availabilityzones = it_group_zones
    io_launchtemplate = lo_launch_template
    iv_minsize = iv_min_size
    iv_maxsize = iv_max_size ).
ENDIF.

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

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

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

```

- For API details, see [CreateAutoScalingGroup](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAutoScalingGroup

The following code example shows how to use DeleteAutoScalingGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

```

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

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

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

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

```

- For API details, see [DeleteAutoScalingGroup](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAutoScalingGroups

The following code example shows how to use DescribeAutoScalingGroups.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

TRY.
  " Build group names parameter

```

```
CREATE OBJECT lo_group_name
  EXPORTING
    iv_value = iv_group_name.
APPEND lo_group_name TO lt_group_names.

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

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

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

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

- For API details, see [DescribeAutoScalingGroups](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAutoScalingInstances

The following code example shows how to use `DescribeAutoScalingInstances`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

```
TRY.  
  DATA(lo_output) = ao_asc->describeautoscalinginstances(  
    it_instanceids = it_instance_ids ).  
  
  ot_output = lo_output->get_autoscalinginstances( ).  
  
  MESSAGE 'Auto Scaling instances information retrieved successfully' TYPE  
'I'.  
  
  CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).  
    RAISE EXCEPTION lo_contention.  
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).  
    RAISE EXCEPTION lo_generic_exception.  
ENDTRY.
```

- For API details, see [DescribeAutoScalingInstances](#) in *AWS SDK for SAP ABAP API reference*.

DescribeScalingActivities

The following code example shows how to use DescribeScalingActivities.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Example: iv_group_name = 'my-auto-scaling-group'  
  
TRY.  
  DATA(lo_output) = ao_asc->describescalingactivities(  
    iv_autoscalinggroupname = iv_group_name ).  
  
  ot_output = lo_output->get_activities( ).  
  
  MESSAGE 'Scaling activities retrieved successfully' TYPE 'I'.  
  
  CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
```

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

- For API details, see [DescribeScalingActivities](#) in *AWS SDK for SAP ABAP API reference*.

DisableMetricsCollection

The following code example shows how to use `DisableMetricsCollection`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Example: iv_group_name = 'my-auto-scaling-group'  
  
TRY.  
    ao_asc->disablemetricscollection(  
        iv_autoscalinggroupname = iv_group_name ).  
  
    MESSAGE 'Metrics collection disabled successfully' TYPE 'I'.  
  
    CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).  
    RAISE EXCEPTION lo_contention.  
    CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).  
    RAISE EXCEPTION lo_generic_exception.  
ENDTRY.
```

- For API details, see [DisableMetricsCollection](#) in *AWS SDK for SAP ABAP API reference*.

EnableMetricsCollection

The following code example shows how to use `EnableMetricsCollection`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

TRY.

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

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

```
CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
```

```
  RAISE EXCEPTION lo_contention.
```

```
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
```

```
  RAISE EXCEPTION lo_generic_exception.
```

```
ENDTRY.
```

- For API details, see [EnableMetricsCollection](#) in *AWS SDK for SAP ABAP API reference*.

SetDesiredCapacity

The following code example shows how to use SetDesiredCapacity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

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

CATCH /aws1/cx_ascscactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.

```

- For API details, see [SetDesiredCapacity](#) in *AWS SDK for SAP ABAP API reference*.

TerminateInstanceInAutoScalingGroup

The following code example shows how to use `TerminateInstanceInAutoScalingGroup`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

  oo_output = lo_output->get_activity( ).

```

```
MESSAGE 'Instance terminated successfully' TYPE 'I'.

CATCH /aws1/cx_ascscsactivityinprg00 INTO DATA(lo_activity_in_progress).
  RAISE EXCEPTION lo_activity_in_progress.
CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
  RAISE EXCEPTION lo_contention.
CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
  RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- For API details, see [TerminateInstanceInAutoScalingGroup](#) in *AWS SDK for SAP ABAP API reference*.

UpdateAutoScalingGroup

The following code example shows how to use UpdateAutoScalingGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

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

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

  CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
    RAISE EXCEPTION lo_contention.
  CATCH /aws1/cx_ascscsactivityinprg00 INTO DATA(lo_activity_in_progress).
```

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

- For API details, see [UpdateAutoScalingGroup](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Bedrock Runtime examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Bedrock Runtime.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

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

Anthropic Claude

InvokeModel

The following code example shows how to send a text message to Anthropic Claude, using the Invoke Model API.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Invoke the Anthropic Claude 2 foundation model to generate text. This example uses features of /US2/CL_JSON which might not be available on some NetWeaver versions.

```
"Claude V2 Input Parameters should be in a format like this:
```

```

*   {
*     "prompt": "\n\nHuman:\n\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\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.

```

Invoke the Anthropic Claude 2 foundation model to generate text using L2 high level client.

```

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.

```

Invoke the Anthropic Claude 3 foundation model to generate text using L2 high level client.

```

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.

```

- For API details, see [InvokeModel](#) in *AWS SDK for SAP ABAP API reference*.

Stable Diffusion

InvokeModel

The following code example shows how to invoke Stability.ai Stable Diffusion XL on Amazon Bedrock to generate an image.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create an image with Stable Diffusion.

```

"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.

```

Invoke the Stability.ai Stable Diffusion XL foundation model to generate images using L2 high level client.

```

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.

```

- For API details, see [InvokeModel](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Bedrock Agents Runtime examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Bedrock Agents Runtime.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

InvokeAgent

The following code example shows how to use `InvokeAgent`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

```

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

```

- For API details, see [InvokeAgent](#) in *AWS SDK for SAP ABAP API reference*.

CloudFront examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with CloudFront.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

ListDistributions

The following code example shows how to use `ListDistributions`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_fnt->listdistributions( ). " oo_result is returned for  
testing purposes. "  
    MESSAGE 'Retrieved list of CloudFront distributions.' TYPE 'I'.  
    CATCH /aws1/cx_fntinvalidargument.  
    MESSAGE 'Invalid argument provided.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListDistributions](#) in *AWS SDK for SAP ABAP API reference*.

UpdateDistribution

The following code example shows how to use `UpdateDistribution`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

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

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

```

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

- For API details, see [UpdateDistribution](#) in *AWS SDK for SAP ABAP API reference*.

CloudWatch examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with CloudWatch.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)
- [Scenarios](#)

Actions

DeleteAlarms

The following code example shows how to use `DeleteAlarms`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DeleteAlarms](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAlarms

The following code example shows how to use DescribeAlarms.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DescribeAlarms](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAlarmsForMetric

The following code example shows how to use `DescribeAlarmsForMetric`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_cwt->describealarmsformetric(           " oo_result is
returned for testing purposes. "
    iv_namespace   = iv_namespace
    iv_metricname  = iv_metric_name
    it_dimensions  = it_dimensions
    iv_statistic   = iv_statistic
    iv_period      = iv_period
    iv_unit        = iv_unit ).
    MESSAGE 'Alarms for metric retrieved.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DescribeAlarmsForMetric](#) in *AWS SDK for SAP ABAP API reference*.

DisableAlarmActions

The following code example shows how to use `DisableAlarmActions`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
"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.
```

- For API details, see [DisableAlarmActions](#) in *AWS SDK for SAP ABAP API reference*.

EnableAlarmActions

The following code example shows how to use `EnableAlarmActions`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
"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'.
    ENDRY.

```

- For API details, see [EnableAlarmActions](#) in *AWS SDK for SAP ABAP API reference*.

GetMetricStatistics

The following code example shows how to use `GetMetricStatistics`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
        oo_result = lo_cwt->getmetricstatistics(           " oo_result is
returned for testing purposes. "
        iv_namespace = iv_namespace
        iv_metricname = iv_metric_name
        iv_starttime = iv_start_time
        iv_endtime = iv_end_time
        iv_period = iv_period
        it_statistics = it_statistics ).
        MESSAGE 'Metric statistics retrieved.' TYPE 'I'.
    CATCH /aws1/cx_cwtinparamvalueex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
    ENDRY.

```

- For API details, see [GetMetricStatistics](#) in *AWS SDK for SAP ABAP API reference*.

ListMetrics

The following code example shows how to use `ListMetrics`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
"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.
```

- For API details, see [ListMetrics](#) in *AWS SDK for SAP ABAP API reference*.

PutMetricAlarm

The following code example shows how to use `PutMetricAlarm`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.

```

- For API details, see [PutMetricAlarm](#) in *AWS SDK for SAP ABAP API reference*.

PutMetricData

The following code example shows how to use PutMetricData.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

DATA lt_metricdata TYPE /aws1/cl_cwtmetricdatum=>tt_metricdata.

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

```

```
INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

Put a set of data into a CloudWatch metric.

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

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

INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

- For API details, see [PutMetricData](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Get started with alarms

The following code example shows how to:

- Create an alarm.
- Disable alarm actions.
- Describe an alarm.
- Delete an alarm.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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'.
  ENDMETHOD.

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

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

  "Describe alarm using the same ABAP internal table."
  TRY.
    oo_result = lo_cwt->describealarms(
      it_alarmnames = lt_alarmnames ).
    " oo_result is
returned for testing purpose "
    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.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [DeleteAlarms](#)
 - [DescribeAlarms](#)

- [DisableAlarmActions](#)
- [PutMetricAlarm](#)

CloudWatch Logs examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with CloudWatch Logs.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

GetQueryResults

The following code example shows how to use `GetQueryResults`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cwl->getqueryresults(  
        iv_queryid = iv_query_id ).  
  
    " Display query status and result count  
    DATA(lv_status) = oo_result->get_status( ).
```

```

DATA(lt_results) = oo_result->get_results( ).
DATA(lv_result_count) = lines( lt_results ).

MESSAGE |Query status: { lv_status }. Retrieved { lv_result_count } log
event(s).| TYPE 'I'.
CATCH /aws1/cx_cwlinvalidparameterex.
  MESSAGE 'Invalid parameter.' TYPE 'E'.
CATCH /aws1/cx_cwlresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cwlserviceunavailex.
  MESSAGE 'Service unavailable.' TYPE 'E'.
ENDTRY.

```

- For API details, see [GetQueryResults](#) in *AWS SDK for SAP ABAP API reference*.

StartQuery

The following code example shows how to use StartQuery.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

```
" Display the query ID for tracking
DATA(lv_query_id) = oo_result->get_queryid( ).
MESSAGE |Query started successfully with ID: { lv_query_id }| TYPE 'I'.
CATCH /aws1/cx_cwlinvalidparameterex.
  MESSAGE 'Invalid parameter.' TYPE 'E'.
CATCH /aws1/cx_cwllimitexceededex.
  MESSAGE 'Limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cwlmalformedqueryex.
  MESSAGE 'Malformed query.' TYPE 'E'.
CATCH /aws1/cx_cwlresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cwlserviceunavailex.
  MESSAGE 'Service unavailable.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartQuery](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Cognito Identity Provider examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Cognito Identity Provider.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AdminInitiateAuth

The following code example shows how to use AdminInitiateAuth.

SDK for SAP ABAP**Note**

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

  " Add SECRET_HASH if provided
  IF iv_secret_hash IS NOT INITIAL.
    INSERT VALUE #(
      key = 'SECRET_HASH'
      value = NEW /aws1/c1_cgpathparamstype_w( iv_secret_hash )
    ) INTO TABLE lt_auth_params.
  ENDIF.

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

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

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

```

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

CATCH /aws1/cx_cgpnauthorizedex INTO DATA(lo_auth_ex).
  MESSAGE 'Not authorized. Check credentials.' TYPE 'E'.
ENDTRY.

```

- For API details, see [AdminInitiateAuth](#) in *AWS SDK for SAP ABAP API reference*.

AdminRespondToAuthChallenge

The following code example shows how to use AdminRespondToAuthChallenge.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Build challenge responses
  DATA(lt_challenge_responses) = VALUE /aws1/
cl_cgpchallengerspsty00=>tt_challengerspsty00(
    ( VALUE /aws1/cl_cgpchallengerspsty00=>ts_challengerspsty00(
      key = 'USERNAME'
      value = NEW /aws1/cl_cgpchallengerspsty00( iv_user_name ) ) )
    ( VALUE /aws1/cl_cgpchallengerspsty00=>ts_challengerspsty00(
      key = 'SOFTWARE_TOKEN_MFA_CODE'
      value = NEW /aws1/cl_cgpchallengerspsty00( iv_mfa_code ) ) )
  ).

  " Add SECRET_HASH if provided
  IF iv_secret_hash IS NOT INITIAL.
    INSERT VALUE #(
      key = 'SECRET_HASH'
      value = NEW /aws1/cl_cgpchallengerspsty00( iv_secret_hash )
    ) INTO TABLE lt_challenge_responses.
  ENDIF.

```

```
DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(  
  iv_userpoolid = iv_user_pool_id  
  iv_clientid = iv_client_id  
  iv_challengename = 'SOFTWARE_TOKEN_MFA'  
  it_challengeresponses = lt_challenge_responses  
  iv_session = iv_session  
).  
  
oo_auth_result = lo_result->get_authenticationresult( ).  
  
IF oo_auth_result IS BOUND.  
  MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.  
ELSE.  
  " Another challenge might be required  
  DATA(lv_next_challenge) = lo_result->get_challengename( ).  
  MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE 'I'.  
ENDIF.  
  
CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).  
  MESSAGE 'Invalid MFA code provided.' TYPE 'E'.  
  
CATCH /aws1/cx_cgpxpiredcodeex INTO DATA(lo_expired_ex).  
  MESSAGE 'MFA code has expired.' TYPE 'E'.  
  
CATCH /aws1/cx_cgpxnotauthorizedex INTO DATA(lo_auth_ex).  
  MESSAGE 'Not authorized. Check MFA configuration.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [AdminRespondToAuthChallenge](#) in *AWS SDK for SAP ABAP API reference*.

AssociateSoftwareToken

The following code example shows how to use AssociateSoftwareToken.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_cgp->associatesoftwaretoken(  
    iv_session = iv_session  
  ).  
  
  ov_secret_code = lo_result->get_secretcode( ).  
  
  MESSAGE 'MFA secret code generated successfully.' TYPE 'I'.  
  
  CATCH /aws1/cx_cgppresourcenotfoundex INTO DATA(lo_ex).  
    MESSAGE 'Session not found or expired.' TYPE 'E'.  
  
  CATCH /aws1/cx_cgppnotauthorizedex INTO DATA(lo_auth_ex).  
    MESSAGE 'Not authorized to associate software token.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [AssociateSoftwareToken](#) in *AWS SDK for SAP ABAP API reference*.

ListUsers

The following code example shows how to use ListUsers.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_cgp->listusers(  
    iv_userpoolid = iv_user_pool_id  
  ).  
  
  ot_users = lo_result->get_users( ).  
  
  MESSAGE |Found { lines( ot_users ) } users in the pool.| TYPE 'I'.  
  
  CATCH /aws1/cx_cgppresourcenotfoundex INTO DATA(lo_ex).
```

```
MESSAGE |User pool { iv_user_pool_id } not found.| TYPE 'E'.

CATCH /aws1/cx_cgpnnotauthorizedex INTO DATA(lo_auth_ex).
MESSAGE 'Not authorized to list users.' TYPE 'E'.
ENDTRY.
```

- For API details, see [ListUsers](#) in *AWS SDK for SAP ABAP API reference*.

VerifySoftwareToken

The following code example shows how to use `VerifySoftwareToken`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

  ov_status = lo_result->get_status( ).

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

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

  CATCH /aws1/cx_cgpenbsoftwaretokmf00 INTO DATA(lo_enabled_ex).
    MESSAGE 'Software token MFA is already enabled.' TYPE 'E'.
ENDTRY.
```

- For API details, see [VerifySoftwareToken](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Comprehend examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Comprehend.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateDocumentClassifier

The following code example shows how to use `CreateDocumentClassifier`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

TRY.

```
oo_result = lo_cpd->createdocumentclassifier(  
    iv_documentclassifiername = iv_classifier_name  
    iv_languagecode = iv_language_code  
    io_inputdataconfig = NEW /aws1/cl_cpdocclifierinpdatt00(  
        iv_s3uri = iv_training_s3_uri
```

```

    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_mode = iv_mode
  ).
  MESSAGE 'Document classifier creation started.' TYPE 'I'.
CATCH /aws1/cx_cpinvalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdrsrclimitexcdex.
  MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
  MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
  MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateDocumentClassifier](#) in *AWS SDK for SAP ABAP API reference*.

DeleteDocumentClassifier

The following code example shows how to use DeleteDocumentClassifier.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_cpd->deletedocumentclassifier(
    iv_documentclassifierarn = iv_classifier_arn
  ).
  MESSAGE 'Document classifier deleted.' TYPE 'I'.
CATCH /aws1/cx_cpinvalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
  MESSAGE 'Too many requests.' TYPE 'E'.

```

```
CATCH /aws1/cx_cpdresourcenotfoundex.  
  MESSAGE 'Resource not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdresourceinuseex.  
  MESSAGE 'Resource in use.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
  MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteDocumentClassifier](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDocumentClassificationJob

The following code example shows how to use `DescribeDocumentClassificationJob`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_cpd->describedocclassificationjob(  
    iv_jobid = iv_job_id  
  ).  
  MESSAGE 'Document classification job described.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
  MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdjobnotfoundex.  
  MESSAGE 'Job not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
  MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
  MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeDocumentClassificationJob](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDocumentClassifier

The following code example shows how to use `DescribeDocumentClassifier`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->describedocumentclassifier(  
        iv_documentclassifierarn = iv_classifier_arn  
    ).  
    MESSAGE 'Document classifier described.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdresourcenotfoundex.  
    MESSAGE 'Resource not found.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeDocumentClassifier](#) in *AWS SDK for SAP ABAP API reference*.

DescribeTopicsDetectionJob

The following code example shows how to use `DescribeTopicsDetectionJob`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->describetopicdetectionjob(  
        iv_jobid = iv_job_id  
    ).  
    MESSAGE 'Topics detection job described.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdjobnotfoundex.  
    MESSAGE 'Job not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeTopicsDetectionJob](#) in *AWS SDK for SAP ABAP API reference*.

DetectDominantLanguage

The following code example shows how to use DetectDominantLanguage.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->detectdominantlanguage( iv_text = iv_text ).  
    MESSAGE 'Languages detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrsizeexceedex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectDominantLanguage](#) in *AWS SDK for SAP ABAP API reference*.

DetectEntities

The following code example shows how to use DetectEntities.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->detectentities(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Entities detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrisizelmtexcdex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundsuppedlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectEntities](#) in *AWS SDK for SAP ABAP API reference*.

DetectKeyPhrases

The following code example shows how to use DetectKeyPhrases.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->detectkeyphrases(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Key phrases detected.' TYPE 'I'.  
    CATCH /aws1/cx_cpdtextrsizeex.   
        MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
    CATCH /aws1/cx_cpdundsupplanguageex.   
        MESSAGE 'Unsupported language.' TYPE 'E'.  
    CATCH /aws1/cx_cpdiinternalsrvrerr.   
        MESSAGE 'Internal server error occurred.' TYPE 'E'.  
    CATCH /aws1/cx_cpdiinvalidrequestex.   
        MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectKeyPhrases](#) in *AWS SDK for SAP ABAP API reference*.

DetectPiiEntities

The following code example shows how to use DetectPiiEntities.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
oo_result = lo_cpd->detectpiientities(  
    iv_text = iv_text  
    iv_languagecode = iv_language_code  
).  
MESSAGE 'PII entities detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrsizeexceedex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundisupportedlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectPiiEntities](#) in *AWS SDK for SAP ABAP API reference*.

DetectSentiment

The following code example shows how to use DetectSentiment.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->detectsentiment(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Sentiment detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrsizeexceedex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundisupportedlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdiinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
```

```
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectSentiment](#) in *AWS SDK for SAP ABAP API reference*.

DetectSyntax

The following code example shows how to use DetectSyntax.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->detectsyntax(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Syntax tokens detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrsizeex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdundisabledlanguageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectSyntax](#) in *AWS SDK for SAP ABAP API reference*.

ListDocumentClassificationJobs

The following code example shows how to use ListDocumentClassificationJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->listdocclassificationjobs( ).  
    MESSAGE 'Document classification jobs listed.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListDocumentClassificationJobs](#) in *AWS SDK for SAP ABAP API reference*.

ListDocumentClassifiers

The following code example shows how to use ListDocumentClassifiers.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_cpd->listdocumentclassifiers( ).  
    MESSAGE 'Document classifiers listed.' TYPE 'I'.
```

```
CATCH /aws1/cx_cpinvalidrequestex.  
  MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
  MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidfilterex.  
  MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
  MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListDocumentClassifiers](#) in *AWS SDK for SAP ABAP API reference*.

ListTopicsDetectionJobs

The following code example shows how to use `ListTopicsDetectionJobs`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_cpd->listtopicsdetectionjobs( ).  
  MESSAGE 'Topics detection jobs listed.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
  MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
  MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidfilterex.  
  MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
  MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListTopicsDetectionJobs](#) in *AWS SDK for SAP ABAP API reference*.

StartDocumentClassificationJob

The following code example shows how to use StartDocumentClassificationJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_cpd->startdocclassificationjob(
    iv_jobname = iv_job_name
    iv_documentclassifierarn = iv_classifier_arn
    io_inputdataconfig = NEW /aws1/cl_cpdinputdataconfig(
      iv_s3uri = iv_input_s3_uri
      iv_inputformat = iv_input_format
    )
    io_outputdataconfig = NEW /aws1/cl_cpdoutputdataconfig(
      iv_s3uri = iv_output_s3_uri
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
  ).
  MESSAGE 'Document classification job started.' TYPE 'I'.
CATCH /aws1/cx_cpdivalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
  MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cpdresourceunavailex.
  MESSAGE 'Resource unavailable.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
  MESSAGE 'KMS key validation error.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
  MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdresrclimitexcdex.
  MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartDocumentClassificationJob](#) in *AWS SDK for SAP ABAP API reference*.

StartTopicsDetectionJob

The following code example shows how to use StartTopicsDetectionJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_cpd->starttopicsdetectionjob(
    iv_jobname = iv_job_name
    io_inputdataconfig = NEW /aws1/cl_cpdinputdataconfig(
      iv_s3uri = iv_input_s3_uri
      iv_inputformat = iv_input_format
    )
    io_outputdataconfig = NEW /aws1/cl_cpdoutputdataconfig(
      iv_s3uri = iv_output_s3_uri
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
  ).
  MESSAGE 'Topics detection job started.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
  MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
  MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
  MESSAGE 'KMS key validation error.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
  MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdresrclimitexcdex.
  MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
  MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartTopicsDetectionJob](#) in *AWS SDK for SAP ABAP API reference*.

AWS Config examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS Config.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

DeleteConfigRule

The following code example shows how to use DeleteConfigRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
lo_cfs->deleteconfigrule( iv_rule_name ).  
MESSAGE 'Deleted AWS Config rule.' TYPE 'I'.
```

- For API details, see [DeleteConfigRule](#) in *AWS SDK for SAP ABAP API reference*.

DescribeConfigRules

The following code example shows how to use DescribeConfigRules.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA(lo_result) = lo_cfs->describeconfigrules(
  it_configrulenames = VALUE /aws1/cl_cfsconfigrulenames_w=>tt_configrulenames(
    ( NEW /aws1/cl_cfsconfigrulenames_w( iv_rule_name ) )
  )
).
ot_cfg_rules = lo_result->get_configrules( ).
MESSAGE 'Retrieved AWS Config rule data.' TYPE 'I'.
```

- For API details, see [DescribeConfigRules](#) in *AWS SDK for SAP ABAP API reference*.

PutConfigRule

The following code example shows how to use PutConfigRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Create a config rule for S3 bucket public read prohibition
lo_cfs->putconfigrule(
  io_configrule = NEW /aws1/cl_cfsconfigrule(
    iv_configrulename = iv_rule_name
```

```

        iv_description = |S3 Public Read Prohibited Bucket Rule|
        io_scope = NEW /aws1/cl_cfsscope(
            it_complianceresourcetypes = VALUE /aws1/
            cl_cfscplnresrctypes_w=>tt_complianceresourcetypes(
                ( NEW /aws1/cl_cfscplnresrctypes_w( |AWS::S3::Bucket| ) )
            )
        )
        io_source = NEW /aws1/cl_cfssource(
            iv_owner = |AWS|
            iv_sourceidentifier = |S3_BUCKET_PUBLIC_READ_PROHIBITED|
        )
        iv_inputparameters = '{}'
        iv_configrulestate = |ACTIVE|
    )
).
MESSAGE 'Created AWS Config rule.' TYPE 'I'.

```

- For API details, see [PutConfigRule](#) in *AWS SDK for SAP ABAP API reference*.

AWS Control Tower examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS Control Tower.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

DisableBaseline

The following code example shows how to use `DisableBaseline`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Disable the baseline
  DATA(lo_output) = io_ctt->disablebaseline(
    iv_enabledbaselineidentifier = iv_enabled_baseline_identifier
  ).

  DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

  " Wait for operation to complete
  DATA lv_status TYPE /aws1/cttbaselineopstatus.
  DO 100 TIMES.
    lv_status = get_baseline_operation(
      io_ctt = io_ctt
      iv_operation_id = lv_operation_id
    ).

    DATA(lv_msg) = |Baseline operation status: { lv_status }|.
    MESSAGE lv_msg TYPE 'I'.

    IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
      EXIT.
    ENDIF.

    " Wait 30 seconds
    WAIT UP TO 30 SECONDS.
  ENDDO.

  ov_operation_id = lv_operation_id.
  MESSAGE 'Baseline disabled successfully.' TYPE 'I'.
  CATCH /aws1/cx_cttconflictexception INTO DATA(lo_conflict).
  " Log conflict but don't fail - return empty operation ID
  DATA(lv_msg2) = |Conflict disabling baseline: { lo_conflict->get_text( ) }.
  Skipping disable step.|.
  MESSAGE lv_msg2 TYPE 'I'.
```

```
CLEAR ov_operation_id.  
ENDTRY.
```

- For API details, see [DisableBaseline](#) in *AWS SDK for SAP ABAP API reference*.

DisableControl

The following code example shows how to use `DisableControl`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Disable the control  
DATA(lo_output) = io_ctt->disablecontrol(  
  iv_controlidentifier = iv_control_arn  
  iv_targetidentifier  = iv_target_identifier  
).  
  
DATA(lv_operation_id) = lo_output->get_operationidentifier( ).  
  
" Wait for operation to complete  
DATA lv_status TYPE /aws1/cttcontrolopstatus.  
DO 100 TIMES.  
  lv_status = get_control_operation(  
    io_ctt = io_ctt  
    iv_operation_id = lv_operation_id  
  ).  
  
  DATA(lv_msg) = |Control operation status: { lv_status }|.  
  MESSAGE lv_msg TYPE 'I'.  
  
  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.  
    EXIT.  
  ENDIF.  
  
" Wait 30 seconds
```

```

    WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Control disabled successfully.' TYPE 'I'.

```

- For API details, see [DisableControl](#) in *AWS SDK for SAP ABAP API reference*.

EnableBaseline

The following code example shows how to use EnableBaseline.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" Prepare parameters for enabling baseline
DATA lt_parameters TYPE /aws1/
cl_cttenbdbaselineparam=>tt_enabledbaselineparameters.

" Add Identity Center baseline parameter if provided
IF iv_identity_center_baseline IS NOT INITIAL.
  " Create a JSON document with the baseline ARN value
  DATA(lv_json) = |\{ "IdentityCenterEnabledBaselineArn":
"{ iv_identity_center_baseline }" \}|.
  DATA(lo_param) = NEW /aws1/cl_cttenbdbaselineparam(
    iv_key = 'IdentityCenterEnabledBaselineArn'
    io_value = /aws1/cl_rt_document=>from_json_str( lv_json )
  ).
  APPEND lo_param TO lt_parameters.
ENDIF.

" Enable the baseline
DATA(lo_output) = io_ctt->enablebaseline(
  iv_baselineidentifier = iv_baseline_identifier
  iv_baselineversion    = iv_baseline_version

```

```

    iv_targetidentifier = iv_target_identifier
    it_parameters       = lt_parameters
  ).

  DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

  " Wait for operation to complete
  DATA lv_status TYPE /aws1/cttbaselineopstatus.
  DO 100 TIMES.
    lv_status = get_baseline_operation(
      io_ctt = io_ctt
      iv_operation_id = lv_operation_id
    ).

    DATA(lv_msg) = |Baseline operation status: { lv_status }|.
    MESSAGE lv_msg TYPE 'I'.

    IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
      EXIT.
    ENDIF.

    " Wait 30 seconds
    WAIT UP TO 30 SECONDS.
  ENDDO.

  ov_enabled_baseline_arn = lo_output->get_arn( ).
  MESSAGE 'Baseline enabled successfully.' TYPE 'I'.

```

- For API details, see [EnableBaseline](#) in *AWS SDK for SAP ABAP API reference*.

EnableControl

The following code example shows how to use `EnableControl`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Enable the control
DATA(lo_output) = io_ctt->enablecontrol(
  iv_controlidentifier = iv_control_arn
  iv_targetidentifier  = iv_target_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttcontrolopstatus.
DO 100 TIMES.
  lv_status = get_control_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Control operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

" Wait 30 seconds
WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Control enabled successfully.' TYPE 'I'.
```

- For API details, see [EnableControl](#) in *AWS SDK for SAP ABAP API reference*.

GetBaselineOperation

The following code example shows how to use GetBaselineOperation.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA(lo_output) = io_ctt->getbaselineoperation(  
    iv_operationidentifier = iv_operation_id  
    ).  
  
ov_status = lo_output->get_baselineoperation( )->get_status( ).
```

- For API details, see [GetBaselineOperation](#) in *AWS SDK for SAP ABAP API reference*.

GetControlOperation

The following code example shows how to use GetControlOperation.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA(lo_output) = io_ctt->getcontroloperation(  
    iv_operationidentifier = iv_operation_id  
    ).  
  
ov_status = lo_output->get_controloperation( )->get_status( ).
```

- For API details, see [GetControlOperation](#) in *AWS SDK for SAP ABAP API reference*.

ListBaselines

The following code example shows how to use ListBaselines.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_baselines TYPE /aws1/cl_cttbaselinessummary=>tt_baselines.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listbaselines(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_baselines( ) TO lt_baselines.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_baselines = lt_baselines.
MESSAGE 'Listed baselines successfully.' TYPE 'I'.
```

- For API details, see [ListBaselines](#) in *AWS SDK for SAP ABAP API reference*.

ListEnabledBaselines

The following code example shows how to use ListEnabledBaselines.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_enabled_baselines TYPE /aws1/
cl_cttenbdbaselinesumm=>tt_enabledbaselines.
DATA lv_nexttoken TYPE /aws1/cttlstenbdbaselinesnex00.

" List all enabled baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listenabledbaselines(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_enabledbaselines( ) TO lt_enabled_baselines.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_enabled_baselines = lt_enabled_baselines.
MESSAGE 'Listed enabled baselines successfully.' TYPE 'I'.
```

- For API details, see [ListEnabledBaselines](#) in *AWS SDK for SAP ABAP API reference*.

ListEnabledControls

The following code example shows how to use ListEnabledControls.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_enabled_controls TYPE /aws1/
cl_cttenabledcontrolsumm=>tt_enabledcontrols.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all enabled controls using pagination
DO.
  DATA(lo_output) = io_ctt->listenabledcontrols(
    iv_targetidentifier = iv_target_identifier
    iv_nexttoken        = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_enabledcontrols( ) TO lt_enabled_controls.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_enabled_controls = lt_enabled_controls.
MESSAGE 'Listed enabled controls successfully.' TYPE 'I'.
```

- For API details, see [ListEnabledControls](#) in *AWS SDK for SAP ABAP API reference*.

ListLandingZones

The following code example shows how to use ListLandingZones.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_landing_zones TYPE /aws1/
cl_cttlandingzonessummary=>tt_landingzonessummaries.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all landing zones using pagination
DO.
  DATA(lo_output) = io_ctt->listlandingzones(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_landingzones( ) TO lt_landing_zones.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_landing_zones = lt_landing_zones.
MESSAGE 'Listed landing zones successfully.' TYPE 'I'.
```

- For API details, see [ListLandingZones](#) in *AWS SDK for SAP ABAP API reference*.

ResetEnabledBaseline

The following code example shows how to use ResetEnabledBaseline.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Reset the enabled baseline
DATA(lo_output) = io_ctt->resetenabledbaseline(
  iv_enabledbaselineidentifier = iv_enabled_baseline_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttbaselineopstatus.
DO 100 TIMES.
  lv_status = get_baseline_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Baseline operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

" Wait 30 seconds
WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Baseline reset successfully.' TYPE 'I'.
```

- For API details, see [ResetEnabledBaseline](#) in *AWS SDK for SAP ABAP API reference*.

Firehose examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Firehose.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

PutRecord

The following code example shows how to use PutRecord.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_record) = NEW /aws1/cl_frhrecord( iv_data = iv_data ).  
  
  DATA(lo_result) = lo_frh->putrecord(  
    iv_deliverystreamname = iv_deliv_stream_name  
    io_record              = lo_record ).  
  
  MESSAGE 'Record sent to Firehose delivery stream.' TYPE 'I'.  
CATCH /aws1/cx_frhresourcenotfoundex.  
  MESSAGE 'Delivery stream not found.' TYPE 'E'.  
CATCH /aws1/cx_frhinvalidargumentex.
```

```

    MESSAGE 'Invalid argument provided.' TYPE 'E'.
  CATCH /aws1/cx_frhserviceunavailex.
    MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
  ENDTRY.

```

- For API details, see [PutRecord](#) in *AWS SDK for SAP ABAP API reference*.

PutRecordBatch

The following code example shows how to use PutRecordBatch.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  DATA(lo_result) = lo_frh->putrecordbatch(
    iv_deliverystreamname = iv_deliv_stream_name
    it_records             = it_records ).

  DATA(lv_failed_count) = lo_result->get_failedputcount( ).

  IF lv_failed_count > 0.
    MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
  ELSE.
    MESSAGE 'All records sent successfully to Firehose delivery stream.' TYPE
'I'.
  ENDIF.
CATCH /aws1/cx_frhresourceindex.
  MESSAGE 'Delivery stream not found.' TYPE 'E'.
CATCH /aws1/cx_frhinvalidargumentex.
  MESSAGE 'Invalid argument provided.' TYPE 'E'.
CATCH /aws1/cx_frhserviceunavailex.
  MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutRecordBatch](#) in *AWS SDK for SAP ABAP API reference*.

DynamoDB examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with DynamoDB.

Basics are code examples that show you how to perform the essential operations within a service.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Basics](#)
- [Actions](#)

Basics

Learn the basics

The following code example shows how to:

- Create a table that can hold movie data.
- Put, get, and update a single movie in the table.
- Write movie data to the table from a sample JSON file.
- Query for movies that were released in a given year.
- Scan for movies that were released in a range of years.
- Delete a movie from the table, then delete the table.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" 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' }| ) ) )
            ) ).
    TRY.

```

```

        lo_resp_putitem = lo_dyn->putitem(
            iv_tablename = iv_table_name
            it_item       = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Speed' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
            ) ).
" TYPE REF TO /AWSEX/CL_AWS1_dyn_PUT_ITEM_OUTPUT
MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
CATCH /aws1/cx_dyncondalcheckfaile00.
MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
CATCH /aws1/cx_dynresourcenotfoundex.
MESSAGE 'The table or index does not exist' TYPE 'E'.
CATCH /aws1/cx_dyntransactconflictex.
MESSAGE 'Another transaction is using the item' TYPE 'E'.
ENDTRY.

" Get item from table.
TRY.
    DATA(lo_resp_getitem) = lo_dyn->getitem(
        iv_tablename       = iv_table_name
        it_key              = VALUE /aws1/cl_dynattributevalue=>tt_key(
            ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
                key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
                key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
'1975' ) ) )
            ) ).
    DATA(lt_attr) = lo_resp_getitem->get_item( ).
    DATA(lo_title) = lt_attr[ key = 'title' ]-value.
    DATA(lo_year) = lt_attr[ key = 'year' ]-value.
    DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
    MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
    MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
    MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.

```

```

CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Query item from table.
TRY.
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
  ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
  DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
  ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
  key = 'year'
  value = NEW /aws1/cl_dyncondition(
  it_attributelist = lt_attributelist
  iv_comparisonoperator = |EQ|
  ) ) ) ).
  DATA(lo_query_result) = lo_dyn->query(
  iv_tablename = iv_table_name
  it_keyconditions = lt_keyconditions ).
  DATA(lt_items) = lo_query_result->get_items( ).
  READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Scan items from table.
TRY.
  DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
  lt_items = lo_scan_result->get_items( ).
  " Read the first item and display the attributes.
  READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.

```

```

        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDRY.

" 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'.
    ENDRY.

" 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'.
    ENDRY.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)
 - [Scan](#)
 - [UpdateItem](#)

Actions

CreateTable

The following code example shows how to use CreateTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.

```

- For API details, see [CreateTable](#) in *AWS SDK for SAP ABAP API reference*.

DeleteItem

The following code example shows how to use DeleteItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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.

```

- For API details, see [DeleteItem](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTable

The following code example shows how to use DeleteTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [DeleteTable](#) in *AWS SDK for SAP ABAP API reference*.

DescribeTable

The following code example shows how to use DescribeTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DescribeTable](#) in *AWS SDK for SAP ABAP API reference*.

GetItem

The following code example shows how to use GetItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_item = lo_dyn->getitem(  
    iv_tablename          = iv_table_name  
    it_key                = it_key ).  
  DATA(lt_attr) = oo_item->get_item( ).  
  DATA(lo_title) = lt_attr[ key = 'title' ]-value.  
  DATA(lo_year) = lt_attr[ key = 'year' ]-value.  
  DATA(lo_rating) = lt_attr[ key = 'rating' ]-value.  
  MESSAGE 'Movie name is: ' && lo_title->get_s( )  
    && 'Movie year is: ' && lo_year->get_n( )  
    && 'Moving rating is: ' && lo_rating->get_n( ) TYPE 'I'.  
  CATCH /aws1/cx_dynresourcenotfoundex.  
    MESSAGE 'The table or index does not exist' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetItem](#) in *AWS SDK for SAP ABAP API reference*.

ListTables

The following code example shows how to use ListTables.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```

oo_result = lo_dyn->listtables( ).
" You can loop over the oo_result to get table properties like this.
LOOP AT oo_result->get_tablenames( ) INTO DATA(lo_table_name).
  DATA(lv_tablename) = lo_table_name->get_value( ).
ENDLOOP.
DATA(lv_tablecount) = lines( oo_result->get_tablenames( ) ).
MESSAGE 'Found ' && lv_tablecount && ' tables' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- For API details, see [ListTables](#) in *AWS SDK for SAP ABAP API reference*.

PutItem

The following code example shows how to use PutItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [PutItem](#) in *AWS SDK for SAP ABAP API reference*.

Query

The following code example shows how to use Query.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.
  ENDLLOOP.
  DATA(lv_count) = oo_result->get_count( ).
  MESSAGE 'Item count is: ' && lv_count TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.

```

```
ENDTRY.
```

- For API details, see [Query](#) in *AWS SDK for SAP ABAP API reference*.

Scan

The following code example shows how to use Scan.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
  ENDLLOOP.
  DATA(lv_count) = oo_scan_result->get_count( ).
  MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
```

```
ENDTRY.
```

- For API details, see [Scan](#) in *AWS SDK for SAP ABAP API reference*.

UpdateItem

The following code example shows how to use UpdateItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [UpdateItem](#) in *AWS SDK for SAP ABAP API reference*.

Amazon EC2 examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon EC2.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AllocateAddress

The following code example shows how to use `AllocateAddress`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [AllocateAddress](#) in *AWS SDK for SAP ABAP API reference*.

AssociateAddress

The following code example shows how to use `AssociateAddress`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [AssociateAddress](#) in *AWS SDK for SAP ABAP API reference*.

AuthorizeSecurityGroupIngress

The following code example shows how to use AuthorizeSecurityGroupIngress.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" Create IP permissions for SSH access (port 22)
" iv_cidr_ip = '192.0.2.0/24'
DATA lt_ip_permissions TYPE /aws1/cl_ec2ippermission=>tt_ippermissionlist.
DATA(lo_ip_permission) = NEW /aws1/cl_ec2ippermission(

```

```

    iv_ipprotocol = 'tcp'
    iv_fromport = 22
    iv_toport = 22
    it_ipranges = VALUE /aws1/cl_ec2iprange=>tt_iprangelist(
      ( NEW /aws1/cl_ec2iprange( iv_cidrip = iv_cidr_ip ) )
    )
  ).
  APPEND lo_ip_permission TO lt_ip_permissions.

  TRY.
    oo_result = lo_ec2->authsecuritygroupingress(          " oo_result is
returned for testing purposes. "
    iv_groupid = iv_group_id
    it_ippermissions = lt_ip_permissions ).
    MESSAGE 'Authorized ingress rule for security group.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
  ENDTRY.

```

- For API details, see [AuthorizeSecurityGroupIngress](#) in *AWS SDK for SAP ABAP API reference*.

CreateKeyPair

The following code example shows how to use CreateKeyPair.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

  TRY.
    oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
      " oo_result is returned for testing purposes. "
    MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).

```

```

        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- For API details, see [CreateKeyPair](#) in *AWS SDK for SAP ABAP API reference*.

CreateSecurityGroup

The following code example shows how to use CreateSecurityGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
        oo_result = lo_ec2->createsecuritygroup(
            iv_description = 'Security group example'
            iv_groupname = iv_security_group_name
            iv_vpcid = iv_vpc_id ).
        MESSAGE 'Security group created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- For API details, see [CreateSecurityGroup](#) in *AWS SDK for SAP ABAP API reference*.

CreateVpc

The following code example shows how to use CreateVpc.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_cidr_block = '10.0.0.0/16'
TRY.
    oo_result = lo_ec2->createvpc( iv_cidrblock = iv_cidr_block ).
oo_result is returned for testing purposes. "
    DATA(lv_vpc_id) = oo_result->get_vpc( )->get_vpcid( ).
    MESSAGE 'Created VPC.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [CreateVpc](#) in *AWS SDK for SAP ABAP API reference*.

CreateVpcEndpoint

The following code example shows how to use CreateVpcEndpoint.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_vpc_id = 'vpc-abc123'
" iv_service_name = 'com.amazonaws.region.service'
TRY.
    oo_result = lo_ec2->createvpcendpoint(
for testing purposes. " " oo_result is returned
```

```

        iv_vpcid = iv_vpc_id
        iv_servicename = iv_service_name
        it_routetableids = it_route_table_ids ).
    DATA(lv_vpc_endpoint_id) = oo_result->get_vpcendpoint( )-
>get_vpcendpointid( ).
    MESSAGE 'Created VPC endpoint.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateVpcEndpoint](#) in *AWS SDK for SAP ABAP API reference*.

DeleteKeyPair

The following code example shows how to use DeleteKeyPair.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [DeleteKeyPair](#) in *AWS SDK for SAP ABAP API reference*.

DeleteSecurityGroup

The following code example shows how to use DeleteSecurityGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ec2->deletesecuritygroup( iv_groupid = iv_security_group_id ).  
  MESSAGE 'Security group deleted.' TYPE 'I'.  
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
  MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteSecurityGroup](#) in *AWS SDK for SAP ABAP API reference*.

DeleteVpc

The following code example shows how to use DeleteVpc.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ec2->deletevpc( iv_vpcid = iv_vpc_id ).  
  MESSAGE 'Deleted VPC.' 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.
```

- For API details, see [DeleteVpc](#) in *AWS SDK for SAP ABAP API reference*.

DeleteVpcEndpoints

The following code example shows how to use DeleteVpcEndpoints.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_ec2->deletevpcendpoints( it_vpcendpointids = it_vpc_endpoint_ids ).
  MESSAGE 'Deleted VPC endpoint(s).' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteVpcEndpoints](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAddresses

The following code example shows how to use DescribeAddresses.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [DescribeAddresses](#) in *AWS SDK for SAP ABAP API reference*.

DescribeAvailabilityZones

The following code example shows how to use DescribeAvailabilityZones.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [DescribeAvailabilityZones](#) in *AWS SDK for SAP ABAP API reference*.

DescribeImages

The following code example shows how to use DescribeImages.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
      oo_result = lo_ec2->describeimages( it_imageids = it_image_ids ).
      " oo_result is returned for testing purposes. "
      DATA(lt_images) = oo_result->get_images( ).
      MESSAGE 'Retrieved information about Amazon Machine Images (AMIs).' TYPE
'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
      ENDTRY.

```

- For API details, see [DescribeImages](#) in *AWS SDK for SAP ABAP API reference*.

DescribeInstanceTypes

The following code example shows how to use DescribeInstanceTypes.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Create filters for architecture and instance type patterns
" iv_architecture = 'x86_64'
DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
APPEND NEW /aws1/cl_ec2filter(
  iv_name = 'processor-info.supported-architecture'
  it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
    ( NEW /aws1/cl_ec2valuestringlist_w( iv_architecture ) )
  )
) TO lt_filters.
" Filter for instance type patterns like '*.micro', '*.small'
APPEND NEW /aws1/cl_ec2filter(
  iv_name = 'instance-type'
  it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
    ( NEW /aws1/cl_ec2valuestringlist_w( '*.micro' ) )
    ( NEW /aws1/cl_ec2valuestringlist_w( '*.small' ) )
  )
) TO lt_filters.

TRY.
  oo_result = lo_ec2->describeinstancetype( it_filters = lt_filters ).
  " oo_result is returned for testing purposes. "
  DATA(lt_instance_types) = oo_result->get_instancetypes( ).
  MESSAGE 'Retrieved information about EC2 instance types.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DescribeInstanceTypes](#) in *AWS SDK for SAP ABAP API reference*.

DescribeInstances

The following code example shows how to use DescribeInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.
    ENDLLOOP.
    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.

```

- For API details, see [DescribeInstances](#) in *AWS SDK for SAP ABAP API reference*.

DescribeKeyPairs

The following code example shows how to use DescribeKeyPairs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DescribeKeyPairs](#) in *AWS SDK for SAP ABAP API reference*.

DescribeRegions

The following code example shows how to use DescribeRegions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.

```

- For API details, see [DescribeRegions](#) in *AWS SDK for SAP ABAP API reference*.

DescribeRouteTables

The following code example shows how to use DescribeRouteTables.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" Create filter for VPC ID
" iv_vpc_id = 'vpc-abc123'
DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
APPEND NEW /aws1/cl_ec2filter(
  iv_name = 'vpc-id'
  it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
    ( NEW /aws1/cl_ec2valuestringlist_w( iv_vpc_id ) )
  )
) TO lt_filters.

TRY.
  oo_result = lo_ec2->describeroutetables( it_filters = lt_filters ).
" oo_result is returned for testing purposes. "
  DATA(lt_route_tables) = oo_result->get_routetables( ).
  MESSAGE 'Retrieved information about route tables.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.

```

```
ENDTRY.
```

- For API details, see [DescribeRouteTables](#) in *AWS SDK for SAP ABAP API reference*.

DescribeSecurityGroups

The following code example shows how to use DescribeSecurityGroups.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DescribeSecurityGroups](#) in *AWS SDK for SAP ABAP API reference*.

MonitorInstances

The following code example shows how to use MonitorInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [MonitorInstances](#) in *AWS SDK for SAP ABAP API reference*.

RebootInstances

The following code example shows how to use RebootInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.

```

- For API details, see [RebootInstances](#) in *AWS SDK for SAP ABAP API reference*.

ReleaseAddress

The following code example shows how to use ReleaseAddress.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [ReleaseAddress](#) in *AWS SDK for SAP ABAP API reference*.

RunInstances

The following code example shows how to use RunInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" Create tags for resource created during instance launch. "
DATA lt_tag specifications TYPE /aws1/
cl_ec2tag specifications=>tt_tag specification list.
DATA ls_tag specifications LIKE LINE OF lt_tag specifications.
ls_tag specifications = NEW /aws1/cl_ec2tag specification(
  iv_resourcetype = 'instance'
  it_tags = VALUE /aws1/cl_ec2tag=>tt_tag list(
    (NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
  ) ).
APPEND ls_tag specifications TO lt_tag specifications.

TRY.
  " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
  oo_result = lo_ec2->runinstances(                                " oo_result is
returned for testing purposes. "
  iv_imageid = iv_ami_id
  iv_instancetype = 't3.micro'
  iv_maxcount = 1
  iv_mincount = 1
  it_tag specifications = lt_tag specifications
  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.

```

- For API details, see [RunInstances](#) in *AWS SDK for SAP ABAP API reference*.

StartInstances

The following code example shows how to use StartInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [StartInstances](#) in *AWS SDK for SAP ABAP API reference*.

StopInstances

The following code example shows how to use StopInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

        DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
        APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
lt_instance_ids.

        "Perform dry run"
        TRY.
            " DryRun is set to true. This checks for the required permissions to stop
the instance without actually making the request. "
            lo_ec2->stopinstances(
                it_instanceids = lt_instance_ids
                iv_dryrun = abap_true ).
        CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
            " If the error code returned is `DryRunOperation`, then you have the
required permissions to stop this instance. "
            IF lo_exception->av_err_code = 'DryRunOperation'.
                MESSAGE 'Dry run to stop instance completed.' TYPE 'I'.
                " DryRun is set to false to stop instance. "
                oo_result = lo_ec2->stopinstances(           " oo_result is returned for
testing purposes. "

```

```
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_false ).
    MESSAGE 'Successfully stopped the EC2 instance.' TYPE 'I'.
    " If the error code returned is `UnauthorizedOperation`, then you don't
    have the required permissions to stop this instance. "
    ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
        MESSAGE 'Dry run to stop instance failed. User does not have permissions
to stop the instance.' TYPE 'E'.
    ELSE.
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDIF.
ENDTRY.
```

- For API details, see [StopInstances](#) in *AWS SDK for SAP ABAP API reference*.

Amazon ECR examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon ECR.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateRepository

The following code example shows how to use `CreateRepository`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_repository_name = 'my-repository'
  oo_result = lo_ecr->createrepository(
    iv_repositoryname = iv_repository_name ).
  DATA(lv_repository_uri) = oo_result->get_repository( )-
>get_repositoryuri( ).
  MESSAGE |Repository created with URI: { lv_repository_uri }| TYPE 'I'.
CATCH /aws1/cx_ecrrepositoryalrexex.
  " If repository already exists, retrieve it
  DATA lt_repo_names TYPE /aws1/
cl_ecrrepositorynames00=>tt_repositorynamelist.
  APPEND NEW /aws1/cl_ecrrepositorynames00( iv_value = iv_repository_name )
TO lt_repo_names.
  DATA(lo_describe_result) = lo_ecr->describerepositories( it_repositorynames
= lt_repo_names ).
  DATA(lt_repos) = lo_describe_result->get_repositories( ).
  IF lines( lt_repos ) > 0.
    READ TABLE lt_repos INDEX 1 INTO DATA(lo_repo).
    oo_result = NEW /aws1/cl_ecrcrrepositoryrsp( io_repository = lo_repo ).
    MESSAGE |Repository { iv_repository_name } already exists.| TYPE 'I'.
  ENDIF.
ENDTRY.

```

- For API details, see [CreateRepository](#) in *AWS SDK for SAP ABAP API reference*.

DeleteRepository

The following code example shows how to use DeleteRepository.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_repository_name = 'my-repository'
  lo_ecr->deleterepository(
    iv_repositoryname = iv_repository_name
    iv_force = abap_true ).
  MESSAGE |Repository { iv_repository_name } deleted.| TYPE 'I'.
CATCH /aws1/cx_ecrrepositorynotfound.
  MESSAGE 'Repository not found.' TYPE 'I'.
ENDTRY.
```

- For API details, see [DeleteRepository](#) in *AWS SDK for SAP ABAP API reference*.

DescribeImages

The following code example shows how to use DescribeImages.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_repository_name = 'my-repository'
  " it_image_ids = VALUE #( ( NEW /aws1/cl_ecrimageidentifier( iv_imagetag =
'latest' ) ) )
  IF it_image_ids IS NOT INITIAL.
    oo_result = lo_ecr->describeimages(
```

```

        iv_repositoryname = iv_repository_name
        it_imageids = it_image_ids ).
    ELSE.
        oo_result = lo_ecr->describeimages(
            iv_repositoryname = iv_repository_name ).
    ENDIF.
    DATA(lt_image_details) = oo_result->get_imagedetails( ).
    MESSAGE |Found { lines( lt_image_details ) } images in repository.| TYPE
'I'.
    CATCH /aws1/cx_ecrrepositorynotfoundex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    CATCH /aws1/cx_ecrimagenotfoundex.
        MESSAGE 'Image not found.' TYPE 'I'.
    CATCH /aws1/cx_ecrinvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'I'.
    ENDTRY.

```

- For API details, see [DescribeImages](#) in *AWS SDK for SAP ABAP API reference*.

DescribeRepositories

The following code example shows how to use DescribeRepositories.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
        " it_repository_names = VALUE #( ( NEW /aws1/
cl_ecrrepositorynames00( iv_value = 'my-repository' ) ) )
        oo_result = lo_ecr->describerepositories(
            it_repositorynames = it_repository_names ).
        DATA(lt_repositories) = oo_result->get_repositories( ).
        MESSAGE |Found { lines( lt_repositories ) } repositories.| TYPE 'I'.
    CATCH /aws1/cx_ecrrepositorynotfoundex.
        MESSAGE 'Repository not found.' TYPE 'I'.

```

```
ENDTRY.
```

- For API details, see [DescribeRepositories](#) in *AWS SDK for SAP ABAP API reference*.

GetAuthorizationToken

The following code example shows how to use `GetAuthorizationToken`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_ecr->getauthorizationtoken( ).  
  DATA(lt_auth_data) = oo_result->get_authorizationdata( ).  
  IF lines( lt_auth_data ) > 0.  
    READ TABLE lt_auth_data INDEX 1 INTO DATA(lo_auth_data).  
    DATA(lv_token) = lo_auth_data->get_authorizationtoken( ).  
    MESSAGE 'Authorization token retrieved.' TYPE 'I'.  
  ENDIF.  
  CATCH /aws1/cx_ecrserverexception.  
    MESSAGE 'Server exception occurred.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [GetAuthorizationToken](#) in *AWS SDK for SAP ABAP API reference*.

GetRepositoryPolicy

The following code example shows how to use `GetRepositoryPolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_repository_name = 'my-repository'
  oo_result = lo_ecr->getrepositorypolicy(
    iv_repositoryname = iv_repository_name ).
  DATA(lv_policy_text) = oo_result->get_policytext( ).
  MESSAGE 'Repository policy retrieved.' TYPE 'I'.
CATCH /aws1/cx_ecrrepositorynotfound.
  MESSAGE 'Repository not found.' TYPE 'I'.
CATCH /aws1/cx_ecrrepositoryplynot00.
  MESSAGE 'Repository policy not found.' TYPE 'I'.
ENDTRY.
```

- For API details, see [GetRepositoryPolicy](#) in *AWS SDK for SAP ABAP API reference*.

PutLifecyclePolicy

The following code example shows how to use PutLifecyclePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_repository_name = 'my-repository'
  " iv_lifecycle_policy_text = '{"rules":
[{"rulePriority":1,"description":"Expire images older than 14 days",...}]}'
```

```

lo_ecr->putlifecyclepolicy(
  iv_repositoryname = iv_repository_name
  iv_lifecyclepolicytext = iv_lifecycle_policy_text ).
MESSAGE |Lifecycle policy set for repository { iv_repository_name }.| TYPE
'I'.
CATCH /aws1/cx_ecrrepositorynotfound.
MESSAGE 'Repository not found.' TYPE 'I'.
CATCH /aws1/cx_ecrvalidationex.
MESSAGE 'Invalid lifecycle policy format.' TYPE 'I'.
ENDTRY.

```

- For API details, see [PutLifecyclePolicy](#) in *AWS SDK for SAP ABAP API reference*.

SetRepositoryPolicy

The following code example shows how to use SetRepositoryPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_repository_name = 'my-repository'
  " iv_policy_text = '{"Version":"2012-10-17",          "Statement":[...]}'
  lo_ecr->setrepositorypolicy(
    iv_repositoryname = iv_repository_name
    iv_policytext = iv_policy_text ).
  MESSAGE |Policy set for repository { iv_repository_name }.| TYPE 'I'.
  CATCH /aws1/cx_ecrrepositorynotfound.
  MESSAGE 'Repository not found.' TYPE 'I'.
ENDTRY.

```

- For API details, see [SetRepositoryPolicy](#) in *AWS SDK for SAP ABAP API reference*.

Amazon EMR examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon EMR.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AddJobFlowSteps

The following code example shows how to use AddJobFlowSteps.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Build args list for Spark submit
  DATA lt_args TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( 'spark-submit' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( '--deploy-mode' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( 'cluster' ) TO lt_args.
  APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_script_uri ) TO lt_args.
  APPEND LINES OF it_script_args TO lt_args.

  " Create step configuration
  DATA(lo_hadoop_jar_step) = NEW /aws1/cl_emrhadoopjarstepcfg(
    iv_jar = 'command-runner.jar'
```

```

        it_args = lt_args
    ).

    DATA(lo_step_config) = NEW /aws1/cl_emrstepconfig(
        iv_name = iv_name
        iv_actiononfailure = 'CONTINUE'
        io_hadoopjarstep = lo_hadoop_jar_step
    ).

    DATA lt_steps TYPE /aws1/cl_emrstepconfig=>tt_stepconfiglist.
    APPEND lo_step_config TO lt_steps.

    DATA(lo_result) = lo_emr->addjobflowsteps(
        iv_jobflowid = iv_cluster_id
        it_steps = lt_steps
    ).

    " Get first step ID
    DATA(lt_step_ids) = lo_result->get_stepids( ).
    READ TABLE lt_step_ids INDEX 1 INTO DATA(lo_step_id_obj).
    IF sy-subrc = 0.
        ov_step_id = lo_step_id_obj->get_value( ).
        MESSAGE |Step added with ID { ov_step_id }| TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- For API details, see [AddJobFlowSteps](#) in *AWS SDK for SAP ABAP API reference*.

DescribeCluster

The following code example shows how to use DescribeCluster.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_emr->describecluster(
    iv_clusterid = iv_cluster_id
  ).
  DATA(lo_cluster) = oo_result->get_cluster( ).
  DATA(lv_cluster_name) = lo_cluster->get_name( ).
  MESSAGE |Retrieved cluster information for { lv_cluster_name }| TYPE 'I'.
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
  lv_error = lo_invalid_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DescribeCluster](#) in *AWS SDK for SAP ABAP API reference*.

DescribeStep

The following code example shows how to use DescribeStep.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_emr->describestep(
    iv_clusterid = iv_cluster_id
    iv_stepid = iv_step_id
  ).
  DATA(lo_step) = oo_result->get_step( ).
  DATA(lv_step_name) = lo_step->get_name( ).
  MESSAGE |Retrieved step information for { lv_step_name }| TYPE 'I'.
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
```

```
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
  lv_error = lo_invalid_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DescribeStep](#) in *AWS SDK for SAP ABAP API reference*.

ListSteps

The following code example shows how to use ListSteps.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_emr->liststeps(
    iv_clusterid = iv_cluster_id
  ).
  DATA(lt_steps) = oo_result->get_steps( ).
  DATA(lv_step_count) = lines( lt_steps ).
  MESSAGE |Retrieved { lv_step_count } steps for cluster| TYPE 'I'.
CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
  lv_error = lo_invalid_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [ListSteps](#) in *AWS SDK for SAP ABAP API reference*.

RunJobFlow

The following code example shows how to use RunJobFlow.

SDK for SAP ABAP**Note**

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create instances configuration
  DATA(lo_instances) = NEW /aws1/cl_emrjobflowinstsconfig(
    iv_masterinstancetype = 'm5.xlarge'
    iv_slaveinstancetype = 'm5.xlarge'
    iv_instancecount = 3
    iv_keepjobflowalivewhennos00 = iv_keep_alive
    iv_emrmanagedmastersecgroup = iv_primary_sec_grp
    iv_emrmanagedslavesecgroup = iv_secondary_sec_grp
  ).

  DATA(lo_result) = lo_emr->runjobflow(
    iv_name = iv_name
    iv_loguri = iv_log_uri
    iv_releaselabel = 'emr-5.30.1'
    io_instances = lo_instances
    it_steps = it_steps
    it_applications = it_applications
    iv_jobflowrole = iv_job_flow_role
    iv_servicerole = iv_service_role
    iv_ebsrootvolumesize = 10
    iv_visibletoallusers = abap_true
  ).

  ov_cluster_id = lo_result->get_jobflowid( ).
  MESSAGE 'EMR cluster created successfully.' TYPE 'I'.
CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
CATCH /aws1/cx_emrclientexc INTO DATA(lo_client_error).
  lv_error = lo_client_error->if_message~get_text( ).
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [RunJobFlow](#) in *AWS SDK for SAP ABAP API reference*.

TerminateJobFlows

The following code example shows how to use TerminateJobFlows.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA lt_cluster_ids TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.  
  APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_cluster_id ) TO lt_cluster_ids.  
  
  lo_emr->terminatejobflows(  
    it_jobflowids = lt_cluster_ids  
  ).  
  MESSAGE 'EMR cluster terminated successfully.' TYPE 'I'.  
CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).  
  DATA(lv_error) = lo_internal_error->if_message~get_text( ).  
  MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [TerminateJobFlows](#) in *AWS SDK for SAP ABAP API reference*.

EventBridge Scheduler examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with EventBridge Scheduler.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateSchedule

The following code example shows how to use CreateSchedule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Constants for time calculations
  DATA lv_start_date TYPE /aws1/scdstartdate.
  DATA lv_end_date TYPE /aws1/scdenndate.
  DATA lv_start_timestamp TYPE timestamp.
  DATA lv_end_timestamp TYPE timestamp.
  DATA lv_hours_to_run TYPE i VALUE 1.

  " Get current timestamp
  GET TIME STAMP FIELD lv_start_timestamp.

  " Add 1 hour to the current timestamp using CL_ABAP_TSTMP
  lv_end_timestamp = cl_abap_tstmp=>add(
    tstmp = lv_start_timestamp
    secs = lv_hours_to_run * 3600 ).

  " Convert timestamps to decimal format for AWS API
  lv_start_date = lv_start_timestamp.
  lv_end_date = lv_end_timestamp.

  " Prepare flexible time window configuration
  DATA lo_flexible_time_window TYPE REF TO /aws1/cl_scdflexibletimewindow.
  IF iv_use_flexible_time_win = abap_true.
    " iv_use_flexible_time_win = ABAP_TRUE
```

```

" Example: Set MaximumWindowInMinutes to 10 for flexible window
lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
  iv_mode = 'FLEXIBLE'
  iv_maximumwindowinminutes = 10 ).
ELSE.
  lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
    iv_mode = 'OFF' ).
ENDIF.

" Prepare target configuration
" Example iv_target_arn = 'arn:aws:sqs:us-east-1:123456789012:my-queue'
" Example iv_role_arn = 'arn:aws:iam::123456789012:role/SchedulerRole'
" Example iv_input = '{"message": "Hello from EventBridge Scheduler"}'
DATA(lo_target) = NEW /aws1/cl_scdtarget(
  iv_arn = iv_target_arn
  iv_rolearn = iv_role_arn
  iv_input = iv_input ).

" Set action after completion if needed
DATA lv_action_after_completion TYPE /aws1/scdactionaftercompletion.
IF iv_delete_after_completion = abap_true.
  " iv_delete_after_completion = ABAP_TRUE
  lv_action_after_completion = 'DELETE'.
ELSE.
  lv_action_after_completion = 'NONE'.
ENDIF.

" Create the schedule
" Example iv_name = 'my-schedule'
" Example iv_schedule_expression = 'rate(15 minutes)'
" Example iv_schedule_group_name = 'my-schedule-group'
DATA(lo_result) = lo_scd->createschedule(
  iv_name = iv_name
  iv_scheduleexpression = iv_schedule_expression
  iv_groupname = iv_schedule_group_name
  io_target = lo_target
  io_flexibletimewindow = lo_flexible_time_window
  iv_startdate = lv_start_date
  iv_enddate = lv_end_date
  iv_actionaftercompletion = lv_action_after_completion ).

ov_schedule_arn = lo_result->get_scheduledarn( ).
MESSAGE 'Schedule created successfully.' TYPE 'I'.

```

```

    CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
      DATA(lv_error) = |Conflict creating schedule: { lo_conflict_ex-
>if_message~get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
      DATA(lv_generic_error) = |Error creating schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
      MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.

```

- For API details, see [CreateSchedule](#) in *AWS SDK for SAP ABAP API reference*.

CreateScheduleGroup

The following code example shows how to use CreateScheduleGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Example iv_name = 'my-schedule-group'
  DATA(lo_result) = lo_scd->createschedulegroup(
    iv_name = iv_name ).

  ov_schedule_group_arn = lo_result->get_schedulegrouparn( ).
  MESSAGE 'Schedule group created successfully.' TYPE 'I'.

  CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
    DATA(lv_error) = |Conflict creating schedule group: { lo_conflict_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
    DATA(lv_generic_error) = |Error creating schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_generic_error TYPE 'I'.
  ENDTRY.

```

- For API details, see [CreateScheduleGroup](#) in *AWS SDK for SAP ABAP API reference*.

DeleteSchedule

The following code example shows how to use DeleteSchedule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Example iv_name = 'my-schedule'
  " Example iv_schedule_group_name = 'my-schedule-group'
  lo_scd->deleteschedule(
    iv_name = iv_name
    iv_groupname = iv_schedule_group_name ).
  MESSAGE 'Schedule deleted successfully.' TYPE 'I'.

  CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
    DATA(lv_error) = |Schedule not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
    DATA(lv_generic_error) = |Error deleting schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
    MESSAGE lv_generic_error TYPE 'I'.
ENDTRY.
```

- For API details, see [DeleteSchedule](#) in *AWS SDK for SAP ABAP API reference*.

DeleteScheduleGroup

The following code example shows how to use DeleteScheduleGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Example iv_name = 'my-schedule-group'
  lo_scd->deleteschedulegroup(
    iv_name = iv_name ).
  MESSAGE 'Schedule group deleted successfully.' TYPE 'I'.

  CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
  DATA(lv_error) = |Schedule group not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
  DATA(lv_generic_error) = |Error deleting schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
  MESSAGE lv_generic_error TYPE 'I'.
ENDTRY.
```

- For API details, see [DeleteScheduleGroup](#) in *AWS SDK for SAP ABAP API reference*.

AWS Glue examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS Glue.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateCrawler

The following code example shows how to use CreateCrawler.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_crawler_name = 'my-crawler'
  " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
  " iv_database_name = 'my-database'
  " iv_table_prefix = 'test_'
  " iv_s3_target = 's3://example-bucket/data/'

  DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
    ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

  DATA(lo_targets) = NEW /aws1/cl_gluclawlerltargets(
    it_s3targets = lt_s3_targets ).

  lo_glu->createcrawler(
    iv_name = iv_crawler_name
    iv_role = iv_role_arn
    iv_databasename = iv_database_name
    iv_tableprefix = iv_table_prefix
    io_targets = lo_targets ).
  MESSAGE 'Crawler created successfully.' TYPE 'I'.
CATCH /aws1/cx_glualreadyexistsex.
  MESSAGE 'Crawler already exists.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
```

```

CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
  DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
  MESSAGE lv_limit_error TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateCrawler](#) in *AWS SDK for SAP ABAP API reference*.

CreateJob

The following code example shows how to use CreateJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_job_name = 'my-etl-job'
  " iv_description = 'ETL job for data transformation'
  " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
  " iv_script_location = 's3://example-bucket/scripts/my-script.py'

  DATA(lo_command) = NEW /aws1/cl_glujobcommand(
    iv_name = 'glueetl'
    iv_scriptlocation = iv_script_location
    iv_pythonversion = '3' ).

  lo_glu->createjob(
    iv_name = iv_job_name
    iv_description = iv_description
    iv_role = iv_role_arn
    io_command = lo_command
    iv_glueversion = '3.0' ).
  MESSAGE 'Job created successfully.' TYPE 'I'.
CATCH /aws1/cx_glualreadyexistsex.
  MESSAGE 'Job already exists.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).

```

```

    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
  CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
    DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
    MESSAGE lv_limit_error TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateJob](#) in *AWS SDK for SAP ABAP API reference*.

DeleteCrawler

The following code example shows how to use DeleteCrawler.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_crawler_name = 'my-crawler'
  lo_glu->deletecrawler( iv_name = iv_crawler_name ).
  MESSAGE 'Crawler deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_glucrawlerrunningex.
    MESSAGE 'Crawler is currently running.' TYPE 'E'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
  CATCH /aws1/cx_gluschdrtransingex.
    MESSAGE 'Scheduler is transitioning.' TYPE 'E'.

```

```
ENDTRY.
```

- For API details, see [DeleteCrawler](#) in *AWS SDK for SAP ABAP API reference*.

DeleteDatabase

The following code example shows how to use DeleteDatabase.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_database_name = 'my-database'  
    lo_glu->deletedatabase( iv_name = iv_database_name ).  
    MESSAGE 'Database deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_gluentitynotfoundex.  
    MESSAGE 'Database does not exist.' TYPE 'E'.  
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).  
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).  
    MESSAGE lv_invalid_error TYPE 'E'.  
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).  
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).  
    MESSAGE lv_internal_error TYPE 'E'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
    MESSAGE lv_timeout_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteDatabase](#) in *AWS SDK for SAP ABAP API reference*.

DeleteJob

The following code example shows how to use DeleteJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " iv_job_name = 'my-etl-job'  
  lo_glu->deletejob( iv_jobname = iv_job_name ).  
  MESSAGE 'Job deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).  
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).  
  MESSAGE lv_invalid_error TYPE 'E'.  
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).  
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).  
  MESSAGE lv_internal_error TYPE 'E'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteJob](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTable

The following code example shows how to use DeleteTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```

" iv_database_name = 'my-database'
" iv_table_name = 'my-table'
lo_glu->deletetable(
  iv_databasename = iv_database_name
  iv_name = iv_table_name ).
MESSAGE 'Table deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Table or database does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.

```

- For API details, see [DeleteTable](#) in *AWS SDK for SAP ABAP API reference*.

GetCrawler

The following code example shows how to use GetCrawler.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
" iv_crawler_name = 'my-crawler'
oo_result = lo_glu->getcrawler( iv_name = iv_crawler_name ).
DATA(lo_crawler) = oo_result->get_crawler( ).
MESSAGE 'Crawler information retrieved.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Crawler does not exist.' TYPE 'I'.

```

```
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- For API details, see [GetCrawler](#) in *AWS SDK for SAP ABAP API reference*.

GetDatabase

The following code example shows how to use GetDatabase.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_database_name = 'my-database'
  oo_result = lo_glu->getdatabase( iv_name = iv_database_name ).
  DATA(lo_database) = oo_result->get_database( ).
  MESSAGE 'Database information retrieved.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Database does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- For API details, see [GetDatabase](#) in *AWS SDK for SAP ABAP API reference*.

GetJobRun

The following code example shows how to use GetJobRun.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_job_name = 'my-etl-job'
  " iv_run_id = 'jr_abcd1234567890abcdef1234567890abcdef12345678'
  oo_result = lo_glu->getjobrun(
    iv_jobname = iv_job_name
    iv_runid = iv_run_id ).
  DATA(lo_job_run) = oo_result->get_jobrun( ).
  MESSAGE 'Job run information retrieved.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Job or job run does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- For API details, see [GetJobRun](#) in *AWS SDK for SAP ABAP API reference*.

GetJobRuns

The following code example shows how to use GetJobRuns.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " iv_job_name = 'my-etl-job'  
  oo_result = lo_glu->getjobruns( iv_jobname = iv_job_name ).  
  DATA(lt_job_runs) = oo_result->get_jobruns( ).  
  MESSAGE 'Job runs retrieved successfully.' TYPE 'I'.  
CATCH /aws1/cx_gluentitynotfoundex.  
  MESSAGE 'Job does not exist.' TYPE 'E'.  
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).  
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).  
  MESSAGE lv_invalid_error TYPE 'E'.  
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).  
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).  
  MESSAGE lv_internal_error TYPE 'E'.  
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).  
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).  
  MESSAGE lv_timeout_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetJobRuns](#) in *AWS SDK for SAP ABAP API reference*.

GetTables

The following code example shows how to use GetTables.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_database_name = 'my-database'
  oo_result = lo_glu->gettables( iv_databasename = iv_database_name ).
  DATA(lt_tables) = oo_result->get_tablelist( ).
  MESSAGE 'Tables retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'Database does not exist.' TYPE 'E'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
  MESSAGE lv_invalid_error TYPE 'E'.
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- For API details, see [GetTables](#) in *AWS SDK for SAP ABAP API reference*.

ListJobs

The following code example shows how to use ListJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_glu->listjobs( ).
  DATA(lt_job_names) = oo_result->get_jobnames( ).
  MESSAGE 'Job list retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_gluentitynotfoundex.
  MESSAGE 'No jobs found.' TYPE 'I'.
CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
  DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
```

```

    MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.

```

- For API details, see [ListJobs](#) in *AWS SDK for SAP ABAP API reference*.

StartCrawler

The following code example shows how to use StartCrawler.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_crawler_name = 'my-crawler'
    lo_glu->startcrawler( iv_name = iv_crawler_name ).
    MESSAGE 'Crawler started successfully.' TYPE 'I'.
    CATCH /aws1/cx_glucrawlerrunningex.
    MESSAGE 'Crawler is already running.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.

```

- For API details, see [StartCrawler](#) in *AWS SDK for SAP ABAP API reference*.

StartJobRun

The following code example shows how to use StartJobRun.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_job_name = 'my-etl-job'
  " iv_input_database = 'my-database'
  " iv_input_table = 'my-table'
  " iv_output_bucket_url = 's3://example-output-bucket/'

  DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
  lt_arguments = VALUE #(
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--input_database'
      value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_database ) ) )
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--input_table'
      value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
    ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
      key = '--output_bucket_url'
      value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

  DATA(oo_result) = lo_glu->startjobrun(
    iv_jobname = iv_job_name
    it_arguments = lt_arguments ).
  ov_job_run_id = oo_result->getjobrunid( ).
  MESSAGE 'Job run started successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluconcurrentrunsex00.
    MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.

```

```
CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
  DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
  MESSAGE lv_internal_error TYPE 'E'.
CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
  DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
  MESSAGE lv_timeout_error TYPE 'E'.
CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
  DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
  MESSAGE lv_limit_error TYPE 'E'.
ENDTRY.
```

- For API details, see [StartJobRun](#) in *AWS SDK for SAP ABAP API reference*.

HealthImaging examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with HealthImaging.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CopyImageSet

The following code example shows how to use CopyImageSet.

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_source_image_set_id = '1234567890123456789012345678901234567890'
```

```

" iv_source_version_id = '1'
" iv_destination_image_set_id =
'1234567890123456789012345678901234567890' (optional)
" iv_destination_version_id = '1' (optional)
" iv_force = abap_false
DATA(lo_source_info) = NEW /aws1/cl_migcpsrcimagesetinf00(
  iv_latestversionid = iv_source_version_id ).
DATA(lo_copy_info) = NEW /aws1/cl_migcpimagesetinfmtion(
  io_sourceimageset = lo_source_info ).
IF iv_destination_image_set_id IS NOT INITIAL AND
  iv_destination_version_id IS NOT INITIAL.
  DATA(lo_dest_info) = NEW /aws1/cl_migcopydstimageset(
    iv_imagesetid = iv_destination_image_set_id
    iv_latestversionid = iv_destination_version_id ).
  lo_copy_info = NEW /aws1/cl_migcpimagesetinfmtion(
    io_sourceimageset = lo_source_info
    io_destinationimageset = lo_dest_info ).
ENDIF.
oo_result = lo_mig->copyimageset(
  iv_datastoreid = iv_datastore_id
  iv_sourceimagesetid = iv_source_image_set_id
  io_copyimagesetinformatoin = lo_copy_info
  iv_force = iv_force ).
DATA(lo_dest_props) = oo_result->get_dstimagesetproperties( ).
DATA(lv_new_id) = lo_dest_props->get_imagesetid( ).
MESSAGE |Image set copied with new ID: { lv_new_id }.| TYPE 'I'.
CATCH /aws1/cx_migaccesssdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migservicequotaexcdex.
  MESSAGE 'Service quota exceeded.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [CopyImageSet](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

CreateDatastore

The following code example shows how to use CreateDatastore.

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_name = 'my-datastore-name'  
  oo_result = lo_mig->createdatastore( iv_datastorename = iv_datastore_name ).  
  DATA(lv_datastore_id) = oo_result->get_datastoreid( ).  
  MESSAGE 'Data store created.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict. Data store may already exist.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migservicequotaexcdex.  
  MESSAGE 'Service quota exceeded.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [CreateDatastore](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

DeleteDatastore

The following code example shows how to use DeleteDatastore.

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '1234567890123456789012345678901234567890'  
  oo_result = lo_mig->deletedatastore( iv_datastoreid = iv_datastore_id ).  
  MESSAGE 'Data store deleted.' TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict. Data store may contain resources.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Data store not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteDatastore](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

DeleteImageSet

The following code example shows how to use DeleteImageSet.

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '1234567890123456789012345678901234567890'  
  " iv_image_set_id = '1234567890123456789012345678901234567890'
```

```

oo_result = lo_mig->deleteimageset(
  iv_datastoreid = iv_datastore_id
  iv_imagesetid = iv_image_set_id ).
MESSAGE 'Image set deleted.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image set not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [DeleteImageSet](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

GetDICOMImportJob

The following code example shows how to use GetDICOMImportJob.

SDK for SAP ABAP

```

TRY.
  " iv_datastore_id = '1234567890123456789012345678901234567890'
  " iv_job_id = '12345678901234567890123456789012'
  oo_result = lo_mig->getdicomimportjob(
    iv_datastoreid = iv_datastore_id
    iv_jobid = iv_job_id ).
  DATA(lo_job_props) = oo_result->get_jobproperties( ).
  DATA(lv_job_status) = lo_job_props->get_jobstatus( ).
  MESSAGE |Job status: { lv_job_status }.| TYPE 'I'.

```

```

CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Job not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [GetDICOMImportJob](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

GetDatastore

The following code example shows how to use GetDatastore.

SDK for SAP ABAP

```

TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  oo_result = lo_mig->getdatastore( iv_datastoreid = iv_datastore_id ).
  DATA(lo_properties) = oo_result->get_datastoreproperties( ).
  DATA(lv_name) = lo_properties->get_datastorename( ).
  DATA(lv_status) = lo_properties->get_datastorestatus( ).
  MESSAGE 'Data store properties retrieved.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Data store not found.' TYPE 'I'.

```

```

CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [GetDatastore](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

GetImageFrame

The following code example shows how to use GetImageFrame.

SDK for SAP ABAP

```

TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_image_set_id = '1234567890123456789012345678901234567890'
  " iv_image_frame_id = '1234567890123456789012345678901234567890'
  oo_result = lo_mig->getimageframe(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id
    io_imageframeinformation = NEW /aws1/cl_migimageframeinfmtion(
      iv_imageframeid = iv_image_frame_id ) ).
  DATA(lv_frame_blob) = oo_result->get_imageframeblob( ).
  MESSAGE 'Image frame retrieved.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Image frame not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.

```

```
MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [GetImageFrame](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

GetImageSet

The following code example shows how to use `GetImageSet`.

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'  
  " iv_image_set_id = '12345678901234567890123456789012345678901234567890'  
  " iv_version_id = '1' (optional)  
  IF iv_version_id IS NOT INITIAL.  
    oo_result = lo_mig->getimageset(  
      iv_datastoreid = iv_datastore_id  
      iv_imagesetid = iv_image_set_id  
      iv_versionid = iv_version_id ).  
  ELSE.  
    oo_result = lo_mig->getimageset(  
      iv_datastoreid = iv_datastore_id  
      iv_imagesetid = iv_image_set_id ).  
  ENDIF.  
  DATA(lv_state) = oo_result->get_imagesetstate( ).  
  MESSAGE |Image set retrieved with state: { lv_state }.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict error.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Image set not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.
```

```
MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- For API details, see [GetImageSet](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

GetImageSetMetadata

The following code example shows how to use `GetImageSetMetadata`.

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
  " iv_version_id = '1' (optional)
  IF iv_version_id IS NOT INITIAL.
    oo_result = lo_mig->getimagesetmetadata(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id
      iv_versionid = iv_version_id ).
  ELSE.
    oo_result = lo_mig->getimagesetmetadata(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id ).
  ENDIF.
  DATA(lv_metadata_blob) = oo_result->get_imagesetmetadatablob( ).
  MESSAGE 'Image set metadata retrieved.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
```

```

    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
  ENDRY.

```

- For API details, see [GetImageSetMetadata](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

ListDICOMImportJobs

The following code example shows how to use ListDICOMImportJobs.

SDK for SAP ABAP

```

TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  oo_result = lo_mig->listdicomimportjobs( iv_datastoreid = iv_datastore_id ).
  DATA(lt_jobs) = oo_result->get_jobsummaries( ).
  DATA(lv_count) = lines( lt_jobs ).
  MESSAGE |Found { lv_count } DICOM import jobs.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
  ENDRY.

```

- For API details, see [ListDICOMImportJobs](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

ListDatastores

The following code example shows how to use ListDatastores.

SDK for SAP ABAP

```
TRY.  
    oo_result = lo_mig->listdatastores( ).  
    DATA(lt_datastores) = oo_result->get_datastoresummaries( ).  
    DATA(lv_count) = lines( lt_datastores ).  
    MESSAGE |Found { lv_count } data stores.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
    MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
    MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
    MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
    MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [ListDatastores](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

ListImageSetVersions

The following code example shows how to use ListImageSetVersions.

SDK for SAP ABAP

```
TRY.  
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'  
  " iv_image_set_id = '12345678901234567890123456789012345678901234567890'  
  oo_result = lo_mig->listimagesetversions(  
    iv_datastoreid = iv_datastore_id  
    iv_imagesetid = iv_image_set_id ).  
  DATA(lt_versions) = oo_result->get_imagesetpropertieslist( ).  
  DATA(lv_count) = lines( lt_versions ).  
  MESSAGE |Found { lv_count } image set versions.| TYPE 'I'.  
CATCH /aws1/cx_migaccessdeniedex.  
  MESSAGE 'Access denied.' TYPE 'I'.  
CATCH /aws1/cx_migconflictexception.  
  MESSAGE 'Conflict error.' TYPE 'I'.  
CATCH /aws1/cx_miginternalserverex.  
  MESSAGE 'Internal server error.' TYPE 'I'.  
CATCH /aws1/cx_migresourcenotfoundex.  
  MESSAGE 'Image set not found.' TYPE 'I'.  
CATCH /aws1/cx_migthrottlingex.  
  MESSAGE 'Request throttled.' TYPE 'I'.  
CATCH /aws1/cx_migvalidationex.  
  MESSAGE 'Validation error.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [ListImageSetVersions](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

ListTagsForResource

The following code example shows how to use ListTagsForResource.

SDK for SAP ABAP

```
TRY.
```

```

    " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
    oo_result = lo_mig->listtagsforresource( iv_resourcearn = iv_resource_arn ).
    DATA(lt_tags) = oo_result->get_tags( ).
    DATA(lv_count) = lines( lt_tags ).
    MESSAGE |Found { lv_count } tags for resource.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [ListTagsForResource](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

SearchImageSets

The following code example shows how to use SearchImageSets.

SDK for SAP ABAP

```

TRY.
    " iv_datastore_id = '1234567890123456789012345678901234567890'
    oo_result = lo_mig->searchimagesets(
        iv_datastoreid = iv_datastore_id
        io_searchcriteria = io_search_criteria ).
    DATA(lt_imagesets) = oo_result->get_imagesetsmetadatasums( ).
    DATA(lv_count) = lines( lt_imagesets ).
    MESSAGE |Found { lv_count } image sets.| TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.

```

```

    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.

```

- For API details, see [SearchImageSets](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

StartDICOMImportJob

The following code example shows how to use StartDICOMImportJob.

SDK for SAP ABAP

```

TRY.
  " iv_job_name = 'import-job-1'
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_role_arn = 'arn:aws:iam::123456789012:role/ImportJobRole'
  " iv_input_s3_uri = 's3://my-bucket/input/'
  " iv_output_s3_uri = 's3://my-bucket/output/'
  oo_result = lo_mig->startdicomimportjob(
    iv_jobname = iv_job_name
    iv_datastoreid = iv_datastore_id
    iv_dataaccessrolearn = iv_role_arn
    iv_inputs3uri = iv_input_s3_uri
    iv_outputs3uri = iv_output_s3_uri ).
  DATA(lv_job_id) = oo_result->get_jobid( ).
  MESSAGE |DICOM import job started with ID: { lv_job_id }.| TYPE 'I'.

```

```

CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_migconflictexception.
  MESSAGE 'Conflict error.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migservicequotaexcdex.
  MESSAGE 'Service quota exceeded.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [StartDICOMImportJob](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

TagResource

The following code example shows how to use TagResource.

SDK for SAP ABAP

```

TRY.
  " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
  lo_mig->tagresource(
    iv_resourcearn = iv_resource_arn
    it_tags = it_tags ).
  MESSAGE 'Resource tagged successfully.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.

```

```

CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [TagResource](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

UntagResource

The following code example shows how to use UntagResource.

SDK for SAP ABAP

```

TRY.
  " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
  lo_mig->untagresource(
    iv_resourcearn = iv_resource_arn
    it_tagkeys = it_tag_keys ).
  MESSAGE 'Resource untagged successfully.' TYPE 'I'.
CATCH /aws1/cx_migaccessdeniedex.
  MESSAGE 'Access denied.' TYPE 'I'.
CATCH /aws1/cx_miginternalserverex.
  MESSAGE 'Internal server error.' TYPE 'I'.
CATCH /aws1/cx_migresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
CATCH /aws1/cx_migthrottlingex.
  MESSAGE 'Request throttled.' TYPE 'I'.
CATCH /aws1/cx_migvalidationex.
  MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [UntagResource](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

UpdateImageSetMetadata

The following code example shows how to use UpdateImageSetMetadata.

SDK for SAP ABAP

```
TRY.
  " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
  " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
  " iv_latest_version_id = '1'
  " iv_force = abap_false
  oo_result = lo_mig->updateimagesetmetadata(
    iv_datastoreid = iv_datastore_id
    iv_imagesetid = iv_image_set_id
    iv_latestversionid = iv_latest_version_id
    io_updateimagesetmetupdates = io_metadata_updates
    iv_force = iv_force ).
  DATA(lv_new_version) = oo_result->get_latestversionid( ).
  MESSAGE |Image set metadata updated to version: { lv_new_version }.| TYPE
'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migservicequotaexcdex.
    MESSAGE 'Service quota exceeded.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- For API details, see [UpdateImageSetMetadata](#) in *AWS SDK for SAP ABAP API reference*.

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

HealthLake examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with HealthLake.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateFHIRDatastore

The following code example shows how to use CreateFHIRDatastore.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

TRY.

```

    " iv_datastore_name = 'MyHealthLakeDataStore'
    oo_result = lo_hll->createfhirdatastore(
        iv_datastorename = iv_datastore_name
        iv_datastoretypeversion = 'R4'
    ).
    MESSAGE 'Data store created successfully.' TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).
    lv_error = |Internal server error: { lo_internal_ex->av_err_code }-
{ lo_internal_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
ENDTRY.

```

- For API details, see [CreateFHIRDatastore](#) in *AWS SDK for SAP ABAP API reference*.

DeleteFHIRDatastore

The following code example shows how to use DeleteFHIRDatastore.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

TRY.

```

    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->deletefhirdatastore(

```

```

        iv_datastoreid = iv_datastore_id
    ).
    MESSAGE 'Data store deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
{ lo_access_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
    CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
        lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
{ lo_conflict_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_conflict_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.

```

- For API details, see [DeleteFHIRDatastore](#) in *AWS SDK for SAP ABAP API reference*.

DescribeFHIRDatastore

The following code example shows how to use DescribeFHIRDatastore.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->describefhirdatastore(
        iv_datastoreid = iv_datastore_id
    ).
    DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).

```

```

    IF lo_datastore_properties IS BOUND.
        DATA(lv_datastore_name) = lo_datastore_properties->get_datastorename( ).
        DATA(lv_datastore_status) = lo_datastore_properties-
>get_datastorestatus( ).
        MESSAGE 'Data store described successfully.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.

```

- For API details, see [DescribeFHIRDatastore](#) in *AWS SDK for SAP ABAP API reference*.

DescribeFHIRExportJob

The following code example shows how to use DescribeFHIRExportJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->describefhirexportjob(
        iv_datastoreid = iv_datastore_id
        iv_jobid = iv_job_id
    ).
    DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).

```

```

    IF lo_export_job_properties IS BOUND.
        DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).
        MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.

```

- For API details, see [DescribeFHIRExportJob](#) in *AWS SDK for SAP ABAP API reference*.

DescribeFHIRImportJob

The following code example shows how to use DescribeFHIRImportJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_hll->describefhirimportjob(
        iv_datastoreid = iv_datastore_id
        iv_jobid = iv_job_id
    ).
    DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
    IF lo_import_job_properties IS BOUND.
        DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
        MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
    ENDIF.

```

```

ENDIF.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
  DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_notfound_ex.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
ENDTRY.

```

- For API details, see [DescribeFHIRImportJob](#) in *AWS SDK for SAP ABAP API reference*.

ListFHIRDatastores

The following code example shows how to use ListFHIRDatastores.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_hll->listfhirdatastores( ).
  DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
  DATA(lv_datastore_count) = lines( lt_datastores ).
  MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
  CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
  lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.

```

```

    RAISE EXCEPTION lo_throttling_ex.
  ENDTRY.

```

- For API details, see [ListFHIRDatastores](#) in *AWS SDK for SAP ABAP API reference*.

ListFHIRExportJobs

The following code example shows how to use ListFHIRExportJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  IF iv_submitted_after IS NOT INITIAL.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
      iv_submittedafter = iv_submitted_after
    ).
  ELSE.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
    ).
  ENDIF.
  DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
  DATA(lv_job_count) = lines( lt_export_jobs ).
  MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
  lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.

```

```

    RAISE EXCEPTION lo_notfound_ex.
  ENDTRY.

```

- For API details, see [ListFHIRExportJobs](#) in *AWS SDK for SAP ABAP API reference*.

ListFHIRImportJobs

The following code example shows how to use ListFHIRImportJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  IF iv_submitted_after IS NOT INITIAL.
    oo_result = lo_hll->listfhirimportjobs(
      iv_datastoreid = iv_datastore_id
      iv_submittedafter = iv_submitted_after
    ).
  ELSE.
    oo_result = lo_hll->listfhirimportjobs(
      iv_datastoreid = iv_datastore_id
    ).
  ENDIF.
  DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).
  DATA(lv_job_count) = lines( lt_import_jobs ).
  MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
  lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.

```

```

        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.

```

- For API details, see [ListFHIRImportJobs](#) in *AWS SDK for SAP ABAP API reference*.

ListTagsForResource

The following code example shows how to use ListTagsForResource.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
    fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    DATA(lo_result) = lo_hll->listtagsforresource(
        iv_resourcearn = iv_resource_arn
    ).
    ot_tags = lo_result->get_tags( ).
    DATA(lv_tag_count) = lines( ot_tags ).
    MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
    { lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
    { lo_notfound_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_notfound_ex.
ENDTRY.

```

- For API details, see [ListTagsForResource](#) in *AWS SDK for SAP ABAP API reference*.

StartFHIRExportJob

The following code example shows how to use StartFHIRExportJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_job_name = 'MyExportJob'
  " iv_output_s3_uri = 's3://my-bucket/export/output/'
  " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
  " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
  oo_result = lo_hll->startfhirexportjob(
    iv_jobname = iv_job_name
    io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
      io_s3configuration = NEW /aws1/cl_hlls3configuration(
        iv_s3uri = iv_output_s3_uri
        iv_kmskeyid = iv_kms_key_id
      )
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_datastoreid = iv_datastore_id
  ).
  DATA(lv_job_id) = oo_result->get_jobid( ).
  MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
  CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
  lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_throttling_ex.
  CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).

```

```

lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_access_ex.
ENDTRY.

```

- For API details, see [StartFHIRExportJob](#) in *AWS SDK for SAP ABAP API reference*.

StartFHIRImportJob

The following code example shows how to use StartFHIRImportJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_job_name = 'MyImportJob'
  " iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'
  " iv_job_output_s3_uri = 's3://my-bucket/import/output/'
  " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
  " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeImportRole'
  oo_result = lo_hll->startfhirimportjob(
    iv_jobname = iv_job_name
    io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =
iv_input_s3_uri )
    io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
      io_s3configuration = NEW /aws1/cl_hlls3configuration(
        iv_s3uri = iv_job_output_s3_uri
        iv_kmskeyid = iv_kms_key_id
      )
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_datastoreid = iv_datastore_id
  ).

```

```

    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
ENDTRY.

```

- For API details, see [StartFHIRImportJob](#) in *AWS SDK for SAP ABAP API reference*.

TagResource

The following code example shows how to use TagResource.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    lo_hll->tagresource(
        iv_resourcearn = iv_resource_arn
        it_tags = it_tags
    ).
    MESSAGE 'Resource tagged successfully.' TYPE 'I'.

```

```

    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
      DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
      lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_notfound_ex.
  ENDTRY.

```

- For API details, see [TagResource](#) in *AWS SDK for SAP ABAP API reference*.

UntagResource

The following code example shows how to use UntagResource.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

  TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    lo_hll->untagresource(
      iv_resourcearn = iv_resource_arn
      it_tagkeys = it_tag_keys
    ).
    MESSAGE 'Resource untagged successfully.' TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).

```

```
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.   
MESSAGE lv_error TYPE 'I'.   
RAISE EXCEPTION lo_notfound_ex.   
ENDTRY.
```

- For API details, see [UntagResource](#) in *AWS SDK for SAP ABAP API reference*.

IAM examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with IAM.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AttachRolePolicy

The following code example shows how to use `AttachRolePolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_iam->attachrolepolicy(  

```

```
        iv_rolename = iv_role_name
        iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to role successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_iamlimitexceeddex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- For API details, see [AttachRolePolicy](#) in *AWS SDK for SAP ABAP API reference*.

AttachUserPolicy

The following code example shows how to use `AttachUserPolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    lo_iam->attachuserpolicy(
        iv_username = iv_user_name
        iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to user successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User or policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_iamlimitexceeddex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- For API details, see [AttachUserPolicy](#) in *AWS SDK for SAP ABAP API reference*.

CreateAccessKey

The following code example shows how to use `CreateAccessKey`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->createaccesskey(  
        iv_username = iv_user_name ).  
    MESSAGE 'Access key created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'User does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceedex.  
    MESSAGE 'Maximum number of access keys reached.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateAccessKey](#) in *AWS SDK for SAP ABAP API reference*.

CreateAccountAlias

The following code example shows how to use CreateAccountAlias.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_iam->createaccountalias(  
        iv_accountalias = iv_account_alias ).  
    MESSAGE 'Account alias created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'Account alias already exists.' TYPE 'E'.
```

```
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Account alias limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateAccountAlias](#) in *AWS SDK for SAP ABAP API reference*.

CreatePolicy

The following code example shows how to use CreatePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->createpolicy(  
        iv_policyname = iv_policy_name  
        iv_policydocument = iv_policy_document  
        iv_description = iv_description ).  
    MESSAGE 'Policy created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'Policy already exists.' TYPE 'E'.  
CATCH /aws1/cx_iammalformedplydocex.  
    MESSAGE 'Policy document is malformed.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Policy limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreatePolicy](#) in *AWS SDK for SAP ABAP API reference*.

CreatePolicyVersion

The following code example shows how to use CreatePolicyVersion.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->createpolicyversion(  
        iv_policyarn = iv_policy_arn  
        iv_policydocument = iv_policy_document  
        iv_setasdefault = iv_set_as_default ).  
    MESSAGE 'Policy version created successfully.' TYPE 'I'.  
    CATCH /aws1/cx_iamnosuchentityex.  
        MESSAGE 'Policy does not exist.' TYPE 'E'.  
    CATCH /aws1/cx_iammalformedplydocex.  
        MESSAGE 'Policy document is malformed.' TYPE 'E'.  
    CATCH /aws1/cx_iamlimitexceeddex.  
        MESSAGE 'Policy version limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreatePolicyVersion](#) in *AWS SDK for SAP ABAP API reference*.

CreateRole

The following code example shows how to use CreateRole.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->createrole(  
        iv_rolename = iv_role_name
```

```

        iv_assumerolepolicydocument = iv_assume_role_policy_document ).
    MESSAGE 'Role created successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamentityalrddyexex.
        MESSAGE 'Role already exists.' TYPE 'E'.
    CATCH /aws1/cx_iammalformedplydocex.
        MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
    CATCH /aws1/cx_iamlimitexceeddex.
        MESSAGE 'Role limit exceeded.' TYPE 'E'.
    ENDRY.

```

- For API details, see [CreateRole](#) in *AWS SDK for SAP ABAP API reference*.

CreateServiceLinkedRole

The following code example shows how to use CreateServiceLinkedRole.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
        oo_result = lo_iam->listpolicyversions(
            iv_policyarn = iv_policy_arn ).
        MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Policy does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
    ENDRY.

```

- For API details, see [CreateServiceLinkedRole](#) in *AWS SDK for SAP ABAP API reference*.

CreateUser

The following code example shows how to use CreateUser.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->createuser(  
        iv_username = iv_user_name ).  
    MESSAGE 'User created successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamentityalrddyexex.  
    MESSAGE 'User already exists.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceeddex.  
    MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Entity does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateUser](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAccessKey

The following code example shows how to use DeleteAccessKey.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_iam->deleteaccesskey(  
        iv_accesskeyid = iv_access_key_id  
        iv_username = iv_user_name ).
```

```
MESSAGE 'Access key deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Access key or user does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteAccessKey](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAccountAlias

The following code example shows how to use DeleteAccountAlias.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_iam->deleteaccountalias(
    iv_accountalias = iv_account_alias ).
MESSAGE 'Account alias deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Account alias does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteAccountAlias](#) in *AWS SDK for SAP ABAP API reference*.

DeletePolicy

The following code example shows how to use DeletePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->deletepolicy( iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
  MESSAGE 'Policy cannot be deleted due to attachments.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeletePolicy](#) in *AWS SDK for SAP ABAP API reference*.

DeletePolicyVersion

The following code example shows how to use DeletePolicyVersion.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->deletepolicyversion(  
    iv_policyarn = iv_policy_arn  
    iv_versionid = iv_version_id ).  
  MESSAGE 'Policy version deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Policy or version does not exist.' TYPE 'E'.
```

```
CATCH /aws1/cx_iamdeleteconflictex.  
    MESSAGE 'Cannot delete default policy version.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceedex.  
    MESSAGE 'Limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeletePolicyVersion](#) in *AWS SDK for SAP ABAP API reference*.

DeleteRole

The following code example shows how to use DeleteRole.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_iam->deleterole( iv_rolename = iv_role_name ).  
    MESSAGE 'Role deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
    MESSAGE 'Role cannot be deleted due to attached resources.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteRole](#) in *AWS SDK for SAP ABAP API reference*.

DeleteUser

The following code example shows how to use DeleteUser.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->deleteuser( iv_username = iv_user_name ).  
  MESSAGE 'User deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'User does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamdeleteconflictex.  
  MESSAGE 'User cannot be deleted due to attached resources.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteUser](#) in *AWS SDK for SAP ABAP API reference*.

DetachRolePolicy

The following code example shows how to use DetachRolePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->detachrolepolicy(  
    iv_rolename = iv_role_name  
    iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy detached from role successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Role or policy does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetachRolePolicy](#) in *AWS SDK for SAP ABAP API reference*.

DetachUserPolicy

The following code example shows how to use `DetachUserPolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->detachuserpolicy(  
    iv_username = iv_user_name  
    iv_policyarn = iv_policy_arn ).  
  MESSAGE 'Policy detached from user successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'User or policy does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetachUserPolicy](#) in *AWS SDK for SAP ABAP API reference*.

GenerateCredentialReport

The following code example shows how to use `GenerateCredentialReport`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->generatecredentialreport( ).  
    MESSAGE 'Credential report generation started.' TYPE 'I'.  
CATCH /aws1/cx_iamlimitexceedex.  
    MESSAGE 'Report generation limit exceeded.' TYPE 'E'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when generating credential report.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GenerateCredentialReport](#) in *AWS SDK for SAP ABAP API reference*.

GetAccessKeyLastUsed

The following code example shows how to use `GetAccessKeyLastUsed`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getaccesskeylastused(  
        iv_accesskeyid = iv_access_key_id ).  
    MESSAGE 'Retrieved access key last used information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Access key does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetAccessKeyLastUsed](#) in *AWS SDK for SAP ABAP API reference*.

GetAccountAuthorizationDetails

The following code example shows how to use `GetAccountAuthorizationDetails`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getaccountauthdetails( ).  
    MESSAGE 'Retrieved account authorization details.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting account authorization details.' TYPE  
'E'.  
ENDTRY.
```

- For API details, see [GetAccountAuthorizationDetails](#) in *AWS SDK for SAP ABAP API reference*.

GetAccountPasswordPolicy

The following code example shows how to use `GetAccountPasswordPolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getaccountpasswordpolicy( ).  
    MESSAGE 'Retrieved account password policy.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'No password policy exists.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting password policy.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetAccountPasswordPolicy](#) in *AWS SDK for SAP ABAP API reference*.

GetAccountSummary

The following code example shows how to use GetAccountSummary.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getaccountsummary( ).  
    MESSAGE 'Retrieved account summary.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when getting account summary.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetAccountSummary](#) in *AWS SDK for SAP ABAP API reference*.

GetCredentialReport

The following code example shows how to use GetCredentialReport.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
oo_result = lo_iam->getcredentialreport( ).
MESSAGE 'Retrieved credential report.' TYPE 'I'.
CATCH /aws1/cx_iamcredrptnotpresen00.
MESSAGE 'Credential report not present.' TYPE 'E'.
CATCH /aws1/cx_iamcredrptexpiredex.
MESSAGE 'Credential report expired.' TYPE 'E'.
CATCH /aws1/cx_iamcredrptnotreadyex.
MESSAGE 'Credential report not ready.' TYPE 'E'.
CATCH /aws1/cx_iamservicefailureex.
MESSAGE 'Service failure when getting credential report.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetCredentialReport](#) in *AWS SDK for SAP ABAP API reference*.

GetPolicy

The following code example shows how to use GetPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
oo_result = lo_iam->getpolicy( iv_policyarn = iv_policy_arn ).
MESSAGE 'Retrieved policy information.' TYPE 'I'.
CATCH /aws1/cx_iamnosuchentityex.
MESSAGE 'Policy does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetPolicy](#) in *AWS SDK for SAP ABAP API reference*.

GetPolicyVersion

The following code example shows how to use GetPolicyVersion.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getpolicyversion(  
        iv_policyarn = iv_policy_arn  
        iv_versionid = iv_version_id ).  
    MESSAGE 'Retrieved policy version information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iaminvalidinputex.  
    MESSAGE 'Invalid input provided.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetPolicyVersion](#) in *AWS SDK for SAP ABAP API reference*.

GetRole

The following code example shows how to use GetRole.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->getrole( iv_rolename = iv_role_name ).  
    MESSAGE 'Retrieved role information.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' TYPE 'E'.
```

```
ENDTRY.
```

- For API details, see [GetRole](#) in *AWS SDK for SAP ABAP API reference*.

ListAccessKeys

The following code example shows how to use ListAccessKeys.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listaccesskeys(  
        iv_username = iv_user_name ).  
    MESSAGE 'Retrieved access key list.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'User does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAccessKeys](#) in *AWS SDK for SAP ABAP API reference*.

ListAccountAliases

The following code example shows how to use ListAccountAliases.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listaccountaliases( ).  
    MESSAGE 'Retrieved account alias list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing account aliases.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAccountAliases](#) in *AWS SDK for SAP ABAP API reference*.

ListAttachedRolePolicies

The following code example shows how to use ListAttachedRolePolicies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listattachedrolepolicies(  
        iv_rolename = iv_role_name ).  
    MESSAGE 'Retrieved attached policy list for role.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Role does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAttachedRolePolicies](#) in *AWS SDK for SAP ABAP API reference*.

ListGroups

The following code example shows how to use ListGroups.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listgroups( ).  
    MESSAGE 'Retrieved group list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing groups.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListGroups](#) in *AWS SDK for SAP ABAP API reference*.

ListPolicies

The following code example shows how to use ListPolicies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listpolicies( iv_scope = iv_scope ).  
    MESSAGE 'Retrieved policy list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing policies.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListPolicies](#) in *AWS SDK for SAP ABAP API reference*.

ListPolicyVersions

The following code example shows how to use ListPolicyVersions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listpolicyversions(  
        iv_policyarn = iv_policy_arn ).  
    MESSAGE 'Retrieved policy versions list.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
    MESSAGE 'Policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListPolicyVersions](#) in *AWS SDK for SAP ABAP API reference*.

ListRolePolicies

The following code example shows how to use ListRolePolicies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listrolepolicies(  

```

```
        iv_rolename = iv_role_name ).
    MESSAGE 'Retrieved inline policy list for role.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'Role does not exist.' TYPE 'E'.
    ENDTRY.
```

- For API details, see [ListRolePolicies](#) in *AWS SDK for SAP ABAP API reference*.

ListRoles

The following code example shows how to use ListRoles.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_iam->listroles( ).
    MESSAGE 'Retrieved role list.' TYPE 'I'.
    CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when listing roles.' TYPE 'E'.
    ENDTRY.
```

- For API details, see [ListRoles](#) in *AWS SDK for SAP ABAP API reference*.

ListSAMLProviders

The following code example shows how to use ListSAMLProviders.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listsamlproviders( ).  
    MESSAGE 'Retrieved SAML provider list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing SAML providers.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListSAMLProviders](#) in *AWS SDK for SAP ABAP API reference*.

ListUsers

The following code example shows how to use ListUsers.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_iam->listusers( ).  
    MESSAGE 'Retrieved user list.' TYPE 'I'.  
CATCH /aws1/cx_iamservicefailureex.  
    MESSAGE 'Service failure when listing users.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListUsers](#) in *AWS SDK for SAP ABAP API reference*.

SetDefaultPolicyVersion

The following code example shows how to use SetDefaultPolicyVersion.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->setdefaultpolicyversion(  
    iv_policyarn = iv_policy_arn  
    iv_versionid = iv_version_id ).  
  MESSAGE 'Default policy version set successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Policy or version does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iaminvalidinputex.  
  MESSAGE 'Invalid input provided.' TYPE 'E'.  
CATCH /aws1/cx_iamlimitexceededex.  
  MESSAGE 'Limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [SetDefaultPolicyVersion](#) in *AWS SDK for SAP ABAP API reference*.

UpdateAccessKey

The following code example shows how to use UpdateAccessKey.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->updateaccesskey(  
    iv_accesskeyid = iv_access_key_id  
    iv_status = iv_status  
    iv_username = iv_user_name ).  
  MESSAGE 'Access key updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'Access key or user does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [UpdateAccessKey](#) in *AWS SDK for SAP ABAP API reference*.

UpdateUser

The following code example shows how to use UpdateUser.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_iam->updateuser(  
    iv_username = iv_user_name  
    iv_newusername = iv_new_user_name ).  
  MESSAGE 'User updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_iamnosuchentityex.  
  MESSAGE 'User does not exist.' TYPE 'E'.  
CATCH /aws1/cx_iamentityalrddyexex.  
  MESSAGE 'New user name already exists.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [UpdateUser](#) in *AWS SDK for SAP ABAP API reference*.

AWS IoT examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS IoT.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AttachThingPrincipal

The following code example shows how to use `AttachThingPrincipal`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    lo_iot->attachthingprincipal(
        iv_thingname = iv_thing_name
        iv_principal = iv_principal ).
    MESSAGE |Principal attached to IoT thing '{ iv_thing_name }'.| TYPE 'I'.
CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE |Resource not found when attaching principal to
'{ iv_thing_name }'.| TYPE 'I'.
```

```
RAISE EXCEPTION lo_ex.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).  
MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.  
RAISE EXCEPTION lo_svc_ex.  
ENDTRY.
```

- For API details, see [AttachThingPrincipal](#) in *AWS SDK for SAP ABAP API reference*.

CreateKeysAndCertificate

The following code example shows how to use CreateKeysAndCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.  
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).  
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).  
TRY.  
    oo_result = lo_iot->createkeysandcertificate( iv_setasactive = abap_true ).  
    MESSAGE |Certificate created: { oo_result->get_certificateid( ) }| TYPE 'I'.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).  
    MESSAGE lo_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [CreateKeysAndCertificate](#) in *AWS SDK for SAP ABAP API reference*.

CreateThing

The following code example shows how to use CreateThing.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    oo_result = lo_iot->creatething(
        iv_thingname = iv_thing_name ).
    MESSAGE |IoT thing created: { oo_result->get_thingname( ) } ARN:
{ oo_result->get_thingarn( ) }| TYPE 'I'.
    CATCH /aws1/cx_iotresrcalrddyexistsex.
    MESSAGE |IoT thing '{ iv_thing_name }' already exists.| TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- For API details, see [CreateThing](#) in *AWS SDK for SAP ABAP API reference*.

CreateTopicRule

The following code example shows how to use CreateTopicRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
```

```

DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    " Build the SNS action that will receive messages matching the rule.
    DATA lo_sns_action TYPE REF TO /aws1/cl_iotsnsaction.
    CREATE OBJECT lo_sns_action
        EXPORTING
            iv_targetarn = iv_sns_action_arn
            iv_rolearn   = iv_role_arn.

    DATA lo_action TYPE REF TO /aws1/cl_iotaction.
    CREATE OBJECT lo_action
        EXPORTING
            io_sns = lo_sns_action.

    DATA lt_actions TYPE /aws1/cl_iotaction=>tt_actionlist.
    APPEND lo_action TO lt_actions.

    " iv_topic = 'my/iot/topic' - The MQTT topic pattern to match
    DATA lo_payload TYPE REF TO /aws1/cl_iottopicrulepayload.
    CREATE OBJECT lo_payload
        EXPORTING
            iv_sql      = |SELECT * FROM '{ iv_topic }'|
            it_actions = lt_actions.

    lo_iot->createtopicrule(
        iv_rulename      = iv_rule_name
        io_topicrulepayload = lo_payload ).
    MESSAGE |IoT topic rule created: { iv_rule_name }| TYPE 'I'.
    CATCH /aws1/cx_iotresrcalrddyexistsex.
        MESSAGE |Topic rule '{ iv_rule_name }' already exists.| TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
        MESSAGE lo_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_ex.
ENDTRY.

```

- For API details, see [CreateTopicRule](#) in *AWS SDK for SAP ABAP API reference*.

DeleteCertificate

The following code example shows how to use DeleteCertificate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    " Certificates must be deactivated before they can be deleted.
    lo_iot->updatecertificate(
        iv_certificateid = iv_certificate_id
        iv_newstatus     = 'INACTIVE' ).
    lo_iot->deletecertificate( iv_certificateid = iv_certificate_id ).
    MESSAGE |Certificate deleted: { iv_certificate_id }| TYPE 'I'.
CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE |Certificate '{ iv_certificate_id }' not found.| TYPE 'I'.
    RAISE EXCEPTION lo_ex.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
    MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_svc_ex.
ENDTRY.
```

- For API details, see [DeleteCertificate](#) in *AWS SDK for SAP ABAP API reference*.

DeleteThing

The following code example shows how to use DeleteThing.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    lo_iot->deletething( iv_thingname = iv_thing_name ).
    MESSAGE |IoT thing deleted: { iv_thing_name }| TYPE 'I'.
CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE |IoT thing '{ iv_thing_name }' not found.| TYPE 'I'.
    RAISE EXCEPTION lo_ex.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
    MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_svc_ex.
ENDTRY.
```

- For API details, see [DeleteThing](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTopicRule

The following code example shows how to use DeleteTopicRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    lo_iot->deletetopicrule( iv_rulename = iv_rule_name ).
    MESSAGE |IoT topic rule deleted: { iv_rule_name }| TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- For API details, see [DeleteTopicRule](#) in *AWS SDK for SAP ABAP API reference*.

DescribeEndpoint

The following code example shows how to use DescribeEndpoint.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
" iv_endpoint_type = 'iot:Data-ATS' - Endpoint type for data operations
TRY.
    DATA(lo_result) = lo_iot->describeendpoint( iv_endpointtype =
iv_endpoint_type ).
    ov_endpoint_address = lo_result->get_endpointaddress( ).
    MESSAGE |Endpoint address: { ov_endpoint_address }| TYPE 'I'.
CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
    MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
    RAISE EXCEPTION lo_throttle_ex.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- For API details, see [DescribeEndpoint](#) in *AWS SDK for SAP ABAP API reference*.

DetachThingPrincipal

The following code example shows how to use DetachThingPrincipal.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    lo_iot->detachthingprincipal(
        iv_thingname = iv_thing_name
        iv_principal = iv_principal ).
    MESSAGE |Principal detached from IoT thing '{ iv_thing_name }'.| TYPE 'I'.
    CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE |Resource not found when detaching principal from
'{ iv_thing_name }'.| TYPE 'I'.
    RAISE EXCEPTION lo_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
    MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_svc_ex.
ENDTRY.
```

- For API details, see [DetachThingPrincipal](#) in *AWS SDK for SAP ABAP API reference*.

ListCertificates

The following code example shows how to use ListCertificates.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    " Collect all certificates by following the pagination marker.
    DATA lv_marker TYPE /aws1/iotmarker.
    DATA lv_count TYPE i.

    DO.
        oo_result = lo_iot->listcertificates( iv_marker = lv_marker ).
        lv_count = lv_count + lines( oo_result->get_certificates( ) ).
        lv_marker = oo_result->get_nextmarker( ).
        IF lv_marker IS INITIAL.
            EXIT.
        ENDIF.
    ENDDO.

    MESSAGE |Retrieved { lv_count } IoT certificates.| TYPE 'I'.
    CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
    MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
    RAISE EXCEPTION lo_throttle_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.

```

- For API details, see [ListCertificates](#) in *AWS SDK for SAP ABAP API reference*.

ListThings

The following code example shows how to use ListThings.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    " Collect all things by following the pagination token.
    DATA lv_nexttoken TYPE /aws1/iotnexttoken.
    DATA lv_count      TYPE i.

    DO.
        oo_result      = lo_iot->listthings( iv_nexttoken = lv_nexttoken ).
        lv_count        = lv_count + lines( oo_result->get_things( ) ).
        lv_nexttoken    = oo_result->get_nexttoken( ).
        IF lv_nexttoken IS INITIAL.
            EXIT.
        ENDIF.
    ENDDO.

    MESSAGE |Retrieved { lv_count } IoT things.| TYPE 'I'.
    CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
    MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
    RAISE EXCEPTION lo_throttle_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.

```

- For API details, see [ListThings](#) in *AWS SDK for SAP ABAP API reference*.

SearchIndex

The following code example shows how to use SearchIndex.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
" iv_query_string = 'thingName:MyThing*' - Fleet indexing query string
TRY.
    oo_result = lo_iot->searchindex( iv_querystring = iv_query_string ).
    MESSAGE |Found { lines( oo_result->get_things( ) ) } IoT things matching
query.| TYPE 'I'.
    CATCH /aws1/cx_iotindexnotreadyex INTO DATA(lo_idx_ex).
    MESSAGE 'Fleet indexing is not ready. Enable indexing with
UpdateIndexingConfiguration first.' TYPE 'I'.
    RAISE EXCEPTION lo_idx_ex.
    CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
    MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
    RAISE EXCEPTION lo_throttle_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.

```

- For API details, see [SearchIndex](#) in *AWS SDK for SAP ABAP API reference*.

UpdateIndexingConfiguration

The following code example shows how to use UpdateIndexingConfiguration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    DATA lo_idx_conf TYPE REF TO /aws1/cl_iotthingindexingconf.
    CREATE OBJECT lo_idx_conf

```

```
EXPORTING
    iv_thingindexingmode = 'REGISTRY'.
lo_iot->updateindexingconfiguration(
    io_thingindexingconf = lo_idx_conf ).
MESSAGE 'IoT thing indexing configuration updated to REGISTRY mode.' TYPE
'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- For API details, see [UpdateIndexingConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

AWS IoT data examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS IoT data.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

GetThingShadow

The following code example shows how to use `GetThingShadow`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    DATA(lo_result) = lo_iop->getthingsshadow( iv_thingname = iv_thing_name ).

    " Convert xstring payload to JSON string
    DATA(lv_payload) = lo_result->get_payload( ).
    ov_shadow = /aws1/cl_rt_util=>xstring_to_string( lv_payload ).
    MESSAGE 'Thing shadow retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_iopresourcenotfoundex.
    MESSAGE 'Thing shadow not found.' TYPE 'E'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).
    DATA(lv_error) = |{ lo_exception->get_text( ) }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [GetThingShadow](#) in *AWS SDK for SAP ABAP API reference*.

UpdateThingShadow

The following code example shows how to use UpdateThingShadow.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    " Convert JSON string to xstring for payload
```

```
DATA(lv_payload) = /aws1/cl_rt_util=>string_to_xstring( iv_shadow_state ).

lo_iop->updatethingshadow(
  iv_thingname = iv_thing_name
  iv_payload = lv_payload ).
MESSAGE 'Thing shadow updated successfully.' TYPE 'I'.
CATCH /aws1/cx_iopresourcenotfoundex.
  MESSAGE 'Thing not found.' TYPE 'E'.
CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).
  DATA(lv_error) = |{ lo_exception->get_text( ) }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [UpdateThingShadow](#) in *AWS SDK for SAP ABAP API reference*.

AWS IoT SiteWise examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS IoT SiteWise.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Get started](#)
- [Actions](#)

Get started

Hello AWS IoT SiteWise

The following code example shows how to get started using AWS IoT SiteWise.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing  
purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
    CATCH /aws1/cx_rt_generic.  
        MESSAGE 'Unable to list asset models.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAssetModels](#) in *AWS SDK for SAP ABAP API reference*.

Actions

BatchPutAssetPropertyValue

The following code example shows how to use BatchPutAssetPropertyValue.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_ios->batchputassetpropertyvalue(  
        it_entries = it_entries  
    ).  
    MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_iosresourcenotfoundex.  
  MESSAGE 'Asset does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [BatchPutAssetPropertyValue](#) in *AWS SDK for SAP ABAP API reference*.

CreateAsset

The following code example shows how to use CreateAsset.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_ios->createasset(  
    iv_assetname = iv_asset_name  
    iv_assetmodelid = iv_asset_model_id  
  ). " oo_result is returned for testing purposes. "  
  MESSAGE 'IoT SiteWise asset created' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
  MESSAGE 'Asset model does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateAsset](#) in *AWS SDK for SAP ABAP API reference*.

CreateAssetModel

The following code example shows how to use CreateAssetModel.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->createassetmodel(  
        iv_assetmodelname = iv_asset_model_name  
        iv_assetmodeldescription = 'This is a sample asset model description.'  
        it_assetmodelproperties = it_properties  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.  
CATCH /aws1/cx_iosresrcalrddyexistsex.  
    MESSAGE 'Asset model already exists.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateAssetModel](#) in *AWS SDK for SAP ABAP API reference*.

CreateGateway

The following code example shows how to use CreateGateway.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->creategateway(  
        iv_gatewayname = iv_gateway_name  
        io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(  
            io_greenrassv2 = NEW /aws1/cl_iosgreenrassv2(  
                iv_gatewayname = iv_gateway_name  
                iv_gatewayplatform = iv_gatewayplatform  
                iv_greenrassv2 = iv_greenrassv2  
            )  
        )  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.  
CATCH /aws1/cx_iosresrcalrddyexistsex.  
    MESSAGE 'Gateway already exists.' TYPE 'E'.  
ENDTRY.
```

```

        iv_coredevicethingname = iv_core_device_thing_name
    )
)
it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
    (
        VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
            key = 'Environment'
            value = NEW /aws1/cl_iostagmap_w( 'Production' )
        )
    )
)
). " oo_result is returned for testing purposes. "
MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
CATCH /aws1/cx_iosresrcalrddyexistsex.
MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateGateway](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAsset

The following code example shows how to use DeleteAsset.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    lo_ios->deleteasset(
        iv_assetid = iv_asset_id
    ).
    MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to delete asset.' TYPE 'E'.
ENDTRY.

```

- For API details, see [DeleteAsset](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAssetModel

The following code example shows how to use DeleteAssetModel.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ios->deleteassetmodel(  
    iv_assetmodelid = iv_asset_model_id  
  ).  
  MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
  MESSAGE 'Unable to delete asset model.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteAssetModel](#) in *AWS SDK for SAP ABAP API reference*.

DeleteGateway

The following code example shows how to use DeleteGateway.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
lo_ios->deletegateway(  
    iv_gatewayid = iv_gateway_id  
).  
MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
    MESSAGE 'Gateway does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteGateway](#) in *AWS SDK for SAP ABAP API reference*.

DescribeGateway

The following code example shows how to use DescribeGateway.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->describegateway(  
        iv_gatewayid = iv_gateway_id  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'Retrieved gateway description.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
    MESSAGE 'Gateway does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeGateway](#) in *AWS SDK for SAP ABAP API reference*.

GetAssetPropertyValue

The following code example shows how to use GetAssetPropertyValue.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->getassetpropertyvalue(  
        iv_assetid = iv_asset_id  
        iv_propertyid = iv_property_id  
    ). " oo_result is returned for testing purposes. "  
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.  
CATCH /aws1/cx_iosresourcenotfoundex.  
    MESSAGE 'Asset or property does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetAssetPropertyValue](#) in *AWS SDK for SAP ABAP API reference*.

ListAssetModelProperties

The following code example shows how to use ListAssetModelProperties.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->listassetmodelproperties(  
        iv_assetmodelid = iv_asset_model_id  
    ). " oo_result is returned for testing purposes. "  
    DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).  
    MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.
```

```
MESSAGE 'Unable to list asset model properties.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAssetModelProperties](#) in *AWS SDK for SAP ABAP API reference*.

ListAssetModels

The following code example shows how to use ListAssetModels.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing  
purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
    CATCH /aws1/cx_rt_generic.  
        MESSAGE 'Unable to list asset models.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListAssetModels](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Keyspaces examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Keyspaces.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateKeyspace

The following code example shows how to use CreateKeyspace.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kys->createkeyspace(  
        iv_keyspacename = iv_keyspace_name ).  
    MESSAGE 'Keyspace created successfully.' TYPE 'I'.  
CATCH /aws1/cx_kysconflictexception.  
    MESSAGE 'Keyspace already exists.' TYPE 'I'.  
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateKeyspace](#) in *AWS SDK for SAP ABAP API reference*.

CreateTable

The following code example shows how to use CreateTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Define schema with columns
  DATA(lt_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
  ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'title' iv_type = 'text' ) )
  ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'year' iv_type = 'int' ) )
  ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'release_date' iv_type =
'timestamp' ) )
  ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'plot' iv_type = 'text' ) )
  ).

  " Define partition keys
  DATA(lt_partition_keys) = VALUE /aws1/
cl_kyspartitionkey=>tt_partitionkeylist(
  ( NEW /aws1/cl_kyspartitionkey( iv_name = 'year' ) )
  ( NEW /aws1/cl_kyspartitionkey( iv_name = 'title' ) )
  ).

  " Create schema definition
  DATA(lo_schema) = NEW /aws1/cl_kysschemadefinition(
    it_allcolumns = lt_columns
    it_partitionkeys = lt_partition_keys ).

  " Enable point-in-time recovery
  DATA(lo_pitr) = NEW /aws1/cl_kyspointintimerec(
    iv_status = 'ENABLED' ).

  oo_result = lo_kys->createtable(
    iv_keyspacename = iv_keyspace_name
    iv_tablename = iv_table_name
    io_schemadefinition = lo_schema
    io_pointintimerecovery = lo_pitr ).
  MESSAGE 'Table created successfully.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).

```

```
DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [CreateTable](#) in *AWS SDK for SAP ABAP API reference*.

DeleteKeyspace

The following code example shows how to use DeleteKeyspace.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_kys->deletekeyspace(
    iv_keyspacename = iv_keyspace_name ).
  MESSAGE 'Keyspace deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteKeyspace](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTable

The following code example shows how to use DeleteTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_kys->deletetable(
    iv_keyspacename = iv_keyspace_name
    iv_tablename = iv_table_name ).
  MESSAGE 'Table deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteTable](#) in *AWS SDK for SAP ABAP API reference*.

GetKeyspace

The following code example shows how to use GetKeyspace.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_kys->getkeyspace(
    iv_keyspacename = iv_keyspace_name ).
  MESSAGE 'Keyspace retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_kysresourcenotfoundex.
  MESSAGE 'Keyspace does not exist.' TYPE 'I'.
```

```

    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
      DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
      MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- For API details, see [GetKeyspace](#) in *AWS SDK for SAP ABAP API reference*.

GetTable

The following code example shows how to use GetTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_kys->gettable(
    iv_keyspacename = iv_keyspace_name
    iv_tablename = iv_table_name ).
  MESSAGE 'Table information retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_kysresourcenotfoundex.
  MESSAGE 'Table does not exist.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- For API details, see [GetTable](#) in *AWS SDK for SAP ABAP API reference*.

ListKeyspaces

The following code example shows how to use ListKeyspaces.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kys->listkeyspaces(  
        iv_maxresults = iv_max_results ).  
    MESSAGE 'Keyspaces listed successfully.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListKeyspaces](#) in *AWS SDK for SAP ABAP API reference*.

ListTables

The following code example shows how to use ListTables.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kys->listtables(  
        iv_keyspacename = iv_keyspace_name ).  
    MESSAGE 'Tables listed successfully.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|. 
```

```
MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- For API details, see [ListTables](#) in *AWS SDK for SAP ABAP API reference*.

RestoreTable

The following code example shows how to use RestoreTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_kys->restoretable(
        iv_sourcekeyspacename = iv_source_keyspace_name
        iv_sourcetablename = iv_source_table_name
        iv_targetkeyspacename = iv_target_keyspace_name
        iv_targettablename = iv_target_table_name
        iv_restoretimestamp = iv_restore_timestamp ).
    MESSAGE 'Table restore initiated successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- For API details, see [RestoreTable](#) in *AWS SDK for SAP ABAP API reference*.

UpdateTable

The following code example shows how to use UpdateTable.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " Add a new column to track watched movies
    DATA(lt_add_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
    ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'watched' iv_type =
'boolean' ) )
    ).

    oo_result = lo_kys->updatetable(
        iv_keyspacename = iv_keyspace_name
        iv_tablename = iv_table_name
        it_addcolumns = lt_add_columns ).
    MESSAGE 'Table updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- For API details, see [UpdateTable](#) in *AWS SDK for SAP ABAP API reference*.

Kinesis examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Kinesis.

Basics are code examples that show you how to perform the essential operations within a service.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Basics](#)
- [Actions](#)

Basics

Learn the basics

The following code example shows how to:

- Create a stream and put a record in it.
- Create a shard iterator.
- Read the record, then clean up resources.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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_knslimitexceeddex.
        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_knskmsstrottlingex.

```

```

    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexclex.
    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_knsprovthruputexclex.
    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_knskmsstrottingex.
  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.

"Delete stream."
TRY.
  lo_kns->deletestream(
    iv_streamname = iv_stream_name ).
  MESSAGE 'Stream deleted.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceedex.
  MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex.
  MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CreateStream](#)
 - [DeleteStream](#)
 - [GetRecords](#)
 - [GetShardIterator](#)
 - [PutRecord](#)

Actions

CreateStream

The following code example shows how to use CreateStream.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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_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.
```

- For API details, see [CreateStream](#) in *AWS SDK for SAP ABAP API reference*.

DeleteStream

The following code example shows how to use DeleteStream.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
lo_kns->deletestream(  
    iv_streamname = iv_stream_name ).  
MESSAGE 'Stream deleted.' TYPE 'I'.  
CATCH /aws1/cx_knslimitexceedex.  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
CATCH /aws1/cx_knsresourceinuseex.  
    MESSAGE 'The request processing has failed because the resource is in use.'  
TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteStream](#) in *AWS SDK for SAP ABAP API reference*.

DescribeStream

The following code example shows how to use DescribeStream.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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_knslimitexceedex.  
        MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
    CATCH /aws1/cx_knsresourceindex.  
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeStream](#) in *AWS SDK for SAP ABAP API reference*.

GetRecords

The following code example shows how to use GetRecords.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    oo_result = lo_kns->getrecords(           " oo_result is returned for
testing purposes. "
        iv_sharditerator = iv_shard_iterator ).
    DATA(lt_records) = oo_result->get_records( ).
    MESSAGE 'Record retrieved.' TYPE 'I'.
CATCH /aws1/cx_knsexpirediteratorex.
    MESSAGE 'Iterator expired.' TYPE 'E'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex.
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex.
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired.
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmsstrottlingex.
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- For API details, see [GetRecords](#) in *AWS SDK for SAP ABAP API reference*.

ListStreams

The following code example shows how to use ListStreams.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kns->liststreams(          " oo_result is returned for testing  
purposes. "  
        "Set Limit to specify that a maximum of streams should be returned."  
        iv_limit = iv_limit ).  
    DATA(lt_streams) = oo_result->get_streamnames( ).  
    MESSAGE 'Streams listed.' TYPE 'I'.  
CATCH /aws1/cx_knslimitexceeddex.  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListStreams](#) in *AWS SDK for SAP ABAP API reference*.

PutRecord

The following code example shows how to use PutRecord.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    oo_result = lo_kns->putrecord(           " oo_result is returned for
testing purposes. "
        iv_streamname = iv_stream_name
        iv_data       = iv_data
        iv_partitionkey = iv_partition_key ).
    MESSAGE 'Record created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex.
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex.
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex.
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex.
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired.
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmssthrrottlingex.
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutRecord](#) in *AWS SDK for SAP ABAP API reference*.

RegisterStreamConsumer

The following code example shows how to use RegisterStreamConsumer.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_kns->registerstreamconsumer(      " oo_result is returned
for testing purposes. "
        iv_streamarn = iv_stream_arn
        iv_consumername = iv_consumer_name ).
    MESSAGE 'Stream consumer registered.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_sgmresource-limitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
CATCH /aws1/cx_sgmresource-inuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresource-notfound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- For API details, see [RegisterStreamConsumer](#) in *AWS SDK for SAP ABAP API reference*.

AWS KMS examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with AWS KMS.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Get started](#)
- [Actions](#)

Get started

Hello AWS KMS

The following code example shows how to get started using AWS Key Management Service.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kms->listkeys( ).  
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListKeys](#) in *AWS SDK for SAP ABAP API reference*.

Actions

CreateAlias

The following code example shows how to use CreateAlias.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_alias_name = 'alias/my-key-alias'
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  lo_kms->createalias(
    iv_aliasname = iv_alias_name
    iv_targetkeyid = iv_key_id
  ).
  MESSAGE 'Alias created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsalreadyexistsex.
  MESSAGE 'Alias already exists.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidaliasnameex.
  MESSAGE 'Invalid alias name.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [CreateAlias](#) in *AWS SDK for SAP ABAP API reference*.

CreateGrant

The following code example shows how to use CreateGrant.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'
  " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'
  oo_result = lo_kms->creategrant(
    iv_keyid = iv_key_id
    iv_granteeprincipal = iv_grantee_principal
    it_operations = it_operations
  ).
  MESSAGE 'Grant created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
  MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateGrant](#) in *AWS SDK for SAP ABAP API reference*.

CreateKey

The following code example shows how to use CreateKey.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_description = 'Created by the AWS SDK for SAP ABAP'
  oo_result = lo_kms->createkey( iv_description = iv_description ).
  MESSAGE 'KMS key created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
CATCH /aws1/cx_kmslimitexceeddex.

```

```
MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateKey](#) in *AWS SDK for SAP ABAP API reference*.

Decrypt

The following code example shows how to use Decrypt.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_ciphertext_blob contains the encrypted data  
    oo_result = lo_kms->decrypt(  
        iv_keyid = iv_key_id  
        iv_ciphertextblob = iv_ciphertext_blob  
    ).  
    MESSAGE 'Text decrypted successfully.' TYPE 'I'.  
    CATCH /aws1/cx_kmsdisabledexception.  
        MESSAGE 'The key is disabled.' TYPE 'E'.  
    CATCH /aws1/cx_kmsincorrectkeyex.  
        MESSAGE 'Incorrect key for decryption.' TYPE 'E'.  
    CATCH /aws1/cx_kmsnotfoundexception.  
        MESSAGE 'Key not found.' TYPE 'E'.  
    CATCH /aws1/cx_kmskmsinternalex.  
        MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [Decrypt](#) in *AWS SDK for SAP ABAP API reference*.

DeleteAlias

The following code example shows how to use DeleteAlias.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_alias_name = 'alias/my-key-alias'  
    lo_kms->deletealias( iv_aliasname = iv_alias_name ).  
    MESSAGE 'Alias deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Alias not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteAlias](#) in *AWS SDK for SAP ABAP API reference*.

DescribeKey

The following code example shows how to use DescribeKey.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```

    oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).
    DATA(lo_key) = oo_result->get_keymetadata( ).
    MESSAGE 'Retrieved key information successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [DescribeKey](#) in *AWS SDK for SAP ABAP API reference*.

DisableKey

The following code example shows how to use `DisableKey`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  lo_kms->disablekey( iv_keyid = iv_key_id ).
  MESSAGE 'KMS key disabled successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [DisableKey](#) in *AWS SDK for SAP ABAP API reference*.

EnableKey

The following code example shows how to use `EnableKey`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekey( iv_keyid = iv_key_id ).  
    MESSAGE 'KMS key enabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [EnableKey](#) in *AWS SDK for SAP ABAP API reference*.

EnableKeyRotation

The following code example shows how to use `EnableKeyRotation`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).  
    MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsdisabledexception.
```

```

    MESSAGE 'The key is disabled.' TYPE 'E'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmsunsupportedopex.
    MESSAGE 'Operation not supported for this key.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [EnableKeyRotation](#) in *AWS SDK for SAP ABAP API reference*.

Encrypt

The following code example shows how to use Encrypt.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_plaintext contains the data to encrypt
  oo_result = lo_kms->encrypt(
    iv_keyid = iv_key_id
    iv_plaintext = iv_plaintext
  ).
  MESSAGE 'Text encrypted successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [Encrypt](#) in *AWS SDK for SAP ABAP API reference*.

GenerateDataKey

The following code example shows how to use `GenerateDataKey`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_keyspec = 'AES_256'
  oo_result = lo_kms->generatedatakey(
    iv_keyid = iv_key_id
    iv_keyspec = 'AES_256'
  ).
  MESSAGE 'Data key generated successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
  MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GenerateDataKey](#) in *AWS SDK for SAP ABAP API reference*.

GetKeyPolicy

The following code example shows how to use `GetKeyPolicy`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  oo_result = lo_kms->getkeypolicy(
    iv_keyid = iv_key_id
    iv_policyname = 'default'
  ).
  MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetKeyPolicy](#) in *AWS SDK for SAP ABAP API reference*.

ListAliases

The following code example shows how to use ListAliases.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  oo_result = lo_kms->listaliases( ).
```

```
MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.
CATCH /aws1/cx_kmskmsinternalex.
MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [ListAliases](#) in *AWS SDK for SAP ABAP API reference*.

ListGrants

The following code example shows how to use ListGrants.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).
  MESSAGE 'Retrieved grants list.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [ListGrants](#) in *AWS SDK for SAP ABAP API reference*.

ListKeyPolicies

The following code example shows how to use ListKeyPolicies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).  
    MESSAGE 'Retrieved key policies list.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListKeyPolicies](#) in *AWS SDK for SAP ABAP API reference*.

ListKeys

The following code example shows how to use ListKeys.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_kms->listkeys( ).  
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListKeys](#) in *AWS SDK for SAP ABAP API reference*.

PutKeyPolicy

The following code example shows how to use PutKeyPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_policy = '{"Version":"2012-10-17",          "Statement": [...]}'
  lo_kms->putkeypolicy(
    iv_keyid = iv_key_id
    iv_policyname = 'default'
    iv_policy = iv_policy
  ).
  MESSAGE 'Key policy updated successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsmalformedplydocex.
  MESSAGE 'Malformed policy document.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [PutKeyPolicy](#) in *AWS SDK for SAP ABAP API reference*.

ReEncrypt

The following code example shows how to use ReEncrypt.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    " iv_source_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_destination_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
    " iv_ciphertext_blob contains the encrypted data
    oo_result = lo_kms->reencrypt(
        iv_sourcekeyid = iv_source_key_id
        iv_destinationkeyid = iv_destination_key_id
        iv_ciphertextblob = iv_ciphertext_blob
    ).
    MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.
    CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
    CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.
    CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [ReEncrypt](#) in *AWS SDK for SAP ABAP API reference*.

RetireGrant

The following code example shows how to use RetireGrant.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " iv_grant_token = 'AQpAM2RhZ...'  
  lo_kms->retiregrant( iv_granttoken = iv_grant_token ).  
  MESSAGE 'Grant retired successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
  MESSAGE 'Grant not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsinvgranttokenex.  
  MESSAGE 'Invalid grant token.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
  MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [RetireGrant](#) in *AWS SDK for SAP ABAP API reference*.

RevokeGrant

The following code example shows how to use RevokeGrant.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
  " iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'
```

```

lo_kms->revokegrant(
  iv_keyid = iv_key_id
  iv_grantid = iv_grant_id
).
MESSAGE 'Grant revoked successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Grant or key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidgrantidex.
  MESSAGE 'Invalid grant ID.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [RevokeGrant](#) in *AWS SDK for SAP ABAP API reference*.

ScheduleKeyDeletion

The following code example shows how to use ScheduleKeyDeletion.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_pending_window_days = 7
  oo_result = lo_kms->schedulekeydeletion(
    iv_keyid = iv_key_id
    iv_pendingwindowdays = iv_pending_window_days
  ).
  MESSAGE 'Key scheduled for deletion.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.

```

```
ENDTRY.
```

- For API details, see [ScheduleKeyDeletion](#) in *AWS SDK for SAP ABAP API reference*.

Sign

The following code example shows how to use Sign.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
    " iv_message contains the message to sign
    " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
    oo_result = lo_kms->sign(
        iv_keyid = iv_key_id
        iv_message = iv_message
        iv_signingalgorithm = iv_signing_algorithm
    ).
    MESSAGE 'Message signed successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidkeyusageex.
    MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [Sign](#) in *AWS SDK for SAP ABAP API reference*.

TagResource

The following code example shows how to use TagResource.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_tag_key = 'Environment'
  " iv_tag_value = 'Production'
  APPEND NEW /aws1/cl_kmstag(
    iv_tagkey = iv_tag_key
    iv_tagvalue = iv_tag_value
  ) TO lt_tags.

  lo_kms->tagresource(
    iv_keyid = iv_key_id
    it_tags = lt_tags
  ).
  MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmstagexception.
  MESSAGE 'Invalid tag format.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [TagResource](#) in *AWS SDK for SAP ABAP API reference*.

UpdateAlias

The following code example shows how to use UpdateAlias.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_alias_name = 'alias/my-key-alias'
  " iv_target_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
  lo_kms->updatealias(
    iv_aliasname = iv_alias_name
    iv_targetkeyid = iv_target_key_id
  ).
  MESSAGE 'Alias updated successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Alias or key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [UpdateAlias](#) in *AWS SDK for SAP ABAP API reference*.

Verify

The following code example shows how to use Verify.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
  " iv_message contains the original message
  " iv_signature contains the signature to verify
  " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
  oo_result = lo_kms->verify(
    iv_keyid = iv_key_id
    iv_message = iv_message
    iv_signature = iv_signature
    iv_signingalgorithm = iv_signing_algorithm
  ).
  DATA(lv_valid) = oo_result->get_signaturevalid( ).
  IF lv_valid = abap_true.
    MESSAGE 'Signature is valid.' TYPE 'I'.
  ELSE.
    MESSAGE 'Signature is invalid.' TYPE 'I'.
  ENDIF.
  CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinvalidsigex.
    MESSAGE 'Invalid signature.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [Verify](#) in *AWS SDK for SAP ABAP API reference*.

Lambda examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Lambda.

Basics are code examples that show you how to perform the essential operations within a service.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Basics](#)
- [Actions](#)

Basics

Learn the basics

The following code example shows how to:

- Create an IAM role and Lambda function, then upload handler code.
- Invoke the function with a single parameter and get results.
- Update the function code and configure with an environment variable.
- Invoke the function with new parameters and get results. Display the returned execution log.
- List the functions for your account, then clean up resources.

For more information, see [Create a Lambda function with the console](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    "Create an AWS Identity and Access Management (IAM) role that grants AWS  
Lambda permission to write to logs."  
    DATA(lv_policy_document) = `{` &&  
        "Version": "2012-10-17",           ` &&  
        "Statement": [ ` &&  
            `{` &&  
                "Effect": "Allow", ` &&
```

```

        "Action": [ ` &&
            "sts:AssumeRole" ` &&
        ], ` &&
        "Principal": { ` &&
            "Service": [ ` &&
                "lambda.amazonaws.com" ` &&
            ] ` &&
        } ` &&
    } ` &&
} ` &&
.

```

TRY.

```

DATA(lo_create_role_output) = lo_iam->createrole(
    iv_rolename = iv_role_name
    iv_assumerolepolicydocument = lv_policy_document
    iv_description = 'Grant lambda permission to write to logs' ).
DATA(lv_role_arn) = lo_create_role_output->get_role( )->get_arn( ).
MESSAGE 'IAM role created.' TYPE 'I'.
WAIT UP TO 10 SECONDS.           " Make sure that the IAM role is ready
for use. "
CATCH /aws1/cx_iamentityalrddyex.
    DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
    lv_role_arn = lo_role->get_role( )->get_arn( ).
CATCH /aws1/cx_iainvalidinputex.
    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_iainvalidinputex.
    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'.
    ENDRTRY.

    " Create a Lambda function and upload handler code. "
    " Lambda function performs 'increment' action on a number. "
    TRY.
        lo_lmd->createfunction(
            iv_functionname = iv_function_name
            iv_runtime = `python3.9`
            iv_role = lv_role_arn
            iv_handler = iv_handler
            io_code = io_initial_zip_file
            iv_description = 'AWS Lambda code example' ).
        MESSAGE 'Lambda function created.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodestorageexcde.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDRTRY.

    " Verify the function is in Active state "
    WHILE lo_lmd->getfunction( iv_functionname = iv_function_name )->
>get_configuration( )->ask_state( ) <> 'Active'.
        IF sy-index = 10.
            EXIT.                " Maximum 10 seconds. "
        ENDIF.
        WAIT UP TO 1 SECONDS.
    ENDWHILE.

    "Invoke the function with a single parameter and get results."
    TRY.
        DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
            `{` &&
            ` "action": "increment",` &&
            ` "number": 10` &&
            `}` ).
        DATA(lo_initial_invoke_output) = lo_lmd->invoke(
            iv_functionname = iv_function_name
            iv_payload = lv_json ).
        ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
        " ov_initial_invoke_payload is returned for testing purposes. "
    
```

```

        DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
        CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
lo_writer_json.
        DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
        CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
        CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        CATCH /aws1/cx_lmdunsuppmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
        ENDTRY.

        " Update the function code and configure its Lambda environment with an
environment variable. "
        " Lambda function is updated to perform 'decrement' action also. "
        TRY.
            lo_lmd->updatefunctioncode(
                iv_functionname = iv_function_name
                iv_zipfile = io_updated_zip_file ).
            WAIT UP TO 10 SECONDS.          " Make sure that the update is
completed. "
            MESSAGE 'Lambda function code updated.' TYPE 'I'.
            CATCH /aws1/cx_lmdcodestorageexcex.
            MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
            CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
            CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        ENDTRY.

        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'.
ENDTRY.

" Delete the Lambda function. "
TRY.
    lo_lmd->deletefunction( iv_functionname = iv_function_name ).
    MESSAGE 'Lambda function deleted.' TYPE 'I'.
CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'W'.
ENDTRY.

" Detach role policy. "
TRY.
    lo_iam->detachrolepolicy(
        iv_rolename = iv_role_name
        iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
    MESSAGE 'Detached policy from the IAM role.' TYPE 'I'.
CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.
CATCH /aws1/cx_iamplynnotattachableex.
    MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
CATCH /aws1/cx_iamunmodableentityex.
    MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
ENDTRY.

" Delete the IAM role. "
TRY.
```

```
    lo_iam->deleterole( iv_rolename = iv_role_name ).  
    MESSAGE 'IAM role deleted.' TYPE 'I'.  
    CATCH /aws1/cx_iamnosuchentityex.  
        MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.  
    CATCH /aws1/cx_iamunmodableentityex.  
        MESSAGE 'Service that depends on the service-linked role is not  
modifiable.' TYPE 'E'.  
    ENDTRY.  
  
    CATCH /aws1/cx_rt_service_generic INTO lo_exception.  
        DATA(lv_error) = lo_exception->get_longtext( ).  
        MESSAGE lv_error TYPE 'E'.  
    ENDTRY.
```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CreateFunction](#)
 - [DeleteFunction](#)
 - [GetFunction](#)
 - [Invoke](#)
 - [ListFunctions](#)
 - [UpdateFunctionCode](#)
 - [UpdateFunctionConfiguration](#)

Actions

CreateFunction

The following code example shows how to use CreateFunction.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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_lmdcodestorageexc dex.
    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.
```

- For API details, see [CreateFunction](#) in *AWS SDK for SAP ABAP API reference*.

DeleteFunction

The following code example shows how to use DeleteFunction.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DeleteFunction](#) in *AWS SDK for SAP ABAP API reference*.

GetFunction

The following code example shows how to use GetFunction.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [GetFunction](#) in *AWS SDK for SAP ABAP API reference*.

Invoke

The following code example shows how to use Invoke.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
        `{` &&
        ` "action": "increment",` &&
        ` "number": 10` &&
        `}` ).
    oo_result = lo_lmd->invoke(
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.

```

- For API details, see [Invoke](#) in *AWS SDK for SAP ABAP API reference*.

ListFunctions

The following code example shows how to use ListFunctions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [ListFunctions](#) in *AWS SDK for SAP ABAP API reference*.

UpdateFunctionCode

The following code example shows how to use UpdateFunctionCode.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

  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.

```

- For API details, see [UpdateFunctionCode](#) in *AWS SDK for SAP ABAP API reference*.

UpdateFunctionConfiguration

The following code example shows how to use UpdateFunctionConfiguration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_lmd->updatefunctionconfiguration(      " oo_result is returned
for testing purposes. "
    iv_functionname = iv_function_name
    iv_runtime = iv_runtime
    iv_description = 'Updated Lambda function'
    iv_memorysize = iv_memory_size ).

  MESSAGE 'Lambda function configuration/settings updated.' TYPE 'I'.
CATCH /aws1/cx_lmdcodesigningcfgno00.
  MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdcodeverification00.

```

```
MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidcodesigex.
MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.
```

- For API details, see [UpdateFunctionConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

Organizations examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Organizations.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

AttachPolicy

The following code example shows how to use AttachPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_org->attachpolicy(
    iv_policyid = iv_policy_id
    iv_targetid = iv_target_id ).
  MESSAGE 'Policy attached to target.' TYPE 'I'.
CATCH /aws1/cx_orgaccessdeniedex.
  MESSAGE 'You do not have permission to attach the policy.' TYPE 'E'.
CATCH /aws1/cx_orgpolicynotfoundex.
  MESSAGE 'The specified policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_orgtargetnotfoundex.
  MESSAGE 'The specified target does not exist.' TYPE 'E'.
CATCH /aws1/cx_orgduplicateplyatta00.
  MESSAGE 'The policy is already attached to the target.' TYPE 'E'.
ENDTRY.
```

- For API details, see [AttachPolicy](#) in *AWS SDK for SAP ABAP API reference*.

CreatePolicy

The following code example shows how to use CreatePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
oo_result = lo_org->createpolicy(      " oo_result is returned for testing
purposes. "
    iv_name          = iv_policy_name
    iv_description   = iv_policy_description
    iv_content       = iv_policy_content
    iv_type          = iv_policy_type ).
MESSAGE 'Policy created.' TYPE 'I'.
CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to create a policy.' TYPE 'E'.
CATCH /aws1/cx_orgduplicatepolicyex.
    MESSAGE 'A policy with this name already exists.' TYPE 'E'.
CATCH /aws1/cx_orgmalformedplydocex.
    MESSAGE 'The policy content is malformed.' TYPE 'E'.
ENDTRY.
```

- For API details, see [CreatePolicy](#) in *AWS SDK for SAP ABAP API reference*.

DeletePolicy

The following code example shows how to use DeletePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    lo_org->deletepolicy(
        iv_policyid = iv_policy_id ).
    MESSAGE 'Policy deleted.' TYPE 'I'.
CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to delete the policy.' TYPE 'E'.
CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
CATCH /aws1/cx_orgpolicyinuseex.
    MESSAGE 'The policy is still attached to one or more targets.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DeletePolicy](#) in *AWS SDK for SAP ABAP API reference*.

DescribePolicy

The following code example shows how to use DescribePolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_org->describepolicy(      " oo_result is returned for testing  
purposes. "  
        iv_policyid = iv_policy_id ).  
    DATA(lo_policy) = oo_result->get_policy( ).  
    MESSAGE 'Retrieved policy details.' TYPE 'I'.  
CATCH /aws1/cx_orgaccessdeniedex.  
    MESSAGE 'You do not have permission to describe the policy.' TYPE 'E'.  
CATCH /aws1/cx_orgpolicynotfoundex.  
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribePolicy](#) in *AWS SDK for SAP ABAP API reference*.

DetachPolicy

The following code example shows how to use DetachPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_org->detachpolicy(  
    iv_policyid = iv_policy_id  
    iv_targetid = iv_target_id ).  
  MESSAGE 'Policy detached from target.' TYPE 'I'.  
CATCH /aws1/cx_orgaccessdeniedex.  
  MESSAGE 'You do not have permission to detach the policy.' TYPE 'E'.  
CATCH /aws1/cx_orgpolicynotfoundex.  
  MESSAGE 'The specified policy does not exist.' TYPE 'E'.  
CATCH /aws1/cx_orgtargetnotfoundex.  
  MESSAGE 'The specified target does not exist.' TYPE 'E'.  
CATCH /aws1/cx_orgpolicynotattex.  
  MESSAGE 'The policy is not attached to the target.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetachPolicy](#) in *AWS SDK for SAP ABAP API reference*.

ListPolicies

The following code example shows how to use ListPolicies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
        oo_result = lo_org->listpolicies(          " oo_result is returned for testing
purposes. "
        iv_filter = iv_filter ).
        DATA(lt_policies) = oo_result->get_policies( ).
        MESSAGE 'Retrieved list of policies.' TYPE 'I'.
    CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to list policies.' TYPE 'E'.
    CATCH /aws1/cx_orgawsorgsnotinuseex.
        MESSAGE 'Your account is not a member of an organization.' TYPE 'E'.
    ENDTRY.
```

- For API details, see [ListPolicies](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Pinpoint examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Pinpoint.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

SendMessages

The following code example shows how to use SendMessages.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Send an email message.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_mesagerequest = NEW /aws1/cl_pptmesagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_html_message
          )
          io_textpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_text_message
          )
        )
      )
    )
  )
)
```

```

)
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Email message sent successfully.' TYPE 'I'.

```

Send an SMS message.

```

" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_body = iv_message
        iv_message_type = iv_message_type
        iv_OriginationNumber = iv_OriginationNumber
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

```

```
MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

Send an email message with an existing email template.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
  )
  io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
    io_emailtemplate = NEW /aws1/cl_ppttemplate(
      iv_name = iv_template_name
      iv_version = iv_template_version
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

Send a text message with an existing SMS template.

```

" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_message_type = iv_message_type
        iv_OriginationNumber = iv_OriginationNumber
      )
    )
  )
  io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
    io_smstemplate = NEW /aws1/cl_ppttemplate(
      iv_name = iv_template_name
      iv_version = iv_template_version
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.

```

- For API details, see [SendMessages](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Pinpoint SMS and Voice API examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Pinpoint SMS and Voice API.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateConfigurationSet

The following code example shows how to use CreateConfigurationSet.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create a new configuration set
  lo_pps->createconfigurationset(
    iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
  ).

  MESSAGE 'Configuration set created successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
  MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
```

```

    RAISE EXCEPTION lo_already_exists_ex.
  CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
  CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
    MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_limit_exceeded_ex.
  CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- For API details, see [CreateConfigurationSet](#) in *AWS SDK for SAP ABAP API reference*.

CreateConfigurationSetEventDestination

The following code example shows how to use `CreateConfigurationSetEventDestination`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Create event destination for the configuration set
  lo_pps->createconfseteventdst(
    iv_configurationsetname = iv_configuration_set_name      " e.g., 'my-config-
set'
    iv_eventdestinationname = iv_event_destination_name     " e.g., 'my-event-
dest'
    io_eventdestination = io_event_destination
  ).

  MESSAGE 'Event destination created successfully.' TYPE 'I'.

```

```

CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
  MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_already_exists_ex.
CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestsex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
  MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
  MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_limit_exceeded_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
  MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- For API details, see [CreateConfigurationSetEventDestination](#) in *AWS SDK for SAP ABAP API reference*.

DeleteConfigurationSet

The following code example shows how to use DeleteConfigurationSet.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Delete the configuration set
  lo_pps->deleteconfigurationset(
    iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set '

```

```
    ).  
  
    MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.  
  
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).  
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_internal_error_ex.  
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).  
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.  
    RAISE EXCEPTION lo_too_many_requests_ex.  
ENDTRY.
```

- For API details, see [DeleteConfigurationSet](#) in *AWS SDK for SAP ABAP API reference*.

DeleteConfigurationSetEventDestination

The following code example shows how to use DeleteConfigurationSetEventDestination.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " Delete the event destination  
    lo_pps->deleteconfseteventdst(  
        iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-  
set'  
        iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-  
dest'  
    ).
```

```

    MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- For API details, see [DeleteConfigurationSetEventDestination](#) in *AWS SDK for SAP ABAP API reference*.

GetConfigurationSetEventDestinations

The following code example shows how to use `GetConfigurationSetEventDestinations`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " Get event destinations for the configuration set
    oo_result = lo_pps->getconfseteventdestinations(
        iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
    ).

    " Process the event destinations
    LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).

```

```

DATA(lv_dest_name) = lo_event_dest->get_name( ).
DATA(lv_enabled) = lo_event_dest->get_enabled( ).

MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
TYPE 'I'.

" Check for CloudWatch Logs destination
DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
IF lo_cloudwatch_dest IS NOT INITIAL.
    DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
    MESSAGE | CloudWatch Logs destination: { lv_log_group_arn }| TYPE 'I'.
ENDIF.

" Check for Kinesis Firehose destination
DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
IF lo_firehose_dest IS NOT INITIAL.
    DATA(lv_delivery_stream) = lo_firehose_dest->get_deliverystreamarn( ).
    MESSAGE | Kinesis Firehose destination: { lv_delivery_stream }| TYPE
'I'.
ENDIF.

" Check for SNS destination
DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
IF lo_sns_dest IS NOT INITIAL.
    DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
    MESSAGE | SNS destination: { lv_topic_arn }| TYPE 'I'.
ENDIF.
ENDLOOP.

CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- For API details, see [GetConfigurationSetEventDestinations](#) in *AWS SDK for SAP ABAP API reference*.

ListConfigurationSets

The following code example shows how to use ListConfigurationSets.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " List all configuration sets
  oo_result = lo_pps->listconfigurationsets(
    iv_nexttoken = iv_next_token      " Optional: Token for pagination
    iv_pagesize = iv_page_size       " Optional: Number of results per page,
e.g., '10'
  ).

  " Process the configuration sets
  LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
    DATA(lv_config_set_name) = lo_config_set->get_value( ).
    MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
  ENDLLOOP.

  " Check if there are more results
  DATA(lv_next_token) = oo_result->get_nexttoken( ).
  IF lv_next_token IS NOT INITIAL.
    MESSAGE |More results available. Next token: { lv_next_token }| TYPE 'I'.
  ENDIF.

CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
  MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).

```

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

- For API details, see [ListConfigurationSets](#) in *AWS SDK for SAP ABAP API reference*.

SendVoiceMessage

The following code example shows how to use `SendVoiceMessage`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create SSML message type object with voice parameters
  DATA(lo_ssml_message) = NEW /aws1/cl_ppssssmlmessagetype(
    iv_languagecode = iv_language_code      " e.g., 'en-US'
    iv_voiceid = iv_voice_id                " e.g., 'Matthew'
    iv_text = iv_ssml_message              " SSML formatted message text
  ).

  " Create voice message content with the SSML message
  DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
    io_ssmlmessage = lo_ssml_message
  ).

  " Send the voice message
  DATA(lo_result) = lo_pps->sendvoicemessage(
    iv_OriginationPhoneNumber = iv_Origination_number " e.g., '+12065550110'
    iv_callerid = iv_caller_id                       " e.g., '+12065550199'
    iv_destinationPhoneNumber = iv_destination_number " e.g., '+12065550142'
    io_content = lo_content
  ).

  " Retrieve the message ID from the response
```

```

    ov_message_id = lo_result->get_messageid( ).

    MESSAGE 'Voice message sent successfully.' TYPE 'I'.

CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
    MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
    MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.

```

- For API details, see [SendVoiceMessage](#) in *AWS SDK for SAP ABAP API reference*.

UpdateConfigurationSetEventDestination

The following code example shows how to use UpdateConfigurationSetEventDestination.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " Update the event destination
    lo_pps->updateconfseteventdst(
        iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        iv_eventdestinationname = iv_event_destination_name   " e.g., 'my-event-
dest'
        io_eventdestination = io_event_destination
    ).

    MESSAGE 'Event destination updated successfully.' TYPE 'I'.

```

```
CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
  MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_internal_error_ex.
CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
  MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
  RAISE EXCEPTION lo_too_many_requests_ex.
ENDTRY.
```

- For API details, see [UpdateConfigurationSetEventDestination](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Polly examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Polly.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

DeleteLexicon

The following code example shows how to use DeleteLexicon.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ply->deletelexicon( iv_name ).  
  MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_plylexiconnotfoundex.  
  MESSAGE 'Lexicon not found.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
  MESSAGE 'Service failure occurred.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteLexicon](#) in *AWS SDK for SAP ABAP API reference*.

DescribeVoices

The following code example shows how to use DescribeVoices.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Only pass optional parameters if they have values  
  IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.  
    oo_result = lo_ply->describevoices(  
      iv_engine = iv_engine  
      iv_languagecode = iv_language ).  
  ELSEIF iv_engine IS NOT INITIAL.
```

```
        oo_result = lo_ply->describevoices(
            iv_engine = iv_engine ).
    ELSEIF iv_language IS NOT INITIAL.
        oo_result = lo_ply->describevoices(
            iv_languagecode = iv_language ).
    ELSE.
        oo_result = lo_ply->describevoices( ).
    ENDIF.
    MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'The NextToken is invalid.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DescribeVoices](#) in *AWS SDK for SAP ABAP API reference*.

GetLexicon

The following code example shows how to use GetLexicon.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_ply->getlexicon( iv_name ).
    DATA(lo_lexicon) = oo_result->get_lexicon( ).
    IF lo_lexicon IS BOUND.
        DATA(lv_lex_name) = lo_lexicon->get_name( ).
        MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.
    ENDIF.
CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
```

```
ENDTRY.
```

- For API details, see [GetLexicon](#) in *AWS SDK for SAP ABAP API reference*.

GetSpeechSynthesisTask

The following code example shows how to use `GetSpeechSynthesisTask`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).  
    DATA(lo_task) = oo_result->get_synthesistask( ).  
    IF lo_task IS BOUND.  
        DATA(lv_status) = lo_task->get_taskstatus( ).  
        MESSAGE |Task status: { lv_status }| TYPE 'I'.  
    ENDIF.  
    CATCH /aws1/cx_plyinvalidtaskidex.  
        MESSAGE 'Invalid task ID.' TYPE 'E'.  
    CATCH /aws1/cx_plyservicefailureex.  
        MESSAGE 'Service failure occurred.' TYPE 'E'.  
    CATCH /aws1/cx_plysynthesistsknotf00.  
        MESSAGE 'Synthesis task not found.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetSpeechSynthesisTask](#) in *AWS SDK for SAP ABAP API reference*.

ListLexicons

The following code example shows how to use `ListLexicons`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_ply->listlexicons( ).
    DATA(lt_lexicons) = oo_result->get_lexicons( ).
    DATA(lv_count) = lines( lt_lexicons ).
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- For API details, see [ListLexicons](#) in *AWS SDK for SAP ABAP API reference*.

ListSpeechSynthesisTasks

The following code example shows how to use ListSpeechSynthesisTasks.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
            iv_maxresults = iv_max_results
```

```

        iv_status = iv_status ).
    ELSEIF iv_max_results IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
            iv_maxresults = iv_max_results ).
    ELSEIF iv_status IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
            iv_status = iv_status ).
    ELSE.
        oo_result = lo_ply->listspeechsynthesistasks( ).
    ENDIF.
    DATA(lt_tasks) = oo_result->get_synthesistasks( ).
    DATA(lv_count) = lines( lt_tasks ).
    MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.

```

- For API details, see [ListSpeechSynthesisTasks](#) in *AWS SDK for SAP ABAP API reference*.

PutLexicon

The following code example shows how to use PutLexicon.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    lo_ply->putlexicon(
        iv_name = iv_name
        iv_content = iv_content ).
    MESSAGE 'Lexicon created successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidlexiconex.
    MESSAGE 'Invalid lexicon.' TYPE 'E'.

```

```

CATCH /aws1/cx_plylexiconsizexc00.
  MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
CATCH /aws1/cx_plymaxlexemelength00.
  MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
CATCH /aws1/cx_plymaxlexiconsnoexc00.
  MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailure00.
  MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plyunsuppdpplsalpha00.
  MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
CATCH /aws1/cx_plyunsuppdpplslangu00.
  MESSAGE 'Unsupported PLS language.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutLexicon](#) in *AWS SDK for SAP ABAP API reference*.

StartSpeechSynthesisTask

The following code example shows how to use StartSpeechSynthesisTask.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " Only pass optional parameters if they have values
  IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
    oo_result = lo_ply->startspeechsynthesistask(
      iv_engine = iv_engine
      iv_outputformat = iv_audio_format
      iv_outputs3bucketname = iv_s3_bucket
      iv_outputs3keyprefix = iv_s3_key_prefix
      iv_text = iv_text
      iv_voiceid = iv_voice_id
      iv_languagecode = iv_lang_code ).
  ELSEIF iv_lang_code IS NOT INITIAL.

```

```
oo_result = lo_ply->startspeechsynthesistask(
  iv_engine = iv_engine
  iv_outputformat = iv_audio_format
  iv_outputs3bucketname = iv_s3_bucket
  iv_text = iv_text
  iv_voiceid = iv_voice_id
  iv_languagecode = iv_lang_code ).
ELSEIF iv_s3_key_prefix IS NOT INITIAL.
oo_result = lo_ply->startspeechsynthesistask(
  iv_engine = iv_engine
  iv_outputformat = iv_audio_format
  iv_outputs3bucketname = iv_s3_bucket
  iv_outputs3keyprefix = iv_s3_key_prefix
  iv_text = iv_text
  iv_voiceid = iv_voice_id ).
ELSE.
oo_result = lo_ply->startspeechsynthesistask(
  iv_engine = iv_engine
  iv_outputformat = iv_audio_format
  iv_outputs3bucketname = iv_s3_bucket
  iv_text = iv_text
  iv_voiceid = iv_voice_id ).
ENDIF.
MESSAGE 'Speech synthesis task started.' TYPE 'I'.
CATCH /aws1/cx_plyinvalids3bucketex.
  MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
CATCH /aws1/cx_plyinvalidssmllex.
  MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
  MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
  MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
  MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartSpeechSynthesisTask](#) in *AWS SDK for SAP ABAP API reference*.

SynthesizeSpeech

The following code example shows how to use SynthesizeSpeech.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Only pass optional language code if it has a value
  IF iv_lang_code IS NOT INITIAL.
    oo_result = lo_ply->synthesizespeech(
      iv_engine = iv_engine
      iv_outputformat = iv_output_fmt
      iv_text = iv_text
      iv_voiceid = iv_voice_id
      iv_languagecode = iv_lang_code ).
  ELSE.
    oo_result = lo_ply->synthesizespeech(
      iv_engine = iv_engine
      iv_outputformat = iv_output_fmt
      iv_text = iv_text
      iv_voiceid = iv_voice_id ).
  ENDIF.
  MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidssmlex.
  MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
  MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
  MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
  MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.
```

- For API details, see [SynthesizeSpeech](#) in *AWS SDK for SAP ABAP API reference*.

Amazon RDS examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon RDS.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateDBParameterGroup

The following code example shows how to use CreateDBParameterGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_dbparametergroupfamily = 'mysql8.0'
" iv_description           = 'My custom DB parameter group for MySQL 8.0'
TRY.
    oo_result = lo_rds->createdbparametergroup(
        iv_dbparametergroupname = iv_dbparametergroupname
        iv_dbparametergroupfamily = iv_dbparametergroupfamily
        iv_description           = iv_description ).
    MESSAGE 'DB parameter group created.' TYPE 'I'.
CATCH /aws1/cx_rdsdbparmgralrxfault.
    MESSAGE 'DB parameter group already exists.' TYPE 'I'.
```

```
CATCH /aws1/cx_rdsdbprmqrquotaexcd00.  
  MESSAGE 'DB parameter group quota exceeded.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [CreateDBParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

DeleteDBParameterGroup

The following code example shows how to use DeleteDBParameterGroup.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_dbparametergroupname = 'mydbparametergroup'  
TRY.  
  lo_rds->deletedbparametergroup(  
    iv_dbparametergroupname = iv_dbparametergroupname ).  
  MESSAGE 'DB parameter group deleted.' TYPE 'I'.  
CATCH /aws1/cx_rdsdbprmqrnotfndfault.  
  MESSAGE 'DB parameter group not found.' TYPE 'I'.  
CATCH /aws1/cx_rdsinvdbprmqrstatef00.  
  MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteDBParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBEngineVersions

The following code example shows how to use DescribeDBEngineVersions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

- For API details, see [DescribeDBEngineVersions](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBParameterGroups

The following code example shows how to use DescribeDBParameterGroups.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
  oo_result = lo_rds->describedbparametergroups(
    iv_dbparametergroupname = iv_dbparametergroupname ).
  MESSAGE 'DB parameter group retrieved.' TYPE 'I'.
```

```
CATCH /aws1/cx_rdsdbprimgrnotfndfault.
  MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- For API details, see [DescribeDBParameterGroups](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDBParameters

The following code example shows how to use DescribeDBParameters.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_source                 = 'user' (optional - filters by parameter source)
TRY.
  oo_result = lo_rds->describedbparameters(
    iv_dbparametergroupname = iv_dbparametergroupname
    iv_source                 = iv_source ).
  DATA(lv_param_count) = lines( oo_result->get_parameters( ) ).
  MESSAGE |Retrieved { lv_param_count } parameters.| TYPE 'I'.
CATCH /aws1/cx_rdsdbprimgrnotfndfault.
  MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- For API details, see [DescribeDBParameters](#) in *AWS SDK for SAP ABAP API reference*.

DescribeOrderableDBInstanceOptions

The following code example shows how to use DescribeOrderableDBInstanceOptions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_engine          = 'mysql'
" iv_engineversion   = '8.0.35'
TRY.
    oo_result = lo_rds->descrorderabledbinoptions(
        iv_engine          = iv_engine
        iv_engineversion   = iv_engineversion ).
    DATA(lv_option_count) = lines( oo_result->get_orderabledbinoptions( ) ).
    MESSAGE |Retrieved { lv_option_count } orderable DB instance options.| TYPE
'I'.
ENDTRY.
```

- For API details, see [DescribeOrderableDBInstanceOptions](#) in *AWS SDK for SAP ABAP API reference*.

ModifyDBParameterGroup

The following code example shows how to use `ModifyDBParameterGroup`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_dbparametergroupname = 'mydbparametergroup'
" it_parameters - table containing parameter objects with:
"   - parametername = 'max_connections'
"   - parametervalue = '100'
```

```
" - applymethod = 'immediate' or 'pending-reboot'  
TRY.  
    oo_result = lo_rds->modifydbparametergroup(  
        iv_dbparametergroupname = iv_dbparametergroupname  
        it_parameters             = it_parameters ).  
    MESSAGE 'DB parameter group modified.' TYPE 'I'.  
CATCH /aws1/cx_rdsdbprmginotfndfault.  
    MESSAGE 'DB parameter group not found.' TYPE 'I'.  
CATCH /aws1/cx_rdsinvdbprmgrstatef00.  
    MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [ModifyDBParameterGroup](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Redshift examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Redshift.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateCluster

The following code example shows how to use CreateCluster.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create the cluster.

```
TRY.
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'
  " Example values: iv_node_type = 'ra3.4xlarge'
  " Example values: iv_master_username = 'awsuser'
  " Example values: iv_master_password = 'AwsUser1000'
  " Example values: iv_publicly_accessible = abap_true
  " Example values: iv_number_of_nodes = 2
  oo_result = lo_rsh->createcluster(
    iv_clusteridentifier = iv_cluster_identifier
    iv_nodetype = iv_node_type
    iv_masterusername = iv_master_username
    iv_masteruserpassword = iv_master_password
    iv_publiclyaccessible = iv_publicly_accessible
    iv_numberofnodes = iv_number_of_nodes
  ).
  MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.
CATCH /aws1/cx_rshclustalrddyexfault.
  MESSAGE 'Cluster already exists.' TYPE 'I'.
CATCH /aws1/cx_rshclstquotaexcdfault.
  MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
ENDTRY.
```

- For API details, see [CreateCluster](#) in *AWS SDK for SAP ABAP API reference*.

DeleteCluster

The following code example shows how to use DeleteCluster.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Delete the cluster.

```
TRY.  
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'  
  lo_rsh->deletecluster(  
    iv_clusteridentifier = iv_cluster_identifier  
    iv_skipfinalclustersnapshot = abap_true  
  ).  
  MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_rshclustnotfoundfault.  
  MESSAGE 'Cluster not found.' TYPE 'I'.  
CATCH /aws1/cx_rshinvcluststatefault.  
  MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteCluster](#) in *AWS SDK for SAP ABAP API reference*.

DescribeClusters

The following code example shows how to use DescribeClusters.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Describe the cluster.

```
TRY.
```

```

" Example values: iv_cluster_identifier = 'my-redshift-cluster' (optional)
oo_result = lo_rsh->describeclusters(
  iv_clusteridentifier = iv_cluster_identifier
).
lt_clusters = oo_result->get_clusters( ).
lv_cluster_count = lines( lt_clusters ).
MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.

```

- For API details, see [DescribeClusters](#) in *AWS SDK for SAP ABAP API reference*.

DescribeStatement

The following code example shows how to use DescribeStatement.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
" Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx'
oo_result = lo_rsd->describestatement(
  iv_id = iv_statement_id
).
lv_status = oo_result->get_status( ).
MESSAGE |Statement status: { lv_status }| TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
MESSAGE 'Statement not found.' TYPE 'I'.
CATCH /aws1/cx_rsdinternalserverex.
MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.

```

- For API details, see [DescribeStatement](#) in *AWS SDK for SAP ABAP API reference*.

ExecuteStatement

The following code example shows how to use ExecuteStatement.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Example values: iv_cluster_identifider = 'redshift-cluster-movies'
  " Example values: iv_database_name = 'dev'
  " Example values: iv_user_name = 'awsuser'
  " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
  " Example values: it_parameter_list - SQL parameters for parameterized
queries

  " Only pass parameters if the list is not empty
  IF it_parameter_list IS NOT INITIAL.
    oo_result = lo_rsd->executestatement(
      iv_clusteridentifider = iv_cluster_identifider
      iv_database = iv_database_name
      iv_dbuser = iv_user_name
      iv_sql = iv_sql
      it_parameters = it_parameter_list
    ).
  ELSE.
    oo_result = lo_rsd->executestatement(
      iv_clusteridentifider = iv_cluster_identifider
      iv_database = iv_database_name
      iv_dbuser = iv_user_name
      iv_sql = iv_sql
    ).
  ENDIF.

  lv_statement_id = oo_result->get_id( ).
  MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
  CATCH /aws1/cx_rsdexecutestatementex.
    MESSAGE 'Statement execution error.' TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
```

```
MESSAGE 'Resource not found.' TYPE 'I'.
ENDTRY.
```

- For API details, see [ExecuteStatement](#) in *AWS SDK for SAP ABAP API reference*.

GetStatementResult

The following code example shows how to use `GetStatementResult`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Check the statement result.

```
TRY.
  " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx'
  " Handle pagination for large result sets

DO.
  lo_result_page = lo_rsd->getstatementresult(
    iv_id = iv_statement_id
    iv_nexttoken = lv_next_token
  ).

  " Collect records from this page
  lt_page_records = lo_result_page->get_records( ).
  APPEND LINES OF lt_page_records TO lt_all_records.

  " Check if there are more pages
  lv_next_token = lo_result_page->get_nexttoken( ).
  IF lv_next_token IS INITIAL.
    EXIT. " No more pages
  ENDIF.
ENDDO.

" For the last call, set oo_result for return value
```

```

    oo_result = lo_result_page.
    lv_record_count = lines( lt_all_records ).
    MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found or results not available.' TYPE 'I'.
  CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  ENDTRY.

```

- For API details, see [GetStatementResult](#) in *AWS SDK for SAP ABAP API reference*.

ListDatabases

The following code example shows how to use ListDatabases.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

  TRY.
    " Example values: iv_cluster_identifer = 'redshift-cluster-movies'
    " Example values: iv_database_name = 'dev'
    " Example values: iv_database_user = 'awsuser'
    oo_result = lo_rsd->listdatabases(
      iv_clusteridentifer = iv_cluster_identifer
      iv_database = iv_database_name
      iv_dbuser = iv_database_user
    ).
    lt_databases = oo_result->get_databases( ).
    lv_db_count = lines( lt_databases ).
    MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
  CATCH /aws1/cx_rsdatabaseconnex.
    MESSAGE 'Database connection error.' TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Cluster not found.' TYPE 'I'.
  ENDTRY.

```

- For API details, see [ListDatabases](#) in *AWS SDK for SAP ABAP API reference*.

ModifyCluster

The following code example shows how to use `ModifyCluster`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Modify a cluster.

```
TRY.  
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'  
  " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'  
  lo_rsh->modifycluster(  
    iv_clusteridentifier = iv_cluster_identifier  
    iv_preferredmaintenancewin00 = iv_pref_maintenance_wn  
  ).  
  MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.  
CATCH /aws1/cx_rshclustnotfoundfault.  
  MESSAGE 'Cluster not found.' TYPE 'I'.  
CATCH /aws1/cx_rshinvcluststatefault.  
  MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [ModifyCluster](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Rekognition examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Rekognition.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CompareFaces

The following code example shows how to use CompareFaces.

For more information, see [Comparing faces in images](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Create S3 object reference for the source image  
  DATA(lo_source_s3obj) = NEW /aws1/cl_reks3object(  
    iv_bucket = iv_source_s3_bucket  
    iv_name = iv_source_s3_key ).  
  
  " Create source image object  
  DATA(lo_source_image) = NEW /aws1/cl_rekimage(  
    io_s3object = lo_source_s3obj ).  
  
  " Create S3 object reference for the target image  
  DATA(lo_target_s3obj) = NEW /aws1/cl_reks3object(  
    iv_bucket = iv_target_s3_bucket  
    iv_name = iv_target_s3_key ).  
  
  " Create target image object
```

```

DATA(lo_target_image) = NEW /aws1/cl_rekimage(
  io_s3object = lo_target_s3obj ).

" Compare faces
oo_result = lo_rek->comparefaces(
  io_sourceimage = lo_source_image
  io_targetimage = lo_target_image
  iv_similaritythreshold = iv_similarity ).

DATA(lt_face_matches) = oo_result->get_facematches( ).
DATA(lt_unmatched_faces) = oo_result->get_unmatchedfaces( ).

" Get counts of matched and unmatched faces
DATA(lv_matched_count) = lines( lt_face_matches ).
DATA(lv_unmatched_count) = lines( lt_unmatched_faces ).

" Output detailed comparison results
DATA(lv_message) = |Face comparison completed: | &&
                  |{ lv_matched_count } matched face(s), | &&
                  |{ lv_unmatched_count } unmatched face(s).|.
MESSAGE lv_message TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CompareFaces](#) in *AWS SDK for SAP ABAP API reference*.

CreateCollection

The following code example shows how to use CreateCollection.

For more information, see [Creating a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_rek->createcollection(  
        iv_collectionid = iv_collection_id ).  
    MESSAGE 'Collection created successfully.' TYPE 'I'.  
CATCH /aws1/cx_rekresrcalrdyexistsex.  
    MESSAGE 'Collection already exists.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateCollection](#) in *AWS SDK for SAP ABAP API reference*.

DeleteCollection

The following code example shows how to use DeleteCollection.

For more information, see [Deleting a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_rek->deletecollection(  
        iv_collectionid = iv_collection_id ).  
    MESSAGE 'Collection deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_rekresourcenotfoundex.  
    MESSAGE 'Collection not found.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteCollection](#) in *AWS SDK for SAP ABAP API reference*.

DeleteFaces

The following code example shows how to use DeleteFaces.

For more information, see [Deleting faces from a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_rek->deletefaces(  
    iv_collectionid = iv_collection_id  
    it_faceids = it_face_ids ).  
  
  DATA(lt_deleted_faces) = oo_result->get_deletedfaces( ).  
  DATA(lv_deleted_count) = lines( lt_deleted_faces ).  
  DATA(lv_msg6) = |{ lv_deleted_count } face(s) deleted successfully.|.  
  MESSAGE lv_msg6 TYPE 'I'.  
  CATCH /aws1/cx_rekresourcenotfoundex.  
    MESSAGE 'Collection not found.' TYPE 'E'.  
  CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteFaces](#) in *AWS SDK for SAP ABAP API reference*.

DescribeCollection

The following code example shows how to use DescribeCollection.

For more information, see [Describing a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_rek->describecollection(  
        iv_collectionid = iv_collection_id ).  
    DATA(lv_face_count) = oo_result->get_facecount( ).  
    DATA(lv_msg) = |Collection described: { lv_face_count } face(s) indexed.|.  
    MESSAGE lv_msg TYPE 'I'.  
CATCH /aws1/cx_rekresourcenotfoundex.  
    MESSAGE 'Collection not found.' TYPE 'E'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeCollection](#) in *AWS SDK for SAP ABAP API reference*.

DetectFaces

The following code example shows how to use DetectFaces.

For more information, see [Detecting faces in an image](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " Create S3 object reference for the image  
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(  

```

```
        iv_bucket = iv_s3_bucket
        iv_name = iv_s3_key ).

" Create image object
DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

" Detect faces in the image with all attributes
DATA(lt_attributes) = VALUE /aws1/cl_rekattributes_w=>tt_attributes( ).
DATA(lo_attr_wrapper) = NEW /aws1/cl_rekattributes_w( iv_value = 'ALL' ).
INSERT lo_attr_wrapper INTO TABLE lt_attributes.

oo_result = lo_rek->detectfaces(
    io_image = lo_image
    it_attributes = lt_attributes ).

DATA(lt_face_details) = oo_result->get_facedetails( ).
DATA(lv_detected_count) = lines( lt_face_details ).
DATA(lv_msg8) = |{ lv_detected_count } face(s) detected in image.|.
MESSAGE lv_msg8 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DetectFaces](#) in *AWS SDK for SAP ABAP API reference*.

DetectLabels

The following code example shows how to use DetectLabels.

For more information, see [Detecting labels in an image](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Create S3 object reference for the image  
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(  
    iv_bucket = iv_s3_bucket  
    iv_name = iv_s3_key ).  
  
  " Create image object  
  DATA(lo_image) = NEW /aws1/cl_rekimage(  
    io_s3object = lo_s3object ).  
  
  " Detect labels in the image  
  oo_result = lo_rek->detectlabels(  
    io_image = lo_image  
    iv_maxlabels = iv_max_labels ).  
  
  DATA(lt_labels) = oo_result->get_labels( ).  
  DATA(lv_label_count) = lines( lt_labels ).  
  DATA(lv_msg9) = |{ lv_label_count } label(s) detected in image.|.  
  MESSAGE lv_msg9 TYPE 'I'.  
  CATCH /aws1/cx_rekinvalids3objectex.  
    MESSAGE 'Invalid S3 object.' TYPE 'E'.  
  CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DetectLabels](#) in *AWS SDK for SAP ABAP API reference*.

DetectModerationLabels

The following code example shows how to use DetectModerationLabels.

For more information, see [Detecting inappropriate images](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect moderation labels
  oo_result = lo_rek->detectmoderationlabels(
    io_image = lo_image ).

  DATA(lt_moderation_labels) = oo_result->get_moderationlabels( ).
  DATA(lv_mod_count) = lines( lt_moderation_labels ).
  DATA(lv_msg10) = |{ lv_mod_count } moderation label(s) detected.|.
  MESSAGE lv_msg10 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DetectModerationLabels](#) in *AWS SDK for SAP ABAP API reference*.

DetectText

The following code example shows how to use DetectText.

For more information, see [Detecting text in an image](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Detect text in the image
  oo_result = lo_rek->detecttext(
    io_image = lo_image ).

  DATA(lt_text_detections) = oo_result->get_textdetections( ).
  DATA(lv_text_count) = lines( lt_text_detections ).
  DATA(lv_msg11) = |{ lv_text_count } text detection(s) found.|.
  MESSAGE lv_msg11 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DetectText](#) in *AWS SDK for SAP ABAP API reference*.

IndexFaces

The following code example shows how to use IndexFaces.

For more information, see [Adding faces to a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Index faces in the image
  oo_result = lo_rek->indexfaces(
    iv_collectionid = iv_collection_id
    io_image = lo_image
    iv_externalimageid = iv_external_id
    iv_maxfaces = iv_max_faces ).

  DATA(lt_face_records) = oo_result->get_facerecords( ).
  DATA(lv_indexed_count) = lines( lt_face_records ).
  DATA(lv_msg2) = |{ lv_indexed_count } face(s) indexed successfully.|.
  MESSAGE lv_msg2 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [IndexFaces](#) in *AWS SDK for SAP ABAP API reference*.

ListCollections

The following code example shows how to use ListCollections.

For more information, see [Listing collections](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_rek->listcollections(  
        iv_maxresults = iv_max_results ).  
  
    DATA(lt_collection_ids) = oo_result->get_collectionids( ).  
    DATA(lv_coll_count) = lines( lt_collection_ids ).  
    DATA(lv_msg7) = |{ lv_coll_count } collection(s) found.|.  
    MESSAGE lv_msg7 TYPE 'I'.  
CATCH /aws1/cx_rekinvalidparameterex.  
    MESSAGE 'Invalid parameter value.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListCollections](#) in *AWS SDK for SAP ABAP API reference*.

ListFaces

The following code example shows how to use ListFaces.

For more information, see [Listing faces in a collection](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_rek->listfaces(  
        iv_collectionid = iv_collection_id
```

```

        iv_maxresults = iv_max_results ).

    DATA(lt_faces) = oo_result->get_faces( ).
    DATA(lv_face_count2) = lines( lt_faces ).
    DATA(lv_msg3) = |{ lv_face_count2 } face(s) found in collection.|.
    MESSAGE lv_msg3 TYPE 'I'.
    CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection not found.' TYPE 'E'.
    CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.

```

- For API details, see [ListFaces](#) in *AWS SDK for SAP ABAP API reference*.

RecognizeCelebrities

The following code example shows how to use `RecognizeCelebrities`.

For more information, see [Recognizing celebrities in an image](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
        iv_bucket = iv_s3_bucket
        iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
        io_s3object = lo_s3object ).

    " Recognize celebrities
    oo_result = lo_rek->recognizecelebrities(
        io_image = lo_image ).

```

```

DATA(lt_celebrity_faces) = oo_result->get_celebrityfaces( ).
DATA(lv_celeb_count) = lines( lt_celebrity_faces ).
DATA(lv_msg12) = |{ lv_celeb_count } celebrity/celebrities recognized.|.
MESSAGE lv_msg12 TYPE 'I'.
CATCH /aws1/cx_rekinvalids3objectex.
  MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.

```

- For API details, see [RecognizeCelebrities](#) in *AWS SDK for SAP ABAP API reference*.

SearchFaces

The following code example shows how to use SearchFaces.

For more information, see [Searching for a face \(face ID\)](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_rek->searchfaces(
    iv_collectionid = iv_collection_id
    iv_faceid = iv_face_id
    iv_facematchthreshold = iv_threshold
    iv_maxfaces = iv_max_faces ).

  DATA(lt_face_matches) = oo_result->get_facematches( ).
  DATA(lv_match_count2) = lines( lt_face_matches ).
  DATA(lv_msg5) = |Face search completed: { lv_match_count2 } match(es)
found.|.
  MESSAGE lv_msg5 TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
  MESSAGE 'Collection or face not found.' TYPE 'E'.

```

```
CATCH /aws1/cx_rekinvalidparameterex.
  MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [SearchFaces](#) in *AWS SDK for SAP ABAP API reference*.

SearchFacesByImage

The following code example shows how to use SearchFacesByImage.

For more information, see [Searching for a face \(image\)](#).

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create S3 object reference for the image
  DATA(lo_s3object) = NEW /aws1/cl_reks3object(
    iv_bucket = iv_s3_bucket
    iv_name = iv_s3_key ).

  " Create image object
  DATA(lo_image) = NEW /aws1/cl_rekimage(
    io_s3object = lo_s3object ).

  " Search for matching faces
  oo_result = lo_rek->searchfacesbyimage(
    iv_collectionid = iv_collection_id
    io_image = lo_image
    iv_facematchthreshold = iv_threshold
    iv_maxfaces = iv_max_faces ).

  DATA(lt_face_matches) = oo_result->get_facematches( ).
  DATA(lv_match_count) = lines( lt_face_matches ).
  DATA(lv_msg4) = |Face search completed: { lv_match_count } match(es)
found.|.
```

```
MESSAGE lv_msg4 TYPE 'I'.
CATCH /aws1/cx_rekresourcenotfoundex.
MESSAGE 'Collection not found.' TYPE 'E'.
CATCH /aws1/cx_rekinvalids3objectex.
MESSAGE 'Invalid S3 object.' TYPE 'E'.
CATCH /aws1/cx_rekinvalidparameterex.
MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- For API details, see [SearchFacesByImage](#) in *AWS SDK for SAP ABAP API reference*.

Amazon S3 examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon S3.

Basics are code examples that show you how to perform the essential operations within a service.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Basics](#)
- [Actions](#)
- [Scenarios](#)

Basics

Learn the basics

The following code example shows how to:

- Create a bucket and upload a file to it.

- Download an object from a bucket.
- Copy an object to a subfolder in a bucket.
- List the objects in a bucket.
- Delete the bucket objects and the bucket.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
  " determine our region from our session
  DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
  DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
  " When in the us-east-1 region, you must not specify a constraint
  " In all other regions, specify the region as the constraint
  IF lv_region = 'us-east-1'.
    CLEAR lo_constraint.
  ELSE.
    lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
  ENDIF.

  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
    io_createbucketconfiguration = lo_constraint ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

```

```
"Upload an object to an S3 bucket."
TRY.
  "Get contents of file from application server."
  DATA lv_file_content TYPE xstring.
  OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
  READ DATASET iv_key INTO lv_file_content.
  CLOSE DATASET iv_key.

  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
    iv_body = lv_file_content ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
  DATA(lo_result) = lo_s3->getobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).
  DATA(lv_object_data) = lo_result->get_body( ).
  MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
  lo_s3->copyobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    iv_copysource = |{ iv_bucket_name }/{ iv_key }| ).
  MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

```
" List objects in the bucket. "
TRY.
  DATA(lo_list) = lo_s3->listobjects(
    iv_bucket = iv_bucket_name ).
  MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ', ' INTO text.
ENDLOOP.
MESSAGE text TYPE'I'.

" Delete the objects in a bucket. "
TRY.
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }| ).
  MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
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.
```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CopyObject](#)
 - [CreateBucket](#)

- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Actions

CopyObject

The following code example shows how to use CopyObject.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [CopyObject](#) in *AWS SDK for SAP ABAP API reference*.

CreateBucket

The following code example shows how to use CreateBucket.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " determine our region from our session
  DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
  DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
  " When in the us-east-1 region, you must not specify a constraint
  " In all other regions, specify the region as the constraint
  IF lv_region = 'us-east-1'.
    CLEAR lo_constraint.
  ELSE.
    lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
  ENDIF.

  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
    io_createbucketconfiguration = lo_constraint ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

- For API details, see [CreateBucket](#) in *AWS SDK for SAP ABAP API reference*.

DeleteBucket

The following code example shows how to use DeleteBucket.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DeleteBucket](#) in *AWS SDK for SAP ABAP API reference*.

DeleteBucketCors

The following code example shows how to use DeleteBucketCors.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  
  lo_s3->deletebucketcors(  
    iv_bucket = iv_bucket_name ).  
  MESSAGE 'Bucket CORS configuration deleted.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteBucketCors](#) in *AWS SDK for SAP ABAP API reference*.

DeleteBucketLifecycle

The following code example shows how to use DeleteBucketLifecycle.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_s3->deletebucketlifecycle(  
    iv_bucket = iv_bucket_name ).  
  MESSAGE 'Bucket lifecycle configuration deleted.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteBucketLifecycle](#) in *AWS SDK for SAP ABAP API reference*.

DeleteBucketPolicy

The following code example shows how to use DeleteBucketPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
lo_s3->deletebucketpolicy(  
    iv_bucket = iv_bucket_name ).  
MESSAGE 'Bucket policy deleted.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteBucketPolicy](#) in *AWS SDK for SAP ABAP API reference*.

DeleteObject

The following code example shows how to use DeleteObject.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DeleteObject](#) in *AWS SDK for SAP ABAP API reference*.

DeleteObjects

The following code example shows how to use DeleteObjects.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->deleteobjects(           " oo_result is returned for  
testing purposes. "  
    iv_bucket = iv_bucket_name  
    io_delete = NEW /aws1/cl_s3_delete( it_objects = it_object_keys ) ).  
    MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteObjects](#) in *AWS SDK for SAP ABAP API reference*.

GetBucketAcl

The following code example shows how to use GetBucketAcl.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->getbucketacl(           " oo_result is returned for testing  
purposes. "  
    iv_bucket = iv_bucket_name ).  
    MESSAGE 'Retrieved bucket ACL.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
ENDTRY.
```

- For API details, see [GetBucketAcl](#) in *AWS SDK for SAP ABAP API reference*.

GetBucketCors

The following code example shows how to use GetBucketCors.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->getbucketcors(           " oo_result is returned for  
testing purposes. "  
    iv_bucket = iv_bucket_name ).  
    MESSAGE 'Retrieved bucket CORS configuration.' TYPE 'I'.  
    CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetBucketCors](#) in *AWS SDK for SAP ABAP API reference*.

GetBucketLifecycleConfiguration

The following code example shows how to use GetBucketLifecycleConfiguration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_s3->getbucketlifecycleconf(           " oo_result is returned
for testing purposes. "
    iv_bucket = iv_bucket_name ).
    MESSAGE 'Retrieved bucket lifecycle configuration.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetBucketLifecycleConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

GetBucketPolicy

The following code example shows how to use GetBucketPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_s3->getbucketpolicy(           " oo_result is returned for
testing purposes. "
    iv_bucket = iv_bucket_name ).
    DATA(lv_policy) = oo_result->get_policy( ).
    MESSAGE 'Retrieved bucket policy.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetBucketPolicy](#) in *AWS SDK for SAP ABAP API reference*.

GetObject

The following code example shows how to use GetObject.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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.

```

- For API details, see [GetObject](#) in *AWS SDK for SAP ABAP API reference*.

GetObjectAcl

The following code example shows how to use GetObjectAcl.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    oo_result = lo_s3->getobjectacl(       " oo_result is returned for testing
purposes. "
        iv_bucket = iv_bucket_name

```

```

        iv_key = iv_object_key ).
    MESSAGE 'Retrieved object ACL.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [GetObjectAcl](#) in *AWS SDK for SAP ABAP API reference*.

GetObjectLegalHold

The following code example shows how to use `GetObjectLegalHold`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    oo_result = lo_s3->getobjectlegalhold(           " oo_result is returned for
testing purposes. "
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key ).
    MESSAGE 'Retrieved object legal hold status.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [GetObjectLegalHold](#) in *AWS SDK for SAP ABAP API reference*.

GetObjectLockConfiguration

The following code example shows how to use `GetObjectLockConfiguration`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->getobjectlockconfiguration(          " oo_result is  
returned for testing purposes. "  
    iv_bucket = iv_bucket_name ).  
    MESSAGE 'Retrieved object lock configuration.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetObjectLockConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

HeadBucket

The following code example shows how to use HeadBucket.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->headbucket(          " oo_result is returned for testing  
purposes. "  
    iv_bucket = iv_bucket_name ).  
    MESSAGE 'Bucket exists and you have access to it.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [HeadBucket](#) in *AWS SDK for SAP ABAP API reference*.

ListBuckets

The following code example shows how to use ListBuckets.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->listbuckets(           " oo_result is returned for testing  
purposes. "  
    ).  
    DATA(lv_bucket_count) = lines( oo_result->get_buckets( ) ).  
    MESSAGE |Retrieved { lv_bucket_count } buckets in all regions.| TYPE 'I'.  
    CATCH /aws1/cx_rt_generic.  
        MESSAGE 'Unable to list buckets.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListBuckets](#) in *AWS SDK for SAP ABAP API reference*.

ListObjectVersions

The following code example shows how to use ListObjectVersions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_s3->listobjectversions(           " oo_result is returned for  
testing purposes. "  
        iv_bucket = iv_bucket_name  
        iv_prefix = iv_prefix ).  
    MESSAGE 'Retrieved object versions.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListObjectVersions](#) in *AWS SDK for SAP ABAP API reference*.

ListObjectsV2

The following code example shows how to use ListObjectsV2.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListObjectsV2](#) in *AWS SDK for SAP ABAP API reference*.

PutBucketAcl

The following code example shows how to use PutBucketAcl.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Example: Grant log delivery access to a bucket  
  " iv_grantwrite = 'uri=http://acs.amazonaws.com/groups/s3/LogDelivery'  
  lo_s3->putbucketacl(  
    iv_bucket = iv_bucket_name  
    iv_grantwrite = iv_grantwrite ).  
  MESSAGE 'Bucket ACL updated.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [PutBucketAcl](#) in *AWS SDK for SAP ABAP API reference*.

PutBucketCors

The following code example shows how to use PutBucketCors.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Example: Allow PUT, POST, DELETE methods from http://www.example.com  
  lo_s3->putbucketcors(  
    iv_bucket = iv_bucket_name  
    io_corsconfiguration = NEW /aws1/cl_s3_corsconfiguration(  

```

```
        it_corsrules = it_cors_rules ) ).  
    MESSAGE 'Bucket CORS configuration set.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [PutBucketCors](#) in *AWS SDK for SAP ABAP API reference*.

PutBucketLifecycleConfiguration

The following code example shows how to use PutBucketLifecycleConfiguration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    " Example: Expire objects with prefix 'logs/' after 30 days  
    lo_s3->putbucketlifecycleconf(  
        iv_bucket = iv_bucket_name  
        io_lifecycleconfiguration = NEW /aws1/cl_s3_bucketlccconf(  
            it_rules = it_lifecycle_rule ) ).  
    MESSAGE 'Bucket lifecycle configuration set.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [PutBucketLifecycleConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

PutBucketPolicy

The following code example shows how to use PutBucketPolicy.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Example policy JSON string
  " iv_policy = '{"Version":"2012-10-17",          "Statement":
[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:user/
user"},"Action":["s3:GetObject"],"Resource":["arn:aws:s3:::bucketname/*"]}]}'
  lo_s3->putbucketpolicy(
    iv_bucket = iv_bucket_name
    iv_policy = iv_policy ).
  MESSAGE 'Bucket policy set.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- For API details, see [PutBucketPolicy](#) in *AWS SDK for SAP ABAP API reference*.

PutBucketVersioning

The following code example shows how to use PutBucketVersioning.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Example: Enable versioning on a bucket
  " iv_status = 'Enabled'
```

```

lo_s3->putbucketversioning(
  iv_bucket = iv_bucket_name
  io_versioningconfiguration = NEW /aws1/cl_s3_versioningconf(
    iv_status = iv_status ) ).
MESSAGE 'Bucket versioning enabled.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutBucketVersioning](#) in *AWS SDK for SAP ABAP API reference*.

PutObject

The following code example shows how to use PutObject.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"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.

```

- For API details, see [PutObject](#) in *AWS SDK for SAP ABAP API reference*.

PutObjectAcl

The following code example shows how to use PutObjectAcl.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Example: Grant read access to an AWS user  
  " iv_grantread = 'emailAddress=user@example.com'  
  lo_s3->putobjectacl(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key  
    iv_grantread = iv_grantread ).  
  MESSAGE 'Object ACL updated.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
CATCH /aws1/cx_s3_nosuchkey.  
  MESSAGE 'Object key does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [PutObjectAcl](#) in *AWS SDK for SAP ABAP API reference*.

PutObjectLegalHold

The following code example shows how to use PutObjectLegalHold.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Example: Set legal hold status to ON  
  " iv_status = 'ON'  
  lo_s3->putobjectlegalhold(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_object_key  
    io_legalhold = NEW /aws1/cl_s3_objlocklegalhold(  
      iv_status = iv_status ) ).  
  MESSAGE 'Object legal hold status set.' TYPE 'I'.  
  CATCH /aws1/cx_s3_nosuchbucket.  
    MESSAGE 'Bucket does not exist.' TYPE 'E'.  
  CATCH /aws1/cx_s3_nosuchkey.  
    MESSAGE 'Object key does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [PutObjectLegalHold](#) in *AWS SDK for SAP ABAP API reference*.

PutObjectLockConfiguration

The following code example shows how to use PutObjectLockConfiguration.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  " Example: Enable object lock with default retention
```

```

" iv_enabled = 'Enabled'
lo_s3->putobjectlockconfiguration(
  iv_bucket = iv_bucket_name
  io_objectlockconfiguration = NEW /aws1/cl_s3_objectlockconf(
    iv_objectlockenabled = iv_enabled ) ).
MESSAGE 'Object lock configuration set.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutObjectLockConfiguration](#) in *AWS SDK for SAP ABAP API reference*.

PutObjectRetention

The following code example shows how to use PutObjectRetention.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
" Example: Set retention mode to GOVERNANCE for 30 days
" iv_mode = 'GOVERNANCE'
" iv_retain_date should be a timestamp in the future
lo_s3->putobjectretention(
  iv_bucket = iv_bucket_name
  iv_key = iv_object_key
  io_retention = NEW /aws1/cl_s3_objectlockret(
    iv_mode = iv_mode
    iv_retainuntildate = iv_retain_date )
  iv_bypassgovernanceretention = abap_true ).
MESSAGE 'Object retention set.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [PutObjectRetention](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Create a presigned URL

The following code example shows how to create a presigned URL for Amazon S3 and upload an object.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create presigned requests to GET S3 objects.

```
" iv_bucket_name is the bucket name
" iv_key is the object name like "myfile.txt"

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

"Upload a nice Hello World file to an S3 bucket."
TRY.
    DATA(lv_contents) = cl_abap_codepage=>convert_to( 'Hello, World' ).
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_contents
        iv_contenttype = 'text/plain' ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" now generate a presigned URL with a 600-second expiration
```

```
DATA(lo_presigner) = lo_s3->get_presigner( iv_expires_sec = 600 ).  
" the presigner getObject() method has the same signature as  
" lo_s3->getObject(), but it doesn't actually make the call.  
" to the service. It just prepares a presigned URL for a future call  
DATA(lo_presigned_req) = lo_presigner->getObject(  
    iv_bucket = iv_bucket_name  
    iv_key = iv_key ).  
  
" You can provide this URL to a web page, user, email etc so they  
" can retrieve the file. The URL will expire in 10 minutes.  
ov_url = lo_presigned_req->get_url( ).
```

Amazon S3 Control examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon S3 Control.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateJob

The following code example shows how to use CreateJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_manifest_arn = 'arn:aws:s3::my-bucket/job-manifest.csv'
  " iv_manifest_etag = 'abc123def456'
  " iv_report_bucket = 'arn:aws:s3::my-report-bucket'
  DATA(lo_result) = lo_s3c->createjob(
    iv_accountid          = iv_account_id
    iv_rolearn            = iv_role_arn
    iv_confirmationrequired = abap_true
    iv_priority           = 10
    iv_description        = 'Batch job for tagging objects'
    io_operation          = NEW /aws1/cl_s3cjoboperation(
      io_s3putobjecttagging = NEW /aws1/cl_s3cs3setobjecttagop(
        it_tagset = VALUE /aws1/cl_s3cs3tag=>tt_s3tagset(
          ( NEW /aws1/cl_s3cs3tag(
            iv_key   = 'BatchTag'
            iv_value = 'BatchValue' ) )
        )
      )
    )
    io_manifest          = NEW /aws1/cl_s3cjobmanifest(
      io_spec            = NEW /aws1/cl_s3cjobmanifestspec(
        iv_format        = 'S3BatchOperations_CSV_20180820'
        it_fields        = VALUE /aws1/
cl_s3cjobmanifestfield00=>tt_jobmanifestfieldlist(
          ( NEW /aws1/cl_s3cjobmanifestfield00( 'Bucket' ) )
          ( NEW /aws1/cl_s3cjobmanifestfield00( 'Key' ) )
        )
      )
      io_location        = NEW /aws1/cl_s3cjobmanifestloc(
        iv_objectarn     = iv_manifest_arn
        iv_etag          = iv_manifest_etag
      )
    )
    io_report            = NEW /aws1/cl_s3cjobreport(
      iv_bucket          = iv_report_bucket
      iv_format          = 'Report_CSV_20180820'
      iv_enabled         = abap_true
      iv_prefix          = 'batch-op-reports'
      iv_reportscope     = 'AllTasks'
    )
  ).
  ov_job_id = lo_result->get_jobid( ).
  MESSAGE |S3 Batch Operations job created: { ov_job_id }| TYPE 'I'.

```

```
CATCH /aws1/cx_s3cbadrequestex INTO DATA(lo_ex_bad).
  MESSAGE lo_ex_bad->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_bad.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
  MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
  MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_srv.
ENDTRY.
```

- For API details, see [CreateJob](#) in *AWS SDK for SAP ABAP API reference*.

DeleteJobTagging

The following code example shows how to use DeleteJobTagging.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_s3c->deletejobtagging(
    iv_accountid = iv_account_id
    iv_jobid      = iv_job_id
  ).
  MESSAGE |Tags deleted from job { iv_job_id }| TYPE 'I'.
CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
  MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_nf.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
  MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
```

```

RAISE EXCEPTION TYPE /aws1/cx_rt_generic
  EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
  MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_srv.
ENDTRY.

```

- For API details, see [DeleteJobTagging](#) in *AWS SDK for SAP ABAP API reference*.

DescribeJob

The following code example shows how to use DescribeJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_s3c->describejob(           " oo_result is returned for testing
purposes.
  iv_accountid = iv_account_id
  iv_jobid      = iv_job_id
  ).
  DATA(lo_job) = oo_result->get_job( ).
  DATA(lv_status) = lo_job->get_status( ).
  DATA(lv_priority) = lo_job->get_priority( ).
  DATA(lo_progress) = lo_job->get_progresssummary( ).
  IF lo_progress IS NOT INITIAL.
    MESSAGE |Job { iv_job_id }: status={ lv_status },
priority={ lv_priority }, | &&
           |total={ lo_progress->get_totalnumberoftasks( ) }, | &&
           |succeeded={ lo_progress->get_numberoftaskssucceeded( ) }, | &&
           |failed={ lo_progress->get_numberoftasksfailed( ) }| TYPE 'I'.
  ELSE.
    MESSAGE |Job { iv_job_id }: status={ lv_status },
priority={ lv_priority }| TYPE 'I'.

```

```

ENDIF.
CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
  MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_nf.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
  MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
  MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_srv.
ENDTRY.

```

- For API details, see [DescribeJob](#) in *AWS SDK for SAP ABAP API reference*.

GetJobTagging

The following code example shows how to use GetJobTagging.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_s3c->getjobtagging(           " oo_result is returned for testing
purposes.
  iv_accountid = iv_account_id
  iv_jobid     = iv_job_id
  ).
  DATA(lt_tags) = oo_result->get_tags( ).
  MESSAGE |Retrieved { lines( lt_tags ) } tag(s) for job { iv_job_id }| TYPE
'I'.
  CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
  MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic

```

```

EXPORTING previous = lo_ex_nf.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
EXPORTING previous = lo_ex_srv.
ENDTRY.

```

- For API details, see [GetJobTagging](#) in *AWS SDK for SAP ABAP API reference*.

PutJobTagging

The following code example shows how to use PutJobTagging.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_s3c->putjobtagging(
    iv_accountid = iv_account_id
    iv_jobid      = iv_job_id
    it_tags       = VALUE /aws1/cl_s3cs3tag=>tt_s3tagset(
      ( NEW /aws1/cl_s3cs3tag(
          iv_key   = 'Environment'
          iv_value = 'Development' ) )
      ( NEW /aws1/cl_s3cs3tag(
          iv_key   = 'Team'
          iv_value = 'DataProcessing' ) )
    )
  ).
  MESSAGE |Tags added to job { iv_job_id }| TYPE 'I'.
CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).

```

```

MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
  EXPORTING previous = lo_ex_nf.
CATCH /aws1/cx_s3ctoomanytagsex INTO DATA(lo_ex_tags).
MESSAGE lo_ex_tags->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
  EXPORTING previous = lo_ex_tags.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
  EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
RAISE EXCEPTION TYPE /aws1/cx_rt_generic
  EXPORTING previous = lo_ex_srv.
ENDTRY.

```

- For API details, see [PutJobTagging](#) in *AWS SDK for SAP ABAP API reference*.

UpdateJobPriority

The following code example shows how to use UpdateJobPriority.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  oo_result = lo_s3c->updatejobpriority( " oo_result is returned for testing
purposes.
  iv_accountid = iv_account_id
  iv_jobid      = iv_job_id
  iv_priority   = 60
  ).
  MESSAGE |Job { oo_result->get_jobid( ) } priority updated to { oo_result-
>get_priority( ) }| TYPE 'I'.

```

```

CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
  MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_nf.
CATCH /aws1/cx_s3cbadrequestex INTO DATA(lo_ex_bad).
  MESSAGE lo_ex_bad->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_bad.
CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
  MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_cli.
CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
  MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
  RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_srv.
ENDTRY.

```

- For API details, see [UpdateJobPriority](#) in *AWS SDK for SAP ABAP API reference*.

UpdateJobStatus

The following code example shows how to use UpdateJobStatus.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_requested_status = 'Cancelled'
  oo_result = lo_s3c->updatejobstatus(      " oo_result is returned for testing
purposes.
  iv_accountid          = iv_account_id
  iv_jobid              = iv_job_id
  iv_requestedjobstatus = iv_requested_status
  ).

```

```
    MESSAGE |Job { oo_result->get_jobid( ) } status updated to { oo_result-
>get_status( ) }| TYPE 'I'.
    CATCH /aws1/cx_s3cjobstatusexception INTO DATA(lo_ex_js).
    MESSAGE lo_ex_js->get_text( ) TYPE 'I'.
    RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_js.
    CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
    MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
    RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_nf.
    CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
    MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
    RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_cli.
    CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
    MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
    RAISE EXCEPTION TYPE /aws1/cx_rt_generic
    EXPORTING previous = lo_ex_srv.
ENDTRY.
```

- For API details, see [UpdateJobStatus](#) in *AWS SDK for SAP ABAP API reference*.

SageMaker AI examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with SageMaker AI.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)
- [Scenarios](#)

Actions

CreateEndpoint

The following code example shows how to use CreateEndpoint.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA oo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.

"Create a production variant as an ABAP object."
"Identifies a model that you want to host and the resources chosen to deploy for
hosting it."
  lo_production_variants = NEW #( iv_variantname = iv_variant_name
                                iv_modelname = iv_model_name
                                iv_initialinstancecount =
iv_initial_instance_count
                                iv_instancetype = iv_instance_type ).

INSERT lo_production_variants INTO TABLE lt_production_variants.

"Create an endpoint configuration."
TRY.
  oo_ep_config_result = lo_sgm->createendpointconfig(
    iv_endpointconfigname = iv_endpoint_config_name
    it_productionvariants = lt_production_variants ).
  MESSAGE 'Endpoint configuration created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Create an endpoint."
```

```

TRY.
    oo_result = lo_sgm->createendpoint(      " oo_result is returned for testing
purposes. "
        iv_endpointconfigname = iv_endpoint_config_name
        iv_endpointname = iv_endpoint_name ).
    MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateEndpoint](#) in *AWS SDK for SAP ABAP API reference*.

CreateModel

The following code example shows how to use CreateModel.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.

"Create an ABAP object for the container image based on input variables."
lo_primarycontainer = NEW #( iv_image = iv_container_image
                            iv_modeldataurl = iv_model_data_url ).

"Create an Amazon SageMaker model."
TRY.
    oo_result = lo_sgm->createmodel(      " oo_result is returned for testing
purposes. "
        iv_executionrolearn = iv_execution_role_arn
        iv_modelname = iv_model_name
        io_primarycontainer = lo_primarycontainer ).
    MESSAGE 'Model created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.

```

```
MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.
```

- For API details, see [CreateModel](#) in *AWS SDK for SAP ABAP API reference*.

CreateTrainingJob

The following code example shows how to use CreateTrainingJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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_sgmalgorithmsspec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.

"Create ABAP internal table for hyperparameters based on input variables."
"These hyperparameters are based on the Amazon SageMaker built-in algorithm,
XGBoost."
lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
```

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

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

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

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

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

lo_hyperparameters_w = NEW #( iv_value = iv_hp_num_round ).
INSERT VALUE #( key = 'num_round' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

"Create ABAP objects for training data sources."
lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                           iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                           iv_s3uri = iv_trn_data_s3uri ).

lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

lo_trn_channel = NEW #( iv_channelname = 'train'
                       io_datasource = lo_trn_datasource
                       iv_compressiontype = iv_trn_data_compressiontype
                       iv_contenttype = iv_trn_data_contenttype ).

INSERT lo_trn_channel INTO TABLE lt_input_data_config.

"Create ABAP objects for validation data sources."
lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                           iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                           iv_s3uri = iv_val_data_s3uri ).
```

```
lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).

INSERT lo_val_channel INTO TABLE lt_input_data_config.

"Create an ABAP object for algorithm specification."
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                    iv_traininginputmode =
iv_training_input_mode ).

"Create an ABAP object for resource configuration."
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                            iv_instancetype = iv_instance_type
                            iv_volumesizeingb = iv_volume_sizeingb ).

"Create an ABAP object for output data configuration."
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

"Create a training job."
TRY.
    oo_result = lo_sgm->createtrainingjob( " oo_result is returned for
testing purposes. "
    iv_trainingjobname           = iv_training_job_name
    iv_rolearn                   = iv_role_arn
    it_hyperparameters           = lt_hyperparameters
    it_inputdataconfig           = lt_input_data_config
    io_algorithmspecification     = lo_algorithm_specification
    io_outputdataconfig          = lo_output_data_config
    io_resourceconfig            = lo_resource_config
    io_stoppingcondition         = lo_stopping_condition ).
    MESSAGE 'Training job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
```

```
MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateTrainingJob](#) in *AWS SDK for SAP ABAP API reference*.

CreateTransformJob

The following code example shows how to use CreateTransformJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA lo_transforminput TYPE REF TO /aws1/cl_sgmtransforminput.  
DATA lo_transformoutput TYPE REF TO /aws1/cl_sgmtransformoutput.  
DATA lo_transformresources TYPE REF TO /aws1/cl_sgmtransformresources.  
DATA lo_datasource TYPE REF TO /aws1/cl_sgmtransformdatasrc.  
DATA lo_s3datasource TYPE REF TO /aws1/cl_sgmtransforms3datasrc.  
  
"Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data  
source."  
lo_s3datasource = NEW #( iv_s3uri = iv_tf_data_s3uri  
                        iv_s3datatype = iv_tf_data_s3datatype ).  
  
"Create an ABAP object for data source."  
lo_datasource = NEW #( io_s3datasource = lo_s3datasource ).  
  
"Create an ABAP object for transform data source."  
lo_transforminput = NEW #( io_datasource = lo_datasource  
                          iv_contenttype = iv_tf_data_contenttype  
                          iv_compressiontype = iv_tf_data_compressiontype ).  
  
"Create an ABAP object for resource configuration."  
lo_transformresources = NEW #( iv_instancecount = iv_instance_count  
                              iv_instancetype = iv_instance_type ).
```

```

"Create an ABAP object for output data configuration."
lo_transformoutput = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create a transform job."
TRY.
    oo_result = lo_sgm->createtransformjob(      " oo_result is returned for
testing purposes. "
        iv_modelname = iv_tf_model_name
        iv_transformjobname = iv_tf_job_name
        io_transforminput = lo_transforminput
        io_transformoutput = lo_transformoutput
        io_transformresources = lo_transformresources ).
    MESSAGE 'Transform job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateTransformJob](#) in *AWS SDK for SAP ABAP API reference*.

DeleteEndpoint

The following code example shows how to use DeleteEndpoint.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"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.

```

- For API details, see [DeleteEndpoint](#) in *AWS SDK for SAP ABAP API reference*.

DeleteModel

The following code example shows how to use DeleteModel.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    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.

```

- For API details, see [DeleteModel](#) in *AWS SDK for SAP ABAP API reference*.

DescribeTrainingJob

The following code example shows how to use DescribeTrainingJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DescribeTrainingJob](#) in *AWS SDK for SAP ABAP API reference*.

ListAlgorithms

The following code example shows how to use ListAlgorithms.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListAlgorithms](#) in *AWS SDK for SAP ABAP API reference*.

ListModels

The following code example shows how to use ListModels.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListModels](#) in *AWS SDK for SAP ABAP API reference*.

ListNotebookInstances

The following code example shows how to use ListNotebookInstances.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListNotebookInstances](#) in *AWS SDK for SAP ABAP API reference*.

ListTrainingJobs

The following code example shows how to use ListTrainingJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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'.
    ENDRY.

```

- For API details, see [ListTrainingJobs](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Get started with models and endpoints

The following code example shows how to:

- Start a training job and create a SageMaker AI model.
- Create an endpoint configuration.
- Create an endpoint, then clean up resources.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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"
lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

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

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

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

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

```

```
lo_hyperparameters_w = NEW #( iv_value = iv_hp_num_round ).
INSERT VALUE #( key = 'num_round' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.
```

"Create ABAP internal table for data based on input variables."

"Training data."

```
lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                             iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                             iv_s3uri = iv_trn_data_s3uri ).
```

```
lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).
```

```
lo_trn_channel = NEW #( iv_channelname = 'train'
                        io_datasource = lo_trn_datasource
                        iv_compressiontype = iv_trn_data_compressiontype
                        iv_contenttype = iv_trn_data_contenttype ).
```

```
INSERT lo_trn_channel INTO TABLE lt_input_data_config.
```

"Validation data."

```
lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                             iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                             iv_s3uri = iv_val_data_s3uri ).
```

```
lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).
```

```
lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).
```

```
INSERT lo_val_channel INTO TABLE lt_input_data_config.
```

"Create an ABAP object for algorithm specification based on input variables."

```
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                     iv_traininginputmode =
iv_training_input_mode ).
```

"Create an ABAP object for resource configuration."

```
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                             iv_instancetype = iv_instance_type
                             iv_volumesizeingb = iv_volume_sizeingb ).
```

"Create an ABAP object for output data configuration."

```
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

TRY.
  lo_sgm->createtrainingjob(
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
    it_inputdataconfig      = lt_input_data_config
    io_algorithmsspecification = lo_algorithm_specification
    io_outputdataconfig     = lo_output_data_config
    io_resourceconfig       = lo_resource_config
    io_stoppingcondition    = lo_stopping_condition ).
  MESSAGE 'Training job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
  MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Wait for training job to be completed."
lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
WHILE lo_training_result->get_trainingjobstatus( ) <> 'Completed'.
  IF sy-index = 30.
    EXIT.              "Maximum 900 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
ENDWHILE.

"Create ABAP object for the container image based on input variables."
lo_primarycontainer = NEW #( iv_image = iv_training_image
                             iv_modeldataurl = lv_model_data_url ).

"Create an Amazon SageMaker model."
TRY.
  lo_sgm->createmodel(
```

```

        iv_executionrolearn = iv_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.

    "Create an endpoint production variant."
    lo_production_variants = NEW #( iv_variantname = iv_ep_variant_name
                                    iv_modelname = iv_model_name
                                    iv_initialinstancecount =
iv_ep_initial_instance_count
                                    iv_instancetype = iv_ep_instance_type ).
    INSERT lo_production_variants INTO TABLE lt_production_variants.

    TRY.
        "Create an endpoint configuration."
        lo_ep_config_result = lo_sgm->createendpointconfig(
            iv_endpointconfigname = iv_ep_cfg_name
            it_productionvariants = lt_production_variants ).
        MESSAGE 'Endpoint configuration created.' TYPE 'I'.

        "Create an endpoint."
        oo_ep_output = lo_sgm->createendpoint(           " oo_ep_output is returned for
testing purposes. "
            iv_endpointconfigname = iv_ep_cfg_name
            iv_endpointname = iv_ep_name ).
        MESSAGE 'Endpoint created.' TYPE 'I'.
        CATCH /aws1/cx_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.
```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CreateEndpoint](#)
 - [CreateEndpointConfig](#)
 - [CreateModel](#)
 - [CreateTrainingJob](#)
 - [DeleteEndpoint](#)
 - [DeleteEndpointConfig](#)
 - [DeleteModel](#)
 - [DescribeEndpoint](#)
 - [DescribeTrainingJob](#)

Secrets Manager examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Secrets Manager.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

BatchGetSecretValue

The following code example shows how to use `BatchGetSecretValue`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " iv_filter_name = 'mySecret'
  DATA(lo_result) = lo_smr->batchgetsecretvalue(
    it_filters = VALUE /aws1/cl_smrfilter=>tt_filterslisttype(
      (
        NEW /aws1/cl_smrfilter(
          iv_key = 'name'
          it_values = VALUE /aws1/
cl_smrfiltvalsstrlist_w=>tt_filtervaluesstringlist(
          ( NEW /aws1/cl_smrfiltvalsstrlist_w( iv_value = iv_filter_name ) )
        )
      )
    )
  ).
  ot_secret_values = lo_result->get_secretvalues( ).
  MESSAGE 'Secrets retrieved successfully.' TYPE 'I'.
CATCH /aws1/cx_smrresourcenotfoundex.
```

```

    MESSAGE 'One or more requested secrets were not found.' TYPE 'E'.
  CATCH /aws1/cx_smrdecryptionfailure.
    MESSAGE 'Failed to decrypt one or more secrets.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
ENDTRY.

```

- For API details, see [BatchGetSecretValue](#) in *AWS SDK for SAP ABAP API reference*.

GetSecretValue

The following code example shows how to use GetSecretValue.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  " iv_secret_name = 'MySecretName'
  DATA(lo_result) = lo_smr->getsecretvalue( iv_secretid = iv_secret_name ).
  ov_secret_value = lo_result->get_secretstring( ).
  MESSAGE 'Secret value retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_smrresourcenotfoundex.
    MESSAGE 'The requested secret was not found.' TYPE 'E'.
  CATCH /aws1/cx_smrdecryptionfailure.
    MESSAGE 'Failed to decrypt the secret.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
ENDTRY.

```

- For API details, see [GetSecretValue](#) in *AWS SDK for SAP ABAP API reference*.

Amazon SES examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon SES.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateReceiptFilter

The following code example shows how to use CreateReceiptFilter.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" iv_allow = abap_true means 'Allow', abap_false means 'Block'
DATA(lv_policy) = COND /aws1/sesreceiptfilterpolicy(
  WHEN iv_allow = abap_true THEN 'Allow'
  ELSE 'Block'
).

DATA(lo_ip_filter) = NEW /aws1/cl_sesreceiptipfilter(
  iv_policy = lv_policy
  iv_cidr = iv_ip_address_or_range
).

DATA(lo_filter) = NEW /aws1/cl_sesreceiptfilter(
```

```

    iv_name = iv_filter_name
    io_ipfilter = lo_ip_filter
  ).

TRY.
  lo_ses->createreceiptfilter( io_filter = lo_filter ).
  MESSAGE 'Receipt filter created successfully' TYPE 'I'.
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
  DATA(lv_error) = |Filter already exists: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [CreateReceiptFilter](#) in *AWS SDK for SAP ABAP API reference*.

CreateReceiptRule

The following code example shows how to use CreateReceiptRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" Create S3 action for copying emails to S3
DATA(lo_s3_action) = NEW /aws1/cl_sess3action(
  iv_bucketname = iv_bucket_name
  iv_objectkeyprefix = iv_prefix
).

" Create receipt action with S3 action
DATA(lo_action) = NEW /aws1/cl_sesreceiptaction(
  io_s3action = lo_s3_action
).

```

```

" Create list of actions
DATA lt_actions TYPE /aws1/cl_sesreceiptaction=>tt_receiptactionslist.
APPEND lo_action TO lt_actions.

" Create receipt rule
DATA(lo_rule) = NEW /aws1/cl_sesrecepitrule(
  iv_name = iv_rule_name
  iv_enabled = abap_true
  it_recipients = it_recipients
  it_actions = lt_actions
).

TRY.
  lo_ses->createrecepitrule(
    iv_rulesetname = iv_rule_set_name
    io_rule = lo_rule
  ).
  MESSAGE 'Receipt rule created successfully' TYPE 'I'.
CATCH /aws1/cx_sesinvalids3confex INTO DATA(lo_ex1).
  DATA(lv_error) = |Invalid S3 configuration: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [CreateReceiptRule](#) in *AWS SDK for SAP ABAP API reference*.

CreateReceiptRuleSet

The following code example shows how to use CreateReceiptRuleSet.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_ses->createreciptruleset( iv_rulesetname = iv_rule_set_name ).
  MESSAGE 'Receipt rule set created successfully' TYPE 'I'.
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
  DATA(lv_error) = |Rule set already exists: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [CreateReceiptRuleSet](#) in *AWS SDK for SAP ABAP API reference*.

CreateTemplate

The following code example shows how to use CreateTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
  lo_ses->createtemplate( io_template = lo_template ).
  MESSAGE 'Template created successfully' TYPE 'I'.
CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
  DATA(lv_error) = |Template already exists: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.

```

```

    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
    lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [CreateTemplate](#) in *AWS SDK for SAP ABAP API reference*.

DeleteIdentity

The following code example shows how to use DeleteIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_ses->deleteidentity( iv_identity = iv_identity ).
  MESSAGE 'Identity deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.

```

- For API details, see [DeleteIdentity](#) in *AWS SDK for SAP ABAP API reference*.

DeleteReceiptFilter

The following code example shows how to use DeleteReceiptFilter.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ses->deletereciptfilter( iv_filtername = iv_filter_name ).  
  MESSAGE 'Receipt filter deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [DeleteReceiptFilter](#) in *AWS SDK for SAP ABAP API reference*.

DeleteReceiptRule

The following code example shows how to use DeleteReceiptRule.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ses->deletereciptrule(  
    iv_rulesetname = iv_rule_set_name  
    iv_rulename = iv_rule_name  
  ).  
  MESSAGE 'Receipt rule deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_sesrulesetdoesnotexex INTO DATA(lo_ex1).
```

```

DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [DeleteReceiptRule](#) in *AWS SDK for SAP ABAP API reference*.

DeleteReceiptRuleSet

The following code example shows how to use DeleteReceiptRuleSet.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_ses->deletereciptruleset( iv_rulesetname = iv_rule_set_name ).
  MESSAGE 'Receipt rule set deleted successfully' TYPE 'I'.
CATCH /aws1/cx_sescannotdeleteex INTO DATA(lo_ex1).
DATA(lv_error) = |Cannot delete rule set: { lo_ex1->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [DeleteReceiptRuleSet](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTemplate

The following code example shows how to use DeleteTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_ses->deletetemplate( iv_templatename = iv_template_name ).  
    MESSAGE 'Template deleted successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [DeleteTemplate](#) in *AWS SDK for SAP ABAP API reference*.

DescribeReceiptRuleSet

The following code example shows how to use DescribeReceiptRuleSet.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ses->describereceiptruleset(  
        iv_rulesetname = iv_rule_set_name
```

```

    ).
    MESSAGE 'Receipt rule set described successfully' TYPE 'I'.
  CATCH /aws1/cx_sesrulesetdoesnotexist INTO DATA(lo_ex1).
    DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.

```

- For API details, see [DescribeReceiptRuleSet](#) in *AWS SDK for SAP ABAP API reference*.

GetIdentityVerificationAttributes

The following code example shows how to use `GetIdentityVerificationAttributes`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

DATA lt_identities TYPE /aws1/cl_sesidentitylist_w=>tt_identitylist.
APPEND NEW /aws1/cl_sesidentitylist_w( iv_value = iv_identity ) TO
lt_identities.

TRY.
  DATA(lo_result) = lo_ses->getidentityverificationattrs(
    it_identities = lt_identities
  ).

  DATA(lt_attrs) = lo_result->get_verificationattributes( ).
  IF lt_attrs IS NOT INITIAL.
    LOOP AT lt_attrs ASSIGNING FIELD-SYMBOL(<ls_attr>).
      ov_status = <ls_attr>-value->get_verificationstatus( ).
      EXIT.
    ENDLOOP.
  ENDIF.
ENDTRY.

```

```

        ENDLOOP.
    ELSE.
        ov_status = 'NotFound'.
    ENDIF.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.

```

- For API details, see [GetIdentityVerificationAttributes](#) in *AWS SDK for SAP ABAP API reference*.

GetTemplate

The following code example shows how to use GetTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

    TRY.
        DATA(lo_result) = lo_ses->gettemplate( iv_templatename = iv_template_name ).
        oo_template = lo_result->get_template( ).
        MESSAGE 'Template retrieved successfully' TYPE 'I'.
    CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
        DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex1.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
        lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.

```

- For API details, see [GetTemplate](#) in *AWS SDK for SAP ABAP API reference*.

ListIdentities

The following code example shows how to use ListIdentities.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_ses->listidentities(  
    iv_identitytype = iv_identity_type  
    iv_maxitems = iv_max_items  
  ).  
  ot_identities = lo_result->get_identities( ).  
  MESSAGE 'Identities retrieved successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [ListIdentities](#) in *AWS SDK for SAP ABAP API reference*.

ListReceiptFilters

The following code example shows how to use ListReceiptFilters.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_ses->listreceiptfilters( ).  
  ot_filters = lo_result->get_filters( ).  
  MESSAGE 'Receipt filters retrieved successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [ListReceiptFilters](#) in *AWS SDK for SAP ABAP API reference*.

ListTemplates

The following code example shows how to use ListTemplates.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_ses->listtemplates( iv_maxitems = iv_max_items ).  
  ot_templates = lo_result->get_templatesmetadata( ).  
  MESSAGE 'Templates retrieved successfully' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
  DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [ListTemplates](#) in *AWS SDK for SAP ABAP API reference*.

SendEmail

The following code example shows how to use SendEmail.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
" Create message object
DATA(lo_subject) = NEW /aws1/cl_sescontent( iv_data = iv_subject ).
DATA(lo_text_body) = NEW /aws1/cl_sescontent( iv_data = iv_text ).
DATA(lo_html_body) = NEW /aws1/cl_sescontent( iv_data = iv_html ).
DATA(lo_body) = NEW /aws1/cl_sesbody(
  io_text = lo_text_body
  io_html = lo_html_body
).
DATA(lo_message) = NEW /aws1/cl_sesmessage(
  io_subject = lo_subject
  io_body = lo_body
).

TRY.
  " Send email
  DATA(lo_result) = lo_ses->sendemail(
    iv_source = iv_source
    io_destination = io_destination
    io_message = lo_message
    it_replytoaddresses = it_reply_tos
  ).
  ov_msg_id = lo_result->get_messageid( ).
  MESSAGE 'Email sent successfully' TYPE 'I'.
CATCH /aws1/cx_sesacctsendingpause00 INTO DATA(lo_ex1).
  DATA(lv_error) = |Account sending paused: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_sesmessagerejected INTO DATA(lo_ex2).
  lv_error = |Message rejected: { lo_ex2->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex2.
```

```
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- For API details, see [SendEmail](#) in *AWS SDK for SAP ABAP API reference*.

SendTemplatedEmail

The following code example shows how to use SendTemplatedEmail.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Send templated email
  DATA(lo_result) = lo_ses->sendtemplatedemail(
    iv_source = iv_source
    io_destination = io_destination
    iv_template = iv_template_name
    iv_templatedata = iv_template_data
    it_replytoaddresses = it_reply_tos
  ).
  ov_msg_id = lo_result->get_messageid( ).
  MESSAGE 'Templated email sent successfully' TYPE 'I'.
CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
  DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- For API details, see [SendTemplatedEmail](#) in *AWS SDK for SAP ABAP API reference*.

UpdateTemplate

The following code example shows how to use UpdateTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
  lo_ses->updatetemplate( io_template = lo_template ).
  MESSAGE 'Template updated successfully' TYPE 'I'.
CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
  DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex1.
CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
  lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex2.
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
  lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- For API details, see [UpdateTemplate](#) in *AWS SDK for SAP ABAP API reference*.

VerifyDomainIdentity

The following code example shows how to use VerifyDomainIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    DATA(lo_result) = lo_ses->verifydomainidentity( iv_domain =  
iv_domain_name ).  
    ov_token = lo_result->get_verificationtoken( ).  
    MESSAGE 'Domain verification initiated' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [VerifyDomainIdentity](#) in *AWS SDK for SAP ABAP API reference*.

VerifyEmailIdentity

The following code example shows how to use VerifyEmailIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    lo_ses->verifyemailidentity( iv_emailaddress = iv_email_address ).  
    MESSAGE 'Email verification initiated' TYPE 'I'.  
CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).  
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_ex.  
ENDTRY.
```

- For API details, see [VerifyEmailIdentity](#) in *AWS SDK for SAP ABAP API reference*.

Amazon SES API v2 examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon SES API v2.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateContact

The following code example shows how to use CreateContact.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_se2->createcontact(
    iv_contactlistname = iv_contact_list_name
    iv_emailaddress = iv_email_address ).
  MESSAGE 'Contact created successfully.' TYPE 'I'.
CATCH /aws1/cx_se2alreadyexistsex.
  MESSAGE 'Contact already exists.' TYPE 'I'.
CATCH /aws1/cx_se2badrequestex.
  MESSAGE 'Bad request.' TYPE 'E'.
CATCH /aws1/cx_se2notfoundexception.
  MESSAGE 'Contact list not found.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateContact](#) in *AWS SDK for SAP ABAP API reference*.

CreateContactList

The following code example shows how to use CreateContactList.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_se2->createcontactlist(
    iv_contactlistname = iv_contact_list_name ).
  MESSAGE 'Contact list created successfully.' TYPE 'I'.
CATCH /aws1/cx_se2alreadyexistsex.
  MESSAGE 'Contact list already exists.' TYPE 'I'.
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
  MESSAGE 'Bad request - contact list limit may be reached.' TYPE 'I'.
  " Re-raise the exception so the caller can handle it
  RAISE EXCEPTION lo_bad_request.
CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
  MESSAGE 'Limit exceeded - contact list limit reached.' TYPE 'I'.
  " Re-raise the exception so the caller can handle it

```

```
RAISE EXCEPTION lo_limit_exceeded.  
ENDTRY.
```

- For API details, see [CreateContactList](#) in *AWS SDK for SAP ABAP API reference*.

CreateEmailIdentity

The following code example shows how to use CreateEmailIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_se2->createemailidentity(  
    iv_emailidentity = iv_email_identity ).  
  MESSAGE 'Email identity created successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2alreadyexistsex.  
  MESSAGE 'Email identity already exists.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.  
CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).  
  MESSAGE lo_limit_exceeded TYPE 'I' DISPLAY LIKE 'E'.  
ENDTRY.
```

- For API details, see [CreateEmailIdentity](#) in *AWS SDK for SAP ABAP API reference*.

CreateEmailTemplate

The following code example shows how to use CreateEmailTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_template_content) = NEW /aws1/cl_se2emailtmplcontent(  
    iv_subject = iv_subject  
    iv_html = iv_html  
    iv_text = iv_text ).  
  
  lo_se2->createemailtemplate(  
    iv_templatename = iv_template_name  
    io_templatecontent = lo_template_content ).  
  MESSAGE 'Email template created successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2alreadyexistsex.  
  MESSAGE 'Email template already exists.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex.  
  MESSAGE 'Bad request.' TYPE 'E'.  
CATCH /aws1/cx_se2limitexceededex.  
  MESSAGE 'Limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [CreateEmailTemplate](#) in *AWS SDK for SAP ABAP API reference*.

DeleteContactList

The following code example shows how to use DeleteContactList.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_se2->deletecontactlist(  
    iv_contactlistname = iv_contact_list_name ).  
  MESSAGE 'Contact list deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Contact list not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- For API details, see [DeleteContactList](#) in *AWS SDK for SAP ABAP API reference*.

DeleteEmailIdentity

The following code example shows how to use DeleteEmailIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_se2->deleteemailidentity(  
    iv_emailidentity = iv_email_identity ).  
  MESSAGE 'Email identity deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Email identity not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- For API details, see [DeleteEmailIdentity](#) in *AWS SDK for SAP ABAP API reference*.

DeleteEmailTemplate

The following code example shows how to use DeleteEmailTemplate.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_se2->deleteemailtemplate(  
    iv_templatename = iv_template_name ).  
  MESSAGE 'Email template deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_se2notfoundexception.  
  MESSAGE 'Email template not found.' TYPE 'I'.  
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).  
  MESSAGE 'Bad request.' TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request.  
ENDTRY.
```

- For API details, see [DeleteEmailTemplate](#) in *AWS SDK for SAP ABAP API reference*.

GetEmailIdentity

The following code example shows how to use GetEmailIdentity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_se2->getemailidentity(  

```

```

        iv_emailidentity = iv_email_identity ).
    MESSAGE |Identity type: { oo_result->get_identitytype( ) }, | &&
        |verified for sending: { oo_result-
>get_verifiedforsendingstatus( ) }| TYPE 'I'.
    CATCH /aws1/cx_se2notfoundexception.
    MESSAGE |Email identity { iv_email_identity } not found.| TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_bad_request.
ENDTRY.

```

- For API details, see [GetEmailIdentity](#) in *AWS SDK for SAP ABAP API reference*.

ListContacts

The following code example shows how to use ListContacts.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    oo_result = lo_se2->listcontacts(
        iv_contactlistname = iv_contact_list_name ).
    DATA(lv_count) = lines( oo_result->get_contacts( ) ).
    MESSAGE |Retrieved { lv_count } contacts from list.| TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
    CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).
    MESSAGE 'Contact list not found.' TYPE 'I'.
    RAISE EXCEPTION lo_not_found.
ENDTRY.

```

- For API details, see [ListContacts](#) in *AWS SDK for SAP ABAP API reference*.

SendBulkEmail

The following code example shows how to use SendBulkEmail.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Build the default template content used for all bulk recipients
  DATA(lo_template) = NEW /aws1/cl_se2template(
    iv_templatename = iv_template_name
    iv_templatedata = iv_template_data ).

  DATA(lo_default_content) = NEW /aws1/cl_se2bulkemailcontent(
    io_template = lo_template ).

  DATA(lo_result) = lo_se2->sendbulkemail(
    iv_fromemailaddress = iv_from_address
    io_defaultcontent    = lo_default_content
    it_bulkemailentries = it_bulk_entries ).

  ot_results = lo_result->get_bulkemailentryresults( ).
  MESSAGE |Bulk email sent to { lines( it_bulk_entries ) } recipient(s).| TYPE
'I'.
  CATCH /aws1/cx_se2messagerejected INTO DATA(lo_rejected).
    MESSAGE lo_rejected TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_rejected.
  CATCH /aws1/cx_se2mailfrmdomnotver00 INTO DATA(lo_not_verified).
    MESSAGE lo_not_verified TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_not_verified.
  CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).
    MESSAGE lo_not_found TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_not_found.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_bad_request.
ENDTRY.
```

- For API details, see [SendBulkEmail](#) in *AWS SDK for SAP ABAP API reference*.

SendEmail

The following code example shows how to use SendEmail.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Sends a message.

```
TRY.
    " Create destination with recipient address
    DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
    APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
TO lt_to_addresses.
    DATA(lo_destination) = NEW /aws1/cl_se2destination(
        it_toaddresses = lt_to_addresses ).

    " Create message content
    DATA(lo_subject) = NEW /aws1/cl_se2content( iv_data = iv_subject ).
    DATA(lo_text_body) = NEW /aws1/cl_se2content( iv_data = iv_text_body ).
    DATA(lo_html_body) = NEW /aws1/cl_se2content( iv_data = iv_html_body ).
    DATA(lo_body) = NEW /aws1/cl_se2body(
        io_text = lo_text_body
        io_html = lo_html_body ).
    DATA(lo_message) = NEW /aws1/cl_se2message(
        io_subject = lo_subject
        io_body = lo_body ).

    DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
        io_simple = lo_message ).

    " Send the email
```

```

lo_se2->sendemail(
  iv_fromemailaddress = iv_from_email_address
  io_destination = lo_destination
  io_content = lo_content ).
MESSAGE 'Email sent successfully.' TYPE 'I'.
CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
MESSAGE 'Account suspended.' TYPE 'I'.
RAISE EXCEPTION lo_account_suspended.
CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
MESSAGE 'Bad request.' TYPE 'I'.
RAISE EXCEPTION lo_bad_request.
CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
RAISE EXCEPTION lo_message_rejected.
ENDTRY.

```

Sends a message using a template.

```

TRY.
  " Create destination with recipient address
  DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
  APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
TO lt_to_addresses.
  DATA(lo_destination) = NEW /aws1/cl_se2destination(
    it_toaddresses = lt_to_addresses ).

  " Create template reference
  DATA(lo_template) = NEW /aws1/cl_se2template(
    iv_templatename = iv_template_name
    iv_templatedata = iv_template_data ).

  DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
    io_template = lo_template ).

  " Create list management options
  DATA(lo_list_mgmt) = NEW /aws1/cl_se2listmanagementopts(
    iv_contactlistname = iv_contact_list_name ).

  " Send the email using template
  lo_se2->sendemail(
    iv_fromemailaddress = iv_from_email_address

```

```
        io_destination = lo_destination
        io_content = lo_content
        io_listmanagementoptions = lo_list_mgmt ).
    MESSAGE 'Email sent using template successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
    MESSAGE 'Account suspended.' TYPE 'I'.
    RAISE EXCEPTION lo_account_suspended.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
    CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
    MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
    RAISE EXCEPTION lo_message_rejected.
ENDTRY.
```

- For API details, see [SendEmail](#) in *AWS SDK for SAP ABAP API reference*.

Amazon SNS examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon SNS.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)
- [Scenarios](#)

Actions

CreateTopic

The following code example shows how to use CreateTopic.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [CreateTopic](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTopic

The following code example shows how to use DeleteTopic.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [DeleteTopic](#) in *AWS SDK for SAP ABAP API reference*.

GetTopicAttributes

The following code example shows how to use GetTopicAttributes.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [GetTopicAttributes](#) in *AWS SDK for SAP ABAP API reference*.

ListSubscriptions

The following code example shows how to use ListSubscriptions.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListSubscriptions](#) in *AWS SDK for SAP ABAP API reference*.

ListTopics

The following code example shows how to use ListTopics.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [ListTopics](#) in *AWS SDK for SAP ABAP API reference*.

Publish

The following code example shows how to use Publish.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

Publish a message with attributes to a topic.

```
TRY.
    oo_result = lo_sns->publish(           " oo_result is returned for
testing purposes. "
    iv_topicarn = iv_topic_arn
    iv_message = iv_message
    it_messageattributes = it_msg_attrs ).
    MESSAGE 'Message with attributes published to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
ENDTRY.
```

Publish a multi-format message to a topic.

```

" Build JSON message structure for multi-format message
DATA(lv_json_message) = |\{ "default": "{ iv_default_message }", "sms":
"{ iv_sms_message }", "email": "{ iv_email_message }" \}|.

TRY.
    oo_result = lo_sns->publish(
testing purposes. "
    iv_topicarn = iv_topic_arn
    iv_message = lv_json_message
    iv_subject = iv_subject
    iv_messagestructure = 'json' ).
    MESSAGE 'Multi-format message published to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
ENDTRY.

```

- For API details, see [Publish](#) in *AWS SDK for SAP ABAP API reference*.

SetSubscriptionAttributes

The following code example shows how to use SetSubscriptionAttributes.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    lo_sns->setsubscriptionattributes(
    iv_subscriptionarn = iv_subscription_arn
    iv_attributename = 'FilterPolicy'
    iv_attributevalue = iv_filter_policy ).
    MESSAGE 'Added filter policy to subscription.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Subscription does not exist.' TYPE 'E'.

```

```
ENDTRY.
```

- For API details, see [SetSubscriptionAttributes](#) in *AWS SDK for SAP ABAP API reference*.

SetTopicAttributes

The following code example shows how to use SetTopicAttributes.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [SetTopicAttributes](#) in *AWS SDK for SAP ABAP API reference*.

Subscribe

The following code example shows how to use Subscribe.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Subscribe an email address to a topic.

```
TRY.  
    oo_result = lo_sns->subscribe(                          "oo_result is returned  
for testing purposes."  
        iv_topicarn = iv_topic_arn  
        iv_protocol = 'email'  
        iv_endpoint = iv_email_address  
        iv_returnsubscriptionarn = abap_true ).  
    MESSAGE 'Email address subscribed to SNS topic.' TYPE 'I'.  
    CATCH /aws1/cx_snsnotfoundexception.  
        MESSAGE 'Topic does not exist.' TYPE 'E'.  
    CATCH /aws1/cx_snssubscriptionlmt00.  
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number  
of subscriptions allowed.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [Subscribe](#) in *AWS SDK for SAP ABAP API reference*.

Unsubscribe

The following code example shows how to use Unsubscribe.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.

```

- For API details, see [Unsubscribe](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Create and publish to a FIFO topic

The following code example shows how to create and publish to a FIFO Amazon SNS topic.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create a FIFO topic, subscribe an Amazon SQS FIFO queue to the topic, and publish a message to an Amazon SNS topic.

```

" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsm_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsm_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsm_w( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

TRY.
    DATA(lo_create_result) = lo_sns->createtopic(
        iv_name = iv_topic_name

```

```

        it_attributes = lt_tpc_attributes ).
    DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
    ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
    MESSAGE 'FIFO topic created' TYPE 'I'.
    CATCH /aws1/cx_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'.
    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.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CreateTopic](#)
 - [Publish](#)
 - [Subscribe](#)

Publish an SMS text message

The following code example shows how to publish SMS messages using Amazon SNS.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

" iv_phone_number = '+12065550101' - Phone number in E.164 format
TRY.
    oo_result = lo_sns->publish(
testing purposes. "
        iv_phonenumber = iv_phone_number
        iv_message = iv_message ).
    MESSAGE 'Message published to phone number.' TYPE 'I'.

```

```
CATCH /aws1/cx_snsnotfoundexception.  
    MESSAGE 'Phone number does not exist.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [Publish](#) in *AWS SDK for SAP ABAP API reference*.

Amazon SQS examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon SQS.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)
- [Scenarios](#)

Actions

CreateQueue

The following code example shows how to use CreateQueue.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create an Amazon SQS standard queue.

```

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.

```

Create an Amazon SQS queue that waits for a message to arrive.

```

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.

```

- For API details, see [CreateQueue](#) in *AWS SDK for SAP ABAP API reference*.

DeleteMessage

The following code example shows how to use DeleteMessage.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_sqs->deletemessage(  
    iv_queueurl = iv_queue_url  
    iv_receipthandle = iv_receipt_handle ).  
  MESSAGE 'Message deleted from SQS queue.' TYPE 'I'.  
CATCH /aws1/cx_sqsinvalididformat.  
  MESSAGE 'The specified receipt handle is not valid.' TYPE 'E'.  
CATCH /aws1/cx_sqsreceipthandleisinv.  
  MESSAGE 'The specified receipt handle is not valid for the current version.'  
TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteMessage](#) in *AWS SDK for SAP ABAP API reference*.

DeleteMessageBatch

The following code example shows how to use DeleteMessageBatch.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
oo_result = lo_sqs->deletemessagebatch(      " oo_result is returned for
testing purposes. "
    iv_queueurl = iv_queue_url
    it_entries = it_entries ).
MESSAGE 'Messages deleted from SQS queue.' TYPE 'I'.
CATCH /aws1/cx_sqsbtcentidsnotdist00.
    MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
CATCH /aws1/cx_sqsemtypbatchrequest.
    MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
CATCH /aws1/cx_sqsinvbatchentryid.
    MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
CATCH /aws1/cx_sqstoomanyentriesin00.
    MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
ENDTRY.
```

- For API details, see [DeleteMessageBatch](#) in *AWS SDK for SAP ABAP API reference*.

DeleteQueue

The following code example shows how to use DeleteQueue.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    lo_sqs->deletequeue( iv_queueurl = iv_queue_url ).
    MESSAGE 'SQS queue deleted' TYPE 'I'.
ENDTRY.
```

- For API details, see [DeleteQueue](#) in *AWS SDK for SAP ABAP API reference*.

GetQueueUrl

The following code example shows how to use `GetQueueUrl`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
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.
```

- For API details, see [GetQueueUrl](#) in *AWS SDK for SAP ABAP API reference*.

ListQueues

The following code example shows how to use `ListQueues`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_sqs->listqueues( ).      " oo_result is returned for  
testing purposes. "  
    MESSAGE 'Retrieved list of queues.' TYPE 'I'.
```

```
ENDTRY.
```

- For API details, see [ListQueues](#) in *AWS SDK for SAP ABAP API reference*.

ReceiveMessage

The following code example shows how to use ReceiveMessage.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Receive a message from an Amazon SQS queue.

```
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.
```

Receive a message from an Amazon SQS queue using long-poll support.

```
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.
```

- For API details, see [ReceiveMessage](#) in *AWS SDK for SAP ABAP API reference*.

SendMessage

The following code example shows how to use SendMessage.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_sqs->sendmessage(           " oo_result is returned for  
testing purposes. "  
    iv_queueurl = iv_queue_url  
    iv_messagebody = iv_message ).  
    MESSAGE 'Message sent to SQS queue.' TYPE 'I'.  
CATCH /aws1/cx_sqsinvalidmsgconts.  
    MESSAGE 'Message contains non-valid characters.' TYPE 'E'.  
CATCH /aws1/cx_sqsunsupportedop.  
    MESSAGE 'Operation not supported.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [SendMessage](#) in *AWS SDK for SAP ABAP API reference*.

SendMessageBatch

The following code example shows how to use SendMessageBatch.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
    oo_result = lo_sqs->sendmessagebatch(          " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        it_entries = it_messages ).
    MESSAGE 'Messages sent to SQS queue.' TYPE 'I'.
CATCH /aws1/cx_sqsbtcentsnotdist00.
    MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
CATCH /aws1/cx_sqsbatchreqtoolong.
    MESSAGE 'The length of all the messages put together is more than the
limit.' TYPE 'E'.
CATCH /aws1/cx_sqsempybatchrequest.
    MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
CATCH /aws1/cx_sqsinvbatchentryid.
    MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
CATCH /aws1/cx_sqstoomanyentriesin00.
    MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
CATCH /aws1/cx_sqsunsupportedop.
    MESSAGE 'Operation not supported.' TYPE 'E'.
ENDTRY.
```

- For API details, see [SendMessageBatch](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Create and publish to a FIFO topic

The following code example shows how to create and publish to a FIFO Amazon SNS topic.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

Create a FIFO topic, subscribe an Amazon SQS FIFO queue to the topic, and publish a message to an Amazon SNS topic.

```

" 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_snssubscriptionlmt00.
        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.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [CreateTopic](#)

- [Publish](#)
- [Subscribe](#)

Step Functions examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Step Functions.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateActivity

The following code example shows how to use CreateActivity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_sfn->createactivity(  
    iv_name = iv_name  
  ).  
  ov_activity_arn = lo_result->get_activityarn( ).  
  MESSAGE 'Activity created successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnactivityalrddyex.  
  MESSAGE 'Activity already exists.' TYPE 'E'.
```

```

CATCH /aws1/cx_sfninvalidname.
  MESSAGE 'Invalid activity name.' TYPE 'E'.
CATCH /aws1/cx_sfnactivitylimitexcd.
  MESSAGE 'Activity limit exceeded.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateActivity](#) in *AWS SDK for SAP ABAP API reference*.

CreateStateMachine

The following code example shows how to use CreateStateMachine.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  DATA(lo_result) = lo_sfn->createstatemachine(
    iv_name = iv_name
    iv_definition = iv_definition
    iv_rolearn = iv_role_arn
  ).
  ov_state_machine_arn = lo_result->get_statemachinearn( ).
  MESSAGE 'State machine created successfully.' TYPE 'I'.
CATCH /aws1/cx_sfnstatemachinealrex.
  MESSAGE 'State machine already exists.' TYPE 'E'.
CATCH /aws1/cx_sfninvaliddefinition.
  MESSAGE 'Invalid state machine definition.' TYPE 'E'.
CATCH /aws1/cx_sfninvalidname.
  MESSAGE 'Invalid state machine name.' TYPE 'E'.
CATCH /aws1/cx_sfninvalidarn.
  MESSAGE 'Invalid role ARN.' TYPE 'E'.
ENDTRY.

```

- For API details, see [CreateStateMachine](#) in *AWS SDK for SAP ABAP API reference*.

DeleteActivity

The following code example shows how to use DeleteActivity.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_sfn->deleteactivity(  
    iv_activityarn = iv_activity_arn  
  ).  
  MESSAGE 'Activity deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid activity ARN.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DeleteActivity](#) in *AWS SDK for SAP ABAP API reference*.

DeleteStateMachine

The following code example shows how to use DeleteStateMachine.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_sfn->deletestatemachine(  
    iv_statemachinarn = iv_state_machine_arn
```

```
    ).  
    MESSAGE 'State machine deleted successfully.' TYPE 'I'.  
  CATCH /aws1/cx_sfninvalidarn.  
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.  
  CATCH /aws1/cx_sfnvalidationex.  
    MESSAGE 'Validation error occurred.' TYPE 'E'.  
  ENDRTRY.
```

- For API details, see [DeleteStateMachine](#) in *AWS SDK for SAP ABAP API reference*.

DescribeExecution

The following code example shows how to use DescribeExecution.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
  TRY.  
    oo_result = lo_sfn->describeexecution(  
      iv_executionarn = iv_execution_arn  
    ).  
    MESSAGE 'Execution described successfully.' TYPE 'I'.  
  CATCH /aws1/cx_sfnexecdoesnotexist.  
    MESSAGE 'Execution does not exist.' TYPE 'E'.  
  CATCH /aws1/cx_sfninvalidarn.  
    MESSAGE 'Invalid execution ARN.' TYPE 'E'.  
  ENDRTRY.
```

- For API details, see [DescribeExecution](#) in *AWS SDK for SAP ABAP API reference*.

DescribeStateMachine

The following code example shows how to use DescribeStateMachine.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_sfn->describestatemachine(  
    iv_statemachearn = iv_state_machine_arn  
  ).  
  MESSAGE 'State machine described successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnstatemachinedoes00.  
  MESSAGE 'State machine does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid state machine ARN.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [DescribeStateMachine](#) in *AWS SDK for SAP ABAP API reference*.

GetActivityTask

The following code example shows how to use `GetActivityTask`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  oo_result = lo_sfn->getactivitytask(  
    iv_activityarn = iv_activity_arn  
  ).  
  MESSAGE 'Activity task retrieved successfully.' TYPE 'I'.
```

```
CATCH /aws1/cx_sfnactivitydoesnotex.  
  MESSAGE 'Activity does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfninvalidarn.  
  MESSAGE 'Invalid activity ARN.' TYPE 'E'.  
CATCH /aws1/cx_sfnactivityworkerlm00.  
  MESSAGE 'Activity worker limit exceeded.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [GetActivityTask](#) in *AWS SDK for SAP ABAP API reference*.

ListActivities

The following code example shows how to use ListActivities.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_sfn->listactivities( ).  
  DATA(lt_activities) = lo_result->get_activities( ).  
  LOOP AT lt_activities INTO DATA(lo_activity).  
    IF lo_activity->get_name( ) = iv_name.  
      ov_activity_arn = lo_activity->get_activityarn( ).  
      EXIT.  
    ENDIF.  
  ENDLOOP.  
  MESSAGE 'Activities listed successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfninvalidtoken.  
  MESSAGE 'Invalid pagination token.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListActivities](#) in *AWS SDK for SAP ABAP API reference*.

ListStateMachines

The following code example shows how to use ListStateMachines.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_sfn->liststatemachines( ).  
  DATA(lt_state_machines) = lo_result->get_statemachines( ).  
  LOOP AT lt_state_machines INTO DATA(lo_state_machine).  
    IF lo_state_machine->get_name( ) = iv_name.  
      ov_state_machine_arn = lo_state_machine->get_statemachinearn( ).  
      EXIT.  
    ENDIF.  
  ENDLOOP.  
  MESSAGE 'State machines listed successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfninvalidtoken.  
  MESSAGE 'Invalid pagination token.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [ListStateMachines](#) in *AWS SDK for SAP ABAP API reference*.

SendTaskSuccess

The following code example shows how to use SendTaskSuccess.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_sfn->sendtasksuccess(  
    iv_tasktoken = iv_task_token  
    iv_output = iv_task_response  
  ).  
  MESSAGE 'Task success sent successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfinvalidtoken.  
  MESSAGE 'Invalid task token.' TYPE 'E'.  
CATCH /aws1/cx_sfntaskdoesnotexist.  
  MESSAGE 'Task does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfinvalidoutput.  
  MESSAGE 'Invalid task output.' TYPE 'E'.  
CATCH /aws1/cx_sfntasktimedout.  
  MESSAGE 'Task timed out.' TYPE 'E'.  
ENDTRY.
```

- For API details, see [SendTaskSuccess](#) in *AWS SDK for SAP ABAP API reference*.

StartExecution

The following code example shows how to use StartExecution.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_sfn->startexecution(  
    iv_statemachearn = iv_state_machine_arn  
    iv_input = iv_input  
  ).  
  ov_execution_arn = lo_result->get_executionarn( ).  
  MESSAGE 'Execution started successfully.' TYPE 'I'.  
CATCH /aws1/cx_sfnstatemachinedoes00.  
  MESSAGE 'State machine does not exist.' TYPE 'E'.  
CATCH /aws1/cx_sfinvalidarn.
```

```
MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
CATCH /aws1/cx_sfninvalidexecinput.
MESSAGE 'Invalid execution input.' TYPE 'E'.
CATCH /aws1/cx_sfnexeclimitexceeded.
MESSAGE 'Execution limit exceeded.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartExecution](#) in *AWS SDK for SAP ABAP API reference*.

Systems Manager examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Systems Manager.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateDocument

The following code example shows how to use CreateDocument.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
lo_ssm->createdocument(  
    iv_name = iv_name  
    iv_content = iv_content  
    iv_documenttype = 'Command' ).  
MESSAGE 'Document created.' TYPE 'I'.  
CATCH /aws1/cx_ssmdocalreadyexists.  
MESSAGE 'Document already exists.' TYPE 'I'.  
CATCH /aws1/cx_ssminvaliddoccontent.  
MESSAGE 'Invalid document content.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [CreateDocument](#) in *AWS SDK for SAP ABAP API reference*.

CreateMaintenanceWindow

The following code example shows how to use CreateMaintenanceWindow.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ssm->createmaintenancewindow(  
        iv_name = iv_name  
        iv_schedule = iv_schedule  
        iv_duration = iv_duration  
        iv_cutoff = iv_cutoff  
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).  
MESSAGE 'Maintenance window created.' TYPE 'I'.  
CATCH /aws1/cx_ssmresrclimitexcdex.  
MESSAGE 'Resource limit exceeded.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [CreateMaintenanceWindow](#) in *AWS SDK for SAP ABAP API reference*.

CreateOpsItem

The following code example shows how to use CreateOpsItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_ssm->createopsitem(  
        iv_title = iv_title  
        iv_source = iv_source  
        iv_category = iv_category  
        iv_severity = iv_severity  
        iv_description = iv_description ).  
    MESSAGE 'OpsItem created.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsitemlimitexcdex.  
    MESSAGE 'You have exceeded your open OpsItem limit.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsitemalrddyexex.  
    MESSAGE 'OpsItem already exists.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [CreateOpsItem](#) in *AWS SDK for SAP ABAP API reference*.

DeleteDocument

The following code example shows how to use DeleteDocument.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ssm->deletedocument( iv_name = iv_name ).  
  MESSAGE 'Document deleted.' TYPE 'I'.  
CATCH /aws1/cx_ssminvaliddocument.  
  MESSAGE 'Invalid document.' TYPE 'I'.  
CATCH /aws1/cx_ssmassocinstances.  
  MESSAGE 'Document has associated instances.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteDocument](#) in *AWS SDK for SAP ABAP API reference*.

DeleteMaintenanceWindow

The following code example shows how to use DeleteMaintenanceWindow.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ssm->deletemaintenancewindow( iv_windowid = iv_window_id ).  
  MESSAGE 'Maintenance window deleted.' TYPE 'I'.  
CATCH /aws1/cx_ssminternalservererr.  
  MESSAGE 'Internal server error occurred.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteMaintenanceWindow](#) in *AWS SDK for SAP ABAP API reference*.

DeleteOpsItem

The following code example shows how to use DeleteOpsItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ssm->deleteopsitem( iv_opsitemid = iv_ops_item_id ).  
  MESSAGE 'OpsItem deleted.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsiteminvparamex.  
  MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [DeleteOpsItem](#) in *AWS SDK for SAP ABAP API reference*.

DescribeDocument

The following code example shows how to use DescribeDocument.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  DATA(lo_result) = lo_ssm->describedocument( iv_name = iv_name ).  
  DATA(lo_document) = lo_result->get_document( ).  
  IF lo_document IS BOUND.  
    rv_status = lo_document->get_status( ).  
    MESSAGE |Document status: { rv_status }| TYPE 'I'.  
  ENDIF.  
CATCH /aws1/cx_ssminvaliddocument.  
  MESSAGE 'Invalid document.' TYPE 'I'.
```

```
ENDTRY.
```

- For API details, see [DescribeDocument](#) in *AWS SDK for SAP ABAP API reference*.

DescribeOpsItems

The following code example shows how to use DescribeOpsItems.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  " Create filter for OpsItem ID
  DATA(lt_filters) = VALUE /aws1/cl_ssmopsitemfilter=>tt_opsitemfilters(
    ( NEW /aws1/cl_ssmopsitemfilter(
      iv_key = 'OpsItemId'
      it_values = VALUE /aws1/
cl_ssmopsitemfiltvals_w=>tt_opsitemfiltervalues(
      ( NEW /aws1/cl_ssmopsitemfiltvals_w( iv_value = iv_ops_item_id ) )
    )
    iv_operator = 'Equal'
  ) )
  ).

  " Use paginator to get all results
  DATA(lo_paginator) = lo_ssm->get_paginator( ).
  DATA(lo_iterator) = lo_paginator->describeopsitems(
    it_opsitemfilters = lt_filters ).

  rv_found = abap_false.

  WHILE lo_iterator->has_next( ).
    DATA(lo_result) = CAST /aws1/cl_ssmdescropsitemrsp( lo_iterator-
>get_next( ) ).
    LOOP AT lo_result->get_opsitemsummaries( ) INTO DATA(lo_item).
      DATA(lv_title) = lo_item->get_title( ).
```

```

        DATA(lv_status) = lo_item->get_status( ).
        MESSAGE |The OpsItem title is { lv_title } and the status is
        { lv_status }| TYPE 'I'.
        rv_found = abap_true.
    ENDLLOOP.
ENDWHILE.
CATCH /aws1/cx_ssminternalservererr.
    MESSAGE 'Internal server error occurred.' TYPE 'I'.
ENDTRY.

```

- For API details, see [DescribeOpsItems](#) in *AWS SDK for SAP ABAP API reference*.

ListCommandInvocations

The following code example shows how to use ListCommandInvocations.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    " Use paginator to get all results
    DATA(lo_paginator) = lo_ssm->get_paginator( ).
    DATA(lo_iterator) = lo_paginator->listcommandinvocations(
        iv_instanceid = iv_instance_id ).

    DATA lv_count TYPE i VALUE 0.

    WHILE lo_iterator->has_next( ).
        DATA(lo_result) = CAST /aws1/cl_ssmlistcmdinvcsresult( lo_iterator-
>get_next( ) ).
        LOOP AT lo_result->get_commandinvocations( ) INTO DATA(lo_invocation).
            lv_count = lv_count + 1.
            DATA(lv_requested_datetime) = lo_invocation->get_requesteddatetime( ).
            MESSAGE |Command invocation requested at: { lv_requested_datetime }|
TYPE 'I'.
        ENDLLOOP.

```

```

        ENDWHILE.

        MESSAGE |{ lv_count } command invocation(s) found for instance
{ iv_instance_id }.| TYPE 'I'.
        CATCH /aws1/cx_ssminvalidinstanceid.
        MESSAGE 'Invalid instance ID.' TYPE 'I'.
        CATCH /aws1/cx_ssminvalidcommandid.
        MESSAGE 'Invalid command ID.' TYPE 'I'.
    ENDTRY.

```

- For API details, see [ListCommandInvocations](#) in *AWS SDK for SAP ABAP API reference*.

SendCommand

The following code example shows how to use SendCommand.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
    DATA(lo_result) = lo_ssm->sendcommand(
        it_instanceids = it_instance_ids
        iv_documentname = iv_document_name
        iv_timeoutseconds = 3600 ).
    DATA(lo_command) = lo_result->get_command( ).
    IF lo_command IS BOUND.
        rv_command_id = lo_command->get_commandid( ).
        MESSAGE 'Command sent successfully.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_ssminvaliddocument.
        MESSAGE 'Invalid document.' TYPE 'I'.
    CATCH /aws1/cx_ssminvalidinstanceid.
        MESSAGE 'Invalid instance ID.' TYPE 'I'.
ENDTRY.

```

- For API details, see [SendCommand](#) in *AWS SDK for SAP ABAP API reference*.

UpdateMaintenanceWindow

The following code example shows how to use UpdateMaintenanceWindow.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ssm->updatemaintenancewindow(  
    iv_windowid = iv_window_id  
    iv_name = iv_name  
    iv_enabled = iv_enabled  
    iv_schedule = iv_schedule  
    iv_duration = iv_duration  
    iv_cutoff = iv_cutoff  
    iv_allowunassociatedtargets = iv_allow_unassociated_targets ).  
  MESSAGE 'Maintenance window updated.' TYPE 'I'.  
CATCH /aws1/cx_ssmdoesnotexistex.  
  MESSAGE 'Maintenance window does not exist.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [UpdateMaintenanceWindow](#) in *AWS SDK for SAP ABAP API reference*.

UpdateOpsItem

The following code example shows how to use UpdateOpsItem.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  lo_ssm->updateopsitem(  
    iv_opsiteid = iv_ops_item_id  
    iv_title = iv_title  
    iv_description = iv_description  
    iv_status = iv_status ).  
  MESSAGE 'OpsItem updated.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsitemnotfoundex.  
  MESSAGE 'OpsItem not found.' TYPE 'I'.  
CATCH /aws1/cx_ssmopsiteminvparamex.  
  MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.  
ENDTRY.
```

- For API details, see [UpdateOpsItem](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Textract examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Textract.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

- [Scenarios](#)

Actions

AnalyzeDocument

The following code example shows how to use AnalyzeDocument.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
"Detects text and additional elements, such as forms or tables,"  
"in a local image file or from in-memory byte data."  
"The image must be in PNG or JPG format."  
  
"Create ABAP objects for feature type."  
"Add TABLES to return information about the tables."  
"Add FORMS to return detected form data."  
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."  
  
DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(  
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )  
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).  
  
"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)  
object."  
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket  
  iv_name = iv_s3object ).  
  
"Create an ABAP object for the document."  
DATA(lo_document) = NEW /aws1/cl_texdocument( io_s3object = lo_s3object ).  
  
"Analyze document stored in Amazon S3."  
TRY.
```

```

        oo_result = lo_tex->analyzedocument(      "oo_result is returned for testing
purposes."
        io_document      = lo_document
        it_featuretypes  = lt_featuretypes ).
LOOP AT oo_result->get_blocks( ) INTO DATA(lo_block).
    IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR,'.
        MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
    ENDIF.
ENDLOOP.
MESSAGE 'Analyze document completed.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
    MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
    MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
    MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texhlquotaexceededex.
    MESSAGE 'Human loop quota exceeded.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
    MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
    MESSAGE 'Request has non-valid parameters.' TYPE 'E'.

CATCH /aws1/cx_texinvalids3objectex.
    MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

```

- For API details, see [AnalyzeDocument](#) in *AWS SDK for SAP ABAP API reference*.

DetectDocumentText

The following code example shows how to use DetectDocumentText.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"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.

```

- For API details, see [DetectDocumentText](#) in *AWS SDK for SAP ABAP API reference*.

GetDocumentAnalysis

The following code example shows how to use GetDocumentAnalysis.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"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_texaccesssdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidjobidex.
  MESSAGE 'Job ID is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
ENDTRY.
```

- For API details, see [GetDocumentAnalysis](#) in *AWS SDK for SAP ABAP API reference*.

StartDocumentAnalysis

The following code example shows how to use StartDocumentAnalysis.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

"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_texaccesssdeniedex.
```

```
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_texlimitexceededex.
```

```

    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
    CATCH /aws1/cx_texpvthruputexclex.
    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.

```

- For API details, see [StartDocumentAnalysis](#) in *AWS SDK for SAP ABAP API reference*.

StartDocumentTextDetection

The following code example shows how to use StartDocumentTextDetection.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"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.
```

- For API details, see [StartDocumentTextDetection](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Get started with document analysis

The following code example shows how to:

- Start asynchronous analysis.
- Get document analysis.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name   = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

"Start document analysis."
TRY.
  DATA(lo_start_result) = lo_tex->startdocumentanalysis(
    io_documentlocation   = lo_documentlocation
    it_featuretypes       = lt_featuretypes ).
  MESSAGE 'Document analysis started.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texidempotentprmmis00.
  MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.

```

```

    MESSAGE 'Internal server error.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidkmskeyex.
    MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidparameterex.
    MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
  CATCH /aws1/cx_texinvalids3objectex.
    MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
  CATCH /aws1/cx_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.

"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.

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [GetDocumentAnalysis](#)
 - [StartDocumentAnalysis](#)

Amazon Transcribe examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Transcribe.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)

Actions

CreateVocabulary

The following code example shows how to use CreateVocabulary.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      it_phrases = it_phrases ).
  ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      iv_vocabularyfileuri = iv_vocab_file_uri ).
```

```

ENDIF.
MESSAGE 'Custom vocabulary created.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
MESSAGE lo_bad_request_ex TYPE 'I'.
RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnblimitexceeddex INTO DATA(lo_limit_ex).
MESSAGE lo_limit_ex TYPE 'I'.
RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
MESSAGE lo_conflict_ex TYPE 'I'.
RAISE EXCEPTION lo_conflict_ex.
ENDTRY.

```

- For API details, see [CreateVocabulary](#) in *AWS SDK for SAP ABAP API reference*.

DeleteTranscriptionJob

The following code example shows how to use DeleteTranscriptionJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

TRY.
  lo_tnb->deletetranscriptionjob( iv_job_name ).
  MESSAGE 'Transcription job deleted.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnblimitexceeddex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).

```

```
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- For API details, see [DeleteTranscriptionJob](#) in *AWS SDK for SAP ABAP API reference*.

DeleteVocabulary

The following code example shows how to use DeleteVocabulary.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  lo_tnb->deletevocabulary( iv_vocabulary_name ).
  MESSAGE 'Vocabulary deleted.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex TYPE 'I'.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- For API details, see [DeleteVocabulary](#) in *AWS SDK for SAP ABAP API reference*.

GetTranscriptionJob

The following code example shows how to use GetTranscriptionJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
    oo_result = lo_tnb->gettranscriptionjob( iv_job_name ).  
    DATA(lo_job) = oo_result->get_transcriptionjob( ).  
    MESSAGE 'Retrieved transcription job details.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
    MESSAGE lo_internal_ex TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- For API details, see [GetTranscriptionJob](#) in *AWS SDK for SAP ABAP API reference*.

GetVocabulary

The following code example shows how to use GetVocabulary.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
```

```
oo_result = lo_tnb->getvocabulary( iv_vocabulary_name ).
MESSAGE 'Retrieved vocabulary details.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
MESSAGE lo_bad_request_ex TYPE 'I'.
RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
MESSAGE lo_not_found_ex TYPE 'I'.
RAISE EXCEPTION lo_not_found_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
MESSAGE lo_internal_ex TYPE 'I'.
RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- For API details, see [GetVocabulary](#) in *AWS SDK for SAP ABAP API reference*.

ListTranscriptionJobs

The following code example shows how to use ListTranscriptionJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  IF iv_job_filter IS NOT INITIAL.
    oo_result = lo_tnb->listtranscriptionjobs( iv_jobnamecontains =
iv_job_filter ).
  ELSE.
    oo_result = lo_tnb->listtranscriptionjobs( ).
  ENDIF.
MESSAGE 'Retrieved transcription jobs list.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
MESSAGE lo_bad_request_ex TYPE 'I'.
RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
MESSAGE lo_internal_ex TYPE 'I'.
```

```
    RAISE EXCEPTION lo_internal_ex.  
  ENDTRY.
```

- For API details, see [ListTranscriptionJobs](#) in *AWS SDK for SAP ABAP API reference*.

ListVocabularies

The following code example shows how to use ListVocabularies.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.  
  IF iv_vocab_filter IS NOT INITIAL.  
    oo_result = lo_tnb->listvocabularies( iv_namecontains = iv_vocab_filter ).  
  ELSE.  
    oo_result = lo_tnb->listvocabularies( ).  
  ENDIF.  
  MESSAGE 'Retrieved vocabularies list.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
  MESSAGE lo_bad_request_ex TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
  MESSAGE lo_internal_ex TYPE 'I'.  
  RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- For API details, see [ListVocabularies](#) in *AWS SDK for SAP ABAP API reference*.

StartTranscriptionJob

The following code example shows how to use StartTranscriptionJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
  DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
  IF iv_vocabulary_name IS NOT INITIAL.
    lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
  ENDIF.

  oo_result = lo_tnb->starttranscriptionjob(
    iv_transcriptionjobname = iv_job_name
    io_media = lo_media
    iv_mediaformat = iv_media_format
    iv_languagecode = iv_language_code
    io_settings = lo_settings ).

  MESSAGE 'Transcription job started.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnblimitexceeddex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
  CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
  MESSAGE lo_conflict_ex TYPE 'I'.
  RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- For API details, see [StartTranscriptionJob](#) in *AWS SDK for SAP ABAP API reference*.

UpdateVocabulary

The following code example shows how to use UpdateVocabulary.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
TRY.
  IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      it_phrases = it_phrases ).
  ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      iv_vocabularyfileuri = iv_vocab_file_uri ).
  ENDIF.
  MESSAGE 'Vocabulary updated.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
CATCH /aws1/cx_tnbblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex TYPE 'I'.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
  MESSAGE lo_conflict_ex TYPE 'I'.
  RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- For API details, see [UpdateVocabulary](#) in *AWS SDK for SAP ABAP API reference*.

Amazon Translate examples using SDK for SAP ABAP

The following code examples show you how to perform actions and implement common scenarios by using the AWS SDK for SAP ABAP with Amazon Translate.

Actions are code excerpts from larger programs and must be run in context. While actions show you how to call individual service functions, you can see actions in context in their related scenarios.

Scenarios are code examples that show you how to accomplish specific tasks by calling multiple functions within a service or combined with other AWS services.

Each example includes a link to the complete source code, where you can find instructions on how to set up and run the code in context.

Topics

- [Actions](#)
- [Scenarios](#)

Actions

DescribeTextTranslationJob

The following code example shows how to use `DescribeTextTranslationJob`.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```
"Gets the properties associated with an asynchronous batch translation job."  
"Includes properties such as name, ID, status, source and target languages, and  
input/output Amazon Simple Storage Service (Amazon S3) buckets."  
TRY.  
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned  
for testing purposes."  
    iv_jobid      = iv_jobid ).
```

```

    MESSAGE 'Job description retrieved.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex.
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex.
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
  ENDTRY.

```

- For API details, see [DescribeTextTranslationJob](#) in *AWS SDK for SAP ABAP API reference*.

ListTextTranslationJobs

The following code example shows how to use ListTextTranslationJobs.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

DATA lo_filter TYPE REF TO /aws1/cl_xl8textxl8tionjobfilt.

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

TRY.
  oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned for
testing purposes."
    io_filter      = lo_filter ).
  MESSAGE 'Jobs retrieved.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex.
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8invalidfilterex.
    MESSAGE 'The filter specified for the operation is not valid. Specify a
different filter.' TYPE 'E'.

```

```

CATCH /aws1/cx_xl8invalidrequestex.
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.

```

- For API details, see [ListTextTranslationJobs](#) in *AWS SDK for SAP ABAP API reference*.

StartTextTranslationJob

The following code example shows how to use StartTextTranslationJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
    oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
        io_inputdataconfig = lo_inputdataconfig
        io_outputdataconfig = lo_outputdataconfig
        it_targetlanguagecodes = lt_targetlanguagecodes
        iv_dataaccessrolelearn = iv_dataaccessrolelearn
        iv_jobname = iv_jobname
        iv_sourcelanguagecode = iv_sourcelanguagecode ).
    MESSAGE 'Translation job started.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex.
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
CATCH /aws1/cx_xl8invparamvalueex.
    MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex.
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00.
    MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
ENDTRY.
```

- For API details, see [StartTextTranslationJob](#) in *AWS SDK for SAP ABAP API reference*.

StopTextTranslationJob

The following code example shows how to use StopTextTranslationJob.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

```

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

```

- For API details, see [StopTextTranslationJob](#) in *AWS SDK for SAP ABAP API reference*.

TranslateText

The following code example shows how to use TranslateText.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

```

"Translates input text from the source language to the target language."
TRY.
    oo_result = lo_xl8->translatetext(      "oo_result is returned for testing
purposes."
        iv_text          = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode ).
    MESSAGE 'Translation completed.' TYPE 'I'.
CATCH /aws1/cx_xl8detectedlanguage00.

```

```
MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
CATCH /aws1/cx_xl8internalserverex.
MESSAGE 'An internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex.
MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8serviceunavailex.
MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
CATCH /aws1/cx_xl8textsizefmtexcdex.
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.
```

- For API details, see [TranslateText](#) in *AWS SDK for SAP ABAP API reference*.

Scenarios

Get started with translate jobs

The following code example shows how to:

- Start an asynchronous batch translation job.
- Wait for the asynchronous job to complete.
- Describe the asynchronous job.

SDK for SAP ABAP

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Code Examples Repository](#).

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

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

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

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

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

```

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

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

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

```

- For API details, see the following topics in *AWS SDK for SAP ABAP API reference*.
 - [DescribeTextTranslationJob](#)
 - [StartTextTranslationJob](#)

Using the AWS SDK for SAP ABAP Knowledge MCP Server

The AWS SDK for SAP ABAP Knowledge MCP Server is a specialized knowledge resource that enables AI coding assistants to generate accurate ABAP code for AWS integrations. By connecting your AI-enabled IDE to this server, you can describe what you want to build in natural language and receive syntactically correct ABAP code that properly invokes the AWS SDK for SAP ABAP.

The MCP server uses the [Model Context Protocol \(MCP\)](#), an open standard for connecting AI assistants to external knowledge sources. The server is updated daily in sync with AWS SDK for SAP ABAP releases, so your IDE always has access to current, accurate SDK information.

The AWS SDK for SAP ABAP Knowledge MCP Server is available at no additional cost. You only pay for the AWS resources and services that you consume in your SAP applications with the AWS SDK for SAP ABAP.

Topics

- [Benefits](#)
- [Setting up the MCP server](#)
- [What the MCP server can do](#)
- [Important considerations](#)

Benefits

The AWS SDK for SAP ABAP Knowledge MCP Server provides the following benefits for ABAP developers.

- **Accurate code generation** – Your AI coding assistant gains deep knowledge of AWS SDK for SAP ABAP-specific patterns, method signatures, data types, and exception handling. This eliminates common errors and significantly reduces compilation errors and debugging time.
- **Instant SDK discovery** – The MCP server exposes the complete AWS SDK for SAP ABAP knowledge base, covering 200+ AWS services. You can discover available services, operations, and data types without manually searching documentation.
- **Always up to date** – The knowledge base is updated daily in sync with SDK releases. New services and updated method signatures are immediately available to your IDE.
- **Zero installation** – Setup requires only pasting a URL into your IDE's MCP configuration. No local software installation, no dependency management, and no IT approval process is required.

Setting up the MCP server

To connect your IDE to the AWS SDK for SAP ABAP Knowledge MCP Server, add the following configuration to your IDE's MCP settings. The configuration format and file location may vary by IDE; refer to your IDE's MCP documentation for details.

The following example shows the configuration for Kiro. Add this to your workspace MCP config file `.kiro/settings/mcp.json` or your user-level config file `~/.kiro/settings/mcp.json`:

```
{
  "mcpServers": {
    "abap-sdk-knowledge": {
      "url": "https://sdk-for-sap-abap-knowledge-mcp.global.api.aws/mcp",
      "disabled": false,
      "autoApprove": ["*"]
    }
  }
}
```

The `autoApprove` setting allows your IDE to use the MCP server's tools without prompting for approval on each request. This is safe because the MCP server is read-only and does not modify any data or execute code.

After saving the configuration, your IDE will connect to the MCP server automatically. No restart is required in most IDEs. You can verify the connection by asking your AI assistant a question about the AWS SDK for SAP ABAP, such as *"What AWS services are available in the AWS SDK for SAP ABAP?"*

If your IDE does not connect to the MCP server, verify that your IDE supports the Model Context Protocol and that the configuration file syntax matches your IDE's requirements.

Note

The MCP server endpoint uses HTTPS and does not require authentication. Your IDE communicates with the server only when you ask your AI assistant questions related to AWS SDK for SAP ABAP usage. Your proprietary business logic and SAP application code remain within your development environment.

What the MCP server can do

Once connected, your AI coding assistant can use the following capabilities provided by the MCP server.

- **Service discovery** – List and search all AWS services available in the AWS SDK for SAP ABAP, including service metadata and factory class names.
- **Operation details** – Retrieve ABAP method signatures, input parameters, return types, and exception types for any SDK operation.
- **Data type definitions** – Look up ABAP structure definitions for SDK data types, including field names and types.
- **Code examples** – Retrieve working ABAP code examples for specific operations, covering common use cases for each service.
- **Session and client creation** – Generate boilerplate ABAP code for creating SDK sessions, initializing service clients, and calling operations with correct exception handling.

For example, you can prompt your AI assistant with requests such as the following.

- *"Write ABAP code to read a message from an SQS queue."*
- *"Generate ABAP code to invoke an Amazon Bedrock model for text generation."*
- *"Show me how to upload a file to Amazon S3 from ABAP."*

The AI assistant uses the MCP server's knowledge to generate syntactically correct ABAP code that follows AWS SDK for SAP ABAP conventions, including proper session creation, client initialization, and exception handling patterns.

Important considerations

Keep the following in mind when using the AWS SDK for SAP ABAP Knowledge MCP Server.

- The MCP server provides read-only access to SDK knowledge. It does not execute ABAP code or connect to SAP systems.
- The MCP server covers only the AWS SDK for SAP ABAP. It does not provide general ABAP programming assistance unrelated to AWS SDK for SAP ABAP usage.
- The MCP server does not support custom or third-party ABAP libraries.

- The MCP server delivers documentation based on the latest version of the SDK, which may not match the version installed in your SAP system.
- Generated code examples are starting points. Review and test all generated code before using it in production.
- The quality and format of generated code may vary depending on your IDE's AI assistant capabilities.

Separately from the AWS SDK for SAP ABAP Knowledge MCP Server described on this page, AWS also offers the [AWS Knowledge MCP Server](#), which provides broad AWS service information, architectural guidance, and troubleshooting across all AWS services. The two servers complement each other and are especially effective when used together.

Security in AWS SDK for SAP ABAP

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to AWS SDK for SAP ABAP, see [AWS services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This section covers the following topics.

Topics

- [SAP system authentication on AWS](#)
- [Best practices for IAM Security](#)
- [SAP authorizations](#)
- [Secure operations](#)
- [Using Secret Access Key Authentication with SSF Encryption](#)
- [Using certificates with IAM Roles Anywhere](#)
- [Using Source Profile for Cross-Account Access](#)
- [Using SAP Credential Store](#)

SAP system authentication on AWS

Before an SAP system can make calls to AWS on behalf of SAP users, the SAP system must authenticate itself to AWS. AWS SDK for SAP ABAP supports the following three base authentication methods that are selected in the SDK profile settings in IMG.

AWS SDK for SAP ABAP - BTP edition can only be authenticated with the [the section called "Secret access key authentication"](#) method using SAP Credential Store.

For cross-account access scenarios, SDK for SAP ABAP also supports source profile, which enables chaining multiple IAM role assumptions across accounts using any of the base authentication methods. For more information, see [the section called "Source profile authentication for cross-account access"](#).

Topics

- [Amazon EC2 instance metadata authentication](#)
- [Secret access key authentication](#)
- [Certificate-based authentication using IAM Roles Anywhere](#)
- [Source profile authentication for cross-account access](#)
- [Next step](#)

Amazon EC2 instance metadata authentication

SAP systems running on Amazon EC2 can acquire short-lived, automatically-rotating credentials from Amazon EC2 instance metadata. For more information, see [Using credentials for Amazon EC2 instance metadata](#).

We strongly recommend this method of authentication while using SDK for SAP ABAP. To enable, the Basis administrator must enable outbound HTTP communication. No further Basis configuration is required.

Note

This method of authentication applies only if your SAP systems are running on Amazon EC2. SAP systems hosted on-premises or in other cloud environments cannot authenticate using this method.

Secret access key authentication

With this method, you use an Access Key ID and a Secret Access Key to authenticate your SAP system on AWS. The SAP system logs into AWS using an IAM user. For more information, see [Managing Access Keys for IAM Users](#).

The Basis administrator receives an Access Key ID and a Secret Access Key from the AWS IAM administrator. Your SAP system must be configured to store the Access Key ID and Secret Access Key.

- **Secure, store, and forward (SSF)**
 - Use the SSF functionality to authenticate AWS SDK for SAP ABAP. For more information, see [Digital Signatures and Encryption](#).
 - You can also test SSF's envelope and develop functionality with the SSF02 report. For more information, see [Testing the SSF Installation](#).
 - The steps for configuring SSF for SDK for SAP ABAP are described in the /AWS1/IMG transaction. Go to **Technical Prerequisites**, and then select **Additional Settings** for On-Premises Systems. For detailed configuration steps, see [Using Secret Access Key Authentication with SSF Encryption](#).
- **SAP Credential Store**
 - Use SAP Credential Store to authenticate AWS SDK for SAP ABAP - BTP edition. For more information, see [What Is SAP Credential Store?](#)
 - See [Using SAP Credential Store](#) for configuration steps.

Certificate-based authentication using IAM Roles Anywhere

An X.509 certificate issued by your certificate authority (CA) can be used for authentication with AWS Identity and Access Management Roles Anywhere. The certificate must be configured in STRUST. The CA must be registered with IAM Roles Anywhere as a trust anchor, and a profile must be created to specify the roles and policies that IAM Roles Anywhere would assume. For more information, see [Creating a trust anchor and profile in AWS Identity and Access Management Roles Anywhere](#).

For detailed steps on how to use IAM Roles Anywhere with SDK for SAP ABAP, see [Using certificates with IAM Roles Anywhere](#).

Note

Certificate revocation is only supported through the use of imported certificate revocation lists. For more information, see [Revocation](#).

Source profile authentication for cross-account access

Source profile is an advanced feature that enables you to chain multiple IAM role assumptions across AWS accounts. With this method, one profile assumes a role, which then assumes another role, and so on, similar to the `source_profile` parameter in AWS CLI.

Source profile works with any of the three base authentication methods (instance metadata, secret access key, or certificate-based). The first profile in the chain must use one of these base methods, and subsequent profiles in the chain use the credentials from the previous profile to assume the next role.

This is useful for cross-account access scenarios where you need to traverse multiple AWS accounts to reach your target resources. For detailed configuration steps, see [Using Source Profile for Cross-Account Access](#).

Next step

After authenticating your SAP system in AWS, SDK for SAP ABAP automatically performs an `sts:assumeRole` to assume the appropriate IAM role for the SAP user's business function.

Best practices for IAM Security

The IAM administrator will be responsible for the following three key areas.

- Ensuring that the SAP system can authenticate itself with Amazon EC2 metadata or Secret Key credentials.
- Ensuring that the SAP system has the permissions it needs to elevate itself with `sts:assumeRole`.
- For each logical IAM role, creating an IAM role for SAP users with the permissions required to perform the business functions (for example, the necessary permissions for Amazon S3, DynamoDB, or other services). These are the roles that SAP users will assume.

For more information, see the [Security](#) chapter in the SAP Lens: AWS Well-Architected Framework.

Topics

- [Best practice for Amazon EC2 instance profile](#)
- [IAM roles for SAP users](#)
- [Source Profile Security Considerations](#)

Best practice for Amazon EC2 instance profile

The Amazon EC2 instance on which your SAP system runs has a set of authorizations based on its instance profile. Generally, the instance profile only needs to have permissions to call `sts:assumeRole`, to allow the SAP system to assume business-specific IAM roles as needed. This elevation to other roles ensures that an ABAP program can assume a role that gives the user the least privilege needed to do their job. For example, an instance profile might contain the following statement.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::012345678912:role/finance-cfo",
        "arn:aws:iam::012345678912:role/finance-auditor",
        "arn:aws:iam::012345678912:role/finance-reporting"
      ]
    }
  ]
}
```

This preceding example allows the SAP system to assume the IAM roles for the CFO, AUDITOR, or REPORTING user. AWS SDK will choose the correct IAM role for the user based on the user's PFCG role in SAP.

Amazon EC2 instance profile can also be used for other functions.

- [AWS Backint Agent for SAP HANA](#)
- [SAP on AWS High Availability with Overlay IP Address Routing](#)

These solutions may also require `sts:assumeRole` permissions to roles specific to backup or failover or they may require permissions to be assigned directly to the instance profile.

IAM roles for SAP users

The ABAP program needs permissions to perform the user's job: read a DynamoDB table, invoke Amazon Textract on a PDF object in Amazon S3, run an AWS Lambda function. The same security model is used in all AWS SDKs. You can use an existing IAM role that was used for another AWS SDK.

The SAP business analyst will ask the IAM administrator for the `arn:aws:` of an IAM role for each logical role needed. For example, in a financial scenario, the business analyst may define the following logical IAM roles.

- CFO
- AUDITOR
- REPORTING

The IAM administrator will define IAM roles for each logical IAM role.

CFO

- `arn:aws:iam::0123456789:role/finance-cfo`
- read and write permissions to an Amazon S3 bucket
- read and write permissions to a DynamoDB database

AUDITOR

- `arn:aws:iam::0123456789:role/finance-auditor`

- read permissions to an Amazon S3 bucket
- read permissions to a DynamoDB database

REPORTING

- `arn:aws:iam::0123456789:role/finance-reporting`
- read permissions to a DynamoDB database
- no permission for the Amazon S3 bucket

The business analyst will enter the IAM roles into a mapping table to map the logical IAM roles with the physical IAM roles.

IAM roles for SAP users need to allow the `sts:assumeRole` action for trusted principals. The trusted principals can vary based on how the SAP system is authenticated on AWS. For more details, see [Specifying a principal](#).

The following are some examples of the most common SAP scenarios.

- **SAP system running on Amazon EC2 with an instance profile assigned** – here, an Amazon EC2 instance profile is attached to an IAM role.

JSON

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

- **SAP systems running on Amazon EC2 without an instance profile** – here, Amazon EC2 assumes roles for SAP users.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "Service": [ "ec2.amazonaws.com" ]
      }
    }
  ]
}
```

- **SAP systems running on-premises** – SAP systems that run on-premises can only authenticate using the Secret Access Key. For more information, see [SAP system authentication on AWS](#).

Here, any IAM role assumed by an SAP user must have a trust relationship that trusts the SAP user.

JSON

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

Source Profile Security Considerations

When using source profile:

IAM Role Management

Critical: IAM roles in source profile chains must be strictly managed to prevent unauthorized access and privilege escalation:

- **Apply least privilege principle** - Grant only the minimum permissions required for each role's specific purpose
- **Regularly audit role permissions** - Review and update role policies quarterly or when requirements change
- **Monitor role usage** - Use to track AssumeRole API calls and identify unusual patterns
- **Limit trust relationships** - Restrict which principals can assume each role to only those that absolutely need access
- **Use conditions in trust policies** - Add conditions like source IP, MFA requirements, or time-based restrictions where appropriate
- **Document role purposes** - Maintain clear documentation of each role's intended use case and required permissions

Authorization and Access Control

- Ensure all intermediate profiles in the chain have appropriate trust policies configured
- Users must have `/AWS1/SESS` authorization for ALL profiles in the chain, including intermediate profiles
- Each IAM role must explicitly trust the previous role in the chain

Technical Safeguards

- The SDK enforces a maximum chain depth of 5 profiles to prevent excessive STS API calls
- Circular references are automatically detected and prevented
- Base profile authentication method is validated to ensure it uses a standard method (INST, SSF, or RLA)

For more information about configuring source profile, see [Using Source Profile for Cross-Account Access](#).

SAP authorizations

The authorization required to configure the SDK is dependent on the SDK edition.

Topics

- [Authorizations for configuration](#)
- [SAP authorizations for end users](#)

Authorizations for configuration

See the following tabs for more details.

SDK for SAP ABAP

The following authorizations are required to configure SDK for SAP ABAP.

- S_TCODE
 - TCD = /AWS1/IMG
- S_TABU_DIS
 - ACTVT = 02, 03
- DICBERCLS

Choose from the following authorization groups.

- /AWS1/CFG - AWS SDK for SAP ABAP v1 - Config
- /AWS1/MOD - AWS SDK for SAP ABAP v1 - Runtime
- /AWS1/PFL - AWS SDK for SAP ABAP v1 - SDK Profile
- /AWS1/RES - AWS SDK for SAP ABAP v1 - Logical Resources
- /AWS1/TRC - AWS SDK for SAP ABAP v1 - Trace

SDK for SAP ABAP - BTP edition

Use the following steps to allow SDK for SAP ABAP - BTP edition access to the configuration.

1. Create a new business role using the `SAP_BR_BPC_EXPERT` business role template. This template provides access to the Cutsom Business Configuration application.
2. Under **General Role Details**, go to **Access Categories**, and choose **Unrestricted** for *Read, Write, Value Help*.
3. Go to the **Business Catalog** tab, and assign the `/AWS1/RTBTP_BCAT` business catalog to provide access to the SDK configuration.
4. Go to the **Business Users** tab, and assign business users to grant access to the SDK configuration.

SAP authorizations for end users

Prerequisite: Define SDK Profiles

Before the SAP security administrator can define their roles, the Business Analyst will define SDK profiles in transaction `/AWS1/IMG` for AWS SDK for SAP ABAP or the Custom Business Configuration application for SDK for SAP ABAP - BTP edition. Typically, an SDK profile will be named according to its business function: ZFINANCE, ZBILLING, ZMFG, ZPAYROLL, etc. For each SDK profile, the Business Analyst will define logical IAM roles with short names, such as CFO, AUDITOR, REPORTING. These will be mapped to the real IAM roles by the IAM security administrator.

Define PFCG or Business Roles

Note

PFCG roles are called Business Roles in SAP BTP, ABAP environment.

The SAP security administrator will then add authorization object `/AWS1/SESS` to grant access to an SDK profile.

Auth Object `/AWS1/SESS`

- Field `/AWS1/PROF` = ZFINANCE

Users should also be mapped to logical IAM roles for each SDK profile, depending on their job function. For example, a financial auditor with reporting access might be authorized for a logical IAM role called AUDITOR.

Auth Object /AWS1/LR0L

- Field /AWS1/PROF = ZFINANCE
- Field /AWS1/LR0L = AUDITOR

Meanwhile, the CFO, with read/write authorizations, might have a PFCG role authorizing them the logical role of CFO.

Auth Object /AWS1/LR0L

- Field /AWS1/PROF = ZFINANCE
- Field /AWS1/LR0L = CFO

In general, a user should be authorized for only one logical IAM role per SDK profile. If a user is authorized for more than one IAM role (for example, if the CFO is authorized for both CFO and AUDITOR logical IAM roles), then AWS SDK breaks the tie by ensuring that the higher priority (lower sequence number) role takes effect.

As with all security scenarios, users should be given least privilege to perform their job functions. A simple strategy for managing PFCG roles would be to name Single PFCG roles according to the SDK profile and logical role they authorize. For example, role Z_AWS_PFL_ZFINANCE_CFO grants access to profile ZFINANCE and logical IAM role CFO. These single roles can then be assigned to composite roles that define job functions. Each company has their own strategy for role management, and we encourage you to define a PFCG strategy that works for you.

Secure operations

Encryption Of Data At Rest

AWS Secret Access Keys are used for authenticating the SDK. They are encrypted using the SSF or Credential Store functionality by SAP.

Encryption Of Data In Transit

All calls to AWS services are encrypted with HTTPS. The SAP ICM manages the HTTPS connection. AWS certificates must be trusted in STRUST.

API Usage

When an ABAP user assumes a role using `sts:assumeRole`, the session name is titled `USERID-SID-MANDT`, where:

- `USERID` is the ABAP user from `SY-UNAME` variable.
- `SID` is the ABAP system ID from `SY-SYSID` variable.
- `MANDT` is the ABAP client from `SY-MANDT` variable.

The session name appears in CloudTrail as *user name*. This ensures that API calls from an ABAP user can be traced back to the system, client, and user that initiated the call. For more information, see [What is AWS CloudTrail?](#)

Using Secret Access Key Authentication with SSF Encryption

On-premises SAP systems (or systems running in other clouds) can be authenticated on AWS by using secret access key authentication with AWS Identity and Access Management. SAP's [Secure Store and Forward Mechanism](#) (SSF) is used to encrypt and securely store AWS credentials (Access Key ID and a Secret Access Key) of an IAM user. The SAP system logs into AWS using an IAM user, see [Managing Access Keys for IAM Users](#) for information.

Prerequisites

The following prerequisites must be met before commencing the configuration:

- IAM roles for SAP users must be created by the IAM administrator. The roles must have permissions to call the required AWS services. For more information, see [Best practices for IAM Security](#).
- Create authorization to run `/AWS1/IMG` transaction. For more information, see [Authorizations for configuration](#).

Procedure

Follow along these instructions to configure SSF-encrypted credential storage:

Steps

- [Step 1 – Define an SSF application for Credential Storage](#)

- [Step 2 – Set the encryption parameters for the SSF application](#)
- [Step 3 – Create PSE for SSF Application](#)
- [Step 4 – Assign an SSF application to the AWS SDK for SAP ABAP](#)
- [Step 5 – Configure SDK profile to use SSF-encrypted credentials](#)

Step 1 – Define an SSF application for Credential Storage

1. Execute transaction code SE16 to define an SSF application.
2. Enter SSFAPPLIC table name, and select **New Entries**.
3. Enter following details:
 - **APPLIC**: ZAWS1 (name for the SSF application).
 - **DESCRIPT**: SSF Encryption for the AWS SDK for SAP ABAP (description).
 - Choose Selected(X) option for the remaining fields.
4. Select Save.

Step 2 – Set the encryption parameters for the SSF application

1. Execute the transaction code /n/AWS1/IMG to launch the Implementation Guide (IMG) for AWS SDK for SAP ABAP.
2. Expand the IMG node **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises systems**.
3. Execute the **Set SSF Parameters** IMG activity.
4. Select **New Entries**, and choose the SSF application created in the previous step. Select **Save**.
5. Modify the hash algorithm to **SHA256** (or higher), and the encryption algorithm to **AES256-CBC**. Retain the other settings as default, and select **Save**.

These encryption settings will be used to securely encrypt AWS credentials.

Step 3 – Create PSE for SSF Application

1. Execute the /n/AWS1/IMG transaction, and select **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises systems**.
2. Execute the IMG activity **Create PSE for SSF Application**, which will direct you to the STRUST transaction. Select **Edit**.

3. Right-select the SSF application created in [the section called "Step 1"](#), and choose **Create**. Retain all other default settings, and select **Continue**. Ensure you choose RSA and not DSA as the algorithm.

Step 4 – Assign an SSF application to the AWS SDK for SAP ABAP

1. Execute the /n/AWS1/IMG transaction, and select **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises systems**.
2. Execute the IMG activity **Assign an SSF application to the AWS SDK for SAP ABAP**.
3. Select **New Entries** and enter the SSF application created in [the section called "Step 1"](#). Select **Save**.

Step 5 – Configure SDK profile to use SSF-encrypted credentials

1. Execute the /n/AWS1/IMG transaction, and select **AWS SDK for SAP ABAP Settings > Application Configurations**.
2. Execute the IMG activity **SDK Profile**.
3. Select **New Entries**. Enter profile name and description. Select **Save**.
4. Highlight the entry that you created and click on the **Authentication And Settings tree** branch.
5. Select **New Entries** and enter following details:
 - **SID**: The system ID of the SAP system.
 - **Client**: The client of the SAP system.
 - **Scenario ID**: Select the DEFAULT scenario created by your Basis administrator.
 - **AWS Region**: AWS Region that you want to make calls to.
 - **Authentication Method**: Select **Credentials from SSF Storage** from the dropdown and select **Save**. Select **Set Credentials** and enter the Access Key ID and Secret Access Key of the IAM user.
 - **Disable IAM roles**: Keep this as default i.e. unchecked.
 - Select **Save**.
6. Click on the IAM Role Mapping tree branch. Select **New Entries**. Enter sequence number, name for logical IAM role and IAM Role ARN provided by the AWS IAM Administrator. Select **Save**.

For more information, see [Application configuration](#).

Using certificates with IAM Roles Anywhere

SAP system can be authenticated on AWS by using certificated-based authentication with AWS Identity and Access Management Roles Anywhere. You must setup the certificate in STRUST, and configure the SDK profile in /AWS1/IMG.

Prerequisites

The following prerequisites must be met before commencing setup for certification.

- The X.509 certificate issued by your certificate authority (CA) must meet the following requirements.
 - The signing certificate must be a v3 certificate.
 - The chain must not exceed 5 certificates.
 - The certificate must support RSA or ECDSA algorithms.
- Register your CA with IAM Roles Anywhere as a trust anchor, and create a profile to specify the roles/policies for IAM Roles Anywhere. For more information, see [Creating a trust anchor and profile in AWS Identity and Access Management Roles Anywhere](#).
- IAM roles for SAP users must be created by the IAM administrator. The roles must have permissions to call the required AWS services. For more information, see [Best practices for IAM Security](#).
- Create authorization to run /AWS1/IMG transaction. For more information, see [Authorizations for configuration](#).

Procedure

Follow along these instructions to setup certificate-based authentication.

Steps

- [Step 1 – Define an SSF application by using SAP's Secure Store and Forward \(SSF\)](#)
- [Step 2 – Set SSF parameters](#)
- [Step 3 – Create the PSE and certificate request](#)
- [Step 4 – Import certificate response into the relevant PSE](#)
- [Step 5 – Configuring SDK profile to use IAM Roles Anywhere](#)

Step 1 – Define an SSF application by using SAP's Secure Store and Forward (SSF)

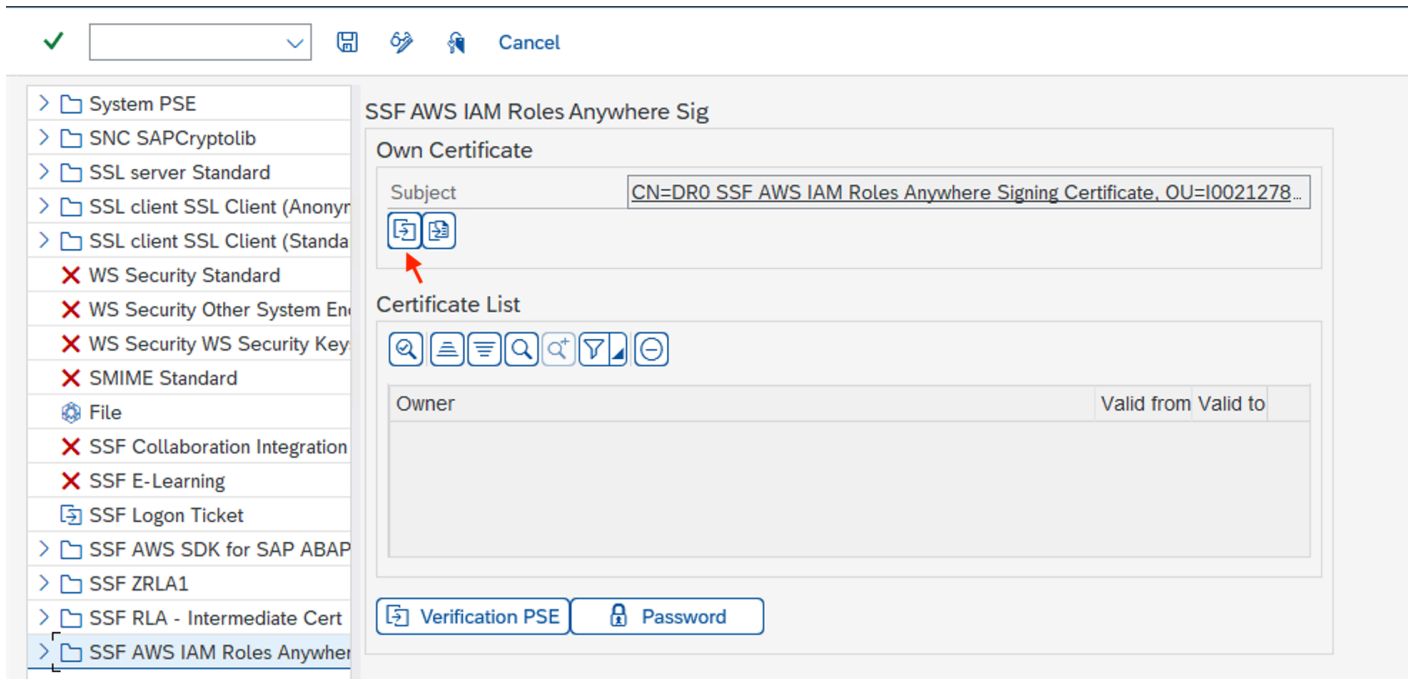
1. Run transaction code SE16 to define an SSF application.
2. Enter SSFAPPLIC table name, and select **New Entries**.
3. Enter a name for the SSF application in the APPLIC filed, a description in the DESCRIPT filed, and select Selected (X) option for the remaining fields.

Step 2 – Set SSF parameters

1. Run the /n/AWS1/IMG to launch AWS SDK for SAP ABAP Implementation Guide (IMG).
2. Select **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises Systems**.
3. Run the **Set SSF Parameters** IMG activity.
4. Select **New Entries**, and choose the SSF application created in the previous step. Select **Save**.
5. Modify the hash algorithm to **SHA256**, and the encryption algorithm to **AES256-CBC**. Retain the other settings as default, and select **Save**.

Step 3 – Create the PSE and certificate request

1. Run the /n/AWS1/IMG transaction, and select **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises systems**.
2. Run the **Create PSE for SSF Application** IMG activity.
3. Select **Edit** for the STRUST transaction.
4. Right-select the SSF application created in [the section called "Step 1"](#), and choose **Create**. Retain all other default settings, and select **Continue**.
5. Select **Create Certificate Request**. See the following image. Retain the default options, and select **Continue**. Copy or export the generated certificate request, and provide it to your CA. Your CA verifies the request, and responds with a signed public-key certificate.



The signing process varies based on your CA, and the technology used by them. See [Issuing private end-entity certificates](#) with AWS Private Certificate Authority for an example.

Step 4 – Import certificate response into the relevant PSE

1. Run the `/n/AWS1/IMG` transaction, and select **AWS SDK for SAP ABAP Settings > Technical Prerequisites > Additional Settings for On-Premises systems**.
2. Run the `Create PSE for SSF Application IMG` activity.
3. Select **Edit** for the STRUST transaction.
4. Choose the SSF application, and then select **Import Certificate Response** located in the PSE section below the subject. Either copy and paste the certificate response into text box or import the file from the file system. Select **Continue > Save**.
5. The certificate details can be viewed by selecting the subject twice. The information is displayed in the certificate section.

Step 5 – Configuring SDK profile to use IAM Roles Anywhere

1. Run the `/n/AWS1/IMG` transaction, and select **AWS SDK for SAP ABAP Settings > Application Configurations**.
2. Create a new SDK profile, and name it.

3. Choose IAM Roles Anywhere as the authentication method.

- In the left pane, select **Authentication and Settings**.
- Create a new entry, and enter the information for your SAP system, and AWS Region.
- Select **IAM Roles Anywhere** for the authentication method, and select **Save**.
- Select **Enter Details**, and in the pop-up window, choose the SSF application created in [the section called “Step 1”](#). Enter the **Trust Anchor ARN**, and **Profile ARN** that were created in [the section called “Prerequisites”](#). See the following image. Select **Continue**.

The screenshot shows a configuration window titled "SAP" with a close button (X) in the top right corner. The window contains the following fields and text:

- Header: "Select Signing Certificate issued by your certificate authority (CA) from SSF"
- Field: "Certificate (SSF Application)" with the value "ZSIGN1" and a copy icon.
- Section: "Enter your IAM Roles Anywhere details"
- Field: "Trust Anchor ARN" with the value "arn:aws:rolesanywhere:us-east-1:111111111111:trust-anchor/447a78db-299a-46cb-b466-43b71513ebda"
- Field: "Profile ARN" with the value "arn:aws:rolesanywhere:us-east-1:111111111111:profile/b8601d15-3ee8-405a-b5fd-1e7363e77943"
- Bottom right corner: A green checkmark and a red X icon.

4. In the left pane, select **IAM Role Mapping**. Enter a name, and provide the IAM role's ARN provided by your IAM administrator.

For more information, see [Application configuration](#).

Using Source Profile for Cross-Account Access

Source profile enables SAP systems to access AWS resources across multiple accounts by chaining IAM role assumptions. One profile assumes a role, which then assumes another role, and so on, similar to the `source_profile` parameter in AWS CLI. This is useful for cross-account access scenarios where you need to traverse multiple AWS accounts to reach your target resources.

Example: Your SAP system runs in Account A (111111111111) and needs to access Amazon S3 buckets in Account C (333333333333). You configure three profiles:

1. DEV_BASE gets base credentials from Amazon EC2 instance metadata and assumes Role P in Account A
2. SHARED_SERVICES uses DEV_BASE credentials to assume Role Q in Account B (222222222222)
3. PROD_S3_ACCESS uses SHARED_SERVICES credentials to assume Role R in Account C

When your application uses PROD_S3_ACCESS, the SDK automatically executes the chain: get credentials from instance metadata → assume Role P → assume Role Q → assume Role R.

Prerequisites

The following prerequisites must be met before configuring source profile:

- IAM roles for each step in the chain must be created by the IAM administrator. Each role must have:
 - Permissions to call the required AWS services
 - Trust relationship configured to allow the previous role in the chain to assume it

For more information, see [Best practices for IAM Security](#).

- Create authorization to run /AWS1/IMG transaction. For more information, see [Authorizations for configuration](#).
- Users must have /AWS1/SESS authorization for ALL profiles in the chain, including intermediate profiles.

Procedure

Follow these instructions to configure source profile.

Steps

- [Step 1 – Configure the base profile](#)
- [Step 2 – Configure chained profiles](#)

Step 1 – Configure the base profile

The base profile is the first profile in the chain and must use a standard authentication method.

1. Run the `/n/AWS1/IMG` transaction to launch AWS SDK for SAP ABAP Implementation Guide (IMG).
2. Select **AWS SDK for SAP ABAP Settings > Application Configurations > SDK Profile**.
3. Create a new profile to use as your base profile by selecting **New Entries** and enter profile name and description. Select **Save**.

 **Note**

If you are using an existing profile that is already configured with a standard authentication method (INST, SSF, or RLA), you can skip the remaining steps in this section and proceed directly to [the section called “Step 2”](#).

4. Select the profile you created, then select **Authentication and Settings > New Entries**, and enter the following details:
 - **SID**: The system ID of the SAP system
 - **Client**: The client of the SAP system
 - **Scenario ID**: Select the DEFAULT scenario created by your Basis administrator
 - **AWS Region**: AWS Region that you want to make calls to
 - **Authentication Method**: Select one of the following:
 - **Instance Role via Metadata** for SAP systems running on Amazon EC2
 - **Credentials from SSF Storage** for on-premises or other cloud systems
 - **IAM Roles Anywhere** for certificate-based authentication

Select **Save**.

5. Select **IAM Role Mapping > New Entries**, and enter:
 - **Sequence number**: 1
 - **Logical IAM Role**: A descriptive name (e.g., DEV_BASE_ROLE)
 - **IAM Role ARN**: The ARN of the IAM role in the first account (e.g., `arn:aws:iam::111111111111:role/DevBaseRole`)

Select **Save**.

Step 2 – Configure chained profiles

Configure each intermediate and final profile in the chain.

For SHARED_SERVICES profile (chains from DEV_BASE):

1. Run the `/n/AWS1/IMG` transaction.
2. Select **AWS SDK for SAP ABAP Settings > Application Configurations > SDK Profile**.
3. Select **New Entries**. Enter profile name (e.g., SHARED_SERVICES) and description. Select **Save**.
4. Select the profile you created, then select **Authentication and Settings > New Entries** and enter the following details:
 - **SID**: The system ID of the SAP system
 - **Client**: The client of the SAP system
 - **Scenario ID**: Select the DEFAULT scenario created by your Basis administrator
 - **AWS Region**: AWS Region that you want to make calls to
 - **Authentication Method**: Select **Source Profile** from the dropdown
 - **Source Profile ID**: Enter the profile ID of the base profile (e.g., DEV_BASE)

Select **Save**.

5. Select **IAM Role Mapping > New Entries**, and enter:
 - **Sequence number**: 1
 - **Logical IAM Role**: A descriptive name (e.g., SHARED_ROLE)
 - **IAM Role ARN**: `arn:aws:iam::222222222222:role/SharedServicesRole`

Select **Save**.

For PROD_S3_ACCESS profile (chains from SHARED_SERVICES):

Repeat the same steps as SHARED_SERVICES, but:

- Use PROD_S3_ACCESS as the name
- Set **Source Profile ID** to SHARED_SERVICES
- Use PROD_S3_ROLE and `arn:aws:iam::333333333333:role/ProdS3AccessRole` in IAM Role Mapping

For security best practices including IAM role management, trust policy configuration, and authorization requirements, see [Best practices for IAM Security](#).

Using SAP Credential Store

SAP Credential Store is used in SAP Business Technology Platform to securely store credentials for secret access key authentication to AWS. You must have a subscription to use the service.

The following instructions assume that you have already configured an SDK profile. For more information, see [Configuring AWS SDK for SAP ABAP](#).

Before commencing the configuration, ensure that you meet the prerequisites. For more information, see [SAP Credential Store](#).

Topics

- [Configuration steps](#)
- [Using SAP Credential Store with the SDK](#)

Configuration steps

Steps

- [Step 1: Configure settings for authentication](#)
- [Step 2: Create a service key](#)
- [Step 3: Convert service key into .p12 format](#)
- [Step 4: Connect to SAP BTP, ABAP environment](#)

Step 1: Configure settings for authentication

Use the following steps to configure the Credential Store settings for authentication.

1. Navigate to the **Settings** tab of the SAP Credential Store instance.
2. Select **Edit Configurations**:
 - Choose **Mutual TLS** as the Default Authentication Type.
 - Select **Disabled** for Payload Encryption Status. The payload is encrypted in transit with HTTPS. However, the payload cannot currently be double-encrypted.
3. Select **Save**.

Step 2: Create a service key

Use the following steps to create a service key for Credential Store.

1. In the left pane of the SAP Credential Store application, navigate to **Service Keys**.
2. Select **Create Service Key**.
3. Enter a name for the service key, and select **Create**.

The service key is created on the basis of the chosen authentication type. Download the service key, and keep it secure for later usage.

Step 3: Convert service key into .p12 format

A client certificate in the .p12 format is required to create an outbound user for communication system. Use the following steps to generate a .p12 certificate from the certificate details provided in the Credential Store Service key.

1. Download the **SAP Cloud Root CA** certificate (required by SAP) from [SAP Trust Center Services](#).
2. Open the SAP Cloud Root CA certificate in any text file format. At the end of the file, press Enter, and copy-paste the certificate from the certificate field of the service key. Replace new line characters \n with actual new line (Enter), and save the entire certificate in .cer file format.
3. Copy the key from the key field of the service key. This private key must be treated as sensitive data. Paste it in a text file, and replace new line characters \n with actual new line (Enter). Save the private key in a text file.
4. With the certificate and private key generated in the previous steps, run the following command to generate a .p12 certificate.

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in  
<certificate.cer>
```

The command required verification of the export password. Retain the password for further use.

Delete the .key text file saved in your private key.

Step 4: Connect to SAP BTP, ABAP environment

Configure SAP BTP, ABAP environment to connect with SAP Credential Store.

Topics

- [Communication system](#)
- [Communication arrangement](#)

Communication system

Use the following steps to create a communication system that enables communication from SAP BTP, ABAP environment to SAP Credential Store.

1. Open the Fiori launchpad of the ABAP environment system.
2. Select the **Communication Systems** tile to open the application.
3. Select **New**.
4. Enter a name and ID for the communication system, and select **Create**. For example, you can name the system ZSAP_CREDSTORE.
5. Enter other required information:
 - **Host name:** Copy the host name from the Service Key URL. For example, if the URL is `https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`, then the host name is `credstore.mesh.cf.us10.hana.ondemand.com`.
 - **Users for Outbound Communication:** Select + to add a new user.
 - a. Select **SSL Client Certificate** as the Authentication mechanism.
 - b. Select **Upload New Certificate**:
 - Browse the `.p12` certificate generated in the preceding step.
 - Enter a description.
 - Enter the export password that was used to generate the `.p12` certificate.
 - Select **Upload**.
 - c. Select **Create** to create an outbound user.
6. Select **Save**.
7. Delete the service key downloaded in the previous step.

Communication arrangement

Use the following steps to create a communication arrangement to provide a communication scenario for outbound communication.

1. Open the Fiori launchpad of the ABAP environment system.
2. Select the **Communication Arrangements** tile to open the application.
3. Select **New**.
4. Select communication scenario /AWS1/CRED_COMM_SCENARIO0, and enter a name for the communication arrangement. For example, Z_AWS_SDK_TO_SAP_CREDSTORE.
5. Select **Create**.
6. In the Communication System field, browse for the the Communication System created in the previous step. Other information is auto-populated post selection of the system.
7. Select **Save**.
8. Select **Check Connection** to test your connection.

Once this setup is complete, the ABAP environment can use the communication arrangement to use the SAP Credential Store service via outbound service (HTTP).

Using SAP Credential Store with the SDK

Steps

- [Step 1: Create a namespace and credential\(s\)](#)
- [Step 2: Configure Custom Business Configuration application](#)

Step 1: Create a namespace and credential(s)

Create a namespace and credential in SAP Credential Store with SAP Help – [Create, Edit, and Delete a Credential](#).

Enter the following details to create a credential of type **Key**.

- **Namespace** – Enter a name for the namespace, and group related credentials together.
- **Name** – Enter a name for the key. We recommend `aws-0123456789012-username`, where:
 - `0123456789012` is the AWS account ID to which the credential grants access
 - `username` is the IAM user name to which the credential belongs
- **Value** – Enter a base-64 encoded secret access key. Use the following command to base-64 encode your secret access key.

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

```
MySecretAccessKey  
Ctrl-D
```

The command reads the secret access key from standard input, and passes it to base64 without a trailing newline. It outputs the base-64 encoded secret access key to the screen. Clear or close your terminal after copying the value into SAP Credential Store.

- **Username** – Enter your access key ID.
- Select **Create**.

A new namespace with one credential is created, and credentials can be added, deleted or modified within this namespace.

Follow the principle of least privilege to manage access to the credentials stored in the namespace.

Step 2: Configure Custom Business Configuration application

Use the following steps to configure Custom Business Configuration application to define the credential to use for authentication by the SDK.

1. Open the Fiori launchpad of the ABAP environment system.
2. Browse **Custom Business Configuration** tile to open the application.
3. Open **SDK Profile** Business Configuration.
4. Select the SDK profile for which authentication settings must be configured for SAP Credential Store.
5. In the **Authentication and Settings** tab for the selected profile, select **Edit**, and enter the following details:
 - **Authentication Method** – Select **Credentials from SAP Credential Store**.
 - **Namespace** – Enter the namespace created in SAP Credential Store. For more information, see [the section called “Step 1: Create a namespace and credential\(s\)”](#).
 - **Key Name** – Enter the name of the created key credential. For more information, see [the section called “Step 1: Create a namespace and credential\(s\)”](#).
 - **Communication Arrangement** – Enter the name of the created communication arrangement. For more information, see [the section called “Communication arrangement”](#).
6. Select **Apply** to go to the **AWS SDK Profile** screen.

7. Select **Select Transport** to select the transport using the value help.
8. Select **Save**.

Troubleshoot AWS SDK for SAP ABAP

This section provides troubleshooting steps for possible error scenarios.

Topics

- [Import failure](#)
- [Unspecified location constraint](#)
- [SSL errors](#)
- [Profile configuration](#)
- [IAM authorization](#)
- [Authorization for performing required actions](#)
- [Active scenario](#)
- [Special characters in code](#)
- [Connectivity](#)

Import failure

Problem – Class 'CL_SYSTEM_UUID' doesn't contain an interface 'IF_SYSTEM_UUID_RFC4122_STATIC'

Cause – SAP Note 0002619546 is missing on your system.

Resolution – Ensure that the [SAP Note 0002619546](#) is applied to your system.

Unspecified location constraint

Problem – The unspecified location constraint is incompatible for the region specific endpoint this request was sent to

Cause – Your Amazon S3 bucket is missing the AWS Region in `io_createbucketconfiguration` parameter.

Resolution – When creating a bucket in any Region, except `us-east-1`, specify your Amazon S3 bucket's Region using `io_createbucketconfiguration` parameter in `createbucket()`. You don't have to specify a constraint for `us-east-1`.

The following example shows a correctly configured `io_createbucketconfiguration` parameter.

```
createbucket(  
  iv_bucket = 'amzn-s3-demo-bucket'  
  io_createbucketconfiguration = NEW /aws1/c1_s3_createbucketconf( 'us-west-1' )  
).
```

SSL errors

Problem – SSL Server Certificate Hostname Mismatch or SSL handshake with docs.aws.amazon.com:443 failed: SSSLERR_NO_SSL_RESPONSE

Cause – `icm/HTTPS/client_sni_enabled` parameter is not set to TRUE in the DEFAULT profile.

Resolution – Use the following steps to troubleshoot the given problems or any other SSL-related problem.

1. Open the SAPGUI and go to the command bar.
2. Run transaction RZ10.
3. Go to **Profile** and choose DEFAULT profile. The version is populated automatically.
4. In the **Edit Profile** section, select **Extended maintenance**, and then select **Change**.
5. Search for the `icm/HTTPS/client_sni_enabled` parameter.
 - If the parameter exists, edit the **Parameter value** and set it to TRUE.
 - If the parameter doesn't exist, create a parameter using the following steps.
 1. Select **Parameter**.

Note

Ensure that you are selecting the Parameter for creation, and not editing (pencil icon).

2. Enter `icm/HTTPS/client_sni_enabled` in the **Parameter Name** field.
3. Enter TRUE in the **Parameter value** field.
4. Select **Save**.

6. Save these changes in the DEFAULT profile, and Exit.

Profile configuration

Problem – Could not find configuration under profile <profile_name> with scenario DEFAULT for <sid>:<client>

Causes – The <profile_name> is incorrect or hasn't been configured.

Resolution – Use the following steps to configure the profile.

1. Open SAPGUI and run transaction /n/AWS1/IMG.
2. Go to **Application Configuration > SDK Profile**.
 - If your profile is configured, verify that the profile name is correct.
 - If your profile is not configured, follow along the steps to configure a profile.
3. Select **New Entries**.
 - a. Enter a Name and Description for the profile.
 - b. Select **Save**.
4. Choose the entry you created in the previous step, and then select **Authentication and Settings**.
5. Select **New Entries**, enter the following details, and then select **Save**.
 - SID
 - Client
 - Scenario ID
 - AWS Region
 - Authentication Method
 - Select *Instance Role via Metadata* for SAP systems running in AWS.
 - Select *Credentials from SSF Storage* for SAP systems running on-premises or other cloud.
6. Select **IAM Role Mapping > New Entries**, enter the following details, and select **Save**.
 - Sequence number
 - Logical IAM Role
 - IAM Role ARN

IAM authorization

Problem – Could not assume role <iam_role_arn> or User: <user_arn> is not authorized to perform: sts:AssumeRole on resource:<iam_role_arn>

Causes – the following may be the possible reasons for this error.

- Incorrect IAM role ARN has been specified
- IAM user lacks permission to access the IAM role
- Lack of trust relationship between the assumed IAM role and the assuming IAM role or IAM user

Resolution – Use the following steps to ensure that the IAM role ARN is correct.

1. Open SAPGUI and run transaction /n/AWS1/IMG.
2. Go to **Application Configuration > SDK Profile**, and choose the profile that has been configured with your IAM role.
3. Select **IAM Role Mapping** and verify or correct your IAM role ARN.
 - If your IAM role ARN is correct, ensure that your IAM role has been configured properly. For more information, see [Troubleshooting IAM roles](#).

Authorization for performing required actions

Problem – User <user_arn> is not authorized to perform: <action> on resource: <resource_arn>

Cause – User does not have permissions to perform an action.

Resolution – user_arn must be set up with required permissions on resource_arn to perform a specified action. For more information, see [Permissions required to access IAM resources](#).

Active scenario


Problem – No active scenario configured

Cause – The setup of active scenario was missed.

Resolution – See [Runtime settings](#) to configure an active scenario.

Special characters in code

Warning – The character 0x00A0 cannot be part of an ABAP word

 **Note**

This warning may be preceded by varied error messages.

Cause – Copying and pasting code from different sources can insert special characters in your code.

Resolution – When you paste any code in the ABAP source code editor, you see the following pop-up.

Non-breaking space characters were detected. Convert to spaces?

Choose **Yes** to answer this question. Also, we recommend selecting the code to copy it, instead of using the copy button in code boxes.

Connectivity

Problem – SCLNT_HTTP(411) : Direct connect to tla.region.amazonaws.com:443 failed:
NIECONN_REFUSED(-10)

Cause – The SAP system does not have internet connectivity, and cannot establish a TCP/IP connection to port 443 of tla.region.amazonaws.com.

Resolution – The SAP system must be able to establish connection to AWS endpoints on HTTPS port 443, either directly or through a proxy server. You can establish/verify internet connectivity with one of the following options.

- Direct outbound connection to internet through a NAT or internet gateway
- Connection through a proxy server

For more information, see [Connection through a proxy server](#).

Additional topics

This section covers the following topics.

Topics

- [AWS SDK for SAP ABAP releases](#)
- [SAP licensing](#)

AWS SDK for SAP ABAP releases

AWS SDK for SAP ABAP is delivered in transports, and AWS SDK for SAP ABAP - BTP edition is delivered as add-ons. The mechanism to import transports and add-ons is different but the technical functionality is the same. For more information, see [Setting up](#).

Topics

- [Release strategy](#)
- [Best practices](#)
- [Patching SDK for SAP ABAP](#)
- [Installing an additional module](#)
- [Uninstalling SDK for SAP ABAP](#)

Release strategy

Version 1 of AWS SDK for SAP ABAP is updated frequently. New patches are released weekly or daily based on the releases and updates of AWS services. The patches for AWS services can include bug fixes and other changes that update patch level of the SDK. For more information, see [AWS SDKs and Tools maintenance policy](#).

Best practices

We recommended retaining same patch level of SDK for SAP ABAP for all SAP systems (development, QA, and production).

When patching the SDK, import the latest version in your sandbox. You can then import it to the development, QA, and production systems, following your normal change control procedures.

Patching SDK for SAP ABAP

Each SDK for SAP ABAP release is delivered as a set of cumulative transports, including all bug fixes, features, and updates. There is no difference between a patch and an installation transport. You must import the latest transports to patch SDK for SAP ABAP.

Due to dependencies of `core` Runtime and API modules, individual API modules cannot be patched separately. When patching, you must update the `core` module and all other API modules that have been installed, even if certain modules are no longer in use.

The following scenarios describe the patching process:

- **Scenario 1: Regular SDK Version Update**

When patching to the latest SDK version:

- Import all transports (core and installed API modules) simultaneously
- No specific import sequence is required
- For example, if you imported the `core`, `ec2`, and `lmd` transports when you installed the SDK, you must import the latest transports for `core`, `ec2`, and `lmd` when patching.

- **Scenario 2: Adding a New API Module**

When importing a new API module (e.g., Amazon Textract `tex`) from a different SDK release than your currently installed modules:

- All existing modules must be patched to match the SDK release version of the new module
- Import simultaneously (no specific sequence):
 - Latest transports for all existing modules (`core` and installed API modules)
 - Transports for the new API module `tex`

- **Scenario 3: Updating an Existing API Module**

When updating an existing API module (e.g., Amazon Translate `x18`) to the latest SDK version, individual module updates are not supported due to interdependencies. You must patch the entire SDK by updating the `core` module and all installed API modules together, following the process outlined in [Regular SDK Version Update \(p. 591\)](#).

Installing an additional module

Import the transport for the new module at the same patch level as your existing core and modules to install an additional API module in your SAP system. Follow the guidelines in [the section called "Patching SDK for SAP ABAP"](#) if you want to import a more recent version of the module. This ensures that the patch levels are compatible across all SDK modules.

Uninstalling SDK for SAP ABAP

To uninstall SDK for SAP ABAP, you must download a *deletion transports* kit from <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip>.

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

You can download a signature file from <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.sig>. To validate the file, see [Verify SDK for SAP ABAP](#).

For each SDK module installed on your SAP system, the corresponding *deletion transport* must be imported from the preceding ZIP file. You can remove a single module without uninstalling the entire SDK. You can do so by importing only the *deletion transport* for module you want to remove. If you are uninstalling the entire SDK with all of its modules, then the *core deletion transport* must be imported last.

We recommend that you test the uninstallation in a sandbox before attempting in development, QA or production systems.

Considerations

Before uninstalling the SDK, see the following considerations.

- SDK configuration settings from will be lost. The IMG must be reconfigured on installation.
- If you have Z programs that rely on the SDK, they will generate syntax errors after the removal of the SDK.
- PFCG or Business roles containing SDK authorization references will have invalid authorizations after the removal of the SDK. Remove SDK authorization references from the PFCG roles before uninstalling the SDK.

Note

AWS SDK for SAP ABAP - BTP edition cannot be uninstalled during the developer preview.

SAP licensing

The use of SAP software is subject to SAP's terms. You are responsible for complying with SAP licensing terms, including software distribution and indirect licensing conditions. Any information provided is not legal advice, and should not be relied upon for licensing compliance purposes. If you have questions about your licensing or rights to SAP software, consult your legal team, SAP, and/or your SAP reseller.

Question : Will SDK for SAP ABAP usage affect my SAP license?

Answer : AWS SDK for SAP ABAP enables you to consume AWS services with your own ABAP code. It is used in integration scenarios between an SAP system and AWS services. Any scenario where data from the SAP system is sent to a third-party (non-SAP) system, or created by that system, may have implications for indirect licensing. SAP has multiple approaches for defining indirect access, such as user-based calculations and outcome-based calculations. The methodology to define indirect access depends on your contract with SAP. You must be aware of the guidance provided in your contract with SAP, and you can further discuss this with SAP or their reseller.

In 2018, SAP released two documents – *Indirect Access Guide for SAP Installed Base Customers* and *SAP ERP Pricing for Digital Age - Addressing Indirect/Digital Access*. These documents can be found on SAP websites, and are examples of indirect licensing approaches. However, these documents do not reflect your particular agreement with SAP.

Document history for AWS SDK for SAP ABAP Developer Guide

The following table describes the documentation releases for AWS SDK for SAP ABAP.

Change	Description	Date
New content	Added Per-service proxy server override .	May 15, 2026
New content	Added Using the AWS SDK for SAP ABAP Knowledge MCP Server .	April 24, 2026
New content	Added Using Source Profile for Cross-Account Access .	March 4, 2026
New content	Added Using Secret Access Key Authentication with SSF Encryption .	January 12, 2025
New content	Developer preview of SDK for SAP ABAP - BTP edition.	May 31, 2024
New content	Added Using certificates with IAM Roles Anywhere .	December 1, 2023
New content	Added Building products with SDK .	December 1, 2023
New content	Added Retry behavior .	December 1, 2023
New content	Added SAP licensing .	September 22, 2023
Public release	Initial relaunch of AWS SDK for SAP ABAP Developer Guide.	June 30, 2023

New content	Added AWS SDK for SAP ABAP features .	May 30, 2023
New content	Added Troubleshoot AWS SDK for SAP ABAP .	February 17, 2023
Developer preview	Developer preview of AWS SDK for SAP ABAP Developer Guide.	November 17, 2022