aws

Guía para desarrolladores

AWS SDK para SAP ABAP

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

AWS SDK para SAP ABAP: Guía para desarrolladores

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS SDK para SAP ABAP? 1
Características de AWS SDK para SAP ABAP 1
Mantenimiento 1
Referencia de la API 2
Precios 2
Recursos 2
Introducción 3
Paso 1: Prepare su cuenta AWS 3
Rol de IAM para los usuarios de SAP 3
Autenticación 5
Paso 2: Instalar el SDK 6
Paso 3: Configurar el SDK 6
Paso 4: Configuración funcional 8
Paso 5: Autorizar a los usuarios de SAP 10
Paso 6: Escribe el código 12
Paso 7: ejecuta la aplicación 15
Configuración 17
Requisitos previos de SAP 17
SDK para SAP ABAP 17
SDK para SAP ABAP: edición BTP 21
Instalación AWS SDK para SAP ABAP 22
Descargar el SDK 22
Verificación del archivo 22
AWS Transportes del SDK 23
Instalación del SDK: edición BTP 27
Instale el SDK para SAP ABAP: edición BTP 27
Módulos 28
SDK de parches para SAP ABAP: edición BTP 29
Configuración 30
Configuración global 31
Configuración técnica 32
Configuración de escenarios 32
Configuración de aplicaciones 33
Perfil del SDK 33

Solucionador lógico de recursos ... 35
Ejemplo ... 35
Configuración del tiempo de ejecución ... 36
Registro y rastreo ... 36
OPT-IN: telemetría mejorada ... 36
Escenario activo ... 37
Escenarios de conectividad avanzada ... 37
Conexión a través de un servidor proxy ... 37
Conexión a través de un firewall de inspección de paquetes ... 38
Puntos de conexión de la puerta de enlace ... 38
Puntos de conexión personalizados ... 39
Enrutamiento avanzado ... 40
Per-service anulación del servidor proxy ... 41
Acceso a puntos de conexión en varias regiones ... 42
Configuración del proveedor de servicios ... 43
Actualización, rastreo y telemetría ... 44
Actualización del sistema SAP ... 44
Rastreo ... 45
Telemetría ... 46
Uso del SDK ... 47
Representación de datos ... 47
Tipos de datos ... 48
AWS tipos de datos ... 50
Programa de ejemplo ... 51
Requisitos previos ... 51
Código ... 52
Secciones de código ... 52
Conceptos ... 55
Clases de API ... 55
Objetos adicionales ... 56
Establecimiento de estructura de clases ... 56
Matrices ... 58
Mapas ... 59
Funciones de nivel superior ... 59
Características ... 1
Configuración mediante programación ... 60

Esperadores 61
Paginadores 62
Comportamiento de los reintentos 63
Prefirmantes 64
Encadenamiento de roles de IAM entre cuentas 65
Creación de productos 65
Configuración de un ID de producto 65
Personalice las solicitudes HTTP para AWS 66
Implemente una mejora 66
Filtra la mejora 66
Codifique la mejora 66
Limitaciones 67
Ejemplos de código 69
ACM 71
Acciones 71
API Gateway 79
Acciones 71
Controlador de recuperación de aplicaciones 88
Acciones 71
Aurora 93
Acciones 71
escalado automático 99
Acciones 71
Amazon Bedrock Runtime 108
Anthropic Claude 109
Stable Diffusion 112
Tiempo de ejecución de agentes de Amazon Bedrock 115
Acciones 71
CloudFront 116
Acciones 71
CloudWatch 119
Acciones 71
Escenarios 127
CloudWatch Registra 129
Acciones 71
Amazon Cognito Identity Provider 132

Acciones 71
Amazon Comprehend 137
Acciones 71
AWS Config 151
Acciones 71
AWS Control Tower 154
Acciones 71
Firehose 165
Acciones 71
DynamoDB 167
Conceptos básicos 168
Acciones 71
Amazon EC2 182
Acciones 71
Amazon ECR 205
Acciones 71
Amazon EMR 212
Acciones 71
EventBridge Planificador 218
Acciones 71
AWS Glue 223
Acciones 71
HealthImaging 236
Acciones 71
HealthLake 254
Acciones 71
IAM 267
Acciones 71
AWS IoT data 293
Acciones 71
AWS IoT SiteWise 295
Introducción 295
Acciones 71
Amazon Keyspaces 304
Acciones 71
Kinesis 312

Conceptos básicos 168
Acciones 71
AWS KMS 322
Introducción 295
Acciones 71
Lambda 343
Conceptos básicos 168
Acciones 71
Organizations 357
Acciones 71
Amazon Pinpoint 362
Acciones 71
API de SMS y voz de Amazon Pinpoint 367
Acciones 71
Amazon Polly 377
Acciones 71
Amazon RDS 385
Acciones 71
Amazon Redshift 391
Acciones 71
Amazon Rekognition 398
Acciones 71
Amazon S3 413
Conceptos básicos 168
Acciones 71
Escenarios 127
SageMaker IA 440
Acciones 71
Escenarios 127
Secrets Manager 458
Acciones 71
Amazon SES 460
Acciones 71
Amazon SES API v2 477
Acciones 71
Amazon SNS 486

Acciones 71
Escenarios 127
Amazon SQS 497
Acciones 71
Escenarios 127
Step Functions 507
Acciones 71
Systems Manager 515
Acciones 71
Amazon Textract 525
Acciones 71
Escenarios 127
Amazon Transcribe 536
Acciones 71
Amazon Translate 544
Acciones 71
Escenarios 127
Servidor MCP 553
Ventajas 553
Configuración del servidor MCP 554
Qué puede hacer el servidor MCP 555
Consideraciones importantes 556
Seguridad 557
Autenticación del sistema 558
Autenticación de metadatos 558
Autenticación de clave de acceso secreta 559
Autenticación basada en certificados mediante funciones de IAM en cualquier lugar 559
Autenticación del perfil de origen para el acceso entre cuentas 560
Siguiente paso 560
Prácticas recomendadas de seguridad de IAM 560
Práctica recomendada para el perfil de instancia de Amazon EC2 561
Roles de IAM para usuarios de SAP 562
Consideraciones de seguridad del perfil de origen 565
Autorizaciones de SAP 566
Autorizaciones de configuración 566
Autorizaciones de SAP para usuarios finales 567

Operaciones seguras 569
Cifrado de datos en reposo 569
Cifrado de datos en tránsito 569
Uso de la API 2
Clave secreta (SSF) 569
Requisitos previos 569
Procedimiento 570
IAM Roles Anywhere 572
Requisitos previos 572
Procedimiento 573
Perfil de origen 65
Requisitos previos 577
Procedimiento 577
Tienda de credenciales 21
Pasos de configuración 580
Uso de SAP Credential Store con el SDK 582
Solución de problemas 586
Error de importación 586
Restricción de ubicación no especificada 586
Errores de SSL 587
Perfil de configuración 588
Autorización de IAM 589
Autorización de acciones 589
Escenario activo 37
Caracteres especiales 590
Conectividad 590
Temas adicionales 592
Versiones 592
Estrategia de versión 592
Prácticas recomendadas 560
Aplicación de parches en el SDK de SAP ABAP 593
Instalación de un módulo adicional 594
Desinstalación del SDK de SAP ABAP 594
Licencias de SAP 595
Historial de revisión 597
..... dxcix

¿Qué es AWS SDK para SAP ABAP?

AWS SDK para SAP ABAP proporciona una interfaz para los servicios ofrecidos AWS en el lenguaje ABAP. Con el SDK, puede implementar ABAP BADIs, informes, transacciones, OData servicios y otros artefactos ABAP en Servicios de AWS, como Amazon Simple Storage Service (Amazon S3),, etc. Amazon DynamoDB Amazon Translate También puede desarrollar sistemas basados en ABAP, a partir de SAP NetWeaver 7.4 y en un entorno de plataforma tecnológica empresarial SAP. Para obtener más información, consulte Instalación AWS del SDK para SAP ABAP: edición BTP.

Temas

- Características de AWS SDK para SAP ABAP
- Mantenimiento y compatibilidad de las versiones principales del SDK
- Referencia de la API
- Precios
- Recursos adicionales

Características de AWS SDK para SAP ABAP

AWS SDK para SAP ABAP ha sido diseñado para que los desarrolladores de SAP lo conozcan y lo entiendan de forma natural. Por ejemplo, aunque todos Servicios de AWS utilizan las `false` cadenas `true` y para representar datos booleanos en estructuras XML y JSON, el SDK para SAP ABAP los convierte en valores nativos de ABAP y de un solo carácter`'X'.' '` El SDK de SAP ABAP utiliza constructos ABAP nativos en la medida de lo posible, incluso en tipos de datos y formatos de marca de tiempo. Como resultado, el programador de ABAP no tiene que preocuparse por la serialización subyacente de JSON y XML ni por el formato de comunicación del protocolo de la API.

Mantenimiento y compatibilidad de las versiones principales del SDK

Para obtener información sobre el mantenimiento y el soporte de las versiones principales del SDK y sus dependencias subyacentes, consulte lo siguiente en la Guía de referencia de herramientas y herramientas:AWS SDKs

- AWS SDKs y política de mantenimiento de herramientas

- AWS SDKs y matriz de soporte de herramientas y versiones

Referencia de la API

Para ver una lista completa AWS SDK para SAP ABAP APIs, consulte AWS SDK para SAP ABAP la Guía de referencia de la API.

Para ver una lista completa de módulos AWS SDK para SAP ABAP TLAs, consulte AWS SDK para SAP ABAP - Lista de módulos.

Para ver una lista completa de módulos del SDK para SAP ABAP (versión preliminar para desarrolladores) TLAs, consulte AWS SDK for SAP ABAP (edición BTP), lista de módulos.

Precios

AWS SDK para SAP ABAP está disponible sin coste adicional. Solo pagas por AWS los recursos y servicios que consumas con el SDK.

Recursos adicionales

Además de esta guía, están disponibles los siguientes recursos online para el SDK de SAP ABAP.

- AWS SDK para SAP ABAP Knowledge MCP Server: conecte su IDE habilitado para IA para generar un código ABAP preciso para las integraciones. AWS
- AWS Biblioteca de ejemplos de códigos del SDK
- SAP sobre la AWS documentación
- Blog de desarrolladores de AWS
- AWS foros de desarrolladores
- @awsdevelopers(Twitter)

Empezar con AWS SDK para SAP ABAP

En esta sección se describe cómo empezar a usar el SDK. Incluye información sobre la instalación del SDK, la configuración básica y la creación de un ejemplo de código de Hello World que traduce una frase de un idioma a otro. Si es la primera vez que utiliza el AWS SDK, le recomendamos que realice estos pasos en un entorno sandbox.

Steps

- Paso 1: Prepare su cuenta AWS
- Paso 2: Instalar el SDK
- Paso 3: Configurar el SDK
- Paso 4: Configuración funcional
- Paso 5: Autorizar a los usuarios de SAP
- Paso 6: Escribe el código
- Paso 7: ejecuta la aplicación

Paso 1: Prepare su cuenta AWS

Para empezar a utilizar el SDK de SAP ABAP, debe tener una Cuenta de AWS activa. Lo necesita Cuenta de AWS incluso si su sistema SAP está alojado en las instalaciones, en la plataforma de tecnología empresarial (BTP) de SAP o con otro proveedor de servicios en la nube.

Si su sistema SAP se ejecuta en AWS la nube, realizará llamadas a los AWS servicios de su empresa. Cuenta de AWS

Temas

- Rol de IAM para los usuarios de SAP
- Autenticación

Rol de IAM para los usuarios de SAP

- Cree un rol de IAM con las instrucciones que se proporcionan en la Guía del usuario de AWS Identity and Access Management . Para obtener más información, consulte Crear un rol para

delegar permisos a un AWS servicio. Anote el nombre de recurso de Amazon (ARN) del rol de IAM para uso posterior.

- Seleccione Amazon EC2 como caso de uso.
- Utilice `SapDemoTranslate` como nombre del rol.
- Adjunte el perfil de `TranslateReadOnly` al rol.
- El rol debe tener las siguientes entidades para que el sistema SAP pueda asumirlo. Sustituya *"111122223333"* por su nombre de cuenta de AWS .

  JSON

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

  En este ejemplo se muestra que cualquier director de la Cuenta de AWS *"111122223333"* puede asumir la función. Este es un permiso amplio adecuado para proof-of-concept. Puede utilizar una entidad principal más limitada para la producción, como en los ejemplos siguientes.

  - Un usuario específico: cuando el sistema SAP utiliza una de las siguientes opciones:
    - Credenciales cifradas con SSF de un sistema SAP local
    - Credenciales del servicio SAP Credential Store en un entorno SAP BTP y ABAP
  - Un rol específico: cuando el sistema SAP está en Amazon EC2 y hay un perfil de instancia.
  - Amazon EC2: cuando el sistema SAP está en Amazon EC2 y no hay perfil de instancia.

Para obtener más información, consulte Prácticas recomendadas de seguridad de IAM.

Autenticación

La autenticación depende del lugar donde esté alojado el sistema SAP.

Ubicaciones

- En la nube AWS
- En las instalaciones, SAP BTP u otra nube

En la nube AWS

Asegúrese de que la instancia EC2 en la que se ejecuta el sistema SAP tenga un perfil de instancia con los siguientes permisos.

JSON

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

Añada el ARN que anotó en el paso anterior.

Este permiso permite que su sistema SAP asuma el rol de `SapDemoTranslate` en nombre del usuario de ABAP.

En las instalaciones, SAP BTP u otra nube

Si su sistema SAP está ubicado de forma local, en SAP BTP o en otra nube, siga los siguientes pasos para establecer una conexión de autenticación mediante una clave de acceso secreta.

1. Cree un usuario de IAM. Para obtener más información, consulte Creación de usuarios de IAM (consola).

2. Utilice `SapDemoSID` como nombre del usuario de IAM. `SID` es el ID de sistema de su sistema SAP.
3. Asigne el rol `SapDemoTranslate` a este usuario.

Conserve la `yaccess_key`. `secret_access_key` Debe configurar estas credenciales en su sistema SAP.

> **Note**
>
> Si su sistema SAP está ubicado en las instalaciones, en SAP BTP o en otra nube, puede autenticarse mediante una de las siguientes opciones.
>
> - Autenticación de claves de acceso secretas mediante SSF o SAP Credential Store
> - Uso de certificados con IAM Roles Anywhere

Paso 2: Instalar el SDK

Consulte las siguientes pestañas para ver las instrucciones de instalación.

SDK for SAP ABAP

Importe el SDK para los transportes SAP ABAP en su sistema SAP. Puede importar los transportes a cualquier cliente. Para obtener más información, consulte Instalación del SDK para SAP ABAP.

SDK for SAP ABAP - BTP edition

Instale el SDK para SAP ABAP, edición BTP, mediante la aplicación Deploy Product. Para obtener más información, consulte Instalación del SDK para SAP ABAP: edición BTP.

Paso 3: Configurar el SDK

Antes de configurar el SDK, asegúrese de contar con las autorizaciones necesarias. Para obtener más información, consulte Autorizaciones de SAP.

Consulte las siguientes pestañas para obtener instrucciones de configuración.

SDK for SAP ABAP

Ejecute la `/AWS1/IMG` transacción para abrir la Guía de implementación del SDK para SAP ABAP. Para ejecutar esta transacción, escriba `/n/AWS1/IMG` en la barra de comandos de su sistema SAP y, a continuación, pulse Intro.

Complete las siguientes configuraciones.

- Vaya a Requisitos técnicos previos.
  - Revise los parámetros recomendados y la conectividad HTTPS.
- Vaya a Configuración global → Configurar escenarios.
  - Cambie la configuración según las recomendaciones de la configuración global.
- Vaya a Configuración global → Configuración técnica.
  - Cambie la configuración según las recomendaciones de la configuración global.
- Vaya a Configuración de tiempo de ejecución → Registrar y rastrear.
  - Seleccione Nuevas entradas.
    - Nivel de rastreo: sin rastreo.
    - Número máximo de líneas de volcado: `100`.
    - OPT-IN: telemetría mejorada: deje este campo en blanco.
  - Seleccione Guardar.
- Vaya a Configuración de tiempo de ejecución → Escenario activo.
  - En Nuevo escenario, seleccione `DEFAULT`.
  - Seleccione Confirmar cambio de escenario.
  - Acepte la solicitud.

Requisitos previos para sistemas en las instalaciones

Si su sistema SAP se ejecuta en las instalaciones o en otra nube, las credenciales deben almacenarse en la base de datos de SAP. Las credenciales se cifran mediante SAP SSF y requieren una biblioteca criptográfica configurada, como la de SAP. CommonCryptoLib

Los pasos para configurar SSF para el SDK de SAP ABAP se describen en la transacción `/AWS1/IMG`.

> **Note**
>
> El requisito previo anterior no se aplica si el sistema SAP se ejecuta en Amazon EC2. Los sistemas SAP que se ejecutan en Amazon EC2 pueden recuperar credenciales de corta duración que rotan automáticamente a partir de metadatos de instancias de Amazon EC2.

SDK for SAP ABAP - BTP edition

Abra su entorno ABAP en un navegador web y navegue hasta la aplicación Custom Business Configurations.

Complete las siguientes configuraciones.

- Vaya a Configurar escenarios.
  - Cambie la configuración según las recomendaciones de la configuración global.
- Vaya a la configuración técnica.
  - Cambie la configuración según las recomendaciones de la configuración global.

Paso 4: Configuración funcional

Consulte las siguientes pestañas para obtener instrucciones de configuración.

SDK for SAP ABAP

Ejecute la transacción `/AWS1/IMG` (introdúzcala `/n/AWS1/IMG` en la barra de comandos y elija Entrar) para abrir la guía de implementación AWS del SDK.

- Vaya a Configuración de la aplicación → Perfil del SDK.
  - Seleccione Nuevas entradas.
    - Perfil: `DEMO`.
    - Descripción: `Demo profile`.
    - Seleccione Guardar.
- Resalte la entrada que creó y haga clic en la rama del árbol de autenticación y configuración.
  - Seleccione Nuevas entradas.
    - SID: el ID del sistema SAP en el que se encuentra actualmente.

- Cliente: el cliente del sistema SAP en el que se encuentra actualmente.
- ID de escenario: la lista desplegable donde encontrará el escenario PREDETERMINADO creado por su administrador de Basis.
- AWS Región: introduce la AWS región a la que quieres hacer llamadas. Si su sistema SAP está funcionando AWS, introduzca la AWS región en la que se ejecuta.
- Método de autenticación:
  - Seleccione Rol de instancia mediante los metadatos si su sistema SAP se ejecuta en Amazon EC2.
  - Si su sistema SAP se ejecuta en las instalaciones o en otra nube, seleccione Credenciales del almacenamiento de SSF.
    - Seleccione Establecer credenciales.
    - Escriba el ID de clave de acceso y la clave de acceso secreta que ha creado en el paso anterior.
- Mantenga en blanco la opción Desactivar roles de IAM.
- Seleccione Guardar.

- Haga clic en la rama del árbol de asignación de roles de IAM.
  - Seleccione Nuevas entradas.
    - Escriba el número de secuencia: 010.
    - Escriba el rol de IAM lógico: TESTUSER.
    - Escriba el ARN del rol de IAM: escriba el arn:aws: del rol de IAM que contiene la política `TranslateRead0nly` creada en el paso anterior.

SDK for SAP ABAP - BTP edition

Configure la autenticación mediante SAP Credential Store. Para obtener más información, consulte Uso de SAP Credential Store.

Abra su entorno ABAP en un navegador web y navegue hasta la aplicación Custom Business Configurations.

- Ve al perfil del SDK.
  - Selecciona Editar para crear un perfil nuevo.
    - Perfil: `DEM0`.
    - Descripción: `Demo profile`.

- Seleccione la tecla de flecha derecha situada junto a la entrada creada para ir a la pestaña Autenticación y configuración.

  Seleccione Nuevas entradas.

  - SID: el ID del sistema SAP en el que se encuentra actualmente.
  - Cliente: el cliente del sistema SAP en el que se encuentra actualmente.
  - ID de escenario: la lista desplegable donde encontrará el escenario PREDETERMINADO creado por su administrador de Basis.
  - AWS Región: introduce la AWS región a la que quieres hacer llamadas. Si su sistema SAP está funcionando AWS, introduzca la AWS región en la que se ejecuta.
  - Método de autenticación: seleccione las credenciales del almacén de credenciales de SAP.
  - Introduzca el espacio de nombres y el nombre clave de las credenciales almacenadas en SAP Credentials Store.
  - Introduzca el nombre del acuerdo de comunicación creado para establecer la comunicación entre el SDK para SAP ABAP (edición BTP) y SAP Credential Store.
  - Mantenga en blanco la opción Desactivar roles de IAM.
- Haga clic con el botón derecho en la tecla de flecha derecha situada junto a la entrada creada para acceder a la pestaña de mapeo de roles de IAM.

  Seleccione Nuevas entradas.

  - Escriba el número de secuencia: 010.
  - Escriba el rol de IAM lógico: TESTUSER.
  - Escriba el ARN del rol de IAM: escriba el arn:aws: del rol de IAM que contiene la política `TranslateReadOnly` creada en el paso anterior.

Paso 5: Autorizar a los usuarios de SAP

Los usuarios de SAP no están autorizados a utilizar AWS la funcionalidad de forma predeterminada. Los usuarios deben estar autorizados de forma explícita mediante autorizaciones de SAP. Consulte las siguientes pestañas para obtener más información.

SDK for SAP ABAP

Cree un rol de PFCG

- Vaya a la transacción `PFCG`
- Escriba el nombre del rol `ZAWS_SDK_DEMO_TESTUSER` y seleccione Crear un rol único.
  - Descripción: `Role for demo AWS SDK functionality`.
  - Vaya a la pestaña Autorizaciones.
  - Seleccione Cambiar datos de autorización y acepte la ventana emergente informativa.
  - En la ventana emergente Elegir plantilla, seleccione No seleccionar plantillas.
  - Seleccione Añadir manualmente en la barra de herramientas.
  - Añada los siguientes objetos de autorización:
    - `/AWS1/LROL`
    - `/AWS1/SESS`
  - En el árbol de autorizaciones, ingrese:
    - Perfil para acceder a AWS APIs: `DEMO`
    - Rol de IAM lógico: `TESTUSER`
  - Seleccione Guardar.
  - Seleccione Generar.
  - Seleccione Atrás.
  - Seleccione Guardar para guardar el rol.

Asignar el rol de PFCG a los usuarios de SAP

Cualquier usuario que tenga el `ZAWS_SDK_DEMO_TESTUSER` rol asignado estará autorizado a usar las funciones AWS del SDK con los ajustes configurados en el perfil `DEMO` del SDK. El usuario autorizado también asumirá el rol de IAM asignado al rol de IAM lógico `TESTUSER` en ese perfil.

- Ejecute la transacción `SU01`.
  - Introduzca el seudónimo de un usuario de SAP que probará la funcionalidad AWS del SDK.
  - Seleccione Cambiar.
  - Vaya a la pestaña Roles y asigne el rol `ZAWS_SDK_DEMO_TESTUSER` al usuario.
  - Seleccione Guardar.

SDK for SAP ABAP - BTP edition

Cree un rol empresarial

- Abra su entorno ABAP en un navegador web y navegue hasta la aplicación Maintain Business Roles.
- Seleccione Crear a partir de una plantilla e introduzca los siguientes detalles.
  - Plantilla: elija**`/AWS1/RT_BTP_ENDUSER`**.
  - ID de nuevo rol empresarial: introduzca un ID.
  - Descripción del nuevo rol empresarial: introduzca una descripción.
- Seleccione Aceptar para ver la página del rol empresarial.
- En la pestaña Detalles generales del rol, vaya a Categorías de acceso y establezca el campo de ayuda para escribir, leer y valorar como restringido.
- Seleccione Mantener restricciones y expanda los tipos de restricciones asignados en el panel de navegación izquierdo. Actualice el siguiente campo en la sección Restricciones y valores.
  - En Elegir sesión del SDK, selecciona el icono del lápiz situado junto al perfil del SDK y navega hasta la pestaña Rangos. **`DEMO`**Ingresa y selecciona Agregar.
  - En Elegir función de IAM lógica, seleccione el icono de lápiz situado junto a la función de IAM lógica y vaya a la pestaña Rangos. Introduzca **`TESTUSER`** y seleccione Añadir.

    Selecciona el icono del lápiz situado junto al perfil del SDK y navega hasta la pestaña Rangos. **`DEMO`**Ingresa y selecciona Agregar
- Vuelva a la plantilla de rol empresarial y abra la pestaña Usuarios empresariales. Seleccione Agregar para asignar el rol empresarial recién creado a un usuario empresarial de SAP que probará la funcionalidad del SDK. Seleccione Guardar.

Cualquier usuario empresarial asignado al rol empresarial creado estará autorizado a utilizar las funciones del AWS SDK con los ajustes configurados en el perfil `DEM0` del SDK. El usuario autorizado también asumirá el rol de IAM asignado al rol de IAM lógico `TESTUSER` en ese perfil.

Paso 6: Escribe el código

Consulte las siguientes pestañas para obtener más información.

SDK for SAP ABAP

1. Abra la transacción SE38.
   - Escriba ZDEMO_TRANSLATE_HELLO_WORLD como nombre del programa.
   - Seleccione Create.
   - Escriba AWS SDK Hello World In Any Language como título.
   - Tipo: elija Programa ejecutable.
   - Estado: elija Programa de prueba.
   - Seleccione Guardar.
   - Guarde el programa como un objeto local.

Añada el código siguiente.

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

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

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

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

      WRITE: / 'Source Phrase: ', pv_text.
```

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

SDK for SAP ABAP - BTP edition

1. Haga clic con el botón derecho en el paquete en el que desee crear la clase ABAP y, a continuación, seleccione Nuevo > Clase ABAP.
2. Introduzca el nombre **`ZCL_DEMO_XL8_SIMPLE`** de la clase y añada una descripción de la clase. Seleccione Siguiente.
3. Cree o elija una solicitud de transporte. Selecciona Finalizar.

Añada el código siguiente.

```
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_xl8unsuppedlanguage00 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.
```

Para obtener más información sobre cómo escribir código ABAP que utilice el SDK, consulte Uso AWS SDK para SAP ABAP.

Paso 7: ejecuta la aplicación

Consulte las siguientes pestañas para obtener más información.

SDK for SAP ABAP

Ejecute la aplicación en SE38. Si la operación se realiza correctamente, el resultado será el siguiente.

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

Si faltan autorizaciones, la configuración o los requisitos previos de Basis, es posible que aparezca un mensaje de error. Consulte el siguiente ejemplo.

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

Si su rol de SAP le autoriza a utilizar un perfil del SDK y asignarlo a un rol de IAM lógico mientras sus permisos de IAM no están configurados para que el sistema SAP asuma el rol de IAM, el resultado será el siguiente.

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

En este caso, revise los permisos de IAM y la configuración de confianza en los roles de IAM, los usuarios o ambos definidos en the section called “Paso 1: Prepare su cuenta AWS”.

SDK for SAP ABAP - BTP edition

Ejecute la aplicación en la aplicación Eclipse > Ejecutar como > ABAP (consola). Si la operación se realiza correctamente, el resultado será el siguiente.

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

Si faltan autorizaciones, la configuración o los requisitos previos de Basis, es posible que aparezca un mensaje de error. Consulte el siguiente ejemplo.

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

Si su rol de SAP le autoriza a utilizar un perfil del SDK y asignarlo a un rol de IAM lógico mientras sus permisos de IAM no están configurados para que el sistema SAP asuma el rol de IAM, el resultado será el siguiente.

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

En este caso, revise los permisos de IAM y la configuración de confianza en los roles de IAM, los usuarios o ambos definidos en the section called “Paso 1: Prepare su cuenta AWS”.

Configuración

En esta sección se ofrece información sobre cómo configurar su entorno de desarrollo para utilizar AWS SDK para SAP ABAP.

Temas

- Requisitos previos de SAP
- Instalación AWS SDK para SAP ABAP
- Instalación AWS del SDK para SAP ABAP - edición BTP

Requisitos previos de SAP

Los siguientes requisitos previos para instalar el SDK se aplican cuando los sistemas SAP están alojados en AWS ellos.

Temas

- Requisitos previos del AWS SDK para SAP ABAP
- Requisitos previos del AWS SDK para SAP ABAP: edición BTP

Requisitos previos del AWS SDK para SAP ABAP

Los siguientes son los requisitos previos del AWS SDK para SAP ABAP.

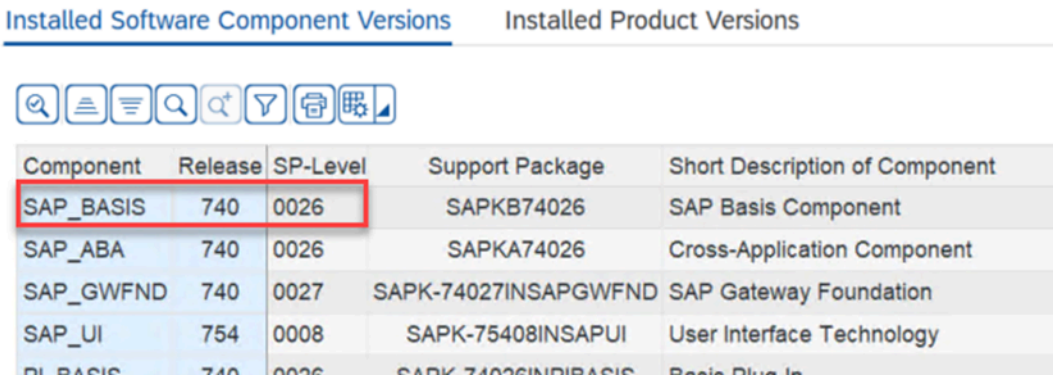
Temas

- Versión de Basis
- Versión del kernel
- Parámetros
- Notas
- Conectividad saliente
- Conectividad HTTPS
- Acceso a los metadatos de las EC2 instancias de Amazon

Versión de Basis

El SDK para SAP ABAP es compatible con SAP NetWeaver 7.4 y versiones posteriores. El SDK de SAP ABAP no afecta a ninguna tabla de aplicaciones de SAP. Es completamente independiente de las aplicaciones, como Planificación de recursos empresariales de SAP y SAP Landscape Transformation Replication Server.

El nivel de SP mínimo admitido para `SAP_BASIS 740` es `SP 0008`. Para obtener más información, consulte la nota de SAP 1856171: se admiten campos de formulario con el mismo nombre en CL_HTTP_ENTITY (requiere acceso al portal de SAP). En función de sus requisitos empresariales, puede elegir un nivel de SP superior, como se muestra en la siguiente imagen.


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

No hay ningún requisito mínimo de nivel de SP para `SAP_BASIS 750` y versiones superiores.

Versión del kernel

El SDK de SAP ABAP y las herramientas que utilizan Internet Communication Manager (ICM) para la conectividad HTTP se basan en el kernel de SAP para sus funciones criptográficas, HTTP, XML y JSON. Recomendamos utilizar la última versión del núcleo que sea compatible con su NetWeaver plataforma SAP. El requisito mínimo es la versión del kernel 741. Para obtener más información, consulte la nota de SAP 2083594: versiones del kernel de SAP y niveles de parches del kernel de SAP (requiere acceso al portal de SAP).

Si utiliza las versiones 741 o 742 del kernel, se requieren los siguientes niveles de parche:

- 741, parche n.º 212
- 742, parche n.º 111

Parámetros

Su sistema SAP debe admitir la indicación de nombre de servidor (SNI), tal como se describe en las siguientes notas de SAP (requiere acceso al portal de SAP).

- Nota de SAP 2124480: Distribuidor web/ICM: indicación del nombre del servidor (SNI) con extensión TLS como cliente
- Nota de SAP 2582368: actualización de SapSSL para el envío de la extensión TLS de SNI por parte del cliente mediante saphttp, sapkprotp, sldreg

Configure el siguiente parámetro en el archivo `DEFAULT.PFL`.

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

Notas

Aplique la siguiente nota de SAP a su sistema.

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

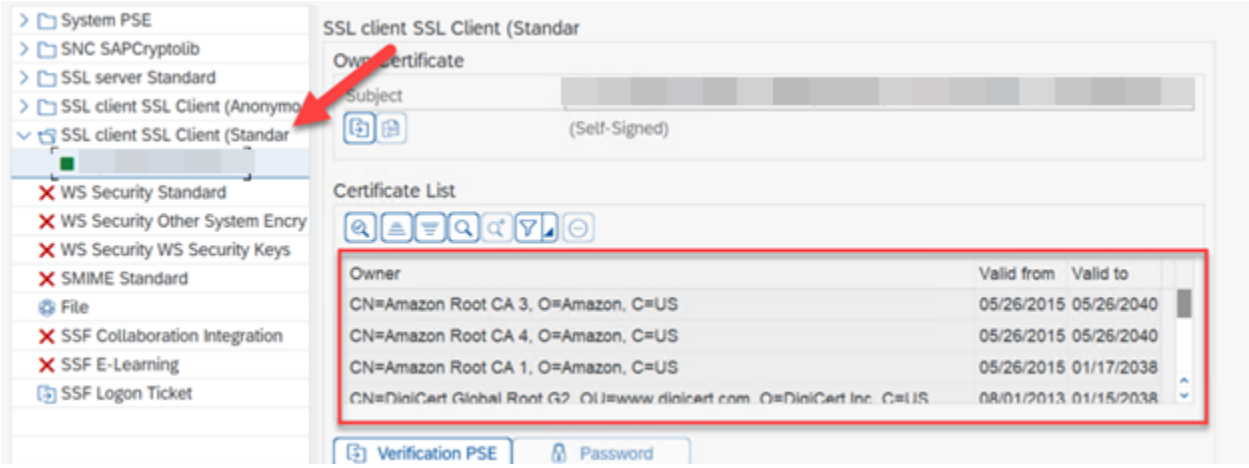
Conectividad saliente

El SDK de SAP ABAP es un cliente HTTPS. El sistema SAP envía los mensajes HTTPS salientes. No se requiere conectividad entrante.

Conectividad HTTPS

Todas las llamadas a AWS la API se realizan con canales HTTPS cifrados. El sistema SAP debe configurarse para que confíe en AWS los certificados a fin de establecer una conexión HTTPS saliente.

1. Vaya a https://www.amazontrust.com/repository/.
2. En Root CAs, descargue todos los certificados mediante el enlace PEM.
3. Importe estos certificados en `SSL Client (Standard) PSE` de su `STRUST` en cada uno de sus sistemas SAP, como se muestra en la siguiente imagen.



Acceso a los metadatos de las EC2 instancias de Amazon

El sistema ABAP establece conexiones HTTP sin cifrar a localhost (http://169.254.169.254) para habilitar los metadatos de las instancias de Amazon EC2 . El canal HTTP solo se usa para recuperar AWS las credenciales del servidor local. El tráfico HTTP permanece dentro del host.

Los metadatos permiten que un sistema SAP se autentique de forma segura sin almacenar una clave secreta en el SAP Secure Store. AWS Esta función solo se aplica a los sistemas SAP alojados en Amazon EC2.

Configure el archivo `DEFAULT.PFL` con el siguiente parámetro para permitir que su sistema SAP establezca una conexión HTTP saliente sin cifrar.

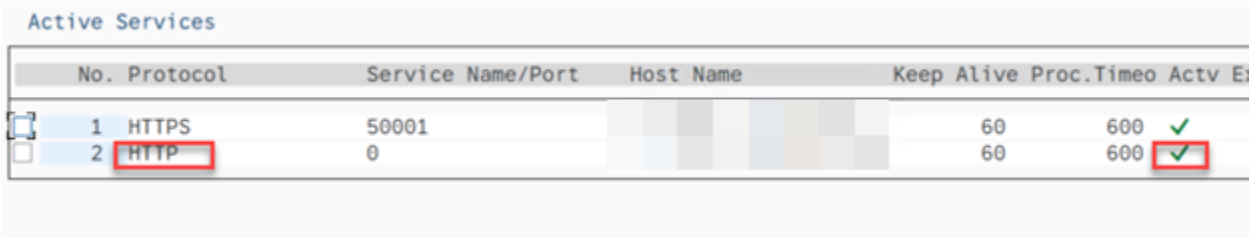
```
icm/server_port_<xx> = PROT=HTTP,PORT=8000,TIMEOUT=60,PROCTIMEOUT=600
```

Utilice el siguiente parámetro para habilitar la conexión HTTP saliente sin habilitar la conexión entrante.

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

Verifique que su sistema SAP esté configurado para las conexiones HTTP salientes con los siguientes pasos:

1. Ejecute la transacción SMICM.
2. Vaya a Servicios activos.
3. Verifique que ve una marca de verificación verde en la fila HTTP, en la columna Activa, como se muestra en la siguiente imagen.



Requisitos previos del AWS SDK para SAP ABAP: edición BTP

Los siguientes son los únicos requisitos previos para el AWS SDK para SAP ABAP: edición BTP.

Temas

- SAP Landscape Portal: edición BTP
- SAP Credential Store: edición BTP

SAP Landscape Portal: edición BTP

Este requisito previo solo se aplica al AWS SDK para SAP ABAP: edición BTP.

SAP Landscape Portal es el único mecanismo compatible para instalar complementos en un entorno SAP BTP. Asegúrese de estar suscrito para utilizar este servicio. Para obtener más información, consulte Landscape Portal.

SAP Credential Store: edición BTP

Este requisito previo solo se aplica al AWS SDK para SAP ABAP: edición BTP.

En la versión preliminar para desarrolladores, la autenticación con clave de acceso secreta es el único mecanismo compatible para autenticar el AWS SDK de la edición BTP de SAP ABAP. El SDK lee las credenciales del almacén de credenciales y almacena la clave de acceso secreta de forma segura.

Debe cumplir los siguientes requisitos previos.

- Suscripción a Credential Store.
- La tienda de credenciales se asigna como un derecho a su subcuenta de BTP. Consulte Configuración inicial para obtener más información.
- Una instancia de servicio con un plan estándar para Credential Store. Consulte Crear una instancia de servicio para obtener más información.

Para obtener más información, consulte Uso de SAP Credential Store.

El servicio SAP Credential Store se ejecuta en SAP BTP fuera del sistema ABAP BTP. Consulte SAP Credential Store para obtener más información.

Instalación AWS SDK para SAP ABAP

Temas

- Descargar el SDK de SAP ABAP
- Verificación del archivo del SDK de SAP ABAP (opcional)
- AWS Transportes del SDK

Descargar el SDK de SAP ABAP

Descargue el SDK desde https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip https://sdk-for-sapabap.amazonaws.cn/ .

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

Cuando se complete la descarga, le recomendamos que descomprima el archivo descargado en un directorio, por ejemplo `/tmp/awssdk`.

Verificación del archivo del SDK de SAP ABAP (opcional)

Este paso opcional de validar la firma del archivo del SDK le ayuda a confirmar que el SDK no se ha manipulado. Siga los pasos que se indican a continuación para verificar el archivo SDK.

1. Descargue el archivo SDK SIGNATURE con el siguiente comando.

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

2. Copie la siguiente clave pública y guárdela en un archivo denominado `abapsdk-signing-key.pem`.

   ```
   -----BEGIN PUBLIC KEY-----
   MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAmS3oN3wKBh4HJOGaOtye
   15RR5909nuw0JxOvEDCT7O9wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bVl6U3gDv
   ```

```
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2UrlSYijCRB
BWZFW+Ux0OkILz+8vCFSXMZ6Z0qtLIlZFbGrn6A5adbwwzfOqkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrxUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QYl
1FSZwAhQmk8hUPgUaqOOQRC6z3TRzIGKOA/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
O7DthUpj8b7QcQaiOpnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuiscJ2lMPkz0
cHOR3lMNsMLzm+d/gVkLt31R/JwAcFCkXTWvR8/VOWNGZZXdVUbefrfI/k7fP60B
bzUrIlN4poq16rc4Tk5Derg+wQ7rOWjXkXop2kiCMjbYo0ol0kS/At64PLjpz8dH
Zg25o79U9EJln+lpqZ297Ks+HoctOv2GPbeeh0s7+N0fRTyOr81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4exOGqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. Verifique el archivo ZIP del SDK descargado con el comando siguiente. El comando requiere `openssl` que forme parte de muchas distribuciones de Linux.

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

4. Verifique que el resultado del comando anterior es `Verified OK`.
5. Si el resultado es `Verification Failure`, repita los pasos anteriores. Si sigue recibiendo un resultado fallido, no instale el SDK ni se ponga en contacto con Soporte.

AWS Transportes del SDK

Temas

- Contenido
- Importación
- Namespace

Contenido

La instalación del SDK de SAP ABAP se completa a través de los transportes de ABAP. Debe importar estos transportes a su entorno de desarrollo o a un entorno aislado.

Cada versión del SDK de SAP ABAP reemplaza por completo a la anterior. No es necesario aplicar transportes incrementales. Los transportes se agrupan en un archivo ZIP. La estructura del archivo ZIP es la siguiente.

```
transports/
```

```
transports/core/
transports/core/Knnnnnn.AWS
transports/core/Rnnnnnn.AWS
transports/tla1/
transports/tla1/Knnnnnn.AWS
transports/tla1/Rnnnnnn.AWS
transports/tla2/
transports/tla2/Knnnnnn.AWS
transports/tla2/Rnnnnnn.AWS
.
.
.
```

La carpeta `transports` contiene una subcarpeta `core`. La subcarpeta `core` contiene los principales transportes en tiempo de ejecución y una subcarpeta para cada módulo, denominada con la abreviatura de tres letras del módulo. Para obtener una lista completa de los módulos TLAs, consulte AWS SDK para SAP ABAP : Lista de módulos.

AWS Los transportes del SDK son solicitudes de Workbench. En función de la configuración de las rutas del TMS, es posible que el SDK no las reenvíe automáticamente a las colas de control de calidad y producción después de importarlas al sistema anterior. Debe añadirlas manualmente a la cola de cada sistema.

Cuando tu proyecto esté listo para la siguiente fase, podrás importar el AWS SDK junto con otros transportes que contengan tu propio Z código y con funcionalidad empresarial. Si utiliza un sistema de control de cambios, como SAP Change Request Management (ChaRM), consulte a su administrador de ChaRM para gestionar correctamente los transportes de terceros.

Importación

Temas

- Indicadores clave
- Tiempo de importación

AWS Los transportes del SDK son independientes del cliente. El transporte principal es obligatorio y contiene el código de tiempo de ejecución del SDK, la API para AWS Security Token Service y la API para Amazon Simple Storage Service. Cada uno de los módulos del SDK restantes se entregan en un transporte independiente. Para que el SDK no ocupe demasiado espacio en su sistema, cada

módulo del SDK es opcional. Puede instalar módulos adicionales más adelante, si es necesario para su lógica empresarial.

Por ejemplo, si desea utilizar el transporte APIs para Amazon S3 e Amazon Translate importar el `core` transporte (que contiene el tiempo de ejecución principal, Amazon S3 y AWS STS los módulos) y el `xl8` transporte (que contiene el módulo Amazon Translate).

Para ver una lista completa del SDK para SAP ABAP APIs, consulte SDK for SAP ABAP: Guía de referencia de API.

Los siguientes son algunos consejos clave a la hora de importar AWS transportes del SDK.

- Cada transporte se entrega como `Knnnnnn.AWS` y `Rnnnnnn.AWS`
  - `Knnnnnn.AWS` debe copiarse en `/usr/sap/trans/cofiles`
  - `Rnnnnnn.AWS` debe copiarse en `/usr/sap/trans/data`.
- Al importar transportes, debe seleccionar Ignorar versión de componentes no válidos que se encuentra en Importar solicitud de transporte > Opciones > Opciones de importación.
- Todos los transportes deseados se pueden importar simultáneamente.
- Si se importan los transportes por separado, primero se debe importar el transporte `core`.
- El nivel de versión de todos los transportes debe ser idéntico.

Tiempo de importación

AWS La importación de los transportes del SDK puede tardar varios minutos. Los transportes se realizan correctamente si el STMS muestra una luz verde (`RC=0`) o amarilla (`RC=4`).

- Una luz roja (`RC=8`) indica que la importación tuvo un error de sintaxis.
  - Seleccione Solicitud → Mostrar → Registros para examinar el error de importación.
  - Durante la importación, si se produce un error debido a la falta de una interfaz `IF_SYSTEM_UUID_RFC4122_STATIC`, asegúrese de que la nota de SAP 2619546 se aplique al sistema. Para obtener más información, consulte Notas.
  - Si se desconoce la causa del error, póngase en contacto con Soporte.
- Un rayo rojo (`RC=12`) indica que los archivos de transporte no se han cargado correctamente en `/usr/sap/trans` o no tienen los permisos necesarios.

Indicadores clave

Los siguientes son algunos consejos clave a la hora de importar transportes AWS del SDK.

- Cada transporte se entrega como `Knnnnnn.AWS` y `Rnnnnnn.AWS`
  - `Knnnnnn.AWS` debe copiarse en `/usr/sap/trans/cofiles`
  - `Rnnnnnn.AWS` debe copiarse en `/usr/sap/trans/data`.
- Al importar transportes, debe seleccionar Ignorar versión de componentes no válidos que se encuentra en Importar solicitud de transporte > Opciones > Opciones de importación.
- Todos los transportes deseados se pueden importar simultáneamente.
- Si se importan los transportes por separado, primero se debe importar el transporte `core`.
- El nivel de versión de todos los transportes debe ser idéntico.

Tiempo de importación

AWS La importación de los transportes del SDK puede tardar varios minutos. Los transportes se realizan correctamente si el STMS muestra una luz verde (`RC=0`) o amarilla (`RC=4`).

- Una luz roja (`RC=8`) indica que la importación tuvo un error de sintaxis.
  - Seleccione Solicitud → Mostrar → Registros para examinar el error de importación.
  - Durante la importación, si se produce un error debido a la falta de una interfaz `IF_SYSTEM_UUID_RFC4122_STATIC`, asegúrese de que la nota de SAP 2619546 se aplique al sistema. Para obtener más información, consulte Notas.
  - Si se desconoce la causa del error, póngase en contacto con Soporte.
- Un rayo rojo (`RC=12`) indica que los archivos de transporte no se han cargado correctamente en `/usr/sap/trans` o no tienen los permisos necesarios.

Namespace

El SDK de SAP ABAP usa el espacio de nombres `/AWS1/` y no modifica los objetos de SAP ni ningún otro objeto del sistema, con la siguiente excepción.

- AWS `auth`los objetos están en una clase de objetos de autenticación. Las clases de objeto de autenticación están limitadas a cuatro caracteres y no admiten espacios de nombres. El SDK de SAP ABAP usa la clase de objeto de autenticación es `YAW1`. Si ya tiene una clase de objeto de

autenticación `YAW1` en transacción`SU21`, póngase en contacto con nosotros Soporte antes de la instalación.

Instalación AWS del SDK para SAP ABAP - edición BTP

La edición BTP se encuentra en versión preliminar para desarrolladores y se puede instalar uniéndose a la versión preliminar. Para instalar el SDK, rellene el formulario de participación en AWS SDK for SAP ABAP: versión preliminar para desarrolladores de la edición BTP.

Antes de instalar el SDK para SAP ABAP (edición BTP), asegúrese de cumplir los requisitos previos necesarios. Para obtener más información, consulte SAP Landscape Portal y SAP Credential Store.

Temas

- Instale el SDK para SAP ABAP: edición BTP
- Módulos
- SDK de parches para SAP ABAP: edición BTP

Instale el SDK para SAP ABAP: edición BTP

1. Vaya a su instancia de SAP Landscape Portal e inicie la aplicación Deploy Product fiori.
2. En Productos, **`/AWS1/SDK_OMNI`**seleccione en Productos de socios.

   Póngase en contacto con nosotros Soporte si no aparece *`/AWS1/SDK_OMNI`* después de haber sido aceptado en la versión preliminar para desarrolladores.
3. En la versión de destino, elija la versión del SDK para SAP ABAP (edición BTP) que desee instalar en su sistema.
4. En Sistemas disponibles, marque las casillas de verificación de todos aquellos SIDs en los que desee instalar el SDK.
5. Seleccione Implementar, introduzca los detalles de la programación y seleccione Programar. Puede supervisar el progreso en el estado de despliegue de la versión del producto.

   La instalación puede tardar entre 30 y 45 minutos e incluye el tiempo de inactividad del sistema. Para obtener más información, consulte Implementar el producto.

Módulos

Los siguientes módulos se incluyen en la versión preliminar para desarrolladores del AWS SDK para SAP ABAP: edición BTP.

- Amazon API Gateway [agw]
- Amazon Athena [] ath
- Tiempo de ejecución de Amazon Bedrock [] 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
- Servicio de almacenamiento simple de Amazon [s3]
- AWS Systems Manager [ssm]
- Amazon Textract [] tex
- Amazon Transcribe [] tnb
- Amazon Translate [xl8]
- AWS CloudTrail [trl]
- AWS IoT [iot]
- AWS KMS [kms]
- AWS Lambda [lmd]
- AWS Secrets Manager [smr]
- AWS Security Token Service [sts]
- AWS Transfer Family [trn]
- Funciones de IAM en cualquier lugar [] rla
- API de datos de Amazon Redshift [] rsd

SDK de parches para SAP ABAP: edición BTP

El proceso de aplicación de parches del SDK para SAP ABAP, edición BTP, es similar al proceso de instalación. Si instala el SDK en un sistema que ya tiene instalada una versión anterior, el SDK se parcheará según la nueva versión que elija.

Configuración AWS SDK para SAP ABAP

Antes de usarlo AWS SDK para SAP ABAP, debe configurar el SDK con los ajustes técnicos y funcionales necesarios para las operaciones del SDK. Algunas configuraciones son transportables y otras son configuraciones de tiempo de ejecución. Muchas de las configuraciones son directamente análogas a las configuraciones definidas en `.INI` los archivos para otras SDKs.

Las configuraciones del SDK, excepto la configuración de tiempo de ejecución, se deben completar en el entorno de desarrollo. Puede transportar las configuraciones a control de calidad y producción siguiendo las reglas habituales de transporte y control de cambios. No se recomienda la configuración transportable para entornos de producción.

Si no tiene permisos para configurar el AWS SDK, consulte Autorizaciones de SAP.

Configurando AWS SDK para SAP ABAP

Para ejecutar la transacción de configuración, escriba `/n/AWS1/IMG` en la barra de comandos de la SAPGUI.

Configuración AWS del SDK para SAP ABAP: edición BTP

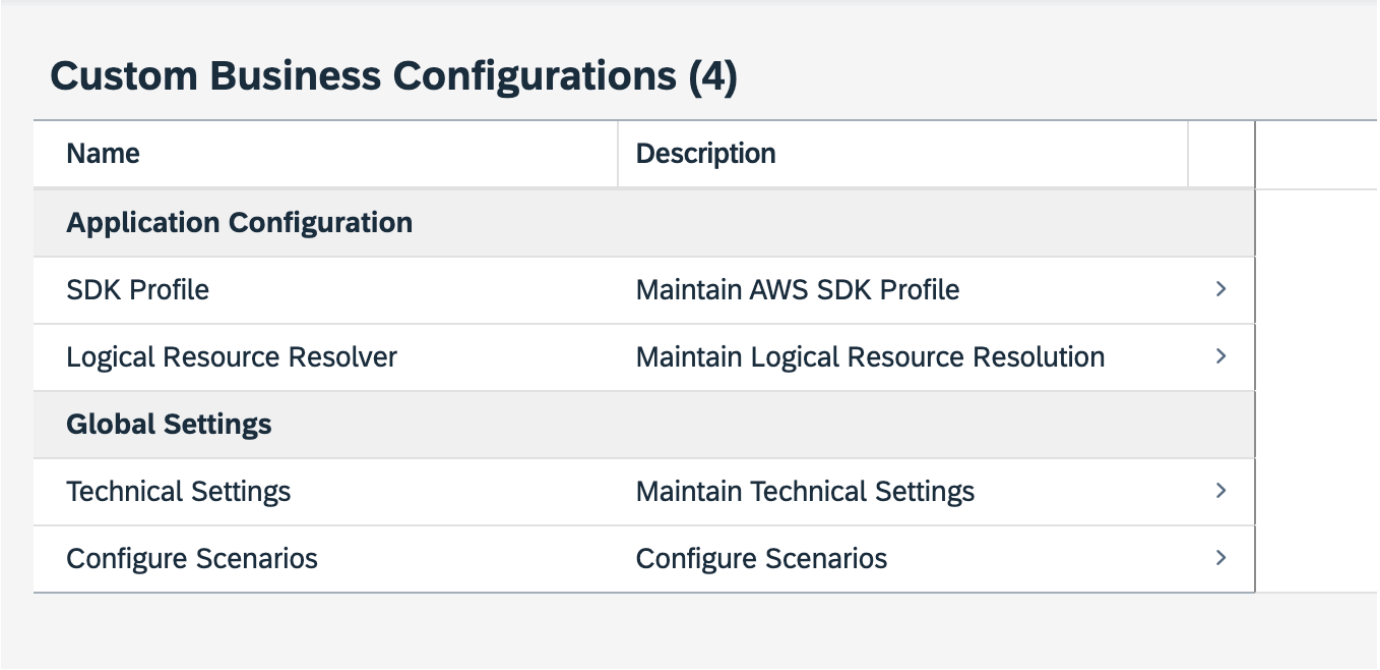
Siga los siguientes pasos para configurar el SDK para SAP ABAP: edición BTP.

1. Abra su entorno ABAP en un navegador web.
2. Navegue hasta la aplicación de configuraciones empresariales personalizadas.

Para crear una solicitud de personalización mediante la aplicación Export Customizing Transports, consulte Trabajar en la aplicación Export Customizing Transports: Create Request.

En la aplicación de configuración empresarial personalizada, puede agrupar las configuraciones según el tipo de configuración del SDK. Siga los siguientes pasos para agrupar las configuraciones.

1. Abra su entorno ABAP en un navegador web y navegue hasta la aplicación Custom Business Configurations.
2. Seleccione Configuración > Grupo y elija Grupo de configuración en la lista desplegable. Seleccione OK (Aceptar).
3. Las configuraciones ahora están disponibles en una estructura jerárquica, como se muestra en la imagen. Para guardar la vista, consulte Vistas (gestión de variantes) - Componentes.



Esta sección abarca los siguientes temas.

Temas

- Configuración global
- Configuración de aplicaciones
- Configuración del tiempo de ejecución
- Escenarios de conectividad avanzada
- Configuración del proveedor de servicios
- Temas de actualización, rastreo y telemetría para AWS SDK para SAP ABAP

Configuración global

Utilice `/n/AWS1/IMG` IMG transacation para el AWS SDK para SAP ABAP y la aplicación de configuración empresarial personalizada para el AWS SDK para SAP ABAP: edición BTP para

configurar los ajustes globales. En este tema se utilizan IMG y Custom Business Configuration indistintamente.

Esta sección abarca los siguientes temas.

Temas

- Configuración técnica
- Configuración de escenarios

Configuración técnica

La configuración global de la transacción `/AWS1/IMG` afecta al comportamiento de todo el SDK. Por lo general, estos ajustes los configura un administrador de Basis. Puede establecer estos valores con las siguientes configuraciones recomendadas.

- Seleccione Nuevas entradas.
  - Regionalización de S3: acceda a los buckets us-east-1 mediante s3.amazonaws.com.
  - Regionalización de STS: acceda a STS mediante un punto de conexión global.
  - Deshabilitar EC2 los metadatos: deje este campo en blanco. Este campo es de solo lectura en la edición BTP y está configurado en «Sí» de forma predeterminada.
  - Modo de punto final de metadatos: utilice el punto final de metadatos. IPv4 Este campo es de solo lectura en la edición BTP y se actualiza automáticamente.
  - URL del punto de conexión de metadatos: deje este campo en blanco. Este campo es de solo lectura en la edición BTP.
- Seleccione Guardar.

Configuración de escenarios

Los escenarios permiten al SDK de AWS cambiar la configuración de manera más eficiente al hacer pruebas de recuperación de desastres entre regiones o en escenarios de pruebas de recuperación de desastres. Es posible que no necesite esta característica y, en su lugar, solo necesite configurar el siguiente escenario PREDETERMINADO.

- Seleccione Nuevas entradas.
  - ID de escenario:`DEFAULT`
  - Descripción del escenario: escenario predeterminado

- Seleccione Guardar.

Si tiene una configuración de recuperación de desastres entre varias regiones u otro tipo de casuística especial que requiera un cambio rápido de configuración, puede configurar varios escenarios.

- `DEFAULT`: operación estándar.
- `DR`: configuración especial si un desastre requiere mover todo el sistema a otra región.
- `DR_TEST`: configuración especial para simular un desastre, por ejemplo, en un clon temporal de producción.

Configuración de aplicaciones

La configuración del SDK de SAP ABAP es similar a la configuración de otras aplicaciones basadas en ABAP. Está organizado en diferentes perfiles para agrupar las configuraciones de varios escenarios. Un perfil del SDK para ABAP define la configuración necesaria para un escenario de aplicación específico. Por ejemplo, si las transacciones `ZVA01``ZVA02`, y `ZVA03` están relacionadas con la facturación, se mejoran y se ejecutan Servicios de AWS, como Amazon S3, y AWS Lambda, entonces Amazon SageMaker AI, se `ZINVOICE` puede crear un perfil de SDK llamado. Este perfil puede agrupar la configuración técnica, las autorizaciones de SAP y las asignaciones de roles de IAM para la funcionalidad relacionada con la facturación.

Utilice la `/n/AWS1/IMG` transacación para el AWS SDK para SAP ABAP y la aplicación de configuración empresarial personalizada para el AWS SDK para SAP ABAP (edición BTP) para configurar los ajustes globales. En este tema se utilizan IMG y Custom Business Configuration de forma intercambiable.

Temas

- Perfil del SDK
- Solucionador lógico de recursos
- Ejemplo

Perfil del SDK

Un perfil del SDK de ABAP define lo siguiente para cada SID y cliente.

> **Note**
>
> El cliente siempre está al 100% en un entorno SAP BTP y ABAP.

- La AWS región predeterminada para todas las llamadas a la API. Por ejemplo, si sus sistemas SAP se ejecutan en la `us-east-1` región, es probable que sus otros AWS recursos también estén en la misma región, y esta debería ser su región predeterminada. Su código ABAP puede anular la región predeterminada.
- Método de autenticación
  - Para los sistemas SAP que se ejecutan en Amazon EC2, recomendamos encarecidamente elegir los metadatos de los roles de instancia para aprovechar las credenciales efímeras que rotan automáticamente.
  - Para los sistemas SAP que se ejecutan en las instalaciones o en otra nube, debe elegir las credenciales del almacenamiento de SSF.
  - Para los sistemas ABAP que se ejecutan en SAP BTP, debe elegir las credenciales de la tienda de credenciales de SAP. Para obtener más información, consulte Uso del almacén de credenciales de SAP para la autenticación.
  - Para escenarios de encadenamiento de roles entre cuentas, elija el perfil de origen y especifique un ID de perfil de origen. Esto permite la resolución y ejecución automáticas de las cadenas de asunción de roles. Para obtener más información, consulte Uso del perfil de origen para el acceso entre cuentas.
- Una asignación de los roles de IAM lógicos a los roles de IAM.
  - Esta asignación está ordenada por orden de prioridad descendente.
  - Se seleccionará automáticamente para el usuario un rol de IAM de máxima prioridad para el que está autorizado un usuario en un rol de PFCG.
- Un mapeo opcional de los servicios a puntos finales personalizados. Esta configuración se describe en Escenarios de conectividad avanzada

> **Note**
>
> Las funciones de PFGC se denominan funciones de negocio en el entorno SAP BTP y ABAP.

Cuando un programa de ABAP desee conectarse a un Servicio de AWS, especificará un perfil del SDK de ABAP que incluye los ajustes necesarios. Se realizará una `AUTHORIZATION-CHECK` para confirmar que el usuario tiene permisos para acceder al perfil del SDK. El administrador de seguridad de SAP puede definir un rol de PFCG que le otorgue acceso a los usuarios correspondientes.

Solucionador lógico de recursos

El solucionador lógico de recursos permite almacenar los nombres de los recursos en un lugar estándar. Viene con el SDK de SAP ABAP. Su acción es similar a la forma en que la transacción de `FILE` asigna los nombres de los archivos lógicos a los nombres de los archivos físicos.

Un recurso lógico define el concepto de AWS recurso, como el depósito de Amazon S3 que contiene nuestras facturas. Este recurso lógico, por ejemplo, puede llamarse `ZINVOICES_OUTBOUND` y asignarse a un nombre de bucket físico diferente, en función de si el sistema SAP es de desarrollo, de control de calidad o de producción.

El SDK de SAP ABAP está configurado de tal manera que un sistema de control de calidad convierte los recursos lógicos en recursos físicos de control de calidad, incluso después de una actualización del sistema desde producción. Las asignaciones de recursos para TODOS los sistemas se definen en su sistema SAP de desarrollo y se transportan hacia adelante. Este enfoque es diferente de la configuración habitual en los sistemas SAP, en la que la asignación se gestiona como datos maestros y se establece en cada sistema. La ventaja del solucionador lógico de recursos que ofrece el SDK de SAP ABAP es que las probabilidades de que se produzca un error de transporte tras una actualización del sistema son prácticamente nulas.

Ejemplo

Hay cuatro buckets de Amazon S3 independientes, uno para desarrollo, producción y control de calidad, así como un segundo bucket de control de calidad para pruebas de regresión.

Cuando el SDK resuelve un recurso lógico, como `ZINVOICE_OUTBOUND`, en un recurso físico, comprueba `SY-SYSID` y `SY-MANDT` para preguntar: ¿En qué SID y cliente me ejecuto? y selecciona automáticamente el recurso físico correcto.

Si es necesario cambiar la asignación de un recurso en producción, debe cambiarla en el `IMG` del sistema de desarrollo y transportarlo hacia adelante. Esto garantiza que la reasignación de AWS recursos a un sistema SAP esté sujeta al control de cambios, como ocurre con cualquier otro transporte.

> **Note**
>
> Como la configuración del SDK depende del cliente, la reasignación de recursos se transporta en una solicitud de personalización y el transporte debe importarse a cada cliente.

Configuración del tiempo de ejecución

Esta sección abarca los siguientes temas.

> **Note**
>
> Estos ajustes no se pueden transportar y son locales para cada sistema SAP.

Temas

- Registro y rastreo
- OPT-IN: telemetría mejorada
- Escenario activo

Registro y rastreo

Puede activar un rastreo para depuración. Se recomienda mantener el nivel de rastreo en Sin rastreo, a menos que se diagnostique un problema técnico. Para obtener más información, consulte funcionamiento seguro.

Esta configuración no se aplica al SDK para SAP ABAP - edición BTP.

OPT-IN: telemetría mejorada

Todos SDKs envían información de telemetría a para AWS fines de soporte. Puede optar por activar la telemetría mejorada. Esto resulta especialmente útil cuando contactas Soporte para identificar el origen de una llamada a la API en particular. Para obtener más información, consulte Rastreo y Telemetría.

Esta configuración no se aplica al SDK para SAP ABAP, edición BTP.

Escenario activo

Active su escenario `DEFAULT` en esta transacción. Esta activación solo se requiere una vez para cada sistema y no debe cambiarse a menos que el sistema esté en proceso de recuperación de desastres en varias regiones. En una configuración multirregional, puede usar esta configuración para cambiar su sistema SAP a un entorno de recuperación de desastres o a escenarios de prueba de recuperación de desastres.

Escenarios de conectividad avanzada

AWS SDK para SAP ABAP consume Servicios de AWS realizando llamadas HTTPS a los AWS puntos finales. En general, los puntos de conexión de AWS son accesibles a través de Internet. Un sistema SAP debe poder conectarse a Internet para establecer estas conexiones salientes. El SDK de SAP ABAP nunca requiere una conexión entrante desde Internet al sistema SAP.

Los siguientes escenarios ofrecen diferentes formas de establecer la conexión saliente.

Escenarios

- Conexión a través de un servidor proxy
- Conexión a través de un firewall de inspección de paquetes
- Puntos de conexión de la puerta de enlace
- Puntos de conexión personalizados
- Enrutamiento avanzado
- Per-service anulación del servidor proxy
- Acceso a puntos de conexión en varias regiones

Conexión a través de un servidor proxy

Para establecer una conexión a través de un servidor proxy, siga estos pasos.

1. En el SDK, vaya a Transacción **`SICF`**.
2. Elija Ejecutar.
3. En el menú, seleccione Cliente > Servidor proxy.
4. Establezca la configuración del proxy como Activa.

5. En el campo para No hay proxy para las siguientes direcciones, muestre las excepciones separadas por punto y coma.
6. En los campos Protocolo HTTP y Protocolo HTTPS, especifique los detalles de conexión del servidor proxy.

El SDK desconoce el servidor proxy y no requiere ninguna configuración para utilizar la configuración del servidor proxy del sistema SAP.

> **Note**
>
> Si utiliza la autenticación de metadatos de la instancia de Amazon EC2, el sistema SAP no podrá utilizar el servidor proxy para acceder a los metadatos de la instancia local en `http://169.254.169.254`. Debe incluir `169.254.169.254` en el campo para No hay proxy para las siguientes direcciones.

> **Note**
>
> Puede anular el comportamiento del servidor proxy por servicio en la sección Enrutamiento avanzado de. `/AWS1/IMG` Para obtener más información, consulte the section called “Per-service anulación del servidor proxy”.

Conexión a través de un firewall de inspección de paquetes

Puede configurar un firewall de inspección de paquetes para la conexión saliente. Estos firewalls descifran el tráfico SSL y, a continuación, lo vuelven a cifrar antes de pasarlo al punto de conexión. Esta configuración normalmente requiere que el firewall emita sus propios certificados para el sistema SAP que consume un Servicio de AWS. Debe instalar el certificado de CA de su firewall en `STRUST`. Para obtener más información, consulte Conectividad HTTPS.

Puntos de conexión de la puerta de enlace

Algunos Servicios de AWS ofrecen puntos finales de puerta de enlace para proporcionar a una VPC un acceso de alto rendimiento sin Internet. Estos puntos de conexión son transparentes para el SDK de SAP ABAP y no requieren ninguna configuración.

Para obtener más información, consulte Puntos de conexión de puerta de enlace.

Puntos de conexión personalizados

Si necesita anular la resolución de punto de conexión predeterminado por un punto de conexión personalizado, puede usar un punto de conexión de interfaz para proporcionar a su VPC acceso de alto rendimiento sin Internet. Para obtener más información, consulte Configurar un punto de conexión de interfaz.

Cuando no se utiliza un DNS privado, estos puntos de conexión tienen sus propias direcciones DNS y un programa ABAP debe anular de forma explícita la lógica habitual de resolución de puntos de conexión. Para obtener más información, consulte AWS re:Post : ¿Por qué no puedo resolver los nombres de dominio de servicio para un punto final de VPC de interfaz?

En el siguiente ejemplo, se crea un punto final de interfaz para AWS STS y Amazon Translate. El sistema SAP no utiliza un DNS privado y llama a los servicios con un punto de conexión personalizado. Los recursos lógicos definidos en `/AWS1/IMG` representan las direcciones de punto de conexión de la interfaz física, por ejemplo, `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`. Esto evita la codificación rígida del DNS en código.

En el código siguiente, los recursos lógicos en `/AWS1/IMG` se resuelven primero en los nombres de los puntos de conexión físicos. A continuación, se proporcionan a los métodos de fábrica de la clase de AWS sesión (que se utiliza AWS STS para asumir una función de IAM) y a la clase de API de traducción.

```
" 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...
```

Como se muestra en el ejemplo, cualquier llamada a un método en `go_xl8` va al punto de conexión `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`. También es posible definir el punto final personalizado de enrutamiento en la configuración de IMG en lugar de hacerlo en el código, como se muestra en la siguiente sección.

Enrutamiento avanzado

En la sección anterior, mostramos cómo se puede especificar un punto final personalizado en el `iv_custom_endpoint` argumento de los métodos de fábrica para los módulos del SDK. A medida que aumenta el número de programas ABAP que utilizan el SDK, esto puede resultar difícil de gestionar. Es posible configurar un mapeo desde un punto final Servicio de AWS a uno personalizado en el perfil del SDK. Para cada SID, cliente y escenario, la abreviatura de tres letras (TLA) del servicio se puede asignar a la URL de un punto final:

TLA	URL de punto final personalizada
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

Con esta configuración, no es necesario especificar las llamadas a `iv_custom_endpoint` los métodos de fábrica. El punto final personalizado se selecciona automáticamente en la tabla de configuración. La configuración es específica del perfil del SDK, por lo que puede crear varios perfiles con diferentes enrutamientos en función de sus necesidades. Al igual que con otras configuraciones de perfiles del SDK, el enrutamiento es específico del SID y del cliente, por lo que se puede definir un enrutamiento independiente para diferentes sistemas.

Per-service anulación del servidor proxy

De forma predeterminada, el SDK usa la configuración del servidor proxy configurada en Transaction `SICF` (consultethe section called “Conexión a través de un servidor proxy”). La configuración del proxy se `SICF` aplica globalmente a todas las conexiones HTTP salientes del sistema SAP. En algunos entornos, es posible que necesite un control más preciso sobre cuáles Servicios de AWS utilizan el servidor proxy y cuáles se conectan directamente.

En la sección Enrutamiento avanzado de`/AWS1/IMG`, puede configurar la opción Usar servidor proxy para cada servicio. Esta configuración controla si el SDK enruta las solicitudes de ese servicio a través del servidor proxy definido en`SICF`, independientemente de la configuración global de activación o filtro del proxy.

Están disponibles los siguientes valores:

- Predeterminado: utilice el comportamiento del proxy en el que esté configurado`SICF`. Si el proxy está activo y el filtro no excluye el punto final, se utiliza el proxy. Este es el comportamiento predeterminado.
- Siempre: dirija siempre las solicitudes de este servicio a través del servidor proxy definido en`SICF`, independientemente de la configuración global de activación o filtro.
- Nunca: nunca dirija las solicitudes de este servicio a través del servidor proxy, independientemente de la configuración global de activación o filtro. Úselo cuando se pueda acceder a un punto final de servicio directamente sin un proxy, por ejemplo, cuando se utiliza un punto final de VPC.

Por ejemplo, puede configurar Amazon S3 para que nunca utilice el proxy (ya que se accede AWS STS a ellos a través de los puntos de enlace de la puerta de enlace de VPC), mientras que Amazon SNS siempre usa el proxy porque solo se puede acceder a él a través de Internet.

TLA	Utilice un servidor proxy
EC2	Predeterminado
S3	Nunca
SNS	Siempre
STS	Nunca

Esta configuración se define por perfil, SID, cliente y escenario del SDK. Puede crear varios perfiles con diferentes enrutamientos de proxy para adaptarse a diferentes entornos o casos de uso.

> **Note**
>
> El host y el puerto del servidor proxy siempre se definen en la pestaña Protocolo HTTPS de la configuración del proxy en Transaction`SICF`. La anulación por servicio solo controla si se usa el proxy, no qué servidor proxy se usa.

Acceso a puntos de conexión en varias regiones

AWS El punto final se determina automáticamente a partir del valor predeterminado Región de AWS definido en el perfil del SDK. También puede especificar una región mediante programación, anulando la región predeterminada. Esto se puede anular en el método `CREATE( )` de fábrica o, posteriormente, con el objeto de configuración del SDK. Para obtener más información, consulte Configuración programática.

En el siguiente ejemplo, se utiliza el método `CREATE( )` de fábrica para establecer la región y mostrar las colas de Amazon SQS en las regiones `us-east-1` y `us-west-2`.

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

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

```
).

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

Configuración del proveedor de servicios

Los administradores de Basis a veces necesitan controlar determinadas funciones del SDK en todo el sistema, desde el cliente`000`. Este es un escenario común para los proveedores de alojamiento y servicios que operan los sistemas por AWS cuenta propia en nombre de sus clientes. AWS El SDK para SAP ABAP admite la configuración del proveedor de servicios. Estos ajustes se configuran en el cliente `000` y afectan al SDK en todos los clientes. La configuración del proveedor de servicios no se admite en la edición BTP del SDK para SAP ABAP.

Los ajustes del proveedor de servicios se configuran en la transacción `/AWS1/IMG` y se deben configurar en el cliente. `000` Se omite la configuración del proveedor de servicios en otros clientes. La configuración del cliente se `000` aplica a todos los clientes y reemplaza a las demás `IMG` configuraciones en caso de conflicto.

Siga los pasos siguientes para configurar los ajustes del proveedor de servicios en el cliente`000`.

1. Amplíe la rama de configuración del proveedor de servicios en la transacción`/AWS1/IMG`.
2. Elija Service Provider Guardrails
3. Seleccione Nuevas entradas y ajuste la configuración en función de los requisitos de su empresa.

    - Inhabilitar EC2 los metadatos: impide que el SDK acceda a los metadatos de la EC2 instancia en todos los clientes, incluso si el perfil del SDK está configurado para autenticarse mediante los metadatos de la EC2 instancia. El SDK genera una excepción si un programa ABAP intenta acceder a los metadatos de la instancia mediante el SDK.
4. Seleccione Guardar.

Temas de actualización, rastreo y telemetría para AWS SDK para SAP ABAP

Esta sección abarca los siguientes temas.

Temas

- Actualización del sistema SAP
- Rastreo
- Telemetría

Actualización del sistema SAP

Tras una actualización del sistema, el principal desafío para un administrador de Basis es garantizar que los distintos sistemas no accedan a los recursos de los demás. Por ejemplo, es posible que desee asegurarse de que su sistema SAP de control de calidad no acceda a los recursos, como un bucket de S3, de su entorno de producción.

El SDK de SAP ABAP ofrece un enfoque de recursos lógicos centrado en la seguridad para hacer frente a este desafío. Un analista de negocios puede seguir los pasos que se indican a continuación:

1. Defina un recurso lógico, como, por ejemplo, `ZINVOICE_OUTBOUND`.
2. Asigne todos los sistemas y clientes del sistema de desarrollo.
3. Transporte la configuración de TODOS los sistemas al entorno de producción.

Pasos básicos después de una actualización

1. Comprobar la autenticación

- Si el sistema utiliza la autenticación con clave de acceso secreta, las credenciales cifradas con SSF no serán válidas porque están almacenadas en los datos maestros. Se deben volver a introducir las credenciales, lo que puede requerir la regeneración de una nueva clave de acceso secreta. https://console.aws.amazon.com/iam/
- Si el sistema se autentica con los metadatos de la EC2 instancia, no es necesario realizar ningún paso.

Comprobar la configuración de rastreo

- En `/AWS1/IMG`, asegúrese de que la configuración de rastreo es la que desea. Estos ajustes no son transportables.

Rastreo

La salida de rastreo se controla en la configuración del tiempo de ejecución de IMG.

Los niveles de rastreo que puede utilizar son:

- Sin rastreo
- Rastrear llamadas a la API
- Rastrear llamadas a la API y carga útil

  Esta opción contiene información de carga útil no cifrada.
- Rastrear llamadas a la API, carga útil y transformación XML interna

  Esta opción contiene información de carga útil no cifrada.

Si el rastreo de la API está activado, los rastreos se escriben en `DIR_WORK` en el archivo `aws1_trace-`*`YYYY-MM-DD`*`.log`.

Si el rastreo de carga útil está activado adicionalmente, se crean archivos adicionales con el título `aws1_payload_*` para cada componente de llamada y carga útil. La longitud del rastreo de la carga útil se puede limitar y el límite de longitud se aplica a cada error de rastreo de carga útil individual.

Los rastreos de carga útil están destinados principalmente a recopilar la información que se proporcionará Soporte en caso de un error de serialización. Le recomendamos que elija Sin rastreo, a menos que esté intentando diagnosticar un error del SDK.

> **Note**
>
> Los rastreos de carga útil pueden contener información empresarial no cifrada. Recomendamos activar estos rastreos solo cuando AWS Support lo solicite para ayudarlo a solucionar problemas. Puede desactivar estos rastreos una vez resueltos los problemas. Los rastreos no se eliminan automáticamente y el administrador del sistema debe eliminarlos cuando ya no sean necesarios.

Esta configuración no se aplica al SDK para SAP ABAP, edición BTP.

Telemetría

SDKs enviar información de telemetría a. Soporte El SDK de SAP ABAP recopila la siguiente información:

- Versión del sistema operativo y nivel de parche
- Versión y nivel de parche de `SAP_BASIS`
- Versión del kernel y nivel de parche de SAP

Puede optar por enviar la siguiente información a Soporte.

- SID de SAP y nombre de la instancia (`host_sid_nn`)
- Cliente SAP (`SY-MANDT`)
- Código de transacción (`SY-TCODE`) e informe (`SY-REPID`)

La información adicional Soporte permite ayudarlo mejor. Soporte puede detectar el motivo por el que se ha realizado una determinada llamada a la API y, además, puede encontrar la transacción correspondiente en un sistema SAP.

La telemetría se limita a las versiones SDK y API del SDK para SAP ABAP (edición BTP).

Uso AWS SDK para SAP ABAP

El SDK de SAP ABAP tiene dos componentes principales.

- Tiempo de ejecución del SDK (paquete `/AWS1/RT`): conjunto de objetos que sustentan las funciones de seguridad, autenticación, rastreo, configuración, conversión de datos y otras funciones entre las API. Los módulos de API para Amazon S3 AWS STS, IAM Roles Anywhere y Secrets Manager son obligatorios.
- APIs (paquete `/AWS1/API` y sus subpaquetes): un subpaquete para cada API en el que los objetos de cada API son completamente independientes entre sí, lo que garantiza que un cambio en una API no interrumpa otra API. Para ver una lista completa de ellas AWS SDK para SAP ABAP APIs, consulta la Guía de referencia AWS SDK para SAP ABAP de las API.

Esta sección abarca los siguientes temas.

Temas

- Representación de datos en ABAP
- Programa de ejemplo de Amazon S3
- Conceptos del SDK de SAP ABAP
- AWS SDK para SAP ABAP features
- Creación de productos con SDK
- Personalice las solicitudes HTTP para AWS
- Limitaciones

Representación de datos en ABAP

Esta sección abarca los siguientes temas.

Temas

- Tipos de datos
- AWS tipos de datos

Tipos de datos

Servicios de AWS tienen un conjunto estándar de tipos de datos que deben asignarse a los tipos de datos ABAP. Consulte la tabla siguiente para obtener más detalles.

AWS tipo de datos	Tipo de datos de ABAP	Comentarios
booleano	C	Cualquier carácter "X" y " "
Cadena	STRING	
Byte	INT2	INT2 tiene un rango mayor que 0-255. La mayoría Servicios de AWS truncará los desbordamientos, pero este comportamiento no está definido formalmente.
Short	INT2	
Entero	INT4	
Largo	DEC19	INT8 no estará disponible hasta el ABAP 750. DEC19 se utiliza para garantizar la compatibilidad y la coherenci a en todas las plataformas ABAP compatibles.
Blob	XSTRING	Representa datos binarios
Flotante	STRING	Si bien ABAP es compatibl e DECFLOATs, no puede representar valores como NaN, Infinity y -Infinity. AWS El SDK los represent a internamente como y STRINGs los convierte DECFLOAT16 en tiempo de
Doble	STRING	

AWS tipo de datos	Tipo de datos de ABAP	Comentarios
		ejecución. Si NaN, Infinity o +Infinity están represent ados, el desarrollador puede procesarlos en respuesta a un conjunto especial de excepciones o asignaciones.
bigInteger	STRING	Estos valores representan números de longitud infinita que no se pueden represent ar en ABAP y se STRINGs utilizan en lugar de BigInteger.
bigDecimal	STRING	
Timestamp	`TZNTSTMPS`	`TZNTSTMPS` permite el procesamiento con funciones de marca temporal nativas de ABAP.

Servicios de AWS también devuelven los siguientes tipos de datos agregados.

AWS tipo de datos	Tipo de datos de ABAP	Comentarios
Estructura	Clase	
Unión	Clase	Una unión es lo mismo que una estructura, con la salvedad de que una unión nunca tendrá más de un conjunto de campos. Todos los demás campos se establecerán en Sin valor.
Matriz	TABLA ESTÁNDAR	

AWS tipo de datos	Tipo de datos de ABAP	Comentarios
Hash	TABLA CON HASH	La tabla codificada solo tendrá dos columnas: una CLAVE (cadena) y un VALOR (clase).

AWS tipos de datos

Los siguientes enfoques se han integrado para dar soporte Servicios de AWS al ABAP.

- Algunos tipos AWS de datos no se pueden representar en ABAP. Por ejemplo, el tipo de datos de `float` en ABAP no admite los valores `NaN`, `Infinity` o `-Infinity`. Por lo tanto, el tipo de datos de `float` se representa como `STRING` y se traduce a `DECFLOAT16` en tiempo de ejecución.
- AWS los datos se representan en el cable como JSON o XML y los valores son opcionales. Por ejemplo, consulta los siguientes ejemplos devueltos por un Servicio de AWS archivo JSON.

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

Si Ana no tiene un segundo nombre, el servicio devuelve el siguiente resultado.

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

ABAP no distingue entre una cadena de longitud 0 y una cadena que no tiene ningún valor. Otros lenguajes pueden asignar un valor NULL a la cadena o incluir la cadena en un constructo (por ejemplo, el contenedor `Optional<>` de Java). ABAP no los admite. Por lo tanto, el SDK de SAP ABAP facilita la distinción de valores al proporcionar variantes del método getter.

Programa de ejemplo de Amazon S3

En esta sección, se muestra un ejemplo de programa sencillo para mostrar el contenido de un bucket de Amazon S3 mediante una llamada a `ListObjectsV2`.

Temas

- Requisitos previos
- Código
- Secciones de código

Requisitos previos

Para ejecutar este programa de ejemplo, debe cumplir los siguientes requisitos previos.

- Tiene un bucket de Amazon S3. En este tutorial, el bucket se denomina `demo-invoices.customer.com`.
- Transacción `/AWS1/IMG`:
  - Tiene un perfil del SDK definido denominado `DEMO_S3`.
    - En el perfil del SDK, El rol de IAM lógico `TESTUSER` debe asignarse a un rol de IAM, por ejemplo, `arn:aws:iam::111122223333:role/SapDemoFinance`, que otorga a `s3:ListBucket` permiso para mostrar el contenido de su bucket de Amazon S3.
  - Tiene un recurso lógico llamado `DEMO_BUCKET` que se asigna a su bucket de Amazon S3 con el SID y el cliente de su sistema SAP.
- Su usuario tiene un rol de PFCG que:
  - Autoriza al usuario acceder al perfil del SDK de `DEMO_S3` a través del objeto de autenticación: `/AWS1/SESS`.
  - Autoriza al usuario a `TESTUSER` acceder al rol de IAM lógico mediante el objeto de autenticación: `/AWS1/LROL`.
- Su sistema SAP puede autenticarse AWS mediante el método definido en el perfil del SDK.
- El perfil de su EC2 instancia de Amazon otorga a su sistema SAP los derechos para `sts:assumeRole` ocupar el rol de IAM `arn:aws:iam::111122223333:role/SapDemoFinance` asignado en el perfil del SDK.

Código

El siguiente bloque de código muestra el aspecto que tendrá su código.

```
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.
```

Secciones de código

La siguiente es una revisión del código en secciones.

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

Los usuarios no pueden especificar un nombre de bucket físico. Especifican un bucket lógico y los administradores del sistema (en concreto, el analista empresarial), en coordinación con el administrador de AWS , asignan los buckets lógicos a los buckets físicos en /AWS1/IMG. En la mayoría de los escenarios empresariales, el usuario no tiene la oportunidad de elegir el bucket lógico: el ID del recurso lógico está codificado de forma rígida o configurado en una tabla de configuración personalizada.

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

Esta línea establece una sesión de seguridad y declara que este programa ABAP espera usar el perfil del SDK de DEMO_S3. Esta llamada es la conexión a la configuración del SDK y extrae la región predeterminada, la configuración de autenticación y el rol de IAM deseado. Se realiza automáticamente una llamada a AUTHORIZATION-CHECK para garantizar que se satisfaga el objeto de autorización de /AWS1/SESS. Además, se realizarán llamadas de AUTHORIZATION-CHECK para determinar cuál es el rol de IAM lógico más potente (número de secuencia inferior) para el que está autorizado el usuario, en función del objeto de autorización de /AWS1/LROL. El SDK asumirá que el rol de IAM está asignado al rol de IAM lógico para el SID y el cliente. A continuación, el objeto de sesión activa el rastreo en función de la configuración de rastreo en la IMG.

Si el usuario no está autorizado para el perfil del SDK solicitado ni para ningún rol de IAM lógico disponible, se generará una excepción.

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

Esta línea convierte el recurso lógico en un nombre de bucket físico. Si el recurso lógico no se puede resolver porque la configuración no tiene ninguna asignación para esta combinación de SID/cliente, se generará una excepción.

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

Esta línea crea un objeto de API para Amazon S3 mediante el método create() de /aws1/cl_s3_factory. El objeto devuelto es de tipo /aws1/if_s3, que es la interfaz de una API de Amazon S3. Se debe crear un objeto de API independiente para cada servicio. Por ejemplo, si un programa ABAP consume Amazon S3 y DynamoDB AWS Lambda, crea objetos /aws1/cl_s3_factory de API a partir de, y. /aws1/cl_lmd_factory /aws1/cl_dyn_factory

El constructor tiene algunos parámetros opcionales que le permiten especificar la región si desea anular la región configurada de forma predeterminada `IMG`. De esta forma, puede haber dos instancias de `/aws1/if_s3`, una para `us-east-1` y otra para `us-west-2`, si desea copiar objetos de un bucket de una región a un bucket de otra región. Del mismo modo, puede crear dos objetos de sesión de seguridad diferentes y utilizarlos para crear dos instancias distintas de `/aws1/cl_s3`, si necesita leer un informe de un bucket relacionado con las finanzas y escribir objetos en un bucket relacionado con la logística.

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

Esta línea es una llamada a `ListObjectsV2`. Requiere argumentos de entrada simples y devuelve un único objeto. Estos objetos pueden representar datos JSON y XML profundos, convertidos por serialización inversa en un constructo ABAP orientado a objetos. En algunos casos, este proceso puede ser bastante complicado. Ahora, solo necesita procesar la salida para mostrar el contenido del 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.
```

Se accede a los datos mediante un método de tipo `GET...()` que oculta la representación interna de los datos. `GET_CONTENTS( )` devuelve una tabla ABAP y cada fila contiene un objeto que representa una única entrada de Amazon S3. En la mayoría de los casos, el AWS SDK adopta este enfoque orientado a objetos y todos los datos se representan como objetos y tablas. El `LastModified` campo se representa como una marca de tiempo que se puede convertir en una fecha con el `CONVERT TIME STAMP` comando nativo de ABAP. `GET_SIZE()` Devuelve una para facilitar las operaciones matemáticas y de formato. `INT4`

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

Todos los errores (de conexión, de cliente 4xx, de servidor 5xx o cualquier error de ABAP, como errores de autorización o configuración) se representan como excepciones. Puede abordar cada excepción por separado. Puede elegir si una excepción debe gestionarse como un error informativo, un reintento, una advertencia, un error de volcado o cualquier otro tipo de gestión.

Conceptos del SDK de SAP ABAP

Esta sección cubre los conceptos básicos de AWS SDK para SAP ABAP.

Temas

- Clases de API
- Objetos adicionales
- Establecimiento de estructura de clases
- Matrices
- Mapas
- Funciones de nivel superior

Clases de API

A cada uno Servicio de AWS se le asigna un acrónimo de tres letras o`TLA`. El servicio está representado por una interfaz en el formato `/AWS1/IF_<TLA>`. A esto lo llamaremos interfaz de servicio. La clase de API está en el paquete `/AWS1/API_<TLA>`. La interfaz de servicio consta de un método para cada AWS operación (llamaremos a estos métodos métodos de operación). Para ver una lista completa de módulos AWS SDK para SAP ABAP TLAs, consulte AWS SDK para SAP ABAP - Lista de módulos.

Cada método de operación tiene algunos argumentos `IMPORTING` y, como máximo, un argumento `RETURNING`. A menudo, estos argumentos son objetos con constructores complicados y un conjunto extenso de métodos `GET…( )`. En muchos casos, los objetos contendrán objetos anidados, referencias recursivas, tablas de objetos, tablas de tablas, etc. Esto se debe a que Servicios de AWS están transmitiendo estructuras XML y JSON profundas, que no se pueden representar mediante un conjunto plano de argumentos.

El `RETURNING` argumento es siempre una clase, incluso si la clase contiene solo un atributo.

Objetos adicionales

Además de contener la clase de API principal, cada paquete de API contiene varios objetos de repositorio y diccionario de datos relacionados.

- Una clase para cada objeto de tipo de estructura.
- Una clase para cualquier tipo de datos primitivo que aparezca en una tabla. Por ejemplo, si un servicio devuelve una tabla de cadenas, la API de ABAP la representará como una tabla de objetos, donde cada objeto es una clase de contenedor que encapsula una cadena. Esto sirve para que la clase de contenedor pueda ocultar los detalles de la representación de una cadena nula que no se pueda representar de forma nativa en ABAP.
- Una clase de excepción para cualquier error específico definido por el servicio.
- Elementos de datos para cada tipo de datos primitivo. Cada tipo de datos tiene su propio elemento de datos para poder documentarse automáticamente.
- Objetos adicionales para el procesamiento interno, como las transformaciones XSLT para realizar la serialización directa e inversa del contenido en archivos XML y JSON.

Establecimiento de estructura de clases

La mayoría de AWS los datos, enviados y recibidos por el servicio, se representan mediante el AWS SDK como clases. Estas clases representan estructuras de datos y ocultan los detalles internos del almacenamiento. En concreto, las clases ocultan la forma en que el SDK representa que este campo no tiene ningún valor.

Para cada campo de una clase de estructura, hay tres métodos.

`GET_field( )`

El método `GET_field( )`

- Devuelve el valor del campo, o
- Si el campo no tiene ningún valor, devuelve un valor predeterminado, que se puede establecer como parámetro opcional.

Por ejemplo, fíjese en el siguiente código que imprime la restricción de ubicación de un bucket.

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

Si el bucket no tiene ninguna restricción de ubicación (como en el caso de `us-east-1`), `GET_LOCATIONCONSTRAINT( )` devolverá la cadena vacía. Puede anular este comportamiento y especificar el valor deseado si el campo no tiene ningún valor.

```
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' ).
```

Ahora el programa escribirá `Bucket Location: assuming us-east-1` si el resultado de `getbucketlocation( )` no devuelve una ubicación.

Es posible pedirle al método GET( ) que devuelva un resultado específico si falta por completo el valor solicitado; consulte el siguiente ejemplo de código.

```
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'  ).
```

En este caso, si no hay ninguna restricción de ubicación, `GET_LOCATIONCONSTRAINT( )` devolverá `NopeNopeNope`.

HAS_field( )

El método `HAS_field( )` es una forma de averiguar si el campo tiene un valor o no. Consulte el siguiente ejemplo.

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

Si se sabe que un campo determinado siempre tiene un valor, no habrá ningún método `HAS_field( )`.

`ASK_field( )`

El método `ASK_field( )` devuelve el valor del campo o genera una excepción si no tiene ningún valor. Esta es una forma cómoda de procesar varios campos y abandonar la lógica y adoptar un enfoque diferente si alguno de los campos no tiene valor.

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

Tenga en cuenta que `/AWS1/CX_RT_VALUE_MISSING` es una excepción estática y recibirá una advertencia si decide no atraparla.

Prácticas recomendadas

En general, puede usar el método `GET_field( )`, ya que trata una cadena nula como una cadena vacía y es la que más se parece a ABAP de las tres opciones. Sin embargo, no permite distinguir fácilmente entre situaciones en las que el campo tiene un valor en blanco y en las que no tiene ningún valor. Si su lógica empresarial depende de distinguir los datos que faltan de los datos en blanco, los métodos `HAS` o `ASK` le permiten gestionar estos casos.

Matrices

Las matrices se representan como tablas de objetos estándar de ABAP.

Una matriz JSON puede contener valores nulos, como la siguiente matriz: `[‘cat’, ‘dog’, null, ‘horse’]`. Esto se conoce como matriz dispersa. Se representa en ABAP como una tabla interna de referencias a objetos y el valor `null` se representa en la tabla como un valor `null` de ABAP verdadero. Al recorrer en iteración una tabla dispersa, debe comprobar los valores `null` para evitar acceder a un objeto `null` y obtener una excepción `CX_SY_REF_IS_INITIAL`. En la práctica, las matrices dispersas son poco frecuentes en los servicios de AWS .

Para inicializar una matriz de objetos, es conveniente utilizar los nuevos constructos de ABAP 7.40. Considere este lanzamiento de una EC2 instancia de Amazon con varios grupos de seguridad asignados:

```
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_ec2secgrpidstrlist_w=>tt_securitygroupidstringlist(
                                    ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-12345678' ) )
                                    ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-55555555' ) )
                                    ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-99999999' ) )
                                )
    iv_subnetid                 = ao_snet->get_subnetid( )
    it_tagspecifications        = make_tag_spec( 'instance' )
)
```

Mapas

Los mapas JSON se representan en ABAP como `Hashed Tables` en que cada fila de la tabla tiene solo dos componentes.

- `KEY`: una cadena que es la `UNIQUE KEY` de la tabla.
- `VALUE`: un objeto que contiene el valor.

Un mapa es uno de los pocos casos en los que el SDK de AWS usa una estructura verdadera, en lugar de una clase. Esto es necesario porque las tablas con hash ABAP no pueden tener una referencia a un objeto como campo clave y las claves del AWS mapa siempre son cadenas no nulas.

Funciones de nivel superior

Las Clases de API descritas en la sección anterior reflejan con precisión el AWS servicio APIs y las representan APIs como clases ABAP conocidas. En algunos casos, el SDK también incluye funciones de nivel superior que se basan en las clases de API para simplificar determinadas operaciones. Las funciones de nivel superior se incluyen para mayor comodidad del programador y no sustituyen a las clases de API de nivel inferior.

Si el SDK incluye funciones de nivel superior para un módulo, se incluyen en el mismo transporte y se puede acceder a ellas a través de una clase de fábrica denominada. `/AWS1/CL_TLA_L2_FACTORY` La clase de fábrica incluye métodos para crear varios clientes de nivel superior para el módulo, que se documentan junto con el resto de la API en la documentación de la API.

AWS SDK para SAP ABAP features

AWS SDK para SAP ABAP proporciona las siguientes funciones.

Temas

- Configuración mediante programación
- Esperadores
- Paginadores
- Comportamiento de los reintentos
- Prefirmantes
- Encadenamiento de roles de IAM entre cuentas

Configuración mediante programación

Utilice `/n/AWS1/IMG` IMG transacation para el AWS SDK para SAP ABAP y la aplicación de configuración empresarial personalizada para el AWS SDK para SAP ABAP (edición BTP) para la configuración programática.

Para iniciar la configuración mediante programación, comience por recuperar un objeto de configuración con el comando `get_config( )`.

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

Cada objeto de configuración implementa la interfaz `/AWS1/IF_RT_CONFIG` que incluye los TER `GET` y `SET` correspondientes a la `IMG`. Por ejemplo, se puede sustituir la región predeterminada. Consulte el siguiente comando de ejemplo.

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

Algunos objetos de configuración no tienen representación de `IMG` y solo se pueden configurar mediante programación, por ejemplo el número máximo de reintentos. Consulte el siguiente comando de ejemplo.

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

El objeto de configuración de también Servicios de AWS puede incluir métodos específicos del servicio que no están representados en. `/aws1/if_rt_config` Por ejemplo, Amazon S3 puede direccionar un bucket denominado `foobucket` mediante un punto de conexión virtual de `foobucket.s3.region.amazonaws.com` o mediante una ruta `s3.region.amazonaws.com/foobucket`. Puede imponer el uso de direccionamiento por ruta con el siguiente comando de ejemplo.

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

Para obtener más información sobre las configuraciones de los servicios, consulte AWS SDK para SAP ABAP la Guía de referencia de la API.

Esperadores

Cuando se trabaja con sistemas asíncronos AWS APIs, hay que esperar a que un determinado recurso esté disponible antes de realizar otras acciones. Por ejemplo, la `CREATETABLE()` API de Amazon DynamoDB responde de inmediato con el estado de la tabla. `CREATING` Puede iniciar operaciones de lectura o escritura solo después de que el estado de la tabla haya cambiado a `ACTIVE`. Los camareros te permiten confirmar que AWS los recursos se encuentran en un estado determinado antes de realizar acciones sobre ellos.

Los camareros utilizan las operaciones de servicio para sondear el estado de AWS los recursos hasta que estos alcancen el estado deseado o hasta que se determine que el recurso no alcanza el estado deseado. Escribir el código para sondear los recursos de AWS de forma continua puede llevar mucho tiempo y se presta a errores. Los esperadores ayudan a simplificar esta complejidad al asumir la responsabilidad de realizar los sondeos.

Consulte el siguiente ejemplo de Amazon S3 con esperador.

```
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|
).
```

- En el siguiente ejemplo, se utiliza el cliente de Amazon S3 para crear un bucket. El comando `get_waiter()` se implementa para especificar si `bucketexists`.
- Debe especificar el parámetro `iv_max_wait_time` para cada esperador. Representa la cantidad total de tiempo que un esperador debe esperar antes de terminar. En el ejemplo anterior, un esperador puede ejecutarse durante 200 segundos.
- Es posible que tenga que proporcionar entradas adicionales para los parámetros necesarios. En el ejemplo anterior, se requiere el nombre del bucket de Amazon S3 para el parámetro `iv_bucket`.
- La excepción `/AWS1/CX_RT_WAITER_FAILURE` indica que el esperador superó el tiempo máximo especificado en el parámetro `iv_max_wait_time`.
- La excepción `/AWS1/CX_RT_WAITER_TIMEOUT` indica que el esperador se ha detenido por no alcanzar el estado deseado.

Paginadores

Algunas Servicio de AWS operaciones ofrecen respuestas paginadas. Están paginadas para devolver una cantidad fija de datos con cada respuesta. Debe realizar las solicitudes posteriores con un token o un marcador para recuperar el conjunto completo de resultados. Por ejemplo, la operación de Amazon S3 de `ListObjectsV2` solo devuelve un máximo de 1000 objetos a la vez. Debe realizar las solicitudes posteriores con el token correspondiente para obtener la siguiente página de resultados.

La paginación es el proceso de enviar solicitudes sucesivas para retomarlas donde quedó una solicitud anterior. Los paginadores son iteradores de los resultados proporcionados por el SDK de SAP ABAP. Puedes usar la paginación APIs con facilidad y sin entender el mecanismo subyacente de la API mediante el uso de tokens de paginación.

Cómo trabajar con paginadores

Puede crear paginadores con el método `get_paginator()` que devuelve un objeto paginador. El objeto paginador llama a la operación que se está paginando. El objeto paginador acepta que se proporcionen los parámetros necesarios a la API subyacente. Este proceso devuelve un objeto iterador que se puede utilizar para iterar sobre los resultados paginados mediante los métodos `has_next()` y `get_next()`.

- `has_next()` devuelve un valor booleano que indica si hay más respuestas o páginas disponibles para la operación llamada.
- `get_next()` devuelve la respuesta de la operación.

En el siguiente ejemplo, se muestran todos los objetos de un bucket de S3 recuperados con paginador.

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

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

Comportamiento de los reintentos

El SDK para SAP ABAP le permite configurar el número máximo de reintentos para solicitudes a Servicios de AWS que fallan debido a errores transitorios o de limitación. El número de reintentos permitidos en el nivel de cliente del servicio, es decir, el número de veces que el SDK reintenta la operación antes de que se produzca un error y se muestre una excepción, se especifica mediante el atributo `AV_MAX_ATTEMPTS` en el objeto de configuración del servicio. Cuando se crea un objeto del cliente de servicio, el SDK configura el atributo `AV_MAX_ATTEMPTS` con un valor predeterminado de

3. El objeto de configuración del servicio se puede utilizar para establecer mediante programación el máximo de reintentos en el valor deseado. Consulte el siguiente ejemplo para obtener más información.

```
" 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**
>
> Aunque el SDK para ABAP del objeto de configuración permite configurar el modo de reintento con el método `/AWS1/IF_RT_CONFIG~SET_RETRY_MODE()`, el SDK solo admite el modo de reintento `standard`. Para obtener más información, consulte el comportamiento de reintento en la Guía de referencia de AWS SDKs las herramientas.

Prefirmantes

Puedes usar prefirmado URLs para conceder acceso por tiempo limitado a algunos. Servicios de AWS Se puede introducir una URL prefirmada en un navegador o puede utilizarla un programa para realizar la operación del servicio. Puede utilizar la URL prefirmada varias veces, hasta la fecha y hora de vencimiento. Para obtener más información, consulte Trabajar con un URLs SDK prefirmado para clientes de SAP ABAP para ver los servicios que admiten a los prefirmantes con un método especial denominado crear un prefirmante `GET_PRESIGNER()` para ese servicio. A continuación, llame a un método del prefirmante, que se corresponda con los métodos del cliente API, salvo que devuelvan una URL prefirmada en lugar de realizar realmente la operación.

```
" 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( ).
```

Encadenamiento de roles de IAM entre cuentas

La compatibilidad con el encadenamiento de funciones de IAM entre cuentas permite un acceso fluido a los recursos de varias AWS cuentas mediante la configuración del perfil de origen. Esta función le permite configurar varios supuestos de roles, en los que un perfil asume un rol y luego asume otro, lo que posibilita patrones complejos de acceso entre cuentas.

Para obtener más información, consulte Uso del perfil de origen para el acceso entre cuentas.

Creación de productos con SDK

Un producto o complemento de ABAP que Servicios de AWS se consuma puede mejorar y ampliar las capacidades del SDK. Puede crear estos productos para usarlos con el SDK.

Temas

- Configuración de un ID de producto

Configuración de un ID de producto

Se recomienda configurar un ID de producto al establecer una sesión dentro de un producto o complemento. Consulte el siguiente ejemplo para obtener más información.

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

El ID de producto solo debe contener letras, números y guiones bajos sin espacios ni caracteres especiales. Puede hacerlo coincidir con el nombre técnico del producto o con cualquier otro identificador. Si desarrolla varios productos o complementos, el ID del producto debe ser único para cada producto. Por ejemplo, el producto IDs para los productos Invoice Analyzer, Tax Calculator y Pricing Engine puede ser `INVOICE_ANALYZERTAX_CALCULATOR`, y. `PRICING_ENGINE`

Añadir un identificador de producto a la sesión mejora la telemetría a la que se envía AWS con cada llamada de servicio. El ID del producto y el espacio de nombres del objeto que realiza la llamada se incluyen en la telemetría. Con esta telemetría, Soporte puede identificar el producto que está

realizando la llamada en caso de que su cliente tenga problemas con el SDK. Puede ayudar a aclarar que la llamada la realiza realmente el producto y no el código del cliente.

Personalice las solicitudes HTTP para AWS

AWS SDK para SAP ABAP Gestiona el proceso de crear una solicitud HTTP, enviar una carga útil y recibir una respuesta. Puede personalizar el comportamiento o el contenido de la solicitud HTTP para que cumpla con sus propios requisitos de TI. El SDK define el punto de mejora `/AWS1/RT_EHN_HTTP_CLIENT` como un lugar central para mejorar la comunicación HTTP. El punto de mejora permite agregar encabezados HTTP a la solicitud realizada a AWS.

Implemente una mejora

SAP proporciona las siguientes instrucciones para implementar un punto de mejora:

- ABAP clásico
- ABAP BTP

Filtra la mejora

El punto de mejora admite múltiples implementaciones que pueden estar activas simultáneamente. Puedes filtrar la ejecución en BAdi función de los siguientes atributos si necesitas asegurarte de que la mejora solo se ejecute en las llamadas a un AWS servicio o acción de API específicos:

- `TLA`- La abreviatura de tres letras del servicio, en mayúsculas.
- `OPERATION`- El nombre de la acción de la API. Por ejemplo, la operación para obtener un objeto de un bucket de S3 es GetObject. El nombre de la acción distingue entre mayúsculas y minúsculas y es posible que no coincida exactamente con el nombre del método ABAP.

Codifique la mejora

La mejora proporciona el siguiente método.

MODIFY_REQ_HEADERS

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

Puede añadir y modificar los encabezados en la tabla interna. `CT_HEADERS` No recomendamos modificar los encabezados, ya que esto altera los datos que utiliza el servicio. AWS El AWS servicio ignora los encabezados que añada, pero su infraestructura de TI, como los servidores proxy u otro middleware, puede procesarlos.

El punto de mejora se indica antes del cálculo de los encabezados de autenticación y telemetría, por lo que la mejora no los puede modificar.

El siguiente es un ejemplo de implementación.

```
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.
```

Limitaciones

AWS SDK para SAP ABAP incluye módulos SDK para todos Servicios de AWS. Algunos de estos módulos pueden tener limitaciones, como se describe aquí.

- Los módulos que se basan en enlaces de `MQTT` protocolo, por ejemplo`iotevents`, no funcionarán. `MQTT`no es un protocolo basado en HTTP y, en la actualidad, no es compatible con. AWS SDK para SAP ABAP
- Se admiten las operaciones que devuelven secuencias de eventos, pero esperarán hasta que se reciba toda la transmisión antes de devolver el control a ABAP. Algunos ejemplos son Amazon Bedrock Agents Runtime `InvokeAgent` y AWS Lambda `InvokeWithResponseStream`
- No se admiten las operaciones que reciben transmisiones de eventos debido a las limitaciones de la plataforma ABAP subyacente. Algunos ejemplos son Amazon Q Business `Chat` y Amazon Lex `StartConversation`

AWS SDK para SAP ABAP tiene las siguientes limitaciones de funciones.

- Las siguientes características de Amazon S3 todavía no son compatibles:
  - Puntos de acceso de varias regiones
  - Cifrado del cliente de Amazon S3

AWS La edición BTP del SDK para SAP ABAP presenta las siguientes limitaciones durante la versión preliminar para desarrolladores.

- Es posible que algunos módulos no estén disponibles.
- No se puede desinstalar.
- Se actualiza con menos frecuencia.

Ejemplos de código de SDK de SAP ABAP

Los ejemplos de código de este tema muestran cómo utilizar el AWS SDK para SAP ABAP con. AWS

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

Algunos servicios contienen categorías de ejemplo adicionales que muestran cómo aprovechar las bibliotecas o funciones específicas del servicio.

Services

- Ejemplos de ACM que utilizan el SDK para SAP ABAP
- Ejemplos de API Gateway que utilizan el SDK para SAP ABAP
- Ejemplos de controladores de recuperación de aplicaciones que utilizan el SDK para SAP ABAP
- Ejemplos de Aurora con el SDK para SAP ABAP
- Ejemplos de Auto Scaling utilizando el SDK para SAP ABAP
- Ejemplos de Amazon Bedrock Runtime con el SDK de SAP ABAP
- Ejemplos de Tiempo de ejecución de Amazon Bedrock Agents con SDK para SAP ABAP
- CloudFront ejemplos de uso del SDK para SAP ABAP
- CloudWatch ejemplos de uso del SDK para SAP ABAP
- CloudWatch Registra ejemplos con el SDK para SAP ABAP
- Ejemplos de proveedores de identidad de Amazon Cognito que utilizan el SDK para SAP ABAP
- Ejemplos de Amazon Comprehend con el SDK para SAP ABAP
- AWS Config ejemplos de uso del SDK para SAP ABAP
- AWS Control Tower ejemplos de uso del SDK para SAP ABAP
- Ejemplos de Firehose utilizando el SDK para SAP ABAP

- Ejemplos de DynamoDB que utilizan el SDK de SAP ABAP
- Ejemplos de Amazon EC2 que utilizan SDK de SAP ABAP
- Ejemplos de Amazon ECR con el SDK para SAP ABAP
- Ejemplos de Amazon EMR con el SDK para SAP ABAP
- EventBridge Ejemplos de planificadores que utilizan el SDK para SAP ABAP
- AWS Glue ejemplos de uso del SDK para SAP ABAP
- HealthImaging ejemplos de uso del SDK para SAP ABAP
- HealthLake ejemplos de uso del SDK para SAP ABAP
- Ejemplos de IAM que utilizan el SDK para SAP ABAP
- AWS IoT data ejemplos de uso del SDK para SAP ABAP
- AWS IoT SiteWise ejemplos de uso del SDK para SAP ABAP
- Ejemplos de Amazon Keyspaces con el SDK para SAP ABAP
- Ejemplos de Kinesis que utilizan el SDK de SAP ABAP
- AWS KMS ejemplos de uso del SDK para SAP ABAP
- Ejemplos de Lambda que utilizan SDK de SAP ABAP
- Ejemplos de organizaciones que utilizan el SDK para SAP ABAP
- Ejemplos de Amazon Pinpoint con el SDK para SAP ABAP
- Ejemplos de API de SMS y voz de Amazon Pinpoint con el SDK para SAP ABAP
- Ejemplos de Amazon Polly con el SDK para SAP ABAP
- Ejemplos de Amazon RDS con el SDK para SAP ABAP
- Ejemplos de Amazon Redshift con el SDK para SAP ABAP
- Ejemplos de Amazon Rekognition con el SDK para SAP ABAP
- Ejemplos de Amazon S3 que utilizan SDK de SAP ABAP
- SageMaker Ejemplos de IA que utilizan el SDK para SAP ABAP
- Ejemplos de Secrets Manager con el SDK para SAP ABAP
- Ejemplos de Amazon SES que utilizan el SDK para SAP ABAP
- Ejemplos de la API v2 de Amazon SES que utilizan el SDK para SAP ABAP
- Ejemplos de Amazon SNS con el SDK de SAP ABAP
- Ejemplos de Amazon SQS que utilizan el SDK de SAP ABAP

- Ejemplos de Step Functions con el SDK para SAP ABAP
- Ejemplos de Systems Manager que utilizan el SDK para SAP ABAP
- Ejemplos de Amazon Textract que utilizan el SDK de SAP ABAP
- Ejemplos de Amazon Transcribe con el SDK para SAP ABAP
- Ejemplos de Amazon Translate que utilizan el SDK de SAP ABAP

Ejemplos de ACM que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con ACM.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AddTagsToCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `AddTagsToCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
```

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

- Para obtener más información sobre la API, consulte AddTagsToCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

- Para obtener más información sobre la API, consulte DeleteCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`GetCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`ImportCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `ImportCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

- Para obtener más información sobre la API, consulte ImportCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`ListCertificates`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListCertificates`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

- Para obtener más información sobre la API, consulte ListCertificatesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTagsForCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTagsForCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListTagsForCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`RemoveTagsFromCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `RemoveTagsFromCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte RemoveTagsFromCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`RequestCertificate`

En el siguiente ejemplo de código, se muestra cómo utilizar `RequestCertificate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte RequestCertificatela referencia sobre la API ABAP del AWS SDK para SAP.

`ResendValidationEmail`

En el siguiente ejemplo de código, se muestra cómo utilizar `ResendValidationEmail`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ResendValidationEmailla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de API Gateway que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el AWS SDK para SAP ABAP con API Gateway.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateDeployment`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDeployment`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateDeploymentla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateResource`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateResourcela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateRestApi`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateRestApi`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateRestApila referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteRestApi`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteRestApi`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteRestApila referencia sobre la API ABAP del AWS SDK para SAP.

`GetResources`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetResources`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetResourcesla referencia sobre la API ABAP del AWS SDK para SAP.

`GetRestApis`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetRestApis`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte GetRestApisla referencia sobre la API ABAP del AWS SDK para SAP.

`PutIntegration`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutIntegration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutIntegrationla referencia sobre la API ABAP del AWS SDK para SAP.

`PutIntegrationResponse`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutIntegrationResponse`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutIntegrationResponsela referencia sobre la API ABAP del AWS SDK para SAP.

`PutMethod`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutMethod`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutMethodla referencia sobre la API ABAP del AWS SDK para SAP.

`PutMethodResponse`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutMethodResponse`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutMethodResponsela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de controladores de recuperación de aplicaciones que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Application Recovery Controller.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`GetRoutingControlState`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetRoutingControlState`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetRoutingControlStatela referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateRoutingControlState`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateRoutingControlState`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateRoutingControlStatela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Aurora con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Aurora.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateDBClusterParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDBClusterParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

- Para obtener más información sobre la API, consulte Crear DBCluster ParameterGroup en el AWS SDK para la referencia sobre la API ABAP de SAP.

`DeleteDBClusterParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDBClusterParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulta Eliminar DBCluster ParameterGroup en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeDBClusterParameterGroups`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBClusterParameterGroups`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_rdsdbprmgrnotfndfault.
ENDTRY.
```

- Para obtener más información sobre la API, consulta la sección Describe DBCluster ParameterGroups en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeDBClusterParameters`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBClusterParameters`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_rdsdbprmgrnotfndfault.
        " Re-raise exception - parameter group not found
        RAISE EXCEPTION TYPE /aws1/cx_rdsdbprmgrnotfndfault.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte Describa DBCluster los parámetros en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`DescribeDBEngineVersions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBEngineVersions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulta la sección Describir DBEngine las versiones en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`DescribeOrderableDBInstanceOptions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeOrderableDBInstanceOptions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulta DescribeOrderableDBInstancelas opciones AWS del SDK para la referencia sobre la API ABAP de SAP.

`ModifyDBClusterParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `ModifyDBClusterParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_rdsinvdbprmgrstatef00.
        " Re-raise exception - invalid state
        RAISE EXCEPTION TYPE /aws1/cx_rdsinvdbprmgrstatef00.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte Modificar DBCluster ParameterGroup en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

Ejemplos de Auto Scaling utilizando el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP with Auto Scaling.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateAutoScalingGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAutoScalingGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

TRY.
    " Create launch template specification
```

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

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

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

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

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

- Para obtener más información sobre la API, consulte CreateAutoScalingGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteAutoScalingGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAutoScalingGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte DeleteAutoScalingGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAutoScalingGroups`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAutoScalingGroups`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeAutoScalingGroupsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAutoScalingInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAutoScalingInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte DescribeAutoScalingInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeScalingActivities`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeScalingActivities`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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.
```

- Para obtener más información sobre la API, consulte DescribeScalingActivitiesla referencia sobre la API ABAP del AWS SDK para SAP.

`DisableMetricsCollection`

En el siguiente ejemplo de código, se muestra cómo utilizar `DisableMetricsCollection`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte DisableMetricsCollectionla referencia sobre la API ABAP del AWS SDK para SAP.

`EnableMetricsCollection`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableMetricsCollection`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte EnableMetricsCollectionla referencia sobre la API ABAP del AWS SDK para SAP.

`SetDesiredCapacity`

En el siguiente ejemplo de código, se muestra cómo utilizar `SetDesiredCapacity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

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

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

- Para obtener más información sobre la API, consulte SetDesiredCapacityla referencia sobre la API ABAP del AWS SDK para SAP.

`TerminateInstanceInAutoScalingGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `TerminateInstanceInAutoScalingGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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_ascscaactivityinprg00 INTO DATA(lo_activity_in_progress).
    RAISE EXCEPTION lo_activity_in_progress.
  CATCH /aws1/cx_ascresrccontionfault INTO DATA(lo_contention).
    RAISE EXCEPTION lo_contention.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_exception).
    RAISE EXCEPTION lo_generic_exception.
ENDTRY.
```

- Para obtener más información sobre la API, consulte TerminateInstanceInAutoScalingGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateAutoScalingGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateAutoScalingGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

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

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

- Para obtener más información sobre la API, consulte UpdateAutoScalingGroupla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Bedrock Runtime con el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Bedrock Runtime.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Anthropic Claude
- Stable Diffusion

Anthropic Claude

InvokeModel

El siguiente ejemplo de código muestra cómo enviar un mensaje de texto a Anthropic Claude con la API de Invoke Model.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Invoque el modelo fundacional Anthropic Claude 2 para generar texto. En este ejemplo, se utilizan funciones de US2 /CL_JSON que podrían no estar disponibles en algunas versiones. NetWeaver

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

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

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

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

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

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

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

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

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

```
    ENDTRY.
```

Invocación del modelo fundacional Anthropic Claude 2 para generar texto con el cliente de alto nivel L2.

```
    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.
```

Invocación del modelo fundacional Anthropic Claude 3 para generar texto con el cliente de alto nivel L2.

```
    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.
```

- Para obtener más información sobre la API, consulte la referencia sobre InvokeModella API ABAP AWS del SDK para SAP.

Stable Diffusion

InvokeModel

El siguiente ejemplo de código muestra cómo invocar el modelo Stability.ai Stable Diffusion XL en Amazon Bedrock para generar una imagen.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree una imagen con 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.
```

Invoque el modelo fundacional Stability.ai Stable Diffusion XL para generar imágenes con el cliente de nivel alto L2.

```
    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.
```

- Para obtener más información sobre la API, consulte InvokeModella referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Tiempo de ejecución de Amazon Bedrock Agents con SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Bedrock Agents Runtime.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`InvokeAgent`

En el siguiente ejemplo de código, se muestra cómo utilizar `InvokeAgent`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte InvokeAgentla referencia sobre la API ABAP del AWS SDK para SAP.

CloudFront ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. CloudFront

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`ListDistributions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDistributions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte ListDistributionsla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateDistribution`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateDistribution`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_origingroups = lo_old_config->get_origingroups( )
          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.
```

- Para obtener más información sobre la API, consulte UpdateDistributionla referencia sobre la API ABAP del AWS SDK para SAP.

CloudWatch ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. CloudWatch

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`DeleteAlarms`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAlarms`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteAlarmsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAlarms`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAlarms`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeAlarmsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAlarmsForMetric`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAlarmsForMetric`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeAlarmsForMetricla referencia sobre la API ABAP del AWS SDK para SAP.

`DisableAlarmActions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DisableAlarmActions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    "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.
```

- Para obtener más información sobre la API, consulte DisableAlarmActionsla referencia sobre la API ABAP del AWS SDK para SAP.

`EnableAlarmActions`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableAlarmActions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    "Enable actions on the specified alarm."
    TRY.
        lo_cwt->enablealarmactions(
          it_alarmnames = it_alarm_names ).
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte EnableAlarmActionsla referencia sobre la API ABAP del AWS SDK para SAP.

`GetMetricStatistics`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetMetricStatistics`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_cwtinvparamvalueex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
```

```
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GetMetricStatisticsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListMetrics`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListMetrics`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "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.
```

- Para obtener más información sobre la API, consulte ListMetricsla referencia sobre la API ABAP del AWS SDK para SAP.

`PutMetricAlarm`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutMetricAlarm`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutMetricAlarmla referencia sobre la API ABAP del AWS SDK para SAP.

`PutMetricData`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutMetricData`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

Coloca un conjunto de datos en una CloudWatch métrica.

```
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.
```

- Para obtener más información sobre la API, consulte PutMetricDataen el AWS SDK la referencia sobre la API ABAP de SAP.

Escenarios

Primeros pasos para usar alarmas

En el siguiente ejemplo de código, se muestra cómo:

- Cree una alarma.
- Desactive las acciones de alarma.
- Describa una alarma.
- Elimine una alarma.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

    "Create an alarm"
```

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

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

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

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

```
    ENDTRY.

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

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

CloudWatch Registra ejemplos con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP with CloudWatch Logs.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`GetQueryResults`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetQueryResults`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetQueryResultsla referencia sobre la API ABAP del AWS SDK para SAP.

`StartQuery`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartQuery`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte StartQueryla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de proveedores de identidad de Amazon Cognito que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Cognito Identity Provider.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AdminInitiateAuth`

En el siguiente ejemplo de código, se muestra cómo utilizar `AdminInitiateAuth`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

      " Add SECRET_HASH if provided
      IF iv_secret_hash IS NOT INITIAL.
        INSERT VALUE #(
          key = 'SECRET_HASH'
          value = NEW /aws1/cl_cgpauthparamstype_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_cgpnotauthorizedex INTO DATA(lo_auth_ex).
      MESSAGE 'Not authorized. Check credentials.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte AdminInitiateAuthla referencia sobre la API ABAP del AWS SDK para SAP.

`AdminRespondToAuthChallenge`

En el siguiente ejemplo de código, se muestra cómo utilizar `AdminRespondToAuthChallenge`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

        DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_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_cgpexpiredcodeex INTO DATA(lo_expired_ex).
      MESSAGE 'MFA code has expired.' TYPE 'E'.

    CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
      MESSAGE 'Not authorized. Check MFA configuration.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte AdminRespondToAuthChallengela referencia sobre la API ABAP del AWS SDK para SAP.

`AssociateSoftwareToken`

En el siguiente ejemplo de código, se muestra cómo utilizar `AssociateSoftwareToken`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

      ov_secret_code = lo_result->get_secretcode( ).

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

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

```
  CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
    MESSAGE 'Not authorized to associate software token.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte AssociateSoftwareTokenla referencia sobre la API ABAP del AWS SDK para SAP.

`ListUsers`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListUsers`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

    ot_users = lo_result->get_users( ).

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

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

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

- Para obtener más información sobre la API, consulte ListUsersla referencia sobre la API ABAP del AWS SDK para SAP.

`VerifySoftwareToken`

En el siguiente ejemplo de código, se muestra cómo utilizar `VerifySoftwareToken`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte VerifySoftwareTokenla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Comprehend con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Comprehend.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateDocumentClassifier`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDocumentClassifier`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_cpd->createdocumentclassifier(
          iv_documentclassifiername = iv_classifier_name
          iv_languagecode = iv_language_code
          io_inputdataconfig = NEW /aws1/cl_cpddocclifierinpdat00(
            iv_s3uri = iv_training_s3_uri
          )
          iv_dataaccessrolearn = iv_data_access_role_arn
          iv_mode = iv_mode
        ).
        MESSAGE 'Document classifier creation started.' TYPE 'I'.
      CATCH /aws1/cx_cpdinvalidrequestex.
        MESSAGE 'Invalid request.' TYPE 'E'.
      CATCH /aws1/cx_cpdresrclimitexcdex.
        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_cpdinternalserverex.
      MESSAGE 'Internal server error occurred.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateDocumentClassifierla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteDocumentClassifier`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDocumentClassifier`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  TRY.
      oo_result = lo_cpd->deletedocumentclassifier(
        iv_documentclassifierarn = iv_classifier_arn
      ).
      MESSAGE 'Document classifier deleted.' TYPE 'I'.
    CATCH /aws1/cx_cpdinvalidrequestex.
      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.
```

- Para obtener más información sobre la API, consulte DeleteDocumentClassifierla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeDocumentClassificationJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDocumentClassificationJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeDocumentClassificationJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeDocumentClassifier`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDocumentClassifier`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_cpd->describedocumentclassifier(
      iv_documentclassifierarn = iv_classifier_arn
    ).
    MESSAGE 'Document classifier described.' TYPE 'I'.
  CATCH /aws1/cx_cpdinvalidrequestex.
    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_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeDocumentClassifierla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeTopicsDetectionJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeTopicsDetectionJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
```

```
    oo_result = lo_cpd->describetopicsdetectionjob(
      iv_jobid = iv_job_id
    ).
    MESSAGE 'Topics detection 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.
```

- Para obtener más información sobre la API, consulte DescribeTopicsDetectionJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectDominantLanguage`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectDominantLanguage`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DetectDominantLanguagela referencia sobre la API ABAP del AWS SDK para SAP.

`DetectEntities`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectEntities`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DetectEntitiesla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectKeyPhrases`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectKeyPhrases`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DetectKeyPhraseslа referencia sobre la API ABAP del AWS SDK para SAP.

`DetectPiiEntities`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectPiiEntities`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DetectPiiEntitiesla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectSentiment`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectSentiment`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

```
      MESSAGE 'Unsupported language.' TYPE 'E'.
    CATCH /aws1/cx_cpdinternalserverex.
      MESSAGE 'Internal server error occurred.' TYPE 'E'.
    CATCH /aws1/cx_cpdinvalidrequestex.
      MESSAGE 'Invalid request.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte DetectSentimentla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectSyntax`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectSyntax`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DetectSyntaxla referencia sobre la API ABAP del AWS SDK para SAP.

`ListDocumentClassificationJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDocumentClassificationJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte ListDocumentClassificationJobsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListDocumentClassifiers`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDocumentClassifiers`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_cpd->listdocumentclassifiers( ).
    MESSAGE 'Document classifiers listed.' TYPE 'I'.
  CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
  CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
  CATCH /aws1/cx_cpdinvalidfilterex.
    MESSAGE 'Invalid filter.' TYPE 'E'.
  CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte ListDocumentClassifiersla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTopicsDetectionJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTopicsDetectionJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

```
    CATCH /aws1/cx_cpdinvalidrequestex.
      MESSAGE 'Invalid request.' TYPE 'E'.
    CATCH /aws1/cx_cpdtoomanyrequestsex.
      MESSAGE 'Too many requests.' TYPE 'E'.
    CATCH /aws1/cx_cpdinvalidfilterex.
      MESSAGE 'Invalid filter.' TYPE 'E'.
    CATCH /aws1/cx_cpdinternalserverex.
      MESSAGE 'Internal server error occurred.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte ListTopicsDetectionJobsla referencia sobre la API ABAP del AWS SDK para SAP.

`StartDocumentClassificationJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartDocumentClassificationJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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_cpdinvalidrequestex.
```

```
      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.
```

- Para obtener más información sobre la API, consulte StartDocumentClassificationJobla referencia sobre la API ABAP del AWS SDK para SAP.

`StartTopicsDetectionJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartTopicsDetectionJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte StartTopicsDetectionJobla referencia sobre la API ABAP del AWS SDK para SAP.

AWS Config ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS Config

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`DeleteConfigRule`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteConfigRule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte DeleteConfigRulela referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeConfigRules`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeConfigRules`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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'.
```

- Para obtener más información sobre la API, consulte DescribeConfigRuleslа referencia sobre la API ABAP del AWS SDK para SAP.

`PutConfigRule`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutConfigRule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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_cfscplncresrctypes_w=>tt_complianceresourcetypes(
            ( NEW /aws1/cl_cfscplncresrctypes_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'.
```

- Para obtener más información sobre la API, consulte PutConfigRulela referencia sobre la API ABAP del AWS SDK para SAP.

AWS Control Tower ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS Control Tower

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`DisableBaseline`

En el siguiente ejemplo de código, se muestra cómo utilizar `DisableBaseline`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DisableBaselinela referencia sobre la API ABAP del AWS SDK para SAP.

`DisableControl`

En el siguiente ejemplo de código, se muestra cómo utilizar `DisableControl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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'.
```

- Para obtener más información sobre la API, consulte DisableControlla referencia sobre la API ABAP del AWS SDK para SAP.

`EnableBaseline`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableBaseline`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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'.
```

- Para obtener más información sobre la API, consulte EnableBaselinela referencia sobre la API ABAP del AWS SDK para SAP.

`EnableControl`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableControl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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'.
```

- Para obtener más información sobre la API, consulte EnableControlla referencia sobre la API ABAP del AWS SDK para SAP.

`GetBaselineOperation`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetBaselineOperation`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  DATA(lo_output) = io_ctt->getbaselineoperation(
    iv_operationidentifier = iv_operation_id
  ).

  ov_status = lo_output->get_baselineoperation( )->get_status( ).
```

- Para obtener más información sobre la API, consulte GetBaselineOperationla referencia sobre la API ABAP del AWS SDK para SAP.

`GetControlOperation`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetControlOperation`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
DATA(lo_output) = io_ctt->getcontroloperation(
  iv_operationidentifier = iv_operation_id
).

ov_status = lo_output->get_controloperation( )->get_status( ).
```

- Para obtener más información sobre la API, consulte GetControlOperationla referencia sobre la API ABAP del AWS SDK para SAP.

`ListBaselines`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListBaselines`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
DATA lt_baselines TYPE /aws1/cl_cttbaselinesummary=>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'.
```

- Para obtener más información sobre la API, consulte ListBaselinesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListEnabledBaselines`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListEnabledBaselines`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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'.
```

- Para obtener más información sobre la API, consulte ListEnabledBaselinesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListEnabledControls`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListEnabledControls`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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'.
```

- Para obtener más información sobre la API, consulte ListEnabledControlsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListLandingZones`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListLandingZones`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    DATA lt_landing_zones TYPE /aws1/
cl_cttlandingzonesummary=>tt_landingzonesummaries.
    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'.
```

- Para obtener más información sobre la API, consulte ListLandingZonesla referencia sobre la API ABAP del AWS SDK para SAP.

`ResetEnabledBaseline`

En el siguiente ejemplo de código, se muestra cómo utilizar `ResetEnabledBaseline`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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'.
```

- Para obtener más información sobre la API, consulte ResetEnabledBaselinela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Firehose utilizando el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP with Firehose.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`PutRecord`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutRecord`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutRecordla referencia sobre la API ABAP del AWS SDK para SAP.

`PutRecordBatch`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutRecordBatch`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        DATA(lo_result) = lo_frh->putrecordbatch(
          iv_deliverystreamname = iv_deliv_stream_name
          it_records            = it_records ).

        DATA(lv_failed_count) = lo_result->get_failedputcount( ).

        IF lv_failed_count > 0.
          MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
        ELSE.
          MESSAGE 'All records sent successfully to Firehose delivery stream.' TYPE
'I'.
        ENDIF.
      CATCH /aws1/cx_frhresourcenotfoundex.
        MESSAGE 'Delivery stream not found.' TYPE 'E'.
      CATCH /aws1/cx_frhinvalidargumentex.
        MESSAGE 'Invalid argument provided.' TYPE 'E'.
      CATCH /aws1/cx_frhserviceunavailex.
        MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte PutRecordBatchla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de DynamoDB que utilizan el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con DynamoDB.

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Conceptos básicos
- Acciones

Conceptos básicos

Conceptos básicos

En el siguiente ejemplo de código, se muestra cómo:

- Creación de una tabla que pueda contener datos de películas.
- Colocar, obtener y actualizar una sola película en la tabla.
- Escribir los datos de películas en la tabla a partir de un archivo JSON de ejemplo.
- Consultar películas que se hayan estrenado en un año determinado.
- Buscar películas que se hayan estrenado en un intervalo de años.
- Eliminación de una película de la tabla y, a continuación, eliminar la tabla.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " Create an Amazon Dynamo DB table.

    TRY.
        DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
        DATA(lo_dyn) = /aws1/cl_dyn_factory=>create( lo_session ).
        DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                              iv_keytype = 'HASH' ) )
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                              iv_keytype = 'RANGE' ) ) ).
        DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                           iv_attributetype = 'N' ) )
```

```
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                           iv_attributetype = 'S' ) ) ).

        " Adjust read/write capacities as desired.
        DATA(lo_dynprovthroughput)  = NEW /aws1/cl_dynprovthroughput(
          iv_readcapacityunits = 5
          iv_writecapacityunits = 5 ).
        DATA(oo_result) = lo_dyn->createtable(
          it_keyschema = lt_keyschema
          iv_tablename = iv_table_name
          it_attributedefinitions = lt_attributedefinitions
          io_provisionedthroughput = lo_dynprovthroughput ).
        " Table creation can take some time. Wait till table exists before
 returning.
        lo_dyn->get_waiter( )->tableexists(
          iv_max_wait_time = 200
          iv_tablename     = iv_table_name ).
        MESSAGE 'DynamoDB Table' && iv_table_name && 'created.' TYPE 'I'.
        " It throws exception if the table already exists.
      CATCH /aws1/cx_dynresourceinuseex INTO DATA(lo_resourceinuseex).
        DATA(lv_error) = |"{ lo_resourceinuseex->av_err_code }" -
{ lo_resourceinuseex->av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

    " Describe table
    TRY.
        DATA(lo_table) = lo_dyn->describetable( iv_tablename = iv_table_name ).
        DATA(lv_tablename) = lo_table->get_table( )->ask_tablename( ).
        MESSAGE 'The table name is ' && lv_tablename TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table does not exist' TYPE 'E'.
    ENDTRY.

    " Put items into the table.
    TRY.
        DATA(lo_resp_putitem) = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Jaws' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
```

```
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
          ) ).
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
 Wars' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) )
          ) ).
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Speed' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
          ) ).
        " TYPE REF TO /AWSEX/CL_AWS1_dyn_PUT_ITEM_OUTPUT
        MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
      CATCH /aws1/cx_dyncondalcheckfaile00.
        MESSAGE 'A condition specified in the operation could not be evaluated.'
 TYPE 'E'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
      CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.
```

```
    " Get item from table.
    TRY.
        DATA(lo_resp_getitem) = lo_dyn->getitem(
          iv_tablename               = iv_table_name
          it_key                     = VALUE /aws1/cl_dynattributevalue=>tt_key(
           ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
             key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Jaws' ) ) )
           ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
             key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
 '1975' ) ) )
          ) ).
        DATA(lt_attr) = lo_resp_getitem->get_item( ).
        DATA(lo_title) = lt_attr[ key = 'title' ]-value.
        DATA(lo_year) = lt_attr[ key = 'year' ]-value.
        DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

    " Query item from table.
    TRY.
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
             ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
        DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
          ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
          key = 'year'
          value = NEW /aws1/cl_dyncondition(
            it_attributevaluelist = lt_attributelist
            iv_comparisonoperator = |EQ|
          ) ) ) ).
        DATA(lo_query_result) = lo_dyn->query(
          iv_tablename = iv_table_name
          it_keyconditions = lt_keyconditions ).
        DATA(lt_items) = lo_query_result->get_items( ).
        READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
        lo_title = lt_item[ key = 'title' ]-value.
        lo_year = lt_item[ key = 'year' ]-value.
        lo_rating = lt_item[ key = 'rating' ]-value.
```

```
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

    " Scan items from table.
    TRY.
        DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
        lt_items = lo_scan_result->get_items( ).
        " Read the first item and display the attributes.
        READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
        lo_title = lt_item[ key = 'title' ]-value.
        lo_year = lt_item[ key = 'year' ]-value.
        lo_rating = lt_item[ key = 'rating' ]-value.
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

    " Update items from table.
    TRY.
        DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
          ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
          key = 'rating' value = NEW /aws1/cl_dynattrvalueupdate(
            io_value  = NEW /aws1/cl_dynattributevalue( iv_n = '7.6' )
            iv_action = |PUT| ) ) ) ).
        DATA(lt_key) = VALUE /aws1/cl_dynattributevalue=>tt_key(
          ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) )
          ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 '1980' ) ) ) ).
        DATA(lo_resp) = lo_dyn->updateitem(
          iv_tablename        = iv_table_name
          it_key              = lt_key
          it_attributeupdates = lt_attributeupdates ).
        MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.
      CATCH /aws1/cx_dyncondalcheckfaile00.
```

```
        MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
      CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

    " Delete table.
    TRY.
        lo_dyn->deletetable( iv_tablename = iv_table_name ).
        lo_dyn->get_waiter( )->tablenotexists(
          iv_max_wait_time = 200
          iv_tablename     = iv_table_name ).
        MESSAGE 'DynamoDB Table deleted.' TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
      CATCH /aws1/cx_dynresourceinuseex.
        MESSAGE 'The table cannot be deleted as it is in use' TYPE 'E'.
    ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Query
  - Scan
  - UpdateItem

Acciones

`CreateTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateTablela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteItemla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteTablela referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeTablela referencia sobre la API ABAP del AWS SDK para SAP.

`GetItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetItemla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTables`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTables`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListTablesla referencia sobre la API ABAP del AWS SDK para SAP.

`PutItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte PutItemla referencia sobre la API ABAP del AWS SDK para SAP.

`Query`

En el siguiente ejemplo de código, se muestra cómo utilizar `Query`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Query movies for a given year .
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
            ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_year }| ) ) ).
        DATA(lt_key_conditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
          ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
          key = 'year'
          value = NEW /aws1/cl_dyncondition(
          it_attributevaluelist = lt_attributelist
          iv_comparisonoperator = |EQ|
          ) ) ) ).
        oo_result = lo_dyn->query(
          iv_tablename = iv_table_name
          it_keyconditions = lt_key_conditions ).
        DATA(lt_items) = oo_result->get_items( ).
        "You can loop over the results to get item attributes.
        LOOP AT lt_items INTO DATA(lt_item).
          DATA(lo_title) = lt_item[ key = 'title' ]-value.
          DATA(lo_year) = lt_item[ key = 'year' ]-value.
        ENDLOOP.
        DATA(lv_count) = oo_result->get_count( ).
        MESSAGE 'Item count is: ' && lv_count TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.
```

- Para obtener detalles sobre la API, consulte Query en la Referencia de la API del SDK de AWS para SAP ABAP.

`Scan`

En el siguiente ejemplo de código, se muestra cómo utilizar `Scan`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Scan movies for rating greater than or equal to the rating specified
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
            ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_rating }| ) ) ).
        DATA(lt_filter_conditions) = VALUE /aws1/
cl_dyncondition=>tt_filterconditionmap(
          ( VALUE /aws1/cl_dyncondition=>ts_filterconditionmap_maprow(
          key = 'rating'
          value = NEW /aws1/cl_dyncondition(
          it_attributevaluelist = lt_attributelist
          iv_comparisonoperator = |GE|
          ) ) ) ).
        oo_scan_result = lo_dyn->scan( iv_tablename = iv_table_name
          it_scanfilter = lt_filter_conditions ).
        DATA(lt_items) = oo_scan_result->get_items( ).
        LOOP AT lt_items INTO DATA(lo_item).
          " You can loop over to get individual attributes.
          DATA(lo_title) = lo_item[ key = 'title' ]-value.
          DATA(lo_year) = lo_item[ key = 'year' ]-value.
        ENDLOOP.
        DATA(lv_count) = oo_scan_result->get_count( ).
        MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.
```

- Para obtener detalles sobre la API, consulte Scan en la Referencia de la API del SDK de AWS para SAP ABAP.

`UpdateItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateItemla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon EC2 que utilizan SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon EC2.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AllocateAddress`

En el siguiente ejemplo de código, se muestra cómo utilizar `AllocateAddress`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte AllocateAddressla referencia sobre la API ABAP del AWS SDK para SAP.

`AssociateAddress`

En el siguiente ejemplo de código, se muestra cómo utilizar `AssociateAddress`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_ec2->associateaddress(                          " oo_result is
 returned for testing purposes. "
            iv_allocationid = iv_allocation_id
            iv_instanceid = iv_instance_id ).
        MESSAGE 'Associated an Elastic IP address with an EC2 instance.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte AssociateAddressla referencia sobre la API ABAP del AWS SDK para SAP.

`AuthorizeSecurityGroupIngress`

En el siguiente ejemplo de código, se muestra cómo utilizar `AuthorizeSecurityGroupIngress`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte AuthorizeSecurityGroupIngressla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateKeyPair`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateKeyPair`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateKeyPairla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateSecurityGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateSecurityGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_ec2->createsecuritygroup(                " oo_result is
 returned for testing purposes. "
          iv_description = 'Security group example'
          iv_groupname = iv_security_group_name
          iv_vpcid = iv_vpc_id ).
        MESSAGE 'Security group created.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateSecurityGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateVpc`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateVpc`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte CreateVpcla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateVpcEndpoint`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateVpcEndpoint`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " iv_vpc_id = 'vpc-abc123'
    " iv_service_name = 'com.amazonaws.region.service'
    TRY.
        oo_result = lo_ec2->createvpcendpoint(             " oo_result is returned
 for testing purposes. "
          iv_vpcid = iv_vpc_id
          iv_servicename = iv_service_name
          it_routetableids = it_route_table_ids ).
        DATA(lv_vpc_endpoint_id) = oo_result->get_vpcendpoint( )-
>get_vpcendpointid( ).
        MESSAGE 'Created VPC endpoint.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateVpcEndpointla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteKeyPair`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteKeyPair`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteKeyPairla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteSecurityGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteSecurityGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteSecurityGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteVpc`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteVpc`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        lo_ec2->deletevpc( iv_vpcid = iv_vpc_id ).
        MESSAGE 'Deleted VPC.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteVpcla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteVpcEndpoints`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteVpcEndpoints`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteVpcEndpointsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAddresses`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAddresses`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeAddresseslа referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeAvailabilityZones`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeAvailabilityZones`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeAvailabilityZonesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeImages`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeImages`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeImagesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeInstanceTypes`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeInstanceTypes`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " Create filters for architecture and instance type patterns
    " iv_architecture = 'x86_64'
    DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'processor-info.supported-architecture'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( iv_architecture ) )
      )
    ) TO lt_filters.
    " Filter for instance type patterns like '*.micro', '*.small'
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'instance-type'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.micro' ) )
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.small' ) )
      )
```

```
    ) TO lt_filters.

    TRY.
        oo_result = lo_ec2->describeinstancetypes( it_filters = lt_filters ).
      " oo_result is returned for testing purposes. "
        DATA(lt_instance_types) = oo_result->get_instancetypes( ).
        MESSAGE 'Retrieved information about EC2 instance types.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeInstanceTypesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_ec2->describeinstances( ).                       " oo_result
 is returned for testing purposes. "

        " Retrieving details of EC2 instances. "
        DATA: lv_istance_id    TYPE /aws1/ec2string,
              lv_status        TYPE /aws1/ec2instancestatename,
              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_istance_id = lo_instance->get_instanceid( ).
```

```
            lv_status = lo_instance->get_state( )->get_name( ).
            lv_instance_type = lo_instance->get_instancetype( ).
            lv_image_id = lo_instance->get_imageid( ).
          ENDLOOP.
        ENDLOOP.
        MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeKeyPairs`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeKeyPairs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeKeyPairsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeRegions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeRegions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeRegionsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeRouteTables`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeRouteTables`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulte DescribeRouteTablesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeSecurityGroups`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeSecurityGroups`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeSecurityGroupsla referencia sobre la API ABAP del AWS SDK para SAP.

`MonitorInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `MonitorInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte MonitorInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`RebootInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `RebootInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte RebootInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`ReleaseAddress`

En el siguiente ejemplo de código, se muestra cómo utilizar `ReleaseAddress`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ReleaseAddressla referencia sobre la API ABAP del AWS SDK para SAP.

`RunInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `RunInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " Create tags for resource created during instance launch. "
    DATA lt_tagspecifications TYPE /aws1/
cl_ec2tagspecification=>tt_tagspecificationlist.
    DATA ls_tagspecifications LIKE LINE OF lt_tagspecifications.
    ls_tagspecifications = NEW /aws1/cl_ec2tagspecification(
      iv_resourcetype = 'instance'
      it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
        ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
      ) ).
    APPEND ls_tagspecifications TO lt_tagspecifications.

    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_tagspecifications = lt_tagspecifications
          iv_subnetid = iv_subnet_id ).
        MESSAGE 'EC2 instance created.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte RunInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`StartInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte StartInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`StopInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `StopInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte StopInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon ECR con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon ECR.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateRepository`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateRepository`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        oo_result = lo_ecr->createrepository(
          iv_repositoryname = iv_repository_name ).
        DATA(lv_repository_uri) = oo_result->get_repository( )-
>get_repositoryuri( ).
        MESSAGE |Repository created with URI: { lv_repository_uri }| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositoryalrexex.
        " If repository already exists, retrieve it
        DATA lt_repo_names TYPE /aws1/
cl_ecrrepositorynamels00=>tt_repositorynamelist.
        APPEND NEW /aws1/cl_ecrrepositorynamels00( 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_ecrcrerepositoryrsp( io_repository = lo_repo ).
          MESSAGE |Repository { iv_repository_name } already exists.| TYPE 'I'.
        ENDIF.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateRepositorylа referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteRepository`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteRepository`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteRepositorylа referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeImages`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeImages`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ecrrepositorynotfndex.
        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.
```

- Para obtener más información sobre la API, consulte DescribeImagesla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeRepositories`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeRepositories`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " it_repository_names = VALUE #( ( NEW /aws1/
cl_ecrrepositorynamels00( 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_ecrrepositorynotfndex.
```

```
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeRepositoriesla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAuthorizationToken`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAuthorizationToken`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetAuthorizationTokenla referencia sobre la API ABAP del AWS SDK para SAP.

`GetRepositoryPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetRepositoryPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrrepositoryplynot00.
        MESSAGE 'Repository policy not found.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GetRepositoryPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`PutLifeCyclePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutLifeCyclePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrvalidationex.
        MESSAGE 'Invalid lifecycle policy format.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte PutLifeCyclePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`SetRepositoryPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `SetRepositoryPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte SetRepositoryPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon EMR con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon EMR.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AddJobFlowSteps`

En el siguiente ejemplo de código, se muestra cómo utilizar `AddJobFlowSteps`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte AddJobFlowStepsla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeCluster`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeCluster`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeClusterla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeStep`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeStep`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeStepla referencia sobre la API ABAP del AWS SDK para SAP.

`ListSteps`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListSteps`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte ListStepsla referencia sobre la API ABAP del AWS SDK para SAP.

`RunJobFlow`

En el siguiente ejemplo de código, se muestra cómo utilizar `RunJobFlow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte RunJobFlowla referencia sobre la API ABAP del AWS SDK para SAP.

`TerminateJobFlows`

En el siguiente ejemplo de código, se muestra cómo utilizar `TerminateJobFlows`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte TerminateJobFlowsla referencia sobre la API ABAP del AWS SDK para SAP.

EventBridge Ejemplos de planificadores que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Scheduler. EventBridge

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateSchedule`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateSchedule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Constants for time calculations
```

```
        DATA lv_start_date TYPE /aws1/scdstartdate.
        DATA lv_end_date TYPE /aws1/scdenddate.
        DATA lv_start_timestamp TYPE timestamp.
        DATA lv_end_timestamp TYPE timestamp.
        DATA lv_hours_to_run TYPE i VALUE 1.

        " Get current timestamp
        GET TIME STAMP FIELD lv_start_timestamp.

        " Add 1 hour to the current timestamp using CL_ABAP_TSTMP
        lv_end_timestamp = cl_abap_tstmp=>add(
          tstmp = lv_start_timestamp
          secs = lv_hours_to_run * 3600 ).

        " Convert timestamps to decimal format for AWS API
        lv_start_date = lv_start_timestamp.
        lv_end_date = lv_end_timestamp.

        " Prepare flexible time window configuration
        DATA lo_flexible_time_window TYPE REF TO /aws1/cl_scdflexibletimewindow.
        IF iv_use_flexible_time_win = abap_true.
          " iv_use_flexible_time_win = ABAP_TRUE
          " Example: Set MaximumWindowInMinutes to 10 for flexible window
          lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
            iv_mode = 'FLEXIBLE'
            iv_maximumwindowinminutes = 10 ).
        ELSE.
          lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
            iv_mode = 'OFF' ).
        ENDIF.

        " Prepare target configuration
        " Example iv_target_arn = 'arn:aws:sqs:us-east-1:123456789012:my-queue'
        " Example iv_role_arn = 'arn:aws:iam::123456789012:role/SchedulerRole'
        " Example iv_input = '{"message": "Hello from EventBridge Scheduler"}'
        DATA(lo_target) = NEW /aws1/cl_scdtarget(
          iv_arn = iv_target_arn
          iv_rolearn = iv_role_arn
          iv_input = iv_input ).

        " Set action after completion if needed
        DATA lv_action_after_completion TYPE /aws1/scdactionaftercompletion.
        IF iv_delete_after_completion = abap_true.
          " iv_delete_after_completion = ABAP_TRUE
```

```
          lv_action_after_completion = 'DELETE'.
        ELSE.
          lv_action_after_completion = 'NONE'.
        ENDIF.

        " Create the schedule
        " Example iv_name = 'my-schedule'
        " Example iv_schedule_expression = 'rate(15 minutes)'
        " Example iv_schedule_group_name = 'my-schedule-group'
        DATA(lo_result) = lo_scd->createschedule(
          iv_name = iv_name
          iv_scheduleexpression = iv_schedule_expression
          iv_groupname = iv_schedule_group_name
          io_target = lo_target
          io_flexibletimewindow = lo_flexible_time_window
          iv_startdate = lv_start_date
          iv_enddate = lv_end_date
          iv_actionaftercompletion = lv_action_after_completion ).

        ov_schedule_arn = lo_result->get_schedulearn( ).
        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.
```

- Para obtener más información sobre la API, consulte CreateSchedulela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateScheduleGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateScheduleGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateScheduleGroupla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteSchedule`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteSchedule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteSchedulela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteScheduleGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteScheduleGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteScheduleGroupla referencia sobre la API ABAP del AWS SDK para SAP.

AWS Glue ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS Glue

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateCrawler`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateCrawler`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
    " iv_database_name = 'my-database'
    " iv_table_prefix = 'test_'
    " iv_s3_target = 's3://example-bucket/data/'

    DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
      ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

    DATA(lo_targets) = NEW /aws1/cl_glucrawlertargets(
      it_s3targets = lt_s3_targets ).

    lo_glu->createcrawler(
      iv_name = iv_crawler_name
      iv_role = iv_role_arn
      iv_databasename = iv_database_name
      iv_tableprefix = iv_table_prefix
      io_targets = lo_targets ).
    MESSAGE 'Crawler created successfully.' TYPE 'I'.
  CATCH /aws1/cx_glualreadyexistsex.
    MESSAGE 'Crawler already exists.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
  CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
    DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
    MESSAGE lv_limit_error TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateCrawlerla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteCrawler`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteCrawler`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteCrawlerla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteDatabase`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDatabase`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteDatabasela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteTablela referencia sobre la API ABAP del AWS SDK para SAP.

`GetCrawler`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetCrawler`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte GetCrawlerla referencia sobre la API ABAP del AWS SDK para SAP.

`GetDatabase`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetDatabase`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte GetDatabasela referencia sobre la API ABAP del AWS SDK para SAP.

`GetJobRun`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetJobRun`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetJobRunla referencia sobre la API ABAP del AWS SDK para SAP.

`GetJobRuns`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetJobRuns`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetJobRunsla referencia sobre la API ABAP del AWS SDK para SAP.

`GetTables`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetTables`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetTablesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListJobsla referencia sobre la API ABAP del AWS SDK para SAP.

`StartCrawler`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartCrawler`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte StartCrawlerla referencia sobre la API ABAP del AWS SDK para SAP.

`StartJobRun`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartJobRun`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " iv_job_name = 'my-etl-job'
        " iv_input_database = 'my-database'
        " iv_input_table = 'my-table'
        " iv_output_bucket_url = 's3://example-output-bucket/'

        DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
        lt_arguments = VALUE #(
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_database'
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_database ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_table'
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--output_bucket_url'
            value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

        DATA(oo_result) = lo_glu->startjobrun(
```

```
          iv_jobname = iv_job_name
          it_arguments = lt_arguments ).
        ov_job_run_id = oo_result->get_jobrunid( ).
        MESSAGE 'Job run started successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluconcurrentrunsex00.
        MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Job does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte StartJobRunla referencia sobre la API ABAP del AWS SDK para SAP.

HealthImaging ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. HealthImaging

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CopyImageSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `CopyImageSet`.

SDK para SAP ABAP

```
   TRY.
       " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
       " iv_source_image_set_id = '12345678901234567890123456789012345678901234567890'
       " iv_source_version_id = '1'
       " iv_destination_image_set_id =
'12345678901234567890123456789012345678901234567890' (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_copyimagesetinformation = lo_copy_info
         iv_force = iv_force ).
       DATA(lo_dest_props) = oo_result->get_dstimagesetproperties( ).
       DATA(lv_new_id) = lo_dest_props->get_imagesetid( ).
       MESSAGE |Image set copied with new ID: { lv_new_id }.| TYPE 'I'.
     CATCH /aws1/cx_migaccessdeniedex.
       MESSAGE 'Access denied.' TYPE 'I'.
     CATCH /aws1/cx_migconflictexception.
       MESSAGE 'Conflict error.' TYPE 'I'.
     CATCH /aws1/cx_miginternalserverex.
       MESSAGE 'Internal server error.' TYPE 'I'.
```

```
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Image set not found.' TYPE 'I'.
    CATCH /aws1/cx_migservicequotaexcdex.
      MESSAGE 'Service quota exceeded.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CopyImageSetla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`CreateDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDatastore`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte CreateDatastorela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`DeleteDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDatastore`.

SDK para SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->deletedatastore( iv_datastoreid = iv_datastore_id ).
    MESSAGE 'Data store deleted.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict. Data store may contain resources.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Data store not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteDatastorela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`DeleteImageSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteImageSet`.

SDK para SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    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.
```

- Para obtener más información sobre la API, consulte DeleteImageSetla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`GetDICOMImportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetDICOMImportJob`.

SDK para SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " 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.
```

- Para obtener más información sobre la API, consulte la referencia Get DICOMImport Job in AWS SDK para la API ABAP de SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`GetDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetDatastore`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte GetDatastorela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`GetImageFrame`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetImageFrame`.

SDK para SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_frame_id = '12345678901234567890123456789012345678901234567890'
    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.
```

- Para obtener más información sobre la API, consulte GetImageFramela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`GetImageSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetImageSet`.

SDK para SAP ABAP

```
  TRY.
      " iv_datastore_id = '1234567890123456789012345678901234567890'
      " iv_image_set_id = '1234567890123456789012345678901234567890'
      " 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.
```

- Para obtener más información sobre la API, consulte GetImageSetla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`GetImageSetMetadata`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetImageSetMetadata`.

SDK para SAP ABAP

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
```

```
      " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
      " iv_version_id = '1' (optional)
      IF iv_version_id IS NOT INITIAL.
        oo_result = lo_mig->getimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id
          iv_versionid = iv_version_id ).
      ELSE.
        oo_result = lo_mig->getimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id ).
      ENDIF.
      DATA(lv_metadata_blob) = oo_result->get_imagesetmetadatablob( ).
      MESSAGE 'Image set metadata retrieved.' TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Image set not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte GetImageSetMetadatala referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`ListDICOMImportJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDICOMImportJobs`.

SDK para 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'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte Listar DICOMImport trabajos en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`ListDatastores`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDatastores`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte ListDatastoresla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`ListImageSetVersions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListImageSetVersions`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte ListImageSetVersionsla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`ListTagsForResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTagsForResource`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte ListTagsForResourcela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`SearchImageSets`

En el siguiente ejemplo de código, se muestra cómo utilizar `SearchImageSets`.

SDK para SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->searchimagesets(
      iv_datastoreid = iv_datastore_id
      io_searchcriteria = io_search_criteria ).
    DATA(lt_imagesets) = oo_result->get_imagesetsmetadatasums( ).
    DATA(lv_count) = lines( lt_imagesets ).
    MESSAGE |Found { lv_count } image sets.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte SearchImageSetsla referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`StartDICOMImportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartDICOMImportJob`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte Start DICOMImport Job in AWS SDK para obtener una referencia sobre la API ABAP de SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`TagResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `TagResource`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte TagResourcela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`UntagResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `UntagResource`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte UntagResourcela referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

`UpdateImageSetMetadata`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateImageSetMetadata`.

SDK para 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.
```

- Para obtener más información sobre la API, consulte UpdateImageSetMetadatala referencia sobre la API ABAP del AWS SDK para SAP.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

HealthLake ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. HealthLake

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateFHIRDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateFHIRDatastore`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte Crear FHIRDatastore en el AWS SDK para la referencia sobre la API ABAP de SAP.

`DeleteFHIRDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteFHIRDatastore`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulta Eliminar FHIRDatastore en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeFHIRDatastore`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeFHIRDatastore`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulta la sección Describe FHIRDatastore en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeFHIRExportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeFHIRExportJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Describe FHIRExport Job en el AWS SDK para la referencia de la API ABAP de SAP.

`DescribeFHIRImportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeFHIRImportJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirimportjob(
          iv_datastoreid = iv_datastore_id
          iv_jobid = iv_job_id
        ).
        DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
        IF lo_import_job_properties IS BOUND.
```

```
          DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
          MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte Describe FHIRImport Job en el AWS SDK para la referencia de la API ABAP de SAP.

`ListFHIRDatastores`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListFHIRDatastores`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre las API, consulta la lista FHIRDatastores en el AWS SDK para obtener información sobre las API ABAP de SAP.

`ListFHIRExportJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListFHIRExportJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Listar FHIRExport trabajos en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`ListFHIRImportJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListFHIRImportJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Listar FHIRImport trabajos en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`ListTagsForResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTagsForResource`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListTagsForResourcela referencia sobre la API ABAP del AWS SDK para SAP.

`StartFHIRExportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartFHIRExportJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Start FHIRExport Job in AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`StartFHIRImportJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartFHIRImportJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Start FHIRImport Job in AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`TagResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `TagResource`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte TagResourcela referencia sobre la API ABAP del AWS SDK para SAP.

`UntagResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `UntagResource`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UntagResourcela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de IAM que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con IAM.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AttachRolePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `AttachRolePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte AttachRolePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`AttachUserPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `AttachUserPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte AttachUserPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateAccessKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAccessKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamlimitexceededex.
    MESSAGE 'Maximum number of access keys reached.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateAccessKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateAccountAlias`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAccountAlias`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_iam->createaccountalias(
      iv_accountalias = iv_account_alias ).
    MESSAGE 'Account alias created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Account alias already exists.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Account alias limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateAccountAliasla referencia sobre la API ABAP del AWS SDK para SAP.

`CreatePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreatePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamentityalrdyexex.
    MESSAGE 'Policy already exists.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreatePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`CreatePolicyVersion`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreatePolicyVersion`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamlimitexceededex.
      MESSAGE 'Policy version limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CreatePolicyVersionla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateRole`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateRole`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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_iamentityalrdyexex.
      MESSAGE 'Role already exists.' TYPE 'E'.
    CATCH /aws1/cx_iammalformedplydocex.
      MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
    CATCH /aws1/cx_iamlimitexceededex.
      MESSAGE 'Role limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateRolela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateServiceLinkedRole`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateServiceLinkedRole`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateServiceLinkedRolela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateUser`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateUser`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_iam->createuser(
      iv_username = iv_user_name ).
    MESSAGE 'User created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'User already exists.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Entity does not exist.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateUserla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteAccessKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAccessKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteAccessKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteAccountAlias`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAccountAlias`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteAccountAliasla referencia sobre la API ABAP del AWS SDK para SAP.

`DeletePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeletePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeletePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DeletePolicyVersion`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeletePolicyVersion`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DeletePolicyVersionla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteRole`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteRole`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteRolela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteUser`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteUser`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteUserla referencia sobre la API ABAP del AWS SDK para SAP.

`DetachRolePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetachRolePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte DetachRolePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DetachUserPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetachUserPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetachUserPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`GenerateCredentialReport`

En el siguiente ejemplo de código, se muestra cómo utilizar `GenerateCredentialReport`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_iam->generatecredentialreport( ).
    MESSAGE 'Credential report generation started.' TYPE 'I'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Report generation limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
```

```
        MESSAGE 'Service failure when generating credential report.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GenerateCredentialReportla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAccessKeyLastUsed`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAccessKeyLastUsed`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetAccessKeyLastUsedla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAccountAuthorizationDetails`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAccountAuthorizationDetails`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetAccountAuthorizationDetailsla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAccountPasswordPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAccountPasswordPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetAccountPasswordPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAccountSummary`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAccountSummary`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetAccountSummaryla referencia sobre la API ABAP del AWS SDK para SAP.

`GetCredentialReport`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetCredentialReport`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetCredentialReportla referencia sobre la API ABAP del AWS SDK para SAP.

`GetPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`GetPolicyVersion`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetPolicyVersion`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetPolicyVersionla referencia sobre la API ABAP del AWS SDK para SAP.

`GetRole`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetRole`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetRolela referencia sobre la API ABAP del AWS SDK para SAP.

`ListAccessKeys`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAccessKeys`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListAccessKeysla referencia sobre la API ABAP del AWS SDK para SAP.

`ListAccountAliases`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAccountAliases`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListAccountAliasesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListAttachedRolePolicies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAttachedRolePolicies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListAttachedRolePoliciesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListGroups`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListGroups`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListGroupsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListPolicies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListPolicies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListPolicieslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListPolicyVersions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListPolicyVersions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListPolicyVersionslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListRolePolicies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListRolePolicies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListRolePolicieslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListRoles`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListRoles`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListRoleslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListSAMLProviders`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListSAMLProviders`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre las API, consulta la lista SAMLProviders en el AWS SDK para obtener información sobre las API ABAP de SAP.

`ListUsers`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListUsers`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListUserslа referencia sobre la API ABAP del AWS SDK para SAP.

`SetDefaultPolicyVersion`

En el siguiente ejemplo de código, se muestra cómo utilizar `SetDefaultPolicyVersion`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte SetDefaultPolicyVersionla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateAccessKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateAccessKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte UpdateAccessKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateUser`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateUser`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iamentityalrdyexex.
      MESSAGE 'New user name already exists.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte UpdateUserla referencia sobre la API ABAP del AWS SDK para SAP.

AWS IoT data ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS IoT data

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`GetThingShadow`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetThingShadow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    DATA(lo_result) = lo_iop->getthingshadow( 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.
```

- Para obtener más información sobre la API, consulte GetThingShadowla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateThingShadow`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateThingShadow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte UpdateThingShadowla referencia sobre la API ABAP del AWS SDK para SAP.

AWS IoT SiteWise ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS IoT SiteWise

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Introducción
- Acciones

Introducción

Hola AWS IoT SiteWise

En el siguiente ejemplo de código se muestra cómo empezar a utilizar AWS IoT SiteWise.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListAssetModelsla referencia sobre la API ABAP del AWS SDK para SAP.

Acciones

`BatchPutAssetPropertyValue`

En el siguiente ejemplo de código, se muestra cómo utilizar `BatchPutAssetPropertyValue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte BatchPutAssetPropertyValuela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateAsset`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAsset`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte CreateAssetla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateAssetModel`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAssetModel`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_iosresrcalrdyexistsex.
    MESSAGE 'Asset model already exists.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateAssetModella referencia sobre la API ABAP del AWS SDK para SAP.

`CreateGateway`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateGateway`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_ios->creategateway(
      iv_gatewayname = iv_gateway_name
      io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(
        io_greengrassv2 = NEW /aws1/cl_iosgreengrassv2(
```

```
            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_iosresrcalrdyexistsex.
    MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateGatewayla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteAsset`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAsset`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteAssetla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteAssetModel`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAssetModel`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteAssetModella referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteGateway`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteGateway`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteGatewayla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeGateway`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeGateway`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeGatewayla referencia sobre la API ABAP del AWS SDK para SAP.

`GetAssetPropertyValue`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetAssetPropertyValue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetAssetPropertyValuela referencia sobre la API ABAP del AWS SDK para SAP.

`ListAssetModelProperties`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAssetModelProperties`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListAssetModelPropertiesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListAssetModels`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAssetModels`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListAssetModelsla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Keyspaces con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Keyspaces.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateKeyspace`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateKeyspace`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateKeyspacela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateTablela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteKeyspace`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteKeyspace`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteKeyspacela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteTablela referencia sobre la API ABAP del AWS SDK para SAP.

`GetKeyspace`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetKeyspace`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetKeyspacela referencia sobre la API ABAP del AWS SDK para SAP.

`GetTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetTablela referencia sobre la API ABAP del AWS SDK para SAP.

`ListKeyspaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListKeyspaces`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListKeyspacesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTables`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTables`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListTablesla referencia sobre la API ABAP del AWS SDK para SAP.

`RestoreTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `RestoreTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte RestoreTablela referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateTable`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateTable`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateTablela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Kinesis que utilizan el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Kinesis.

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Conceptos básicos
- Acciones

Conceptos básicos

Conceptos básicos

En el siguiente ejemplo de código, se muestra cómo:

- Cree un flujo e incluya un registro en él.

- Cree un iterador de partición.
- Lea el registro y, a continuación, limpie los recursos.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   DATA lo_stream_describe_result TYPE REF TO /aws1/cl_knsdescrstreamoutput.
   DATA lo_stream_description TYPE REF TO /aws1/cl_knsstreamdescription.
   DATA lo_sharditerator TYPE REF TO /aws1/cl_knsgetsharditerator01.
   DATA lo_record_result TYPE REF TO /aws1/cl_knsputrecordoutput.

   "Create stream."
   TRY.
       lo_kns->createstream(
           iv_streamname = iv_stream_name
           iv_shardcount = iv_shard_count ).
       MESSAGE 'Stream created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.

   "Wait for stream to becomes active."
   lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
   lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
   WHILE lo_stream_description->get_streamstatus( ) <> 'ACTIVE'.
     IF sy-index = 30.
       EXIT.               "maximum 5 minutes"
     ENDIF.
```

```
      WAIT UP TO 10 SECONDS.
      lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
      lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
    ENDWHILE.

    "Create record."
    TRY.
        lo_record_result = lo_kns->putrecord(
            iv_streamname = iv_stream_name
            iv_data       = iv_data
            iv_partitionkey = iv_partition_key ).
        MESSAGE 'Record created.' TYPE 'I'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
      CATCH /aws1/cx_knskmsoptinrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
      CATCH /aws1/cx_knskmsthrottlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

    "Create a shard iterator in order to read the record."
    TRY.
        lo_sharditerator = lo_kns->getsharditerator(
          iv_shardid = lo_record_result->get_shardid( )
          iv_sharditeratortype = iv_sharditeratortype
          iv_streamname = iv_stream_name ).
        MESSAGE 'Shard iterator created.' TYPE 'I'.
      CATCH /aws1/cx_knsinvalidargumentex.
```

```
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcenotfound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

    "Read the record."
    TRY.
        oo_result = lo_kns->getrecords(                    " oo_result is returned
for testing purposes. "
            iv_sharditerator   = lo_sharditerator->get_sharditerator( ) ).
        MESSAGE 'Shard iterator created.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
      CATCH /aws1/cx_knskmsoptinrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
      CATCH /aws1/cx_knskmsthrottlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

    "Delete stream."
    TRY.
        lo_kns->deletestream(
            iv_streamname = iv_stream_name ).
        MESSAGE 'Stream deleted.' TYPE 'I'.
```

```
      CATCH /aws1/cx_knslimitexceededex.
        MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
      CATCH /aws1/cx_knsresourceinuseex.
        MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
    ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - CreateStream
  - DeleteStream
  - GetRecords
  - GetShardIterator
  - PutRecord

Acciones

`CreateStream`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateStream`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        lo_kns->createstream(
            iv_streamname = iv_stream_name
            iv_shardcount = iv_shard_count ).
        MESSAGE 'Stream created.' TYPE 'I'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knslimitexceededex.
```

```
       MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateStreamla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteStream`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteStream`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   TRY.
       lo_kns->deletestream(
           iv_streamname = iv_stream_name ).
       MESSAGE 'Stream deleted.' TYPE 'I'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteStreamla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeStream`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeStream`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_knslimitexceededex.
        MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeStreamla referencia sobre la API ABAP del AWS SDK para SAP.

`GetRecords`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetRecords`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_knskmsthrottlingex.
       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.
```

- Para obtener más información sobre la API, consulte GetRecordsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListStreams`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListStreams`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte ListStreamsla referencia sobre la API ABAP del AWS SDK para SAP.

`PutRecord`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutRecord`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_knskmsthrottlingex.
        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.
```

- Para obtener más información sobre la API, consulte PutRecordla referencia sobre la API ABAP del AWS SDK para SAP.

`RegisterStreamConsumer`

En el siguiente ejemplo de código, se muestra cómo utilizar `RegisterStreamConsumer`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcenotfound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte RegisterStreamConsumerla referencia sobre la API ABAP del AWS SDK para SAP.

AWS KMS ejemplos de uso del SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con. AWS KMS

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Introducción
- Acciones

Introducción

Hola AWS KMS

En el siguiente ejemplo de código se muestra cómo empezar a utilizar AWS Key Management Service.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListKeysla referencia sobre la API ABAP del AWS SDK para SAP.

Acciones

`CreateAlias`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateAlias`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateAliasla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateGrant`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateGrant`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateGrantla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  TRY.
      " iv_description = 'Created by the AWS SDK for SAP ABAP'
      oo_result = lo_kms->createkey( iv_description = iv_description ).
      MESSAGE 'KMS key created successfully.' TYPE 'I'.
    CATCH /aws1/cx_kmskmsinternalex.
      MESSAGE 'An internal error occurred.' TYPE 'E'.
    CATCH /aws1/cx_kmslimitexceededex.
      MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`Decrypt`

En el siguiente ejemplo de código, se muestra cómo utilizar `Decrypt`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulta Cómo descifrar en el AWS SDK para obtener información sobre la API ABAP de SAP.

`DeleteAlias`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteAlias`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteAliasla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`DisableKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `DisableKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DisableKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`EnableKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte EnableKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`EnableKeyRotation`

En el siguiente ejemplo de código, se muestra cómo utilizar `EnableKeyRotation`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte EnableKeyRotationla referencia sobre la API ABAP del AWS SDK para SAP.

`Encrypt`

En el siguiente ejemplo de código, se muestra cómo utilizar `Encrypt`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Cifrar en el AWS SDK para obtener información sobre la API ABAP de SAP.

`GenerateDataKey`

En el siguiente ejemplo de código, se muestra cómo utilizar `GenerateDataKey`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GenerateDataKeyla referencia sobre la API ABAP del AWS SDK para SAP.

`GetKeyPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetKeyPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetKeyPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`ListAliases`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAliases`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListAliasesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListGrants`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListGrants`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListGrantsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListKeyPolicies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListKeyPolicies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListKeyPoliciesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListKeys`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListKeys`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListKeysla referencia sobre la API ABAP del AWS SDK para SAP.

`PutKeyPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutKeyPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte PutKeyPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`ReEncrypt`

En el siguiente ejemplo de código, se muestra cómo utilizar `ReEncrypt`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ReEncryptla referencia sobre la API ABAP del AWS SDK para SAP.

`RetireGrant`

En el siguiente ejemplo de código, se muestra cómo utilizar `RetireGrant`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte RetireGrantla referencia sobre la API ABAP del AWS SDK para SAP.

`RevokeGrant`

En el siguiente ejemplo de código, se muestra cómo utilizar `RevokeGrant`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte RevokeGrantla referencia sobre la API ABAP del AWS SDK para SAP.

`ScheduleKeyDeletion`

En el siguiente ejemplo de código, se muestra cómo utilizar `ScheduleKeyDeletion`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_pendingwindowindays = 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.
```

- Para obtener más información sobre la API, consulte ScheduleKeyDeletionla referencia sobre la API ABAP del AWS SDK para SAP.

`Sign`

En el siguiente ejemplo de código, se muestra cómo utilizar `Sign`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte la referencia sobre el AWS SDK de inicio de sesión para la API ABAP de SAP.

`TagResource`

En el siguiente ejemplo de código, se muestra cómo utilizar `TagResource`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte TagResourcela referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateAlias`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateAlias`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateAliasla referencia sobre la API ABAP del AWS SDK para SAP.

`Verify`

En el siguiente ejemplo de código, se muestra cómo utilizar `Verify`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte Verificar en el AWS SDK para obtener información sobre la API ABAP de SAP.

Ejemplos de Lambda que utilizan SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el AWS SDK para SAP ABAP con Lambda.

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Conceptos básicos
- Acciones

Conceptos básicos

Conceptos básicos

En el siguiente ejemplo de código, se muestra cómo:

- Crear un rol de IAM y una función de Lambda y, a continuación, cargar el código de controlador.
- Invocar la función con un único parámetro y obtener resultados.
- Actualizar el código de la función y configurar con una variable de entorno.
- Invocar la función con un nuevo parámetro y obtener resultados. Mostrar el registro de ejecución devuelto.
- Enumerar las funciones de su cuenta y, luego, limpiar los recursos.

Para obtener información, consulte Crear una función de Lambda con la consola.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_iamentityalrdyexex.
            DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
            lv_role_arn = lo_role->get_role( )->get_arn( ).
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iammalformedplydocex.
            MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
        ENDTRY.

        TRY.
            lo_iam->attachrolepolicy(
                iv_rolename  = iv_role_name
                iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
            MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
          CATCH /aws1/cx_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'.
        ENDTRY.

        " Create a Lambda function and upload handler code. "
        " Lambda function performs 'increment' action on a number. "
        TRY.
            lo_lmd->createfunction(
                 iv_functionname = iv_function_name
```

```
                iv_runtime = `python3.9`
                iv_role = lv_role_arn
                iv_handler = iv_handler
                io_code = io_initial_zip_file
                iv_description = 'AWS Lambda code example' ).
            MESSAGE 'Lambda function created.' TYPE 'I'.
          CATCH /aws1/cx_lmdcodestorageexcdex.
            MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        ENDTRY.

        " Verify the function is in Active state "
        WHILE lo_lmd->getfunction( iv_functionname = iv_function_name )-
>get_configuration( )->ask_state( ) <> 'Active'.
          IF sy-index = 10.
            EXIT.               " Maximum 10 seconds. "
          ENDIF.
          WAIT UP TO 1 SECONDS.
        ENDWHILE.

        "Invoke the function with a single parameter and get results."
        TRY.
            DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
              `{`  &&
                `"action": "increment",`  &&
                `"number": 10` &&
              `}` ).
            DATA(lo_initial_invoke_output) = lo_lmd->invoke(
                       iv_functionname = iv_function_name
                       iv_payload = lv_json ).
            ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
        " ov_initial_invoke_payload is returned for testing purposes. "
            DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
 if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
 lo_writer_json.
            DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
```

```
          CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
          CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

        " Update the function code and configure its Lambda environment with an
 environment variable. "
        " Lambda function is updated to perform 'decrement' action also. "
        TRY.
            lo_lmd->updatefunctioncode(
                  iv_functionname = iv_function_name
                  iv_zipfile = io_updated_zip_file ).
            WAIT UP TO 10 SECONDS.            " Make sure that the update is
 completed. "
            MESSAGE 'Lambda function code updated.' TYPE 'I'.
          CATCH /aws1/cx_lmdcodestorageexcdex.
            MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        ENDTRY.

        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_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

        " 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_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'.
        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.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - CreateFunction
  - DeleteFunction
  - GetFunction
  - Invoke
  - ListFunctions
  - UpdateFunctionCode
  - UpdateFunctionConfiguration

Acciones

`CreateFunction`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateFunction`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_lmd->createfunction(
        iv_functionname = iv_function_name
        iv_runtime = `python3.9`
        iv_role = iv_role_arn
        iv_handler = iv_handler
        io_code = io_zip_file
        iv_description = 'AWS Lambda code example' ).
```

```
        MESSAGE 'Lambda function created.' TYPE 'I'.
      CATCH /aws1/cx_lmdcodesigningcfgno00.
        MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
      CATCH /aws1/cx_lmdcodestorageexcdex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
      CATCH /aws1/cx_lmdcodeverification00.
        MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvalidcodesigex.
        MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
      CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
      CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
      CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
      CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateFunctionla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteFunction`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteFunction`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteFunctionla referencia sobre la API ABAP del AWS SDK para SAP.

`GetFunction`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetFunction`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetFunctionla referencia sobre la API ABAP del AWS SDK para SAP.

`Invoke`

En el siguiente ejemplo de código, se muestra cómo utilizar `Invoke`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
          `{`  &&
            `"action": "increment",`  &&
            `"number": 10` &&
          `}` ).
        oo_result = lo_lmd->invoke(                  " oo_result is returned for
testing purposes. "
                 iv_functionname = iv_function_name
                 iv_payload = lv_json ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
      CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvalidzipfileex.
        MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
      CATCH /aws1/cx_lmdrequesttoolargeex.
        MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
      CATCH /aws1/cx_lmdresourceconflictex.
```

```
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
      CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
      CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
      CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
      CATCH /aws1/cx_lmdunsuppedmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
    ENDTRY.
```

- Para obtener detalles sobre la API, consulte Invoke en la Referencia de la API del SDK de AWS para SAP ABAP.

`ListFunctions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListFunctions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListFunctionsla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateFunctionCode`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateFunctionCode`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateFunctionCodela referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateFunctionConfiguration`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateFunctionConfiguration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateFunctionConfigurationla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de organizaciones que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el AWS SDK para SAP ABAP with Organizations.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`AttachPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `AttachPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte AttachPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`CreatePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreatePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreatePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DeletePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeletePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeletePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribePolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribePolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte DescribePolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DetachPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetachPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetachPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`ListPolicies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListPolicies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListPolicieslа referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Pinpoint con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Pinpoint.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`SendMessages`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendMessages`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un mensaje de correo electrónico.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_html_message
          )
          io_textpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_text_message
          )
        )
      )
    )
```

```
    )
  ).

  " Extract message IDs from response
  DATA(lo_message_response) = lo_result->get_messageresponse( ).
  ot_message_ids = lo_message_response->get_result( ).

  MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

Enviar un mensaje SMS.

```
  " 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_messagetype = iv_message_type
          iv_originationnumber = iv_origination_number
        )
      )
    )
  ).

  " 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'.
```

Enviar un mensaje de correo electrónico con una plantilla de correo electrónico existente.

```
    " 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'.
```

Enviar un mensaje de texto con una plantilla de SMS existente.

```
" 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_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
    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'.
```

- Para obtener más información sobre la API, consulte SendMessagesla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de API de SMS y voz de Amazon Pinpoint con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con la API de SMS y voz de Amazon Pinpoint.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateConfigurationSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateConfigurationSet`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateConfigurationSetla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateConfigurationSetEventDestination`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateConfigurationSetEventDestination`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
```

```
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        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.
```

- Para obtener más información sobre la API, consulte CreateConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteConfigurationSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteConfigurationSet`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteConfigurationSetla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteConfigurationSetEventDestination`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteConfigurationSetEventDestination`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

`GetConfigurationSetEventDestinations`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetConfigurationSetEventDestinations`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetConfigurationSetEventDestinationsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListConfigurationSets`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListConfigurationSets`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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'.
        ENDLOOP.

        " 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.
```

- Para obtener más información sobre la API, consulte ListConfigurationSetsla referencia sobre la API ABAP del AWS SDK para SAP.

`SendVoiceMessage`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendVoiceMessage`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  TRY.
      " Create SSML message type object with voice parameters
      DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
        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.
```

- Para obtener más información sobre la API, consulte SendVoiceMessagela referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateConfigurationSetEventDestination`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateConfigurationSetEventDestination`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte UpdateConfigurationSetEventDestinationla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Polly con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Polly.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`DeleteLexicon`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteLexicon`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteLexiconla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeVoices`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeVoices`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeVoicesla referencia sobre la API ABAP del AWS SDK para SAP.

`GetLexicon`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetLexicon`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetLexiconla referencia sobre la API ABAP del AWS SDK para SAP.

`GetSpeechSynthesisTask`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetSpeechSynthesisTask`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetSpeechSynthesisTaskla referencia sobre la API ABAP del AWS SDK para SAP.

`ListLexicons`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListLexicons`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListLexiconsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListSpeechSynthesisTasks`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListSpeechSynthesisTasks`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListSpeechSynthesisTasksla referencia sobre la API ABAP del AWS SDK para SAP.

`PutLexicon`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutLexicon`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_plylexiconsizeexcdex.
    MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plymaxlexemelengthe00.
    MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plymaxlexiconsnoexc00.
    MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plyunsuppedplsalpha00.
    MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
  CATCH /aws1/cx_plyunsuppedplslangu00.
    MESSAGE 'Unsupported PLS language.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte PutLexiconla referencia sobre la API ABAP del AWS SDK para SAP.

`StartSpeechSynthesisTask`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartSpeechSynthesisTask`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Only pass optional parameters if they have values
        IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_lang_code IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ELSE.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
```

```
          iv_outputs3bucketname = iv_s3_bucket
          iv_text = iv_text
          iv_voiceid = iv_voice_id ).
      ENDIF.
      MESSAGE 'Speech synthesis task started.' TYPE 'I'.
    CATCH /aws1/cx_plyinvalids3bucketex.
      MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
    CATCH /aws1/cx_plyinvalidssmlex.
      MESSAGE 'Invalid SSML.' TYPE 'E'.
    CATCH /aws1/cx_plylexiconnotfoundex.
      MESSAGE 'Lexicon not found.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
      MESSAGE 'Service failure occurred.' TYPE 'E'.
    CATCH /aws1/cx_plytextlengthexcdex.
      MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte StartSpeechSynthesisTaskla referencia sobre la API ABAP del AWS SDK para SAP.

`SynthesizeSpeech`

En el siguiente ejemplo de código, se muestra cómo utilizar `SynthesizeSpeech`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte SynthesizeSpeechla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon RDS con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon RDS.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateDBParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDBParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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_rdsdbparmgralrexfault.
    MESSAGE 'DB parameter group already exists.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
    MESSAGE 'DB parameter group quota exceeded.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulta Crear un DBParameter grupo en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DeleteDBParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDBParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " iv_dbparametergroupname = 'mydbparametergroup'
    TRY.
        lo_rds->deletedbparametergroup(
          iv_dbparametergroupname = iv_dbparametergroupname ).
        MESSAGE 'DB parameter group deleted.' TYPE 'I'.
      CATCH /aws1/cx_rdsdbprmgrnotfndfault.
        MESSAGE 'DB parameter group not found.' TYPE 'I'.
      CATCH /aws1/cx_rdsinvdbprmgrstatef00.
        MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulta Eliminar DBParameter un grupo en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeDBEngineVersions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBEngineVersions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " 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.
```

- Para obtener más información sobre la API, consulta la sección Describir DBEngine las versiones en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`DescribeDBParameterGroups`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBParameterGroups`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
    oo_result = lo_rds->describedbparametergroups(
      iv_dbparametergroupname = iv_dbparametergroupname ).
    MESSAGE 'DB parameter group retrieved.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre las API, consulta la sección Describir DBParameter los grupos en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

`DescribeDBParameters`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDBParameters`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_source               = 'user' (optional - filters by parameter source)
TRY.
    oo_result = lo_rds->describedbparameters(
      iv_dbparametergroupname = iv_dbparametergroupname
      iv_source               = iv_source ).
    DATA(lv_param_count) = lines( oo_result->get_parameters( ) ).
    MESSAGE |Retrieved { lv_param_count } parameters.| TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulta la sección Describe DBParameters en el AWS SDK para ver la referencia sobre la API ABAP de SAP.

`DescribeOrderableDBInstanceOptions`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeOrderableDBInstanceOptions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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.
```

- Para obtener más información sobre la API, consulta DescribeOrderableDBInstancelas opciones AWS del SDK para la referencia sobre la API ABAP de SAP.

`ModifyDBParameterGroup`

En el siguiente ejemplo de código, se muestra cómo utilizar `ModifyDBParameterGroup`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " iv_dbparametergroupname = 'mydbparametergroup'
    " it_parameters - table containing parameter objects with:
    "   - parametername = 'max_connections'
    "   - parametervalue = '100'
    "   - applymethod = 'immediate' or 'pending-reboot'
    TRY.
        oo_result = lo_rds->modifydbparametergroup(
          iv_dbparametergroupname = iv_dbparametergroupname
          it_parameters           = it_parameters ).
        MESSAGE 'DB parameter group modified.' TYPE 'I'.
      CATCH /aws1/cx_rdsdbprmgrnotfndfault.
        MESSAGE 'DB parameter group not found.' TYPE 'I'.
      CATCH /aws1/cx_rdsinvdbprmgrstatef00.
        MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre las API, consulta Modificar DBParameter un grupo en el AWS SDK para obtener una referencia sobre la API ABAP de SAP.

Ejemplos de Amazon Redshift con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Redshift.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateCluster`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateCluster`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree el clúster.

```
    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_rshclustalrdyexfault.
      MESSAGE 'Cluster already exists.' TYPE 'I'.
    CATCH /aws1/cx_rshclstquotaexcdfault.
      MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateClusterla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteCluster`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteCluster`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine el clúster.

```
  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.
```

- Para obtener más información sobre la API, consulte DeleteClusterla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeClusters`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeClusters`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Describir el clúster.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeClustersla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeStatement`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeStatement`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
    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.
```

- Para obtener más información sobre la API, consulte DescribeStatementla referencia sobre la API ABAP del AWS SDK para SAP.

`ExecuteStatement`

En el siguiente ejemplo de código, se muestra cómo utilizar `ExecuteStatement`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
        " Example values: iv_database_name = 'dev'
        " Example values: iv_user_name = 'awsuser'
        " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
        " Example values: it_parameter_list - SQL parameters for parameterized
queries

        " Only pass parameters if the list is not empty
        IF it_parameter_list IS NOT INITIAL.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
            it_parameters = it_parameter_list
          ).
        ELSE.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
          ).
        ENDIF.

        lv_statement_id = oo_result->get_id( ).
        MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
      CATCH /aws1/cx_rsdexecutestatementex.
        MESSAGE 'Statement execution error.' TYPE 'I'.
      CATCH /aws1/cx_rsdresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte ExecuteStatementla referencia sobre la API ABAP del AWS SDK para SAP.

`GetStatementResult`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetStatementResult`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Compruebe el resultado de la instrucción.

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
    " 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.
```

- Para obtener más información sobre la API, consulte GetStatementResultla referencia sobre la API ABAP del AWS SDK para SAP.

`ListDatabases`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListDatabases`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
    " Example values: iv_database_name = 'dev'
    " Example values: iv_database_user = 'awsuser'
    oo_result = lo_rsd->listdatabases(
      iv_clusteridentifier = iv_cluster_identifier
      iv_database = iv_database_name
      iv_dbuser = iv_database_user
    ).
    lt_databases = oo_result->get_databases( ).
    lv_db_count = lines( lt_databases ).
    MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
  CATCH /aws1/cx_rsddatabaseconnex.
    MESSAGE 'Database connection error.' TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte ListDatabasesla referencia sobre la API ABAP del AWS SDK para SAP.

`ModifyCluster`

En el siguiente ejemplo de código, se muestra cómo utilizar `ModifyCluster`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Modificar un clúster

```
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.
```

- Para obtener más información sobre la API, consulte ModifyClusterla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Rekognition con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Rekognition.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CompareFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `CompareFaces`.

Para obtener información, consulte Comparación de rostros en imágenes.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CompareFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`CreateCollection`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateCollection`.

Para obtener información, consulte Creación de una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateCollectionla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteCollection`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteCollection`.

Para obtener información, consulte Eliminación de una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteCollectionla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteFaces`.

Para obtener información, consulte Eliminación de rostros de una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeCollection`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeCollection`.

Para obtener información, consulte Descripción de una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DescribeCollectionla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectFaces`.

Para obtener información, consulte Detección de rostros en una imagen.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetectFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`DetectLabels`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectLabels`.

Para obtener información, consulte Detección de etiquetas en una imagen.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetectLabelsla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectModerationLabels`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectModerationLabels`.

Para obtener información, consulte Detección de imágenes inapropiadas.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetectModerationLabelsla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectText`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectText`.

Para obtener información, consulte Detección de texto en una imagen.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DetectTextla referencia sobre la API ABAP del AWS SDK para SAP.

`IndexFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `IndexFaces`.

Para obtener información, consulte Adición de rostros a una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte IndexFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListCollections`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListCollections`.

Para obtener información, consulte Enumerar colecciones.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListCollectionsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListFaces`.

Para obtener información, consulte Enumerar rostros en una colección.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`RecognizeCelebrities`

En el siguiente ejemplo de código, se muestra cómo utilizar `RecognizeCelebrities`.

Para obtener información, consulte Reconocimiento de famosos en una imagen.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte RecognizeCelebritiesla referencia sobre la API ABAP del AWS SDK para SAP.

`SearchFaces`

En el siguiente ejemplo de código, se muestra cómo utilizar `SearchFaces`.

Para obtener información, consulte Búsqueda de un rostro (ID de rostro).

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte SearchFaceslа referencia sobre la API ABAP del AWS SDK para SAP.

`SearchFacesByImage`

En el siguiente ejemplo de código, se muestra cómo utilizar `SearchFacesByImage`.

Para obtener información, consulte Búsqueda de un rostro (imagen).

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte SearchFacesByImagela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon S3 que utilizan SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon S3.

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Conceptos básicos
- Acciones
- Escenarios

Conceptos básicos

Conceptos básicos

En el siguiente ejemplo de código, se muestra cómo:

- Creación de un bucket y cargar un archivo en el bucket.
- Descargar un objeto desde un bucket.
- Copiar un objeto en una subcarpeta de un bucket.
- Obtención de una lista de los objetos de un bucket.
- Eliminación del bucket y todos los objetos que incluye.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_bucketalrdyexists.
        MESSAGE 'Bucket name already exists.' TYPE 'E'.
      CATCH /aws1/cx_s3_bktalrdyownedbyyou.
        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.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Acciones

`CopyObject`

En el siguiente ejemplo de código, se muestra cómo utilizar `CopyObject`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CopyObjectla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateBucket`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateBucket`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateBucketla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteBucket`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteBucket`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteBucketla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteBucketCors`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteBucketCors`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteBucketCorsla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteBucketLifecycle`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteBucketLifecycle`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteBucketLifecyclela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteBucketPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteBucketPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteBucketPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteObject`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteObject`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteObjectla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteObjects`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteObjects`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteObjectsla referencia sobre la API ABAP del AWS SDK para SAP.

`GetBucketAcl`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetBucketAcl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetBucketAcll la referencia sobre la API ABAP del AWS SDK para SAP.

`GetBucketCors`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetBucketCors`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetBucketCorsla referencia sobre la API ABAP del AWS SDK para SAP.

`GetBucketLifecycleConfiguration`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetBucketLifecycleConfiguration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetBucketLifecycleConfigurationla referencia sobre la API ABAP del AWS SDK para SAP.

`GetBucketPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetBucketPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte GetBucketPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`GetObject`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetObject`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetObjectla referencia sobre la API ABAP del AWS SDK para SAP.

`GetObjectAcl`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetObjectAcl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetObjectAcllla referencia sobre la API ABAP del AWS SDK para SAP.

`GetObjectLegalHold`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetObjectLegalHold`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetObjectLegalHoldla referencia sobre la API ABAP del AWS SDK para SAP.

`GetObjectLockConfiguration`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetObjectLockConfiguration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetObjectLockConfigurationla referencia sobre la API ABAP del AWS SDK para SAP.

`HeadBucket`

En el siguiente ejemplo de código, se muestra cómo utilizar `HeadBucket`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte HeadBucketla referencia sobre la API ABAP del AWS SDK para SAP.

`ListBuckets`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListBuckets`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte ListBucketsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListObjectVersions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListObjectVersions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte ListObjectVersionsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListObjectsV2`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListObjectsV2`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulta la ListObjectsversión 2 AWS del SDK para ver la referencia sobre la API ABAP de SAP.

`PutBucketAcl`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutBucketAcl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte PutBucketAcll la referencia sobre la API ABAP del AWS SDK para SAP.

`PutBucketCors`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutBucketCors`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutBucketCorsla referencia sobre la API ABAP del AWS SDK para SAP.

`PutBucketLifecycleConfiguration`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutBucketLifecycleConfiguration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Example: Expire objects with prefix 'logs/' after 30 days
        lo_s3->putbucketlifecycleconf(
          iv_bucket = iv_bucket_name
          io_lifecycleconfiguration = NEW /aws1/cl_s3_bucketlcconf(
            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.
```

- Para obtener más información sobre la API, consulte PutBucketLifecycleConfigurationla referencia sobre la API ABAP del AWS SDK para SAP.

`PutBucketPolicy`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutBucketPolicy`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte PutBucketPolicyla referencia sobre la API ABAP del AWS SDK para SAP.

`PutBucketVersioning`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutBucketVersioning`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutBucketVersioningla referencia sobre la API ABAP del AWS SDK para SAP.

`PutObject`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutObject`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
"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.
```

- Para obtener más información sobre la API, consulte PutObjectla referencia sobre la API ABAP del AWS SDK para SAP.

`PutObjectAcl`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutObjectAcl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutObjectAcll la referencia sobre la API ABAP del AWS SDK para SAP.

`PutObjectLegalHold`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutObjectLegalHold`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutObjectLegalHoldla referencia sobre la API ABAP del AWS SDK para SAP.

`PutObjectLockConfiguration`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutObjectLockConfiguration`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutObjectLockConfigurationla referencia sobre la API ABAP del AWS SDK para SAP.

`PutObjectRetention`

En el siguiente ejemplo de código, se muestra cómo utilizar `PutObjectRetention`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte PutObjectRetentionla referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Crear una URL prefirmada

En el siguiente ejemplo de código se muestra cómo crear una URL prefirmada para Amazon S3 y cargar un objeto.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear solicitudes previamente firmadas para obtener (GET) objetos de S3.

```
" 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( ).
```

SageMaker Ejemplos de IA que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con SageMaker IA.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`CreateEndpoint`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateEndpoint`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_sgmresourcelimitexcd.
        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_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateEndpointla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateModel`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateModel`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.

   "Create an ABAP object for the container image based on input variables."
   lo_primarycontainer = NEW #( iv_image = iv_container_image
                                iv_modeldataurl = iv_model_data_url ).

   "Create an Amazon SageMaker model."
   TRY.
       oo_result = lo_sgm->createmodel(        " oo_result is returned for testing
purposes. "
         iv_executionrolearn = iv_execution_role_arn
         iv_modelname = iv_model_name
         io_primarycontainer = lo_primarycontainer ).
       MESSAGE 'Model created.' TYPE 'I'.
     CATCH /aws1/cx_sgmresourcelimitexcd.
       MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateModella referencia sobre la API ABAP del AWS SDK para SAP.

`CreateTrainingJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTrainingJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_sgmalgorithmspec.
   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.
```

- Para obtener más información sobre la API, consulte CreateTrainingJobla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateTransformJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTransformJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte CreateTransformJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteEndpoint`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteEndpoint`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    "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.
```

- Para obtener más información sobre la API, consulte DeleteEndpointla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteModel`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteModel`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DeleteModella referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeTrainingJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeTrainingJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeTrainingJobla referencia sobre la API ABAP del AWS SDK para SAP.

`ListAlgorithms`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListAlgorithms`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListAlgorithmsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListModels`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListModels`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListModelsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListNotebookInstances`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListNotebookInstances`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListNotebookInstancesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTrainingJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTrainingJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
```

```
        oo_result = lo_sgm->listtrainingjobs(       " oo_result is returned for
 testing purposes. "
          iv_namecontains = iv_name_contains
          iv_maxresults = iv_max_results ).
        MESSAGE 'Retrieved list of training jobs.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte ListTrainingJobsla referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Introducción a modelos y puntos de conexión

En el siguiente ejemplo de código, se muestra cómo:

- Inicie un trabajo de formación y cree un modelo de SageMaker IA.
- Crear una configuración de punto de conexión.
- Cree un punto de conexión y, a continuación, limpie los recursos.

SDK para SAP ABAP

> **Note**
>
> Hay más en marcha GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_sgmalgorithmspec.
    DATA lo_resource_config  TYPE REF TO /aws1/cl_sgmresourceconfig.
    DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
    DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
    DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
    DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
    DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
    DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
    DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
    DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
    DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
    DATA lv_model_data_url TYPE /aws1/sgmurl.

    lv_model_data_url = iv_s3_output_path && iv_training_job_name && '/output/
model.tar.gz'.

    "Create ABAP internal table for hyperparameters based on input variables."
    "These hyperparameters are based on Amazon SageMaker built-in algorithm -
 XGBoost"
    lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
    INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w )  INTO TABLE
 lt_hyperparameters.

    lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
    INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w )  INTO TABLE
 lt_hyperparameters.

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

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

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

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

```
   lo_hyperparameters_w = NEW #( iv_value = iv_hp_num_round ).
   INSERT VALUE #( key = 'num_round' value = lo_hyperparameters_w )  INTO TABLE
lt_hyperparameters.

   "Create ABAP internal table for data based on input variables."
   "Training data."
   lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

   lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

   lo_trn_channel = NEW #( iv_channelname = 'train'
                           io_datasource = lo_trn_datasource
                           iv_compressiontype = iv_trn_data_compressiontype
                           iv_contenttype = iv_trn_data_contenttype ).
   INSERT lo_trn_channel INTO TABLE lt_input_data_config.

   "Validation data."
   lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).

   lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

   lo_val_channel = NEW #( iv_channelname = 'validation'
                           io_datasource = lo_val_datasource
                           iv_compressiontype = iv_val_data_compressiontype
                           iv_contenttype = iv_val_data_contenttype ).
   INSERT lo_val_channel INTO TABLE lt_input_data_config.

   "Create an ABAP object for algorithm specification based on input variables."
   lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                       iv_traininginputmode =
iv_training_input_mode ).

   "Create an ABAP object for resource configuration."
   lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                               iv_instancetype = iv_instance_type
                               iv_volumesizeingb = iv_volume_sizeingb ).
```

```
   "Create an ABAP object for output data configuration."
   lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

   "Create an ABAP object for stopping condition."
   lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

   TRY.
       lo_sgm->createtrainingjob(
         iv_trainingjobname           = iv_training_job_name
         iv_rolearn                   = iv_role_arn
         it_hyperparameters           = lt_hyperparameters
         it_inputdataconfig           = lt_input_data_config
         io_algorithmspecification    = lo_algorithm_specification
         io_outputdataconfig          = lo_output_data_config
         io_resourceconfig            = lo_resource_config
         io_stoppingcondition         = lo_stopping_condition ).
       MESSAGE 'Training job created.' TYPE 'I'.
     CATCH /aws1/cx_sgmresourceinuse.
       MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcenotfound.
       MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcelimitexcd.
       MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
   ENDTRY.

   "Wait for training job to be completed."
   lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
   WHILE lo_training_result->get_trainingjobstatus( ) <> 'Completed'.
     IF sy-index = 30.
       EXIT.               "Maximum 900 seconds."
     ENDIF.
     WAIT UP TO 30 SECONDS.
     lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
   ENDWHILE.

   "Create ABAP object for the container image based on input variables."
   lo_primarycontainer = NEW #( iv_image = iv_training_image
                                iv_modeldataurl = lv_model_data_url ).

   "Create an Amazon SageMaker model."
   TRY.
```

```
        lo_sgm->createmodel(
          iv_executionrolearn = iv_role_arn
          iv_modelname = iv_model_name
          io_primarycontainer = lo_primarycontainer ).
        MESSAGE 'Model created.' TYPE 'I'.
      CATCH /aws1/cx_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

    "Create an endpoint production variant."
    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_sgmresourcelimitexcd.
        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.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - CreateEndpoint
  - CreateEndpointConfig
  - CreateModel
  - CreateTrainingJob
  - DeleteEndpoint
  - DeleteEndpointConfig
  - DeleteModel
  - DescribeEndpoint
  - DescribeTrainingJob

Ejemplos de Secrets Manager con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el AWS SDK para SAP ABAP with Secrets Manager.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`BatchGetSecretValue`

En el siguiente ejemplo de código, se muestra cómo utilizar `BatchGetSecretValue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte BatchGetSecretValuela referencia sobre la API ABAP del AWS SDK para SAP.

`GetSecretValue`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetSecretValue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte GetSecretValuela referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon SES que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon SES.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateReceiptFilter`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateReceiptFilter`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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.
```

- Para obtener más información sobre la API, consulte CreateReceiptFilterla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateReceiptRule`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateReceiptRule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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_sesreceiptrule(
  iv_name = iv_rule_name
  iv_enabled = abap_true
  it_recipients = it_recipients
  it_actions = lt_actions
).

TRY.
    lo_ses->createreceiptrule(
      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.
```

- Para obtener más información sobre la API, consulte CreateReceiptRulela referencia sobre la API ABAP del AWS SDK para SAP.

`CreateReceiptRuleSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateReceiptRuleSet`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_ses->createreceiptruleset( 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.
```

- Para obtener más información sobre la API, consulte CreateReceiptRuleSetla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteIdentity`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteIdentity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteIdentityla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteReceiptFilter`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteReceiptFilter`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_ses->deletereceiptfilter( 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.
```

- Para obtener más información sobre la API, consulte DeleteReceiptFilterla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteReceiptRule`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteReceiptRule`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_ses->deletereceiptrule(
      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.
```

- Para obtener más información sobre la API, consulte DeleteReceiptRulela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteReceiptRuleSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteReceiptRuleSet`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_ses->deletereceiptruleset( 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.
```

- Para obtener más información sobre la API, consulte DeleteReceiptRuleSetla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeReceiptRuleSet`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeReceiptRuleSet`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_ses->describereceiptruleset(
      iv_rulesetname = iv_rule_set_name
    ).
    MESSAGE 'Receipt rule set described 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.
```

- Para obtener más información sobre la API, consulte DescribeReceiptRuleSetla referencia sobre la API ABAP del AWS SDK para SAP.

`GetIdentityVerificationAttributes`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetIdentityVerificationAttributes`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
       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.
```

- Para obtener más información sobre la API, consulte GetIdentityVerificationAttributesla referencia sobre la API ABAP del AWS SDK para SAP.

`GetTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`ListIdentities`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListIdentities`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListIdentitiesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListReceiptFilters`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListReceiptFilters`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListReceiptFilterslа referencia sobre la API ABAP del AWS SDK para SAP.

`ListTemplates`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTemplates`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte ListTemplateslа referencia sobre la API ABAP del AWS SDK para SAP.

`SendEmail`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendEmail`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
" 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.
```

- Para obtener más información sobre la API, consulte SendEmailla referencia sobre la API ABAP del AWS SDK para SAP.

`SendTemplatedEmail`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendTemplatedEmail`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte SendTemplatedEmailla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte UpdateTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`VerifyDomainIdentity`

En el siguiente ejemplo de código, se muestra cómo utilizar `VerifyDomainIdentity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte VerifyDomainIdentityla referencia sobre la API ABAP del AWS SDK para SAP.

`VerifyEmailIdentity`

En el siguiente ejemplo de código, se muestra cómo utilizar `VerifyEmailIdentity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte VerifyEmailIdentityla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de la API v2 de Amazon SES que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon SES API v2.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateContact`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateContact`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateContactla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateContactList`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateContactList`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateContactListla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateEmailIdentity`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateEmailIdentity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateEmailIdentityla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateEmailTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateEmailTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateEmailTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteContactList`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteContactList`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte DeleteContactListla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteEmailIdentity`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteEmailIdentity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteEmailIdentityla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteEmailTemplate`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteEmailTemplate`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteEmailTemplatela referencia sobre la API ABAP del AWS SDK para SAP.

`ListContacts`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListContacts`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListContactsla referencia sobre la API ABAP del AWS SDK para SAP.

`SendEmail`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendEmail`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Envía un mensaje.

```
    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.
```

Envía un mensaje mediante una plantilla.

```
    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.
```

- Para obtener más información sobre la API, consulte SendEmailla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon SNS con el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon SNS.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`CreateTopic`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateTopic`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte CreateTopicla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTopic`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTopic`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte DeleteTopicla referencia sobre la API ABAP del AWS SDK para SAP.

`GetTopicAttributes`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetTopicAttributes`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte GetTopicAttributesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListSubscriptions`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListSubscriptions`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte ListSubscriptionsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTopics`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTopics`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte ListTopicsla referencia sobre la API ABAP del AWS SDK para SAP.

`Publish`

En el siguiente ejemplo de código, se muestra cómo utilizar `Publish`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

Publica un mensaje con los atributos de un tema.

```
   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.
```

Publica un mensaje multiformato en un tema.

```
   " 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(              " oo_result is returned for
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.
```

- Para obtener detalles sobre la API, consulte Publish en la Referencia de la API de AWS SDK para SAP ABAP.

`SetSubscriptionAttributes`

En el siguiente ejemplo de código, se muestra cómo utilizar `SetSubscriptionAttributes`.

SDK para SAP ABAP

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte SetSubscriptionAttributesla referencia sobre la API ABAP del AWS SDK para SAP.

`SetTopicAttributes`

En el siguiente ejemplo de código, se muestra cómo utilizar `SetTopicAttributes`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte SetTopicAttributesla referencia sobre la API ABAP del AWS SDK para SAP.

`Subscribe`

En el siguiente ejemplo de código, se muestra cómo utilizar `Subscribe`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Suscriba una dirección de correo electrónico a un tema.

```
   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_snssubscriptionlmte00.
       MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
   ENDTRY.
```

- Para obtener detalles sobre la API, consulte Subscribe en la Referencia de la API de AWS SDK para SAP ABAP.

`Unsubscribe`

En el siguiente ejemplo de código, se muestra cómo utilizar `Unsubscribe`.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener detalles sobre la API, consulte Unsubscribe en la Referencia de la API de AWS SDK para SAP ABAP.

Escenarios

Creación y publicación en un tema FIFO

El siguiente ejemplo de código indica cómo crear y publicar en un tema FIFO de Amazon SNS.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crea un tema de FIFO, suscribe una cola FIFO de Amazon SQS al tema y publica un mensaje en un tema de Amazon SNS.

```
    " Creates a FIFO topic. "
    DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsmap_w=>tt_topicattributesmap.
    DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmap_w=>ts_topicattributesmap_maprow.
    ls_tpc_attributes-key = 'FifoTopic'.
    ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmap_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.
```

- Para obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para SAP ABAP.
  - CreateTopic
  - Publish
  - Subscribe

Publicación de un mensaje SMS

En el siguiente ejemplo de código se muestra cómo publicar mensajes SMS mediante Amazon SNS.

SDK para SAP ABAP

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    " iv_phone_number = '+12065550101' - Phone number in E.164 format
    TRY.
        oo_result = lo_sns->publish(              " oo_result is returned for
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.
```

- Para obtener detalles sobre la API, consulte Publish en la Referencia de la API de AWS SDK para SAP ABAP.

Ejemplos de Amazon SQS que utilizan el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon SQS.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`CreateQueue`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateQueue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear una cola estándar de Amazon SQS.

```
    TRY.
        oo_result = lo_sqs->createqueue( iv_queuename = iv_queue_name ).        "
 oo_result is returned for testing purposes. "
        MESSAGE 'SQS queue created.' TYPE 'I'.
      CATCH /aws1/cx_sqsqueuedeldrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
 queue with the same name.' TYPE 'E'.
      CATCH /aws1/cx_sqsqueuenameexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
    ENDTRY.
```

Crear una cola de Amazon SQS que espere a que llegue un mensaje.

```
    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_queuename = iv_queue_name
                it_attributes = lt_attributes ).
        MESSAGE 'SQS queue created.' TYPE 'I'.
      CATCH /aws1/cx_sqsqueuedeldrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
 queue with the same name.' TYPE 'E'.
      CATCH /aws1/cx_sqsqueuenameexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateQueuela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteMessage`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteMessage`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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.
```

- Para obtener más información sobre la API, consulte DeleteMessagela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteMessageBatch`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteMessageBatch`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_sqsemptybatchrequest.
        MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
      CATCH /aws1/cx_sqsinvbatchentryid.
        MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
      CATCH /aws1/cx_sqstoomanyentriesin00.
        MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteMessageBatchla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteQueue`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteQueue`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        lo_sqs->deletequeue( iv_queueurl = iv_queue_url ).
        MESSAGE 'SQS queue deleted' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteQueuela referencia sobre la API ABAP del AWS SDK para SAP.

`GetQueueUrl`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetQueueUrl`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_sqs->getqueueurl( iv_queuename = iv_queue_name ).        "
oo_result is returned for testing purposes. "
        MESSAGE 'Queue URL retrieved.' TYPE 'I'.
      CATCH /aws1/cx_sqsqueuedoesnotexist.
        MESSAGE 'The requested queue does not exist.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GetQueueUrlla referencia sobre la API ABAP del AWS SDK para SAP.

`ListQueues`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListQueues`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   TRY.
       oo_result = lo_sqs->listqueues( ).        " oo_result is returned for
testing purposes. "
       MESSAGE 'Retrieved list of queues.' TYPE 'I'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte ListQueuesla referencia sobre la API ABAP del AWS SDK para SAP.

`ReceiveMessage`

En el siguiente ejemplo de código, se muestra cómo utilizar `ReceiveMessage`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Recibir un mensaje de una cola de Amazon SQS.

```
   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.
```

Recibir un mensaje de una cola de Amazon SQS mediante el soporte de sondeos largos.

```
   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.
```

- Para obtener más información sobre la API, consulte ReceiveMessagela referencia sobre la API ABAP del AWS SDK para SAP.

`SendMessage`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendMessage`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte SendMessagela referencia sobre la API ABAP del AWS SDK para SAP.

`SendMessageBatch`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendMessageBatch`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   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_sqsbtcentidsnotdist00.
       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_sqsemptybatchrequest.
       MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
     CATCH /aws1/cx_sqsinvbatchentryid.
       MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
     CATCH /aws1/cx_sqstoomanyentriesin00.
       MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
     CATCH /aws1/cx_sqsunsupportedop.
       MESSAGE 'Operation not supported.' TYPE 'E'.
   ENDTRY.
```

- Para obtener más información sobre la API, consulte SendMessageBatchla referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Creación y publicación en un tema FIFO

El siguiente ejemplo de código indica cómo crear y publicar en un tema FIFO de Amazon SNS.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crea un tema de FIFO, suscribe una cola FIFO de Amazon SQS al tema y publica un mensaje en un tema de Amazon SNS.

```
    " Creates a FIFO topic. "
    DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsmap_w=>tt_topicattributesmap.
    DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmap_w=>ts_topicattributesmap_maprow.
    ls_tpc_attributes-key = 'FifoTopic'.
    ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmap_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.
```

- Para obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para SAP ABAP.
  - CreateTopic
  - Publish
  - Subscribe

Ejemplos de Step Functions con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP with Step Functions.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateActivity`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateActivity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_sfnactivityalrdyex.
        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.
```

- Para obtener más información sobre la API, consulte CreateActivityla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateStateMachine`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateStateMachine`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte CreateStateMachinela referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteActivity`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteActivity`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  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.
```

- Para obtener más información sobre la API, consulte DeleteActivityla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteStateMachine`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteStateMachine`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_sfn->deletestatemachine(
      iv_statemachinearn = iv_state_machine_arn
    ).
    MESSAGE 'State machine deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfnvalidationex.
    MESSAGE 'Validation error occurred.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteStateMachinela referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeExecution`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeExecution`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeExecutionla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeStateMachine`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeStateMachine`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    oo_result = lo_sfn->describestatemachine(
      iv_statemachinearn = 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.
```

- Para obtener más información sobre la API, consulte DescribeStateMachinela referencia sobre la API ABAP del AWS SDK para SAP.

`GetActivityTask`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetActivityTask`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetActivityTaskla referencia sobre la API ABAP del AWS SDK para SAP.

`ListActivities`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListActivities`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListActivitiesla referencia sobre la API ABAP del AWS SDK para SAP.

`ListStateMachines`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListStateMachines`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListStateMachinesla referencia sobre la API ABAP del AWS SDK para SAP.

`SendTaskSuccess`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendTaskSuccess`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_sfn->sendtasksuccess(
      iv_tasktoken = iv_task_token
      iv_output = iv_task_response
    ).
    MESSAGE 'Task success sent successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid task token.' TYPE 'E'.
  CATCH /aws1/cx_sfntaskdoesnotexist.
    MESSAGE 'Task does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidoutput.
    MESSAGE 'Invalid task output.' TYPE 'E'.
  CATCH /aws1/cx_sfntasktimedout.
    MESSAGE 'Task timed out.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte SendTaskSuccessla referencia sobre la API ABAP del AWS SDK para SAP.

`StartExecution`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartExecution`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    DATA(lo_result) = lo_sfn->startexecution(
      iv_statemachinearn = iv_state_machine_arn
      iv_input = iv_input
    ).
    ov_execution_arn = lo_result->get_executionarn( ).
    MESSAGE 'Execution started successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnstatemachinedoes00.
    MESSAGE 'State machine does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidexecinput.
    MESSAGE 'Invalid execution input.' TYPE 'E'.
  CATCH /aws1/cx_sfnexeclimitexceeded.
    MESSAGE 'Execution limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte StartExecutionla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Systems Manager que utilizan el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Systems Manager.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateDocument`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateDocument`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateDocumentla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateMaintenanceWindow`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateMaintenanceWindow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte CreateMaintenanceWindowla referencia sobre la API ABAP del AWS SDK para SAP.

`CreateOpsItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateOpsItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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_ssmopsitemalrdyexex.
        MESSAGE 'OpsItem already exists.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateOpsItemla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteDocument`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteDocument`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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_ssmassocdinstances.
        MESSAGE 'Document has associated instances.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteDocumentla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteMaintenanceWindow`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteMaintenanceWindow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteMaintenanceWindowla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteOpsItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteOpsItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteOpsItemla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeDocument`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeDocument`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte DescribeDocumentla referencia sobre la API ABAP del AWS SDK para SAP.

`DescribeOpsItems`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeOpsItems`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        " Create filter for OpsItem ID
        DATA(lt_filters) = VALUE /aws1/cl_ssmopsitemfilter=>tt_opsitemfilters(
          ( NEW /aws1/cl_ssmopsitemfilter(
              iv_key = 'OpsItemId'
              it_values = VALUE /aws1/
cl_ssmopsitemfiltvals_w=>tt_opsitemfiltervalues(
                ( NEW /aws1/cl_ssmopsitemfiltvals_w( iv_value = iv_ops_item_id ) )
              )
              iv_operator = 'Equal'
            ) )
        ).

        " Use paginator to get all results
        DATA(lo_paginator) = lo_ssm->get_paginator( ).
        DATA(lo_iterator) = lo_paginator->describeopsitems(
          it_opsitemfilters = lt_filters ).

        rv_found = abap_false.

        WHILE lo_iterator->has_next( ).
          DATA(lo_result) = CAST /aws1/cl_ssmdescropsitemsrsp( lo_iterator-
>get_next( ) ).
          LOOP AT lo_result->get_opsitemsummaries( ) INTO DATA(lo_item).
            DATA(lv_title) = lo_item->get_title( ).
            DATA(lv_status) = lo_item->get_status( ).
            MESSAGE |The OpsItem title is { lv_title } and the status is
 { lv_status }| TYPE 'I'.
            rv_found = abap_true.
          ENDLOOP.
        ENDWHILE.
      CATCH /aws1/cx_ssminternalservererr.
        MESSAGE 'Internal server error occurred.' TYPE 'I'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DescribeOpsItemsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListCommandInvocations`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListCommandInvocations`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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'.
          ENDLOOP.
        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.
```

- Para obtener más información sobre la API, consulte ListCommandInvocationsla referencia sobre la API ABAP del AWS SDK para SAP.

`SendCommand`

En el siguiente ejemplo de código, se muestra cómo utilizar `SendCommand`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte SendCommandla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateMaintenanceWindow`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateMaintenanceWindow`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte UpdateMaintenanceWindowla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateOpsItem`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateOpsItem`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_ssm->updateopsitem(
        iv_opsitemid = iv_ops_item_id
        iv_title = iv_title
        iv_description = iv_description
        iv_status = iv_status ).
    MESSAGE 'OpsItem updated.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemnotfoundex.
    MESSAGE 'OpsItem not found.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsiteminvparamex.
    MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte UpdateOpsItemla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Textract que utilizan el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Textract.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`AnalyzeDocument`

En el siguiente ejemplo de código, se muestra cómo utilizar `AnalyzeDocument`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "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.
```

- Para obtener más información sobre la API, consulte AnalyzeDocumentla referencia sobre la API ABAP del AWS SDK para SAP.

`DetectDocumentText`

En el siguiente ejemplo de código, se muestra cómo utilizar `DetectDocumentText`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "Detects text in the input document."
   "Amazon Textract can detect lines of text and the words that make up a line of
 text."
   "The input document must be in one of the following image formats: JPEG, PNG,
 PDF, or TIFF."

   "Create an ABAP object for the Amazon S3 object."
   DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
     iv_name   = iv_s3object ).

   "Create an ABAP object for the document."
   DATA(lo_document) = NEW /aws1/cl_texdocument( io_s3object = lo_s3object ).
   "Analyze document stored in Amazon S3."
   TRY.
       oo_result = lo_tex->detectdocumenttext( io_document = lo_document ).
 "oo_result is returned for testing purposes."
       LOOP AT oo_result->get_blocks( ) INTO DATA(lo_block).
         IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR,'.
           MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
         ENDIF.
       ENDLOOP.
       DATA(lo_metadata) = oo_result->get_documentmetadata( ).
       MESSAGE 'The number of pages in the document is ' && lo_metadata-
>ask_pages( ) TYPE 'I'.
       MESSAGE 'Detect document text completed.' TYPE 'I'.
     CATCH /aws1/cx_texaccessdeniedex.
       MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
     CATCH /aws1/cx_texbaddocumentex.
       MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
     CATCH /aws1/cx_texdocumenttoolargeex.
       MESSAGE 'The document is too large.' TYPE 'E'.
     CATCH /aws1/cx_texinternalservererr.
       MESSAGE 'Internal server error.' TYPE 'E'.
```

```
      CATCH /aws1/cx_texinvalidparameterex.
        MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
      CATCH /aws1/cx_texinvalids3objectex.
        MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte DetectDocumentTextla referencia sobre la API ABAP del AWS SDK para SAP.

`GetDocumentAnalysis`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetDocumentAnalysis`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "Gets the results for an Amazon Textract"
   "asynchronous operation that analyzes text in a document."
   TRY.
       oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).
"oo_result is returned for testing purposes."
       WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.
         IF sy-index = 10.
           EXIT.               "Maximum 300 seconds.
         ENDIF.
         WAIT UP TO 30 SECONDS.
         oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).
       ENDWHILE.
```

```
    DATA(lt_blocks) = oo_result->get_blocks( ).
    LOOP AT lt_blocks INTO DATA(lo_block).
      IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR,'.
        MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
      ENDIF.
    ENDLOOP.
    MESSAGE 'Document analysis retrieved.' TYPE 'I'.
  CATCH /aws1/cx_texaccessdeniedex.
    MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
  CATCH /aws1/cx_texinternalservererr.
    MESSAGE 'Internal server error.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidjobidex.
    MESSAGE 'Job ID is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidkmskeyex.
    MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texinvalidparameterex.
    MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
  CATCH /aws1/cx_texinvalids3objectex.
    MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
  CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
  CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte GetDocumentAnalysisla referencia sobre la API ABAP del AWS SDK para SAP.

`StartDocumentAnalysis`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartDocumentAnalysis`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "Starts the asynchronous analysis of an input document for relationships"
   "between detected items such as key-value pairs, tables, and selection
elements."

   "Create ABAP objects for feature type."
   "Add TABLES to return information about the tables."
   "Add FORMS to return detected form data."
   "To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

   DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).
   "Create an ABAP object for the Amazon S3 object."
   DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
     iv_name   = iv_s3object ).
   "Create an ABAP object for the document."
   DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

   "Start async document analysis."
   TRY.
       oo_result = lo_tex->startdocumentanalysis(      "oo_result is returned for
testing purposes."
         io_documentlocation     = lo_documentlocation
         it_featuretypes         = lt_featuretypes ).
       DATA(lv_jobid) = oo_result->get_jobid( ).

       MESSAGE 'Document analysis started.' TYPE 'I'.
     CATCH /aws1/cx_texaccessdeniedex.
       MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
     CATCH /aws1/cx_texbaddocumentex.
       MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
     CATCH /aws1/cx_texdocumenttoolargeex.
       MESSAGE 'The document is too large.' TYPE 'E'.
     CATCH /aws1/cx_texidempotentprmmis00.
       MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
     CATCH /aws1/cx_texinternalservererr.
       MESSAGE 'Internal server error.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidkmskeyex.
       MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidparameterex.
       MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
```

```
      CATCH /aws1/cx_texinvalids3objectex.
        MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texlimitexceededex.
        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.
```

- Para obtener más información sobre la API, consulte StartDocumentAnalysisla referencia sobre la API ABAP del AWS SDK para SAP.

`StartDocumentTextDetection`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartDocumentTextDetection`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    "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_texlimitexceededex.
      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.
```

- Para obtener más información sobre la API, consulte StartDocumentTextDetectionla referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Introducción al análisis de documentos

En el siguiente ejemplo de código, se muestra cómo:

- Inicie el análisis asíncrono.

- Obtenga el análisis de documentos.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "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_texlimitexceededex.
        MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.

    "Get job ID from the output."
    DATA(lv_jobid) = lo_start_result->get_jobid( ).

    "Wait for job to complete."
    oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).     " oo_result
is returned for testing purposes. "
    WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.
      IF sy-index = 10.
        EXIT.               "Maximum 300 seconds."
      ENDIF.
      WAIT UP TO 30 SECONDS.
      oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).
    ENDWHILE.

    DATA(lt_blocks) = oo_result->get_blocks( ).
    LOOP AT lt_blocks INTO DATA(lo_block).
      IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR,'.
        MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
      ENDIF.
    ENDLOOP.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.

- GetDocumentAnalysis
- StartDocumentAnalysis

Ejemplos de Amazon Transcribe con el SDK para SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Transcribe.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones

Acciones

`CreateVocabulary`

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateVocabulary`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    IF it_phrases IS NOT INITIAL.
      oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
```

```
            it_phrases = it_phrases ).
        ELSEIF iv_vocab_file_uri IS NOT INITIAL.
          oo_result = lo_tnb->createvocabulary(
            iv_vocabularyname = iv_vocabulary_name
            iv_languagecode = iv_language_code
            iv_vocabularyfileuri = iv_vocab_file_uri ).
        ENDIF.
        MESSAGE 'Custom vocabulary created.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
        MESSAGE lo_limit_ex TYPE 'I'.
        RAISE EXCEPTION lo_limit_ex.
      CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
        MESSAGE lo_internal_ex TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
      CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
        MESSAGE lo_conflict_ex TYPE 'I'.
        RAISE EXCEPTION lo_conflict_ex.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte CreateVocabularyla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteTranscriptionJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteTranscriptionJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        lo_tnb->deletetranscriptionjob( iv_job_name ).
        MESSAGE 'Transcription job deleted.' TYPE 'I'.
```

```
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Para obtener más información sobre la API, consulte DeleteTranscriptionJobla referencia sobre la API ABAP del AWS SDK para SAP.

`DeleteVocabulary`

En el siguiente ejemplo de código, se muestra cómo utilizar `DeleteVocabulary`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte DeleteVocabularyla referencia sobre la API ABAP del AWS SDK para SAP.

`GetTranscriptionJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetTranscriptionJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetTranscriptionJobla referencia sobre la API ABAP del AWS SDK para SAP.

`GetVocabulary`

En el siguiente ejemplo de código, se muestra cómo utilizar `GetVocabulary`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte GetVocabularyla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTranscriptionJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTranscriptionJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
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.
```

- Para obtener más información sobre la API, consulte ListTranscriptionJobsla referencia sobre la API ABAP del AWS SDK para SAP.

`ListVocabularies`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListVocabularies`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    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.
```

- Para obtener más información sobre la API, consulte ListVocabularieslа referencia sobre la API ABAP del AWS SDK para SAP.

`StartTranscriptionJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartTranscriptionJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
        DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
        IF iv_vocabulary_name IS NOT INITIAL.
          lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
        ENDIF.

        oo_result = lo_tnb->starttranscriptionjob(
          iv_transcriptionjobname = iv_job_name
          io_media = lo_media
          iv_mediaformat = iv_media_format
          iv_languagecode = iv_language_code
          io_settings = lo_settings ).

        MESSAGE 'Transcription job started.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
        MESSAGE lo_limit_ex TYPE 'I'.
```

```
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte StartTranscriptionJobla referencia sobre la API ABAP del AWS SDK para SAP.

`UpdateVocabulary`

En el siguiente ejemplo de código, se muestra cómo utilizar `UpdateVocabulary`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
  TRY.
      IF it_phrases IS NOT INITIAL.
        oo_result = lo_tnb->updatevocabulary(
          iv_vocabularyname = iv_vocabulary_name
          iv_languagecode = iv_language_code
          it_phrases = it_phrases ).
      ELSEIF iv_vocab_file_uri IS NOT INITIAL.
        oo_result = lo_tnb->updatevocabulary(
          iv_vocabularyname = iv_vocabulary_name
          iv_languagecode = iv_language_code
          iv_vocabularyfileuri = iv_vocab_file_uri ).
      ENDIF.
      MESSAGE 'Vocabulary updated.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
```

```
      MESSAGE lo_limit_ex TYPE 'I'.
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex TYPE 'I'.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte UpdateVocabularyla referencia sobre la API ABAP del AWS SDK para SAP.

Ejemplos de Amazon Translate que utilizan el SDK de SAP ABAP

Los siguientes ejemplos de código muestran cómo realizar acciones e implementar escenarios comunes mediante el uso del AWS SDK para SAP ABAP con Amazon Translate.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

En cada ejemplo se incluye un enlace al código de origen completo, con instrucciones de configuración y ejecución del código en el contexto.

Temas

- Acciones
- Escenarios

Acciones

`DescribeTextTranslationJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeTextTranslationJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "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.
```

- Para obtener más información sobre la API, consulte DescribeTextTranslationJobla referencia sobre la API ABAP del AWS SDK para SAP.

`ListTextTranslationJobs`

En el siguiente ejemplo de código, se muestra cómo utilizar `ListTextTranslationJobs`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

   DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.

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

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

- Para obtener más información sobre la API, consulte ListTextTranslationJobsla referencia sobre la API ABAP del AWS SDK para SAP.

`StartTextTranslationJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StartTextTranslationJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "Starts an asynchronous batch translation job."
```

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

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

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

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

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

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

```
    ENDTRY.
```

- Para obtener más información sobre la API, consulte StartTextTranslationJobla referencia sobre la API ABAP del AWS SDK para SAP.

`StopTextTranslationJob`

En el siguiente ejemplo de código, se muestra cómo utilizar `StopTextTranslationJob`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "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.
```

- Para obtener más información sobre la API, consulte StopTextTranslationJobla referencia sobre la API ABAP del AWS SDK para SAP.

`TranslateText`

En el siguiente ejemplo de código, se muestra cómo utilizar `TranslateText`.

SDK para SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

- Para obtener más información sobre la API, consulte TranslateTextla referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Introducción a los trabajos de traducción

En el siguiente ejemplo de código, se muestra cómo:

- Inicie un trabajo de traducción por lotes asincrónico.
- Espere a que el trabajo asíncrono finalice.
- Describa el trabajo asíncrono.

SDK de SAP ABAP

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

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

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

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

    TRY.
```

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

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

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

    TRY.
        oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
 for testing purposes."
          iv_jobid        = lv_jobid ).
```

```
        MESSAGE 'Job description retrieved.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
    ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del AWS SDK para SAP ABAP.
  - DescribeTextTranslationJob
  - StartTextTranslationJob

Uso del AWS SDK para SAP ABAP Knowledge MCP Server

El AWS SDK para SAP ABAP Knowledge MCP Server es un recurso de conocimiento especializado que permite a los asistentes de codificación de IA generar código ABAP preciso para las integraciones. AWS Al conectar su IDE habilitado para IA a este servidor, puede describir lo que desea crear en lenguaje natural y recibir un código ABAP sintácticamente correcto que invoque correctamente el SDK para SAP ABAP. AWS

El servidor MCP utiliza el Model Context Protocol (MCP), un estándar abierto para conectar los asistentes de IA con fuentes de conocimiento externas. El servidor se actualiza a diario en sincronía con las versiones AWS del SDK para SAP ABAP, por lo que su IDE siempre tendrá acceso a información actualizada y precisa del SDK.

El AWS SDK para SAP ABAP Knowledge MCP Server está disponible sin coste adicional. Con el AWS SDK para SAP ABAP, solo paga por los AWS recursos y servicios que consume en sus aplicaciones de SAP.

Temas

- Ventajas
- Configuración del servidor MCP
- Qué puede hacer el servidor MCP
- Consideraciones importantes

Ventajas

El AWS SDK para SAP ABAP Knowledge MCP Server ofrece las siguientes ventajas a los desarrolladores de ABAP.

- Generación de código precisa: su asistente de codificación de IA adquiere un conocimiento profundo del AWS SDK para SAP ABAP: los patrones, las firmas de métodos, los tipos de datos y el manejo de excepciones específicos del SDK para SAP ABAP. Esto elimina los errores más comunes y reduce considerablemente los errores de compilación y el tiempo de depuración.
- Descubrimiento instantáneo del SDK: el servidor MCP expone la base de conocimientos completa sobre el AWS SDK para SAP ABAP, que abarca más de 200 servicios. AWS Puede descubrir los servicios, operaciones y tipos de datos disponibles sin necesidad de buscar manualmente en la documentación.

- Siempre actualizada: la base de conocimientos se actualiza a diario de forma sincronizada con las versiones del SDK. Los nuevos servicios y las firmas de métodos actualizadas están disponibles de forma inmediata en su IDE.
- Instalación cero: la configuración solo requiere pegar una URL en la configuración MCP del IDE. No se requiere la instalación de software local, la administración de dependencias ni un proceso de aprobación de TI.

Configuración del servidor MCP

Para conectar su IDE al AWS SDK para el servidor MCP de SAP ABAP Knowledge, añada la siguiente configuración a los ajustes de MCP de su IDE. El formato de configuración y la ubicación del archivo pueden variar según el IDE; consulte la documentación del MCP del IDE para obtener más información.

El siguiente ejemplo muestra la configuración de Kiro. Añada esto al archivo de configuración MCP de su espacio de trabajo `.kiro/settings/mcp.json` o al archivo de configuración a nivel de usuario: `~/.kiro/settings/mcp.json`

```
{
  "mcpServers": {
    "abap-sdk-knowledge": {
      "url": "https://sdk-for-sap-abap-knowledge-mcp.global.api.aws/mcp",
      "disabled": false,
      "autoApprove": ["*"]
    }
  }
}
```

La `autoApprove` configuración permite que su IDE utilice las herramientas del servidor MCP sin solicitar la aprobación de cada solicitud. Esto es seguro porque el servidor MCP es de solo lectura y no modifica ningún dato ni ejecuta código.

Tras guardar la configuración, el IDE se conectará automáticamente al servidor MCP. En la mayoría IDEs de los casos, no es necesario reiniciar. Puedes verificar la conexión haciendo una pregunta a tu asistente de IA sobre el AWS SDK para SAP ABAP, como «¿Qué AWS servicios están disponibles en el AWS SDK para SAP ABAP?»

Si su IDE no se conecta al servidor MCP, compruebe que es compatible con el Model Context Protocol y que la sintaxis del archivo de configuración coincide con los requisitos del IDE.

> **Note**
>
> El punto final del servidor MCP utiliza HTTPS y no requiere autenticación. Su IDE se comunica con el servidor solo cuando usted hace preguntas a su asistente de IA relacionadas con el uso AWS del SDK para SAP ABAP. Su lógica empresarial patentada y el código de aplicación de SAP permanecen dentro de su entorno de desarrollo.

Qué puede hacer el servidor MCP

Una vez conectado, su asistente de codificación de IA puede utilizar las siguientes capacidades proporcionadas por el servidor MCP.

- Detección de servicios: enumere y busque todos los AWS servicios disponibles en el AWS SDK para SAP ABAP, incluidos los metadatos de los servicios y los nombres de las clases de fábrica.
- Detalles de la operación: recupere las firmas de los métodos ABAP, los parámetros de entrada, los tipos de devolución y los tipos de excepciones para cualquier operación del SDK.
- Definiciones de tipos de datos: busque las definiciones de estructura ABAP para los tipos de datos del SDK, incluidos los nombres y tipos de campos.
- Ejemplos de código: recupera ejemplos de código ABAP en funcionamiento para operaciones específicas, que abarquen los casos de uso más comunes de cada servicio.
- Creación de sesiones y clientes: genere código ABAP repetitivo para crear sesiones del SDK, inicializar los clientes de servicio y llamar a las operaciones con un manejo correcto de las excepciones.

Por ejemplo, puedes enviar solicitudes a tu asistente de IA como las siguientes.

- «Escribe un código ABAP para leer un mensaje de una cola de SQS».
- «Genere código ABAP para invocar un modelo Amazon Bedrock para la generación de texto».
- «Muéstreme cómo subir un archivo a Amazon S3 desde ABAP».

El asistente de IA utiliza los conocimientos del servidor MCP para generar código ABAP sintácticamente correcto que siga las convenciones ABAP del AWS SDK para SAP, incluidas la creación adecuada de sesiones, la inicialización del cliente y los patrones de gestión de excepciones.

Consideraciones importantes

Tenga en cuenta lo siguiente cuando utilice el AWS SDK para SAP ABAP Knowledge MCP Server.

- El servidor MCP proporciona acceso de solo lectura a los conocimientos sobre el SDK. No ejecuta código ABAP ni se conecta a sistemas SAP.
- El servidor MCP solo cubre el AWS SDK para SAP ABAP. No proporciona asistencia general de programación de ABAP que no esté relacionada con el AWS SDK para el uso de SAP ABAP.
- El servidor MCP no admite bibliotecas ABAP personalizadas o de terceros.
- El servidor MCP proporciona la documentación basada en la última versión del SDK, que puede no coincidir con la versión instalada en el sistema SAP.
- Los ejemplos de código generados son puntos de partida. Revise y pruebe todo el código generado antes de usarlo en producción.
- La calidad y el formato del código generado pueden variar en función de las capacidades del asistente de IA del IDE.

Además del AWS SDK para SAP ABAP Knowledge MCP Server que se describe en esta página, AWS también ofrece el AWS Knowledge MCP Server, que proporciona amplia información de AWS servicio, orientación sobre la arquitectura y solución de problemas en todos los servicios. AWS Los dos servidores se complementan entre sí y son especialmente eficaces cuando se utilizan juntos.

Seguridad en AWS SDK para SAP ABAP

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que se ejecuta Servicios de AWS en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS SDK para SAP ABAP, consulte Servicios de AWS Alcance by Compliance Servicios de AWS .
- Seguridad en la nube: su responsabilidad viene determinada por lo Servicio de AWS que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta sección abarca los siguientes temas.

Temas

- Autenticación del sistema SAP activada AWS
- Prácticas recomendadas de seguridad de IAM
- Autorizaciones de SAP
- Operaciones seguras
- Uso de la autenticación de clave de acceso secreta con cifrado SSF
- Uso de certificados con funciones de IAM en cualquier lugar.
- Uso del perfil de origen para el acceso entre cuentas
- Uso de SAP Credential Store

Autenticación del sistema SAP activada AWS

Antes de que un sistema SAP pueda realizar llamadas AWS en nombre de los usuarios de SAP, el sistema SAP debe autenticarse AWS ante él. AWS SDK para SAP ABAP admite los siguientes tres métodos de autenticación básicos que se seleccionan en la configuración del perfil del SDK en`IMG`.

AWS El SDK para SAP ABAP: la edición BTP solo se puede autenticar con el the section called “Autenticación de clave de acceso secreta” método que utiliza SAP Credential Store.

Para escenarios de acceso entre cuentas, el SDK para SAP ABAP también admite el perfil de origen, lo que permite encadenar varios supuestos de funciones de IAM entre cuentas mediante cualquiera de los métodos de autenticación básicos. Para obtener más información, consulte the section called “Autenticación del perfil de origen para el acceso entre cuentas”.

Temas

- Autenticación de metadatos de instancias de Amazon EC2
- Autenticación de clave de acceso secreta
- Autenticación basada en certificados mediante funciones de IAM en cualquier lugar
- Autenticación del perfil de origen para el acceso entre cuentas
- Siguiente paso

Autenticación de metadatos de instancias de Amazon EC2

Los sistemas SAP que se ejecutan en Amazon EC2 pueden adquirir credenciales de corta duración que rotan automáticamente a partir de metadatos de instancias de Amazon EC2. Para obtener más información, consulte Uso de credenciales para metadatos de instancias de Amazon EC2.

Recomendamos encarecidamente este método de autenticación al utilizar el SDK de SAP ABAP. Para habilitarlo, el administrador de Basis debe habilitar la comunicación HTTP saliente. No se necesitan más configuraciones base.

> **Note**
>
> Este método de autenticación solo se aplica si sus sistemas SAP se ejecutan en Amazon EC2. Los sistemas SAP alojados en las instalaciones o en otros entornos de nube no pueden autenticarse mediante este método.

Autenticación de clave de acceso secreta

Con este método, utiliza un ID de clave de acceso y una clave de acceso secreta para autenticar su sistema SAP en AWS. El sistema SAP inicia sesión AWS con un usuario de IAM. Para obtener más información, consulte Administración de las claves de acceso de los usuarios de IAM.

El administrador de Basis recibe un ID de clave de acceso y una clave de acceso secreta del administrador de IAM AWS . El sistema SAP debe estar configurado para almacenar el ID de la clave de acceso y la clave de acceso secreta.

- Proteja, almacene y reenvíe (SSF)
  - Utilice la funcionalidad SSF para autenticar el AWS SDK de SAP ABAP. Para obtener más información, consulte Digital Signatures and Encryption.
  - También puede probar la funcionalidad de los SSF `envelope` y `develope` con el informe `SSF02`. Para obtener más información, consulte Prueba de la instalación de SSF.
  - Los pasos para configurar SSF para el SDK de SAP ABAP se describen en la transacción `/AWS1/IMG`. Vaya a Requisitos técnicos previos y, a continuación, seleccione Configuración adicional para sistemas en las instalaciones. Para ver los pasos de configuración detallados, consulte Uso de la autenticación por clave de acceso secreta con cifrado SSF.
- Almacén de credenciales de SAP
  - Utilice SAP Credential Store para autenticar el AWS SDK de la edición SAP ABAP - BTP. Para obtener más información, consulte ¿Qué es SAP Credential Store?
  - Consulte Uso de SAP Credential Store para ver los pasos de configuración.

Autenticación basada en certificados mediante funciones de IAM en cualquier lugar

Puede utilizar un certificado X.509 emitido por su autoridad de certificación (CA) para la autenticación con AWS Identity and Access Management Roles Anywhere. El certificado debe estar configurado en `STRUST`. La CA debe estar registrada en funciones de IAM en cualquier lugar como un anclaje de confianza y debe crearse un perfil para especificar los roles y políticas que asumirían las funciones de IAM en cualquier lugar. Para obtener más información, consulte Crear un ancla y un perfil de confianza en AWS Identity and Access Management Roles Anywhere.

Para ver los pasos detallados sobre cómo utilizar funciones de IAM en cualquier lugar con el SDK para SAP ABAP, consulte Uso de certificados con funciones de IAM en cualquier lugar.

> **Note**
>
> La revocación de certificados solo se admite mediante el uso de listas de revocación de certificados importadas. Para obtener más información, consulte Revocación.

Autenticación del perfil de origen para el acceso entre cuentas

El perfil de origen es una función avanzada que permite encadenar varios supuestos de funciones de IAM entre AWS cuentas. Con este método, un perfil asume una función y, a continuación, asume otra, y así sucesivamente, de forma similar al `source_profile` parámetro de la AWS CLI.

El perfil de origen funciona con cualquiera de los tres métodos de autenticación básicos (metadatos de la instancia, clave de acceso secreta o basado en certificados). El primer perfil de la cadena debe usar uno de estos métodos básicos y los perfiles subsiguientes de la cadena deben usar las credenciales del perfil anterior para asumir el siguiente rol.

Esto resulta útil en situaciones de acceso entre cuentas en las que es necesario recorrer varias AWS cuentas para llegar a los recursos objetivo. Para ver los pasos de configuración detallados, consulte Uso del perfil de origen para el acceso entre cuentas.

Siguiente paso

Tras autenticar su sistema SAP AWS, el SDK para SAP ABAP asume automáticamente la función de `sts:assumeRole` IAM adecuada para la función empresarial del usuario de SAP.

Prácticas recomendadas de seguridad de IAM

El administrador de IAM será responsable de las siguientes tres áreas clave.

- Garantizar que el sistema SAP pueda autenticarse con metadatos de Amazon EC2 o credenciales de clave secreta.
- Garantizar que el sistema SAP cuente con los permisos que necesita para promocionarse con `sts:assumeRole`.
- Para cada rol de IAM lógico, se crea un rol de IAM para los usuarios de SAP con los permisos necesarios para realizar las funciones empresariales (por ejemplo, los permisos necesarios para Amazon S3, DynamoDB u otros servicios). Estos son los roles que los usuarios de SAP asumirán.

Para obtener más información, consulte el capítulo Seguridad en SAP Lens: Marco de AWS Well-Architected.

Temas

- Práctica recomendada para el perfil de instancia de Amazon EC2
- Roles de IAM para usuarios de SAP
- Consideraciones de seguridad del perfil de origen

Práctica recomendada para el perfil de instancia de Amazon EC2

La instancia de Amazon EC2 en la que se ejecuta el sistema SAP tiene un conjunto de autorizaciones basadas en su perfil de instancia. Por lo general, el perfil de instancia solo necesita tener permisos para realizar llamadas a `sts:assumeRole`, a fin de que el sistema SAP pueda asumir los roles de IAM específicos de la empresa, según sea necesario. Este ascenso a otros roles garantiza que un programa ABAP pueda asumir un rol que otorgue al usuario el privilegio mínimo necesario para realizar su trabajo. Por ejemplo, un perfil de instancia podría contener la siguiente instrucción.

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"
      ]
   }
 ]
}
```

El ejemplo anterior permite que el sistema SAP asuma las funciones de IAM para el CFO, el AUDITOR o el usuario REPORTING. AWS El SDK elegirá la función de IAM correcta para el usuario en función de la función de PFCG del usuario en SAP.

El perfil de instancia de Amazon EC2 también se puede utilizar para otras funciones.

- AWS Backint Agent para SAP HANA
- SAP apuesta por la alta disponibilidad de AWS con enrutamiento de direcciones IP de superposición

Estas soluciones también pueden requerir permisos de `sts:assumeRole` para roles específicos de copia de seguridad o conmutación por error o pueden requerir que los permisos se asignen directamente al perfil de instancia.

Roles de IAM para usuarios de SAP

El programa ABAP necesita permisos para realizar la tarea del usuario: leer una tabla de DynamoDB, invocar Amazon Textract en un objeto PDF en Amazon S3, ejecutar una función. AWS Lambda En todas ellas se utiliza el mismo modelo de seguridad. AWS SDKs Puede usar un rol de IAM existente que se usó para otro SDK de AWS .

El analista de negocios de SAP solicitará al administrador de IAM el arn:aws: de un rol de IAM para cada rol lógico necesario. Por ejemplo, en un escenario financiero, el analista de negocios puede definir los siguientes roles lógicos de IAM.

- `CFO`
- `AUDITOR`
- `REPORTING`

El administrador de IAM definirá los roles de IAM para cada rol lógico de IAM.

`CFO`

- `arn:aws:iam::0123456789:role/finance-cfo`
- Permisos de lectura y escritura para un bucket de Amazon S3
- Permisos de lectura y escritura para una base de datos DynamoDB

`AUDITOR`

- `arn:aws:iam::0123456789:role/finance-auditor`
- Permisos de lectura para un bucket de Amazon S3
- Permisos de lectura para una base de datos DynamoDB

`REPORTING`

- `arn:aws:iam::0123456789:role/finance-reporting`
- Permisos de lectura para una base de datos DynamoDB
- Sin permisos para el bucket de Amazon S3

El analista empresarial ingresará los roles de IAM en una tabla de asignación para asignar los roles de IAM lógicos a los roles de IAM físicos.

Los roles de IAM para los usuarios de SAP deben permitir la acción `sts:assumeRole` para entidades principales de confianza. Las entidades principales de confianza pueden variar según la forma en que se autentique el sistema SAP en AWS. Para obtener más información, consulte Especificación de una entidad principal.

A continuación se muestran algunos ejemplos de los escenarios de SAP más comunes.

- Sistema SAP que se ejecuta en Amazon EC2 con un perfil de instancia asignado; aquí, un perfil de instancia de Amazon EC2 se adjunta a un rol de IAM.

  JSON

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

- Sistemas SAP que se ejecutan en Amazon EC2 sin un perfil de instancia: en este caso, Amazon EC2 asume los roles de los usuarios de SAP.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [ "ec2.amazonaws.com" ]
            }
        }
    ]
}
```

- Sistemas SAP que se ejecutan en las instalaciones: los sistemas SAP que se ejecutan en las instalaciones solo pueden autenticarse mediante la clave de acceso secreta. Para obtener más información, consulte Autenticación del sistema SAP en AWS.

  En este caso, cualquier rol de IAM que asuma un usuario de SAP debe tener una relación de confianza en el usuario de SAP.

  JSON

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

Consideraciones de seguridad del perfil de origen

Al utilizar el perfil de origen:

Administración de roles de IAM

Fundamental: las funciones de IAM en las cadenas de perfiles de origen deben gestionarse estrictamente para evitar el acceso no autorizado y la escalada de privilegios:

- Aplique el principio de privilegios mínimos: conceda solo los permisos mínimos necesarios para el propósito específico de cada función
- Audite periódicamente los permisos de los puestos: revise y actualice las políticas de los puestos trimestralmente o cuando cambien los requisitos
- Supervise el uso de los roles: utilícelo para rastrear las llamadas a la AssumeRole API e identificar patrones inusuales
- Limite las relaciones de confianza: limite los directores que pueden asumir cada función solo a aquellos que necesitan acceso absolutamente
- Utilice las condiciones en las políticas de confianza: añada condiciones como la IP de origen, los requisitos de MFA o las restricciones basadas en el tiempo, según proceda
- Documente los propósitos de las funciones: mantenga una documentación clara sobre el caso de uso previsto de cada función y los permisos necesarios

Autorización y control de acceso

- Asegúrese de que todos los perfiles intermedios de la cadena tengan configuradas las políticas de confianza adecuadas
- Los usuarios deben tener `/AWS1/SESS` autorización para TODOS los perfiles de la cadena, incluidos los perfiles intermedios
- Cada función de IAM debe confiar explícitamente en la función anterior de la cadena

Salvaguardias técnicas

- El SDK impone una profundidad de cadena máxima de 5 perfiles para evitar llamadas excesivas a la API STS
- Las referencias circulares se detectan y evitan automáticamente

- El método de autenticación del perfil base se valida para garantizar que utiliza un método estándar (INST, SSF o RLA)

Para obtener más información sobre la configuración del perfil de origen, consulte Uso del perfil de origen para el acceso entre cuentas.

Autorizaciones de SAP

La autorización necesaria para configurar el SDK depende de la edición del SDK.

Temas

- Autorizaciones de configuración
- Autorizaciones de SAP para usuarios finales

Autorizaciones de configuración

Consulte las siguientes pestañas para obtener más información.

SDK for SAP ABAP

Se requieren las siguientes autorizaciones para configurar el SDK de SAP ABAP.

- S_TCODE
  - TCD = /AWS1/IMG
- S_TABU_DIS
  - ACTVT = 02, 03
  - DICBERCLS

    Elija entre los siguientes grupos de autorización.

    - /AWS1/CFG- AWS SDK para SAP ABAP v1 - Config
    - /AWS1/MOD- AWS SDK para SAP ABAP v1 - Tiempo de ejecución
    - /AWS1/PFL- AWS SDK para SAP ABAP v1 - Perfil del SDK
    - /AWS1/RES- AWS SDK para SAP ABAP v1 - Recursos lógicos
    - /AWS1/TRC- AWS SDK para SAP ABAP v1 - Rastreo

SDK for SAP ABAP - BTP edition

Siga los siguientes pasos para permitir que el SDK para SAP ABAP - edición BTP acceda a la configuración.

1. Cree un nuevo rol empresarial mediante la plantilla de rol `SAP_BR_BPC_EXPERT` empresarial. Esta plantilla proporciona acceso a la aplicación de configuración empresarial personalizada.
2. En Detalles generales de las funciones, vaya a Categorías de acceso y seleccione Sin restricciones para leer, escribir y valorar la ayuda.
3. Vaya a la pestaña Catálogo empresarial y asigne el catálogo `/AWS1/RTBTP_BCAT` empresarial para proporcionar acceso a la configuración del SDK.
4. Vaya a la pestaña Usuarios empresariales y asigne a los usuarios empresariales el acceso a la configuración del SDK.

Autorizaciones de SAP para usuarios finales

Requisito previo: definir los perfiles del SDK

Antes de que el administrador de seguridad de SAP pueda definir sus funciones, el analista de negocios definirá los perfiles del SDK en Transaction `/AWS1/IMG` para el AWS SDK de SAP ABAP o la aplicación de configuración empresarial personalizada para el SDK de SAP ABAP, edición BTP. Por lo general, un perfil del SDK se denominará de acuerdo con su función empresarial: ZFINANCE, ZBILLING, ZMFG, ZPAYROLL, etc. Para cada perfil del SDK, el analista de negocios definirá los roles de IAM lógicos con nombres abreviados, como CFO (Director financiero), AUDITOR o INFORMES. El administrador de seguridad de IAM los asignará a los roles de IAM reales.

Defina las funciones empresariales o de PFCG

> **Note**
>
> Los roles de PFCG se denominan roles de negocio en el entorno SAP BTP o ABAP.

A continuación, el administrador de seguridad de SAP añadirá un objeto de autorización `/AWS1/SESS` para conceder el acceso a un perfil del SDK.

Objeto de autenticación `/AWS1/SESS`

- Campo /AWS1/PROF = ZFINANCE

A los usuarios también se les deben asignar roles de IAM lógicos para cada perfil del SDK, según su función laboral. Por ejemplo, un auditor financiero con acceso a los informes podría estar autorizado para desempeñar un rol de IAM lógico denominado AUDITOR.

Objeto de autenticación /AWS1/LROL

- Campo /AWS1/PROF = ZFINANCE
- Campo /AWS1/LROL = AUDITOR

Mientras tanto, el director financiero, con autorizaciones de lectura/escritura, podría tener un rol de PFCG que le autorice el rol lógico de CFO.

Objeto de autenticación /AWS1/LROL

- Campo /AWS1/PROF = ZFINANCE
- Campo /AWS1/LROL = CFO

En general, un usuario solo debe estar autorizado para un rol de IAM lógico por perfil del SDK. Si un usuario está autorizado para más de un rol de IAM (por ejemplo, si el CFO está autorizado para ambos CFO roles de IAM AUDITOR lógicos), el AWS SDK resuelve el problema asegurándose de que el rol de mayor prioridad (número de secuencia más bajo) surta efecto.

Como ocurre con todos los escenarios de seguridad, los usuarios deben tener el privilegio mínimo para realizar sus funciones laborales. Una estrategia sencilla para gestionar los roles de PFCG sería asignar un nombre a los roles de PFCG únicos según el perfil del SDK y el rol lógico que autoricen. Por ejemplo, el rol Z_AWS_PFL_ZFINANCE_CFO otorga acceso al perfil ZFINANCE y al rol de IAM lógico CFO. Luego, estos roles individuales se pueden asignar a roles compuestos que definan las funciones laborales. Cada empresa tiene su propia estrategia de gestión de roles, y le recomendamos que defina una estrategia de PFCG que se adapte a sus necesidades.

Operaciones seguras

Cifrado de datos en reposo

AWS Las claves de acceso secretas se utilizan para autenticar el SDK. Se cifran mediante la función SSF o el almacén de credenciales de SAP.

Cifrado de datos en tránsito

Todas las llamadas se Servicios de AWS cifran con HTTPS. El ICM de SAP gestiona la conexión HTTPS. AWS Los certificados deben ser de confianza en STRUST.

Uso de la API

Cuando un usuario de ABAP asume un rol con `sts:assumeRole`, el nombre de la sesión se titula `USERID-SID-MANDT`, donde:

- `USERID` es el usuario ABAP de la variable `SY-UNAME`.
- `SID` es el ID del sistema ABAP de la variable `SY-SYSID`.
- `MANDT` es el cliente ABAP de la variable `SY-MANDT`.

El nombre de la sesión aparece CloudTrail como nombre de usuario. Esto garantiza que las llamadas a la API de un usuario de ABAP puedan rastrearse hasta el sistema, el cliente y el usuario que iniciaron la llamada. Para obtener más información, consulte ¿Qué es AWS CloudTrail?

Uso de la autenticación de clave de acceso secreta con cifrado SSF

Los sistemas SAP locales (o los sistemas que se ejecutan en otras nubes) se pueden autenticar AWS mediante la autenticación con clave de acceso secreta. AWS Identity and Access Management El mecanismo de almacenamiento y reenvío seguro (SSF) de SAP se utiliza para cifrar y almacenar de forma segura AWS las credenciales (ID de clave de acceso y clave de acceso secreta) de un usuario de IAM. El sistema SAP inicia sesión AWS con un usuario de IAM; consulte Gestión de claves de acceso para usuarios de IAM para obtener más información.

Requisitos previos

Se deben cumplir los siguientes requisitos previos antes de comenzar la configuración:

- El administrador de IAM debe crear los roles de IAM para los usuarios de SAP. Los roles deben tener permisos para llamar a los necesarios. Servicios de AWS Para obtener más información, consulte Prácticas recomendadas de seguridad de IAM.
- Cree una autorización para ejecutar la transacción `/AWS1/IMG`. Para obtener más información, consulte Autorizaciones para configuración.

Procedimiento

Siga estas instrucciones para configurar el almacenamiento de credenciales cifrado con SSF:

Steps

- Paso 1: Defina una aplicación SSF para el almacenamiento de credenciales
- Paso 2: Defina los parámetros de cifrado para la aplicación SSF
- Paso 3: Crear la aplicación PSE para SSF
- Paso 4: Asigne una aplicación SSF al AWS SDK para SAP ABAP
- Paso 5: Configure el perfil del SDK para usar credenciales cifradas con SSF

Paso 1: Defina una aplicación SSF para el almacenamiento de credenciales

1. Ejecute el código de transacción `SE16` para definir una aplicación SSF.
2. Introduzca el nombre de la tabla `SSFAPPLIC` y seleccione Nuevas entradas.
3. Introduzca los siguientes detalles:
   - APPLIC: `ZAWS1` (nombre de la aplicación SSF).
   - DESCRIPCIÓN: `SSF Encryption for the AWS SDK para SAP ABAP` (descripción).
   - Elija `Selected(X)` una opción para los campos restantes.
4. Seleccione Guardar.

Paso 2: Defina los parámetros de cifrado para la aplicación SSF

1. Ejecute el código de transacción `/n/AWS1/IMG` para iniciar la Guía de implementación (IMG). AWS SDK para SAP ABAP
2. Expanda el nodo IMG AWS SDK para SAP ABAP Configuración > Requisitos técnicos previos > Configuración adicional para sistemas locales.

3. Ejecute la actividad IMG Definir los parámetros de SSF.
4. Seleccione Nuevas entradas y elija la aplicación de SSF creada en el paso anterior. Seleccione Guardar.
5. Modifique el algoritmo hash a SHA256(o una versión superior) y el algoritmo de cifrado a AES256-CBC. Conserve el resto de la configuración de forma predeterminada y seleccione Guardar.

Estos ajustes de cifrado se utilizarán para cifrar AWS las credenciales de forma segura.

Paso 3: Crear la aplicación PSE para SSF

1. Ejecute la `/n/AWS1/IMG` transacción y seleccione AWS SDK para SAP ABAP Configuración > Requisitos técnicos previos > Configuración adicional para sistemas locales.
2. Ejecute la actividad de IMG Create PSE for SSF Application, que lo dirigirá a la. `STRUST transaction` Seleccione Editar.
3. Seleccione con el botón derecho la aplicación de SSF creada en the section called “Paso 1” y elija Crear. Conserve el resto de la configuración predeterminada y seleccione Continuar. Asegúrese de elegir `RSA` y no DSA como algoritmo.

Paso 4: Asigne una aplicación SSF al AWS SDK para SAP ABAP

1. Ejecute la `/n/AWS1/IMG` transacción y seleccione AWS SDK para SAP ABAP Configuración > Requisitos técnicos previos > Configuración adicional para sistemas locales.
2. Ejecute la actividad IMG Asigne una aplicación SSF al. AWS SDK para SAP ABAP
3. Seleccione Nuevas entradas e introduzca la aplicación SSF creada en. the section called “Paso 1” Seleccione Guardar.

Paso 5: Configure el perfil del SDK para usar credenciales cifradas con SSF

1. Ejecute la `/n/AWS1/IMG` transacción y seleccione AWS SDK para SAP ABAP Configuración > Configuraciones de la aplicación.
2. Ejecute el perfil del SDK de actividades de IMG.
3. Seleccione Nuevas entradas. Introduzca el nombre y la descripción del perfil. Seleccione Guardar.
4. Resalte la entrada que creó y haga clic en la rama del árbol de autenticación y configuración.
5. Seleccione Nuevas entradas e introduzca los siguientes detalles:

- SID: el ID del sistema SAP.
- Cliente: el cliente del sistema SAP.
- ID de escenario: seleccione el `DEFAULT` escenario creado por su administrador de Basis.
- AWS Región: AWS región a la que desea realizar llamadas.
- Método de autenticación: `Credentials from SSF Storage` selecciónalo en el menú desplegable y selecciona Guardar. Seleccione Establecer credenciales e introduzca el ID de clave de acceso y la clave de acceso secreta del usuario de IAM.
- Desactivar las funciones de IAM: manténgala como predeterminada, es decir, desmarcada.
- Seleccione Guardar.

6. Haga clic en la rama del árbol de asignación de roles de IAM. Seleccione Nuevas entradas. Introduzca el número de secuencia, el nombre del rol de IAM lógico y el ARN del rol de IAM proporcionados por el administrador de IAM. AWS Seleccione Guardar.

Para obtener más información, consulte Configuración de aplicaciones.

Uso de certificados con funciones de IAM en cualquier lugar.

El sistema SAP se puede autenticar AWS mediante la autenticación basada en certificados con AWS Identity and Access Management Roles Anywhere. Debe configurar el certificado en `STRUST` y el perfil del SDK en `/AWS1/IMG`.

Requisitos previos

Deben cumplirse los siguientes requisitos previos antes de comenzar a configurar la certificación.

- El certificado X.509 emitido por su autoridad de certificación (CA) debe cumplir los siguientes requisitos.
  - El certificado de firma debe ser un certificado v3.
  - La cadena no debe superar los 5 certificados.
  - El certificado debe ser compatible con los algoritmos RSA o ECDSA.
- Registre su entidad de certificación en IAM Roles Anywhere como base de confianza y cree un perfil para especificar las funciones de IAM en cualquier lugar. roles/policies Para obtener más información, consulte Crear un ancla y un perfil de confianza en AWS Identity and Access Management Roles Anywhere.

- El administrador de IAM debe crear los roles de IAM para los usuarios de SAP. Los roles deben tener permisos para llamar a los necesarios Servicios de AWS. Para obtener más información, consulte Prácticas recomendadas de seguridad de IAM.
- Cree una autorización para ejecutar la transacción `/AWS1/IMG`. Para obtener más información, consulte Autorizaciones para configuración.

Procedimiento

Siga estas instrucciones para configurar la autenticación basada en certificados.

Steps

- Paso 1: Definir una aplicación de SSF mediante Secure Store and Forward (SSF) de SAP
- Paso 2: Definir los parámetros de SSF
- Paso 3: Crear el PSE y la solicitud de certificado
- Paso 4: Importar la respuesta del certificado al PSE correspondiente
- Paso 5: Configurar el perfil del SDK para usar funciones de IAM en cualquier lugar

Paso 1: Definir una aplicación de SSF mediante Secure Store and Forward (SSF) de SAP

1. Ejecute el código de transacción `SE16` para definir una aplicación de SSF.
2. Introduzca el nombre de la tabla `SSFAPPLIC` y seleccione Nuevas entradas.
3. Introduzca un nombre para la aplicación de SSF en el campo `APPLIC`, una descripción en el campo `DESCRIPT` y seleccione una opción `Selected (X)` para los campos restantes.

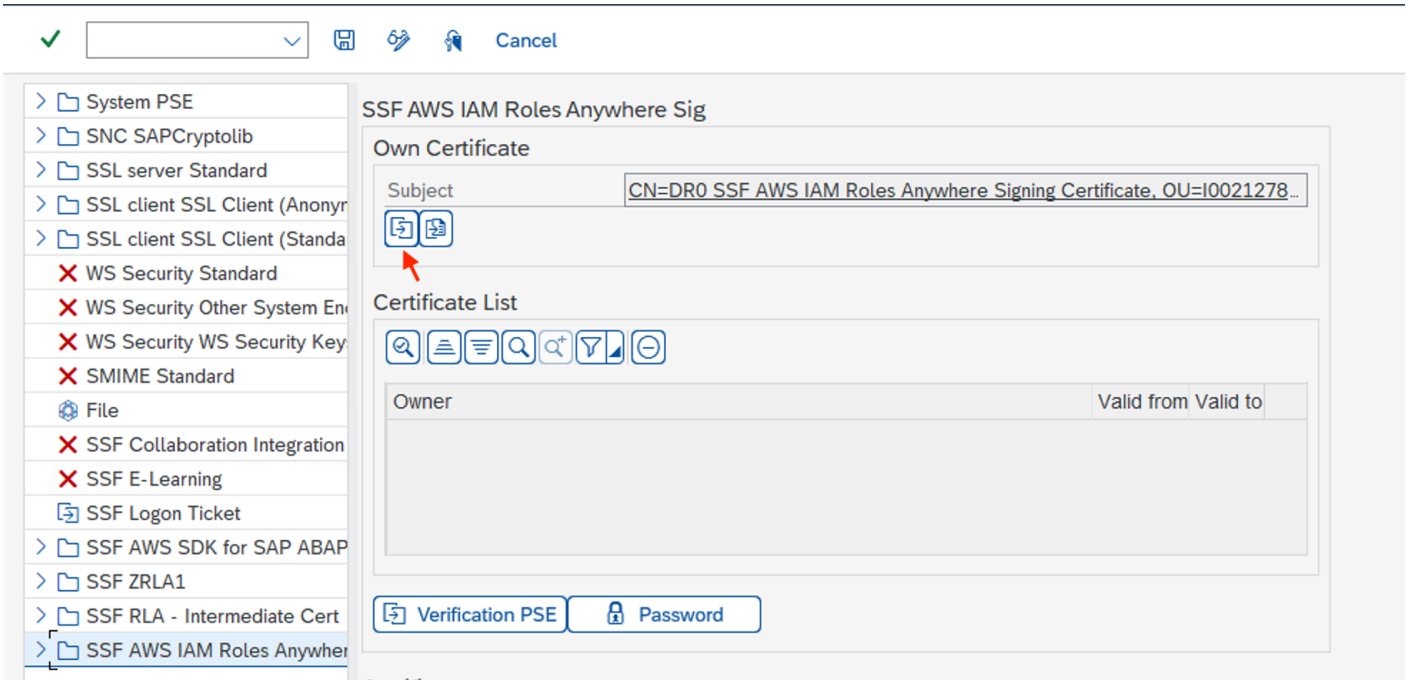
Paso 2: Definir los parámetros de SSF

1. Ejecute `/n/AWS1/IMG` la Guía de AWS SDK para SAP ABAP implementación (IMG) para iniciar.
2. Vaya a Configuración de AWS SDK para SAP ABAP > Requisitos técnicos previos > Configuración adicional para sistemas en las instalaciones.
3. Ejecute la actividad de IMG Establecer parámetros de SSF.
4. Seleccione Nuevas entradas y elija la aplicación de SSF creada en el paso anterior. Seleccione Guardar.

5. Modifique el algoritmo hash a SHA256-CBC y el algoritmo de cifrado a AES256-CBC. Conserve el resto de la configuración de forma predeterminada y seleccione Guardar.

Paso 3: Crear el PSE y la solicitud de certificado

1. Ejecute la transacción de `/n/AWS1/IMG` y seleccione Configuración de AWS SDK para SAP ABAP > Requisitos técnicos previos > Configuración adicional para sistemas en las instalaciones.
2. Ejecute la actividad de IMG `Create PSE for SSF Application`.
3. Seleccione Editar para la transacción `STRUST`.
4. Seleccione con el botón derecho la aplicación de SSF creada en the section called “Paso 1” y elija Crear. Conserve el resto de la configuración predeterminada y seleccione Continuar.
5. Seleccione Crear solicitud de certificado. Vea la siguiente imagen. Conserve las opciones predeterminadas y seleccione Continuar. Copie o exporte la solicitud de certificado generada y proporciónesela a su CA. Su CA verifica la solicitud y responde con un certificado de clave pública firmado.



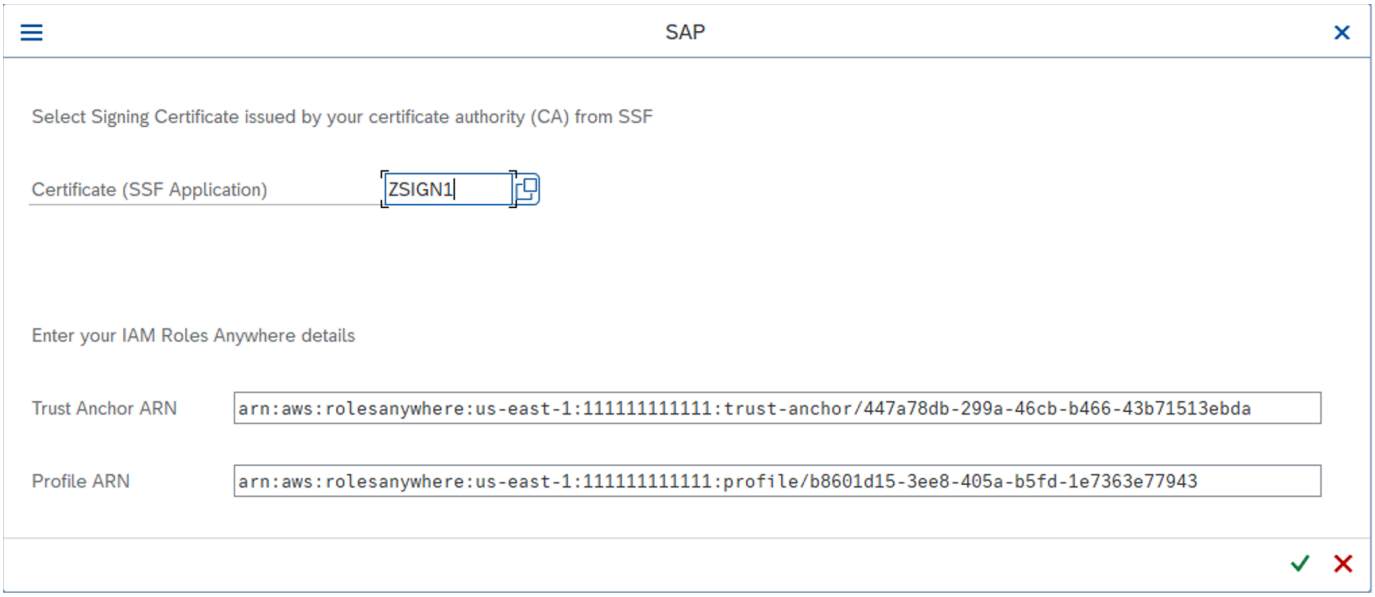
El proceso de firma varía en función de la CA y de la tecnología que utilice. Consulte Emitir certificados privados de entidades finales con una autoridad de certificación AWS privada para ver un ejemplo.

Paso 4: Importar la respuesta del certificado al PSE correspondiente

1. Ejecute la transacción de `/n/AWS1/IMG` y seleccione Configuración de AWS SDK para SAP ABAP > Requisitos técnicos previos > Configuración adicional para sistemas en las instalaciones.
2. Ejecute la actividad de IMG `Create PSE for SSF Application`.
3. Seleccione Editar para la transacción `STRUST`.
4. Elija la aplicación de SSF y, a continuación, seleccione Importar la respuesta del certificado, que se encuentra en la sección de PSE, debajo del asunto. Copie y pegue la respuesta del certificado en el cuadro de texto o importe el archivo desde el sistema de archivos. Seleccione Continuar > Guardar.
5. Los detalles del certificado se pueden ver seleccionando el tema dos veces. La información se muestra en la sección de certificados.

Paso 5: Configurar el perfil del SDK para usar funciones de IAM en cualquier lugar

1. Ejecute la `/n/AWS1/IMG` transacción y seleccione AWS SDK para SAP ABAP Configuración > Configuraciones de la aplicación.
2. Cree un nuevo perfil de SDK y asígnele un nombre.
3. Elija funciones de IAM en cualquier lugar como método de autenticación.
   - En el panel izquierdo, seleccione Autenticación y configuración.
   - Cree una nueva entrada e introduzca la información de su sistema SAP y la Región de AWS.
   - Seleccione Funciones de IAM en cualquier lugar como método de autenticación y luego Guardar.
   - Seleccione Introducir detalles y, en la ventana emergente, elija la aplicación SSF creada en the section called “Paso 1”. Introduzca el ARN del anclaje de confianza y ARN del perfil que se crearon en the section called “Requisitos previos”. Vea la siguiente imagen. Seleccione Continuar.



4. En el panel izquierdo, seleccione Asignación de roles de IAM. Introduzca un nombre y proporcione el ARN del rol de IAM facilitado por su administrador de IAM.

Para obtener más información, consulte Configuración de aplicaciones.

Uso del perfil de origen para el acceso entre cuentas

El perfil de origen permite a los sistemas SAP acceder a AWS los recursos de varias cuentas al encadenar las suposiciones de las funciones de IAM. Un perfil asume un rol, que luego asume otro rol, y así sucesivamente, similar al `source_profile` parámetro de AWS CLI. Esto resulta útil en situaciones de acceso entre cuentas en las que es necesario recorrer varias AWS cuentas para llegar a los recursos objetivo.

Ejemplo: su sistema SAP se ejecuta en la cuenta A (1111) y necesita acceder a los buckets de Amazon S3 en la cuenta C (333333333333). Se configuran tres perfiles:

1. `DEV_BASE`obtiene las credenciales base de los metadatos de la instancia Amazon EC2 y asume la función P en la cuenta A
2. `SHARED_SERVICES`usa `DEV_BASE` credenciales para asumir el rol Q en la cuenta B (222222222222)
3. `PROD_S3_ACCESS`usa `SHARED_SERVICES` credenciales para asumir el rol R en la cuenta C

Cuando su aplicación lo usa`PROD_S3_ACCESS`, el SDK ejecuta automáticamente la cadena: obtener credenciales de los metadatos de la instancia → asumir el rol P → asumir el rol Q → asumir el rol R.

Requisitos previos

Se deben cumplir los siguientes requisitos previos antes de configurar el perfil de origen:

- El administrador de IAM debe crear las funciones de IAM para cada paso de la cadena. Cada función debe tener:
  - Permisos para llamar a los requeridos Servicios de AWS
  - Relación de confianza configurada para permitir que el rol anterior de la cadena la asuma

  Para obtener más información, consulte Prácticas recomendadas de seguridad de IAM.
- Cree una autorización para ejecutar la transacción `/AWS1/IMG`. Para obtener más información, consulte Autorizaciones para configuración.
- Los usuarios deben tener `/AWS1/SESS` autorización para TODOS los perfiles de la cadena, incluidos los perfiles intermedios.

Procedimiento

Siga estas instrucciones para configurar el perfil de origen.

Steps

- Paso 1: Configurar el perfil base
- Paso 2: Configurar perfiles encadenados

Paso 1: Configurar el perfil base

El perfil base es el primer perfil de la cadena y debe utilizar un método de autenticación estándar.

1. Ejecute la `/n/AWS1/IMG` transacción para lanzar AWS SDK para SAP ABAP la Guía de implementación (IMG).
2. Seleccione AWS SDK para SAP ABAP Settings > Configuraciones de aplicaciones > SDK Profile.
3. Cree un perfil nuevo para usarlo como perfil base. Para ello, seleccione Nuevas entradas e introduzca el nombre y la descripción del perfil. Seleccione Guardar.

> **Note**
>
> Si utiliza un perfil existente que ya está configurado con un método de autenticación estándar (INST, SSF o RLA), puede omitir los pasos restantes de esta sección y pasar directamente a. the section called “Paso 2”

4. Seleccione el perfil que creó y, a continuación, seleccione Autenticación y configuración > Nuevas entradas e introduzca los siguientes detalles:
   - SID: el ID del sistema SAP
   - Cliente: El cliente del sistema SAP
   - ID de escenario: seleccione el `DEFAULT` escenario creado por su administrador de Basis
   - AWS Región: AWS región a la que quieres hacer llamadas
   - Método de autenticación: selecciona una de las siguientes opciones:
     - Función de instancia mediante metadatos para sistemas SAP que se ejecutan en Amazon EC2
     - Credenciales de SSF Storage para sistemas locales u otros sistemas en la nube
     - IAM Roles Anywhere para la autenticación basada en certificados

   Seleccione Guardar.
5. Seleccione IAM Role Mapping > Nuevas entradas e introduzca:
   - Número de secuencia: 1
   - Función lógica de IAM: un nombre descriptivo (por ejemplo,`DEV_BASE_ROLE`)
   - ARN del rol de IAM: el ARN del rol de IAM en la primera cuenta (por ejemplo,) `arn:aws:iam::111111111111:role/DevBaseRole`

   Seleccione Guardar.

Paso 2: Configurar perfiles encadenados

Configure cada perfil intermedio y final de la cadena.

Para el **`SHARED_SERVICES`** perfil (cadenas desde**`DEV_BASE`**):

1. Ejecute la `/n/AWS1/IMG` transacción.
2. Seleccione AWS SDK para SAP ABAP Settings > Configuraciones de aplicaciones > SDK Profile.

3. Seleccione Nuevas entradas. Introduzca el nombre del perfil (por ejemplo,`SHARED_SERVICES`) y la descripción. Seleccione Guardar.
4. Seleccione el perfil que ha creado y, a continuación, seleccione Autenticación y configuración > Nuevas entradas e introduzca los siguientes detalles:
   - SID: el ID del sistema SAP
   - Cliente: El cliente del sistema SAP
   - ID de escenario: seleccione el `DEFAULT` escenario creado por su administrador de Basis
   - AWS Región: AWS región a la que quieres hacer llamadas
   - Método de autenticación: selecciona el perfil de origen en el menú desplegable
   - ID de perfil de origen: introduzca el ID de perfil del perfil base (por ejemplo,`DEV_BASE`)

   Seleccione Guardar.
5. Seleccione IAM Role Mapping > Nuevas entradas e introduzca:
   - Número de secuencia: 1
   - Función lógica de IAM: un nombre descriptivo (por ejemplo,`SHARED_ROLE`)
   - ARN del rol de IAM: `arn:aws:iam::222222222222:role/SharedServicesRole`

   Seleccione Guardar.

Para el **`PROD_S3_ACCESS`** perfil (cadenas desde**`SHARED_SERVICES`**):

Repita los mismos pasos que`SHARED_SERVICES`, pero:

- `PROD_S3_ACCESS`Úselo como nombre
- Establezca el ID del perfil de origen en `SHARED_SERVICES`
- Utilice `PROD_S3_ROLE` y `arn:aws:iam::333333333333:role/ProdS3AccessRole` en el mapeo de roles de IAM

Para obtener información sobre las prácticas recomendadas de seguridad, incluida la administración de funciones de IAM, la configuración de la política de confianza y los requisitos de autorización, consulte las prácticas recomendadas para la seguridad de IAM.

Uso de SAP Credential Store

El almacén de credenciales de SAP se utiliza en la plataforma tecnológica empresarial de SAP para almacenar de forma segura las credenciales para la autenticación de claves de acceso secretas. AWS Debe tener una suscripción para utilizar el servicio.

En las siguientes instrucciones se presupone que ya ha configurado un perfil de SDK. Para obtener más información, consulte Configuración AWS SDK para SAP ABAP.

Antes de comenzar la configuración, asegúrese de cumplir con los requisitos previos. Para obtener más información, consulte SAP Credential Store.

Temas

- Pasos de configuración
- Uso de SAP Credential Store con el SDK

Pasos de configuración

Steps

- Paso 1: Configurar los ajustes de autenticación
- Paso 2: Crear una clave de servicio
- Paso 3: Convierte la clave de servicio en .p12 formato
- Paso 4: Conectarse a SAP BTP, entorno ABAP

Paso 1: Configurar los ajustes de autenticación

Siga los siguientes pasos para configurar los ajustes del almacén de credenciales para la autenticación.

1. Vaya a la pestaña Configuración de la instancia de SAP Credential Store.
2. Seleccione Editar configuraciones:
   - Elija el TLS mutuo como el tipo de autenticación predeterminado.
   - Seleccione Desactivado para ver el estado de cifrado de la carga útil. La carga útil se cifra en tránsito con HTTPS. Sin embargo, actualmente la carga útil no puede estar doblemente cifrada.
3. Seleccione Guardar.

Paso 2: Crear una clave de servicio

Siga los siguientes pasos para crear una clave de servicio para el almacén de credenciales.

1. En el panel izquierdo de la aplicación SAP Credential Store, vaya a Service Keys.
2. Seleccione Crear clave de servicio.
3. Introduzca un nombre para la clave de servicio y seleccione Crear.

La clave de servicio se crea en función del tipo de autenticación elegido. Descargue la clave de servicio y manténgala segura para usarla más adelante.

Paso 3: Convierte la clave de servicio en `.p12` formato

Se requiere un certificado de cliente en `.p12` este formato para crear un usuario saliente para el sistema de comunicación. Siga los siguientes pasos para generar un `.p12` certificado a partir de los detalles del certificado proporcionados en la clave del servicio de almacenamiento de credenciales.

1. Descargue el certificado SAP Cloud Root CA (requerido por SAP) desde SAP Trust Center Services.
2. Abra el certificado SAP Cloud Root CA en cualquier formato de archivo de texto. Al final del archivo, pulse Entrar y copie y pegue el certificado del campo de certificado de la clave de servicio. Sustituya los caracteres `\n` de línea nuevos por una nueva línea real (Entrar) y guarde el certificado completo en formato de `.cer` archivo.
3. Copie la clave del campo clave de la clave de servicio. Esta clave privada debe tratarse como información confidencial. Péguela en un archivo de texto y sustituya los caracteres `\n` de línea nuevos por una línea nueva real (Entrar). Guarde la clave privada en un archivo de texto.
4. Con el certificado y la clave privada generados en los pasos anteriores, ejecute el siguiente comando para generar un `.p12` certificado.

   ```
   openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in
    <certificate.cer>
   ```

   El comando requería la verificación de la contraseña de exportación. Conserve la contraseña para usarla más adelante.

Elimine el archivo de `.key` texto guardado en su clave privada.

Paso 4: Conectarse a SAP BTP, entorno ABAP

Configure el entorno SAP BTP y ABAP para conectarse con SAP Credential Store.

Temas

- Sistema de comunicación
- Disposición de comunicación

Sistema de comunicación

Siga los siguientes pasos para crear un sistema de comunicación que permita la comunicación entre SAP BTP y el entorno ABAP y SAP Credential Store.

1. Abra la plataforma de lanzamiento Fiori del sistema de entorno ABAP.
2. Seleccione el mosaico Sistemas de comunicación para abrir la aplicación.
3. Seleccione Nueva.
4. Introduzca un nombre y un identificador para el sistema de comunicación y seleccione Crear. Por ejemplo, puede asignar un nombre al sistema`ZSAP_CREDSTORE`.
5. Introduzca el resto de la información necesaria:
   - Nombre de host: copie el nombre de host de la URL de la clave de servicio. Por ejemplo, si la URL es`https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`, entonces el nombre del host es`credstore.mesh.cf.us10.hana.ondemand.com`.
   - Usuarios para la comunicación saliente: seleccione esta opción + para añadir un nuevo usuario.
     a. Seleccione el certificado de cliente SSL como mecanismo de autenticación.
     b. Seleccione Cargar un certificado nuevo:
        - Examine el `.p12` certificado generado en el paso anterior.
        - Escriba una descripción.
        - Introduzca la contraseña de exportación que se utilizó para generar el `.p12` certificado.
        - Seleccione Cargar.
     c. Seleccione Crear para crear un usuario saliente.
6. Seleccione Guardar.
7. Elimine la clave de servicio descargada en el paso anterior.

Disposición de comunicación

Utilice los siguientes pasos para crear una disposición de comunicación que proporcione un escenario de comunicación para la comunicación saliente.

1. Abra la plataforma de lanzamiento Fiori del sistema de entorno ABAP.
2. Seleccione el mosaico Acuerdos de comunicación para abrir la aplicación.
3. Seleccione Nueva.
4. Seleccione el escenario `/AWS1/CRED_COMM_SCENARIO` de comunicación e introduzca un nombre para el arreglo de comunicación. Por ejemplo, `Z_AWS_SDK_TO_SAP_CREDSTORE`.
5. Seleccione Crear.
6. En el campo Sistema de comunicación, busque el sistema de comunicación creado en el paso anterior. El resto de la información se rellena automáticamente después de la selección del sistema.
7. Seleccione Guardar.
8. Seleccione Comprobar conexión para probar la conexión.

Una vez completada esta configuración, el entorno ABAP puede utilizar el acuerdo de comunicación para utilizar el servicio SAP Credential Store a través del servicio saliente (HTTP).

Uso de SAP Credential Store con el SDK

Steps

- Paso 1: Crear un espacio de nombres y credenciales
- Paso 2: Configure la aplicación de configuración empresarial personalizada

Paso 1: Crear un espacio de nombres y credenciales

Cree un espacio de nombres y una credencial en SAP Credential Store con la ayuda de SAP: cree, edite y elimine una credencial.

Introduzca los siguientes detalles para crear una credencial del tipo Key.

- Espacio de nombres: introduzca un nombre para el espacio de nombres y agrupe las credenciales relacionadas.

- Nombre: introduzca un nombre para la clave. Recomendamos que`aws-0123456789012-username`, cuando:
  - `0123456789012`es el Cuenta de AWS identificador al que la credencial concede acceso
  - `username`es el nombre de usuario de IAM al que pertenece la credencial
- Valor: introduzca una clave de acceso secreta codificada en base 64. Usa el siguiente comando para codificar en base 64 tu clave de acceso secreta.

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

  El comando lee la clave de acceso secreta de la entrada estándar y la pasa a base64 sin una línea nueva al final. Envía la clave de acceso secreta codificada en base 64 a la pantalla. Borre o cierre el terminal después de copiar el valor en SAP Credential Store.
- Nombre de usuario: introduzca su ID de clave de acceso.

- Seleccione Crear.

Se crea un nuevo espacio de nombres con una credencial y las credenciales se pueden agregar, eliminar o modificar dentro de este espacio de nombres.

Siga el principio del privilegio mínimo para administrar el acceso a las credenciales almacenadas en el espacio de nombres.

Paso 2: Configure la aplicación de configuración empresarial personalizada

Siga los siguientes pasos para configurar la aplicación de configuración empresarial personalizada a fin de definir la credencial que se utilizará para la autenticación mediante el SDK.

1. Abra la plataforma de inicio Fiori del sistema de entorno ABAP.
2. Examine el icono de configuración empresarial personalizada para abrir la aplicación.
3. Abra la configuración empresarial de SDK Profile.
4. Seleccione el perfil de SDK para el que se deben configurar los ajustes de autenticación para SAP Credential Store.
5. En la pestaña Autenticación y configuración del perfil seleccionado, seleccione Editar e introduzca los siguientes detalles:

- Método de autenticación: seleccione las credenciales del almacén de credenciales de SAP.
- Espacio de nombres: introduzca el espacio de nombres creado en SAP Credential Store. Para obtener más información, consulte the section called “Paso 1: Crear un espacio de nombres y credenciales”.
- Nombre de clave: introduzca el nombre de la credencial clave creada. Para obtener más información, consulte the section called “Paso 1: Crear un espacio de nombres y credenciales”.
- Acuerdo de comunicación: introduzca el nombre del acuerdo de comunicación creado. Para obtener más información, consulte the section called “Disposición de comunicación”.

6. Seleccione Aplicar para ir a la pantalla del perfil del AWS SDK.
7. Selecciona Seleccionar transporte para seleccionar el transporte mediante la ayuda de valores.
8. Seleccione Guardar.

Solucionar problemas AWS SDK para SAP ABAP

En esta sección se proporcionan los pasos de solución de problemas para posibles escenarios de error.

Temas

- Error de importación
- Restricción de ubicación no especificada
- Errores de SSL
- Perfil de configuración
- Autorización de IAM
- Autorización para realizar las acciones necesarias
- Escenario activo
- Caracteres especiales en el código
- Conectividad

Error de importación

Problema: la clase 'CL_SYSTEM_UUID' no contiene una interfaz 'IF_SYSTEM_UUID_ _STATIC RFC4122

Causa: falta la nota de SAP 0002619546 en su sistema.

Resolución: asegúrese de que la nota de SAP 0002619546 esté instalada en su sistema.

Restricción de ubicación no especificada

Problema: la restricción de ubicación no especificada no es compatible con el punto de conexión específico de la `region` a la que se envió esta solicitud

Causa: a su bucket de Amazon S3 le falta la AWS región en el `io_createbucketconfiguration` parámetro.

Resolución: al crear un bucket en cualquier región, excepto en `us-east-1`, especifique la región de su bucket de Amazon S3 mediante el parámetro `io_createbucketconfiguration` en `createbucket()`. No es necesario que especifique una restricción para `us-east-1`.

En el siguiente ejemplo se muestra un parámetro `io_createbucketconfiguration` configurado correctamente.

```
createbucket(
    iv_bucket = 'amzn-s3-demo-bucket'
    io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )
).
```

Errores de SSL

Problema: SSL Server Certificate Hostname Mismatch o SSL handshake with docs.aws.amazon.com:443 failed: SSSLERR_NO_SSL_RESPONSE

Causa: el parámetro `icm/HTTPS/client_sni_enabled` no está establecido en `TRUE` en el perfil de `DEFAULT`.

Resolución: siga los siguientes pasos para solucionar los problemas indicados o cualquier otro problema relacionado con SSL.

1. Abra la SAPGUI y vaya a la barra de comandos.
2. Ejecute la transacción `RZ10`.
3. Vaya a Perfil y elija el perfil de `DEFAULT`. La versión se rellena automáticamente.
4. En la sección Editar perfil, seleccione Mantenimiento ampliado y, a continuación, seleccione Cambiar.
5. Busque el parámetro `icm/HTTPS/client_sni_enabled`.

   - Si el parámetro existe, edite el valor del parámetro y establézcalo en `TRUE`.
   - Si el parámetro no existe, cree uno siguiendo estos pasos.

     1. Seleccione el parámetro.

        > **Note**
        >
        > Asegúrese de seleccionar el parámetro para creación y no para edición (icono de lápiz).

     2. Ingrese `icm/HTTPS/client_sni_enabled` en el campo Nombre del parámetro.
     3. Ingrese `TRUE` en el campo Valor del parámetro.

4. Seleccione Guardar.

6. Guarde estos cambios en el perfil `DEFAULT` y salga.

Perfil de configuración

Problema: Could not find configuration under profile <profile_name> with scenario DEFAULT for <sid>:<client>

Causas: <nombre_de_perfil> es incorrecto o no se ha configurado.

Resolución: siga los siguientes pasos para configurar el perfil.

1. Abra la SAPGUI y ejecute la transacción `/n/AWS1/IMG`.
2. Vaya a Configuración de la aplicación > Perfil del SDK.
   - Si su perfil está configurado, compruebe que el nombre del perfil sea correcto.
   - Si su perfil no está configurado, siga los pasos para configurar un perfil.
3. Seleccione Nuevas entradas.
   a. Escriba un nombre y la descripción del perfil.
   b. Seleccione Guardar.
4. Elija la entrada que creó en el paso anterior y, a continuación, seleccione Autenticación y configuración.
5. Seleccione Nuevas entradas, escriba los siguientes detalles y, a continuación, seleccione Guardar.
   - SID
   - Cliente
   - ID de escenario
   - AWS Región
   - Método de autenticación
     - Seleccione el rol de instancia a través de los metadatos para los sistemas SAP que se estén ejecutando AWS.
     - Para los sistemas SAP que se ejecutan en las instalaciones o en otra nube, seleccione Credenciales del almacenamiento de SSF.

6. Seleccione Asignación de rol de IAM > Nuevas entradas, escriba los siguientes detalles y seleccione Guardar.

   - Número de secuencia
   - Rol de IAM lógico
   - ARN del rol de IAM

Autorización de IAM

Problema: no se pudo asumir el rol <iam_role_arn>o el usuario: <user_arn>no está autorizado para desempeñar: sts: AssumeRole on resource: <iam_role_arn>

Causas: a continuación se indican las posibles razones de este error.

- Se especificó un ARN de rol de IAM incorrecto
- El usuario de IAM no tiene permiso para acceder al rol de IAM
- Falta de relación de confianza entre el rol de IAM asumido y el rol de IAM o usuario de IAM que lo asume

Resolución: siga los siguientes pasos para asegurarse de que el ARN del rol de IAM sea correcto.

1. Abra la SAPGUI y ejecute la transacción `/n/AWS1/IMG`.
2. Vaya a Configuración de la aplicación > Perfil del SDK y elija el perfil que se ha configurado con su rol de IAM.
3. Seleccione Asignación de rol de IAM y compruebe o corrija el ARN de su rol de IAM.

   - Si el ARN del rol de IAM es correcto, asegúrese de que el rol de IAM se haya configurado correctamente. Para obtener más información, consulte Solución de problemas de roles de IAM.

Autorización para realizar las acciones necesarias

Problema: User <user_arn> is not authorized to perform: <action> on resource: <resource_arn>

Causa: el usuario no tiene permisos para realizar una acción.

Resolución: `user_arn` debe configurarse con los permisos necesarios en `resource_arn` para realizar una `action` específica. Para obtener más información, consulte Permisos necesarios para acceder a los recursos de IAM.

Escenario activo

Problema: No active scenario configured

Causa: no se pudo configurar el escenario activo.

Resolución: consulte la configuración de tiempo de ejecución para configurar un escenario activo.

Caracteres especiales en el código

Advertencia: el carácter 0x00A0 no puede formar parte de una palabra ABAP

> **Note**
>
> Esta advertencia puede ir precedida de varios mensajes de error.

Causa: al copiar y pegar código de orígenes diferentes, se pueden insertar caracteres especiales en el código.

Resolución: al pegar cualquier código en el editor de código fuente de ABAP, aparece la siguiente ventana emergente.

Se detectaron espacios de no separación. ¿Convertir en espacios?

Elija Sí para responder a esta pregunta. Además, le recomendamos seleccionar el código para copiarlo, en lugar de utilizar el botón de copiar de los cuadros de códigos.

Conectividad

Problema: SCLNT_HTTP(411) : Direct connect to tla.region.amazonaws.com:443 failed: NIECONN_REFUSED(-10)

Causa: el sistema SAP no tiene conectividad a Internet y no puede establecer una conexión TCP/IP con el puerto 443 de tla.region.amazonaws.com.

Solución: el sistema SAP debe poder establecer la conexión con los AWS puntos finales en el puerto HTTPS 443, ya sea directamente o a través de un servidor proxy. Puede establecer o verificar la conectividad a Internet con una de las siguientes opciones.

- Conexión saliente directa a Internet a través de una NAT o una puerta de enlace de Internet
- Conexión a través de un servidor proxy

  Para obtener más información, consulte Conexión a través de un servidor proxy.

Temas adicionales

Esta sección abarca los siguientes temas.

Temas

- AWS SDK para SAP ABAP lanzamientos
- Licencias de SAP

AWS SDK para SAP ABAP lanzamientos

AWS El SDK para SAP ABAP se entrega en transporte y el AWS SDK para SAP ABAP (edición BTP) se entrega como complementos. El mecanismo para importar transportes y complementos es diferente, pero la funcionalidad técnica es la misma. Para obtener más información, consulte Configuración.

Temas

- Estrategia de versión
- Prácticas recomendadas
- Aplicación de parches en el SDK de SAP ABAP
- Instalación de un módulo adicional
- Desinstalación del SDK de SAP ABAP

Estrategia de versión

La versión 1 de AWS SDK para SAP ABAP se actualiza con frecuencia. Los nuevos parches se publican semanalmente o diariamente en función de las versiones y actualizaciones de Servicios de AWS. Los parches Servicios de AWS pueden incluir correcciones de errores y otros cambios que actualicen el nivel de parche del SDK. Para obtener más información, consulte la política de mantenimiento de AWS SDKs las herramientas.

Prácticas recomendadas

Recomendamos retener el mismo nivel de parche del SDK de SAP ABAP para todos los sistemas SAP (desarrollo, control de calidad y producción).

Al aplicar los parches en el SDK, importe la versión más reciente a su entorno aislado. A continuación, puede importarlo a los sistemas de desarrollo, control de calidad y producción, siguiendo los procedimientos habituales de control de cambios.

Aplicación de parches en el SDK de SAP ABAP

Cada versión del SDK de SAP ABAP se entrega como un conjunto de transportes acumulativos, que incluye todas las correcciones de errores, características y actualizaciones. No hay diferencia entre el transporte de un parche y el de una instalación. Debe importar los transportes más recientes para aplicar parches en el SDK de SAP ABAP.

Debido a las dependencias de los módulos `core` Runtime y API, los módulos API individuales no se pueden parchear por separado. Al aplicar los parches, debe actualizar el `core` módulo y todos los demás módulos de API que se hayan instalado, incluso si algunos módulos ya no se utilizan.

Los siguientes escenarios describen el proceso de aplicación de parches:

- Escenario 1: actualización periódica de la versión del SDK

  Al aplicar el parche a la última versión del SDK:

  - Importe todos los transportes (módulos API principales e instalados) simultáneamente
  - No se requiere una secuencia de importación específica
  - Por ejemplo, si importó los transportes `core`, `ec2` y `lmd` cuando instaló el SDK, debe importar los transportes más recientes para `core`, `ec2` y `lmd` al aplicar los parches.
- Escenario 2: Añadir un nuevo módulo de API

  Al importar un módulo de API nuevo (p. ej., Amazon Textract`tex`) desde una versión del SDK diferente a la de los módulos actualmente instalados:

  - Todos los módulos existentes deben estar parcheados para que coincidan con la versión de lanzamiento del SDK del nuevo módulo
  - Importación simultánea (sin secuencia específica):
    - Los últimos transportes de todos los módulos existentes (`core`y los módulos API instalados)
    - Transportes para el nuevo módulo de API `tex`
- Escenario 3: Actualización de un módulo de API existente

  Al actualizar un módulo de API existente (por ejemplo, Amazon Translate`xl8`) a la última versión del SDK, no se admiten las actualizaciones de los módulos individuales debido a las

interdependencias. Debe aplicar un parche a todo el SDK actualizando el `core` módulo y todos los módulos de API instalados a la vez, siguiendo el proceso descrito en. Regular SDK Version Update (p. 593)

Instalación de un módulo adicional

Importe el transporte del nuevo módulo al mismo nivel de parche que el `core` existente y los módulos para instalar un módulo API adicional en su sistema SAP. Siga las directrices de the section called “Aplicación de parches en el SDK de SAP ABAP” en caso de que desee importar una versión más reciente del módulo. Esto garantiza que los niveles de parche sean compatibles en todos los módulos del SDK.

Desinstalación del SDK de SAP ABAP

Para desinstalar el SDK para SAP ABAP, debe descargar un kit de transporte de eliminación desde 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"
```

Puede descargar un archivo de firma desde https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/ Release/Uninstall-ABAPSDK-latest.sig. Para validar el archivo, consulte Verificar el SDK de SAP ABAP.

Para cada módulo del SDK instalado en su sistema SAP, el transporte de eliminación correspondiente debe importarse del archivo ZIP anterior. Puede eliminar un solo módulo sin desinstalar todo el SDK. Puede hacerlo importando solo el transporte de eliminación del módulo que desee eliminar. Si va a desinstalar todo el SDK con todos sus módulos, el transporte de eliminación principal debe importarse en último lugar.

Le recomendamos que pruebe la desinstalación en un entorno aislado antes de intentarlo en sistemas de desarrollo, control de calidad o producción.

Consideraciones

Antes de desinstalar el SDK, tenga en cuenta las siguientes consideraciones.

- Se perderán los ajustes de configuración del SDK de. Debe volver a configurarse `IMG` durante la instalación.
- Si tiene programas Z que se basan en el SDK, generarán errores de sintaxis tras la eliminación del SDK.
- Los roles de PFCG o Business que contengan referencias de autorización del SDK tendrán autorizaciones no válidas tras la eliminación del SDK. Elimine las referencias de autorización del SDK de los roles de PFCG antes de desinstalar el SDK.

> **Note**
>
> AWS SDK para SAP ABAP: la edición BTP no se puede desinstalar durante la versión preliminar para desarrolladores.

Licencias de SAP

El uso del software de SAP está sujeto a los términos de SAP. Usted es responsable de cumplir con los términos de licencia de SAP, incluidas las condiciones de distribución de software y de licencia indirecta. La información proporcionada no constituye asesoramiento legal y no se debe confiar en ella para garantizar el cumplimiento de las licencias. Si tiene alguna duda acerca de las licencias o los derechos del software de SAP, consulte con el equipo legal, SAP o su distribuidor de SAP.

Pregunta: ¿El uso del SDK de SAP ABAP afectará a mi licencia de SAP?

Respuesta: AWS SDK para SAP ABAP le permite consumir Servicios de AWS con su propio código ABAP. Se utiliza en escenarios de integración entre un sistema SAP y Servicios de AWS. Cualquier escenario en el que los datos del sistema SAP se envíen a un sistema de terceros (que no sea SAP) o sean creados por ese sistema puede tener implicaciones para la concesión de licencias indirectas. SAP utiliza varios enfoques para definir el acceso indirecto, como los cálculos basados en los usuarios y los cálculos basados en los resultados. La metodología para definir el acceso indirecto depende de su contrato con SAP. Debe conocer las instrucciones que se proporcionan en su contrato con SAP y puede analizarlas más a fondo con SAP o su distribuidor.

En 2018, SAP publicó dos documentos: la Guía de acceso indirecto para los clientes de bases instaladas de SAP y los precios del ERP de SAP para la era digital, que abordaban el acceso indirecto/digital. Estos documentos se pueden encontrar en los sitios web de SAP y son ejemplos

de enfoques de concesión de licencias indirectas. Sin embargo, estos documentos no reflejan su acuerdo particular con SAP.

Historial de documentos para AWS SDK para SAP ABAP Guía para desarrolladores

En la siguiente tabla se describen las versiones de la documentación de AWS SDK para SAP ABAP.

Cambio	Descripción	Fecha
Contenido nuevo	Se agregó la anulación del servidor Per-service proxy.	15 de mayo de 2026
Contenido nuevo	Se agregó mediante el AWS SDK para el servidor MCP de SAP ABAP Knowledge.	24 de abril de 2026
Contenido nuevo	Se agregó el uso del perfil de origen para el Cross-Account acceso.	4 de marzo de 2026
Contenido nuevo	Se agregó el uso de la autenticación por clave de acceso secreta con cifrado SSF.	12 de enero de 2025
Contenido nuevo	Versión preliminar para desarrolladores del SDK para SAP ABAP: edición BTP.	31 de mayo de 2024
Contenido nuevo	Se ha agregado Uso de certificados con funciones de IAM en cualquier lugar.	1 de diciembre de 2023
Contenido nuevo	Se ha agregado Creación de productos con SDK.	1 de diciembre de 2023
Contenido nuevo	Se ha agregado Comportamiento de los reintentos.	1 de diciembre de 2023

Contenido nuevo	Se ha agregado Licencias de SAP.	22 de septiembre de 2023
Versión pública	Relanzamiento inicial de la Guía para desarrolladores. AWS SDK para SAP ABAP	30 de junio de 2023
Contenido nuevo	Se han agregado caracterí sticas de AWS SDK para SAP ABAP.	30 de mayo de 2023
Contenido nuevo	Se agregó Solución de problemas de AWS SDK para SAP ABAP.	17 de febrero de 2023
Vista previa del desarrollador	Vista previa para desarroll adores de la Guía AWS SDK para SAP ABAP para desarroll adores.	17 de noviembre de 2022

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.