



Guía para desarrolladores

AWS SDK para SAP ABAP



AWS SDK para SAP ABAP: Guía para desarrolladores

Copyright © 2024 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	4
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 de 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
SDK de parches para SAP ABAP: edición BTP	27
Configuración	28
Configuración global	29
Configuración técnica	29
Configuración de escenarios	30
Configuración de aplicaciones	30
Perfil del SDK	31
Solucionador lógico de recursos	32

Ejemplo	32
Configuración del tiempo de ejecución	33
Registro y rastreo	33
OPT-IN: telemetría mejorada	33
Escenario activo	34
Actualización, rastreo y telemetría	34
Actualización del sistema SAP	34
Rastreo	35
Telemetría	36
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	38
Acceso a puntos de conexión en varias regiones	39
Uso del SDK	41
Representación de datos	41
Tipos de datos	42
Tipos de datos de AWS	44
Programa de ejemplo	45
Requisitos previos	45
Código	46
Secciones de código	46
Conceptos	49
Clases de API	49
Objetos adicionales	50
Establecimiento de estructura de clases	50
Matrices	52
Mapas	53
Funciones de nivel superior	53
Características	1
Configuración mediante programación	54
Esperadores	55
Paginadores	56
Comportamiento de los reintentos	57
Creación de productos	58

Configuración de un ID de producto	58
Limitaciones	59
Ejemplos de código	60
Acciones y escenarios	60
Amazon Bedrock Runtime	61
CloudWatch	66
DynamoDB	74
Amazon EC2	89
Kinesis	105
Lambda	115
Amazon S3	129
SageMaker	138
Amazon SNS	157
Amazon SQS	166
Amazon Textract	174
Amazon Translate	184
Seguridad	194
Autenticación del sistema	194
Autenticación de metadatos	195
Autenticación de clave de acceso secreta	195
Autenticación basada en certificados mediante funciones de IAM en cualquier lugar	196
Siguiendo el siguiente paso	196
Prácticas recomendadas de seguridad de IAM	197
Práctica recomendada para el perfil de instancia Amazon EC2	197
Roles de IAM para usuarios de SAP	198
Autorizaciones de SAP	201
Autorizaciones de configuración	201
Autorizaciones de SAP para usuarios finales	202
Operaciones seguras	203
Cifrado de datos en reposo	203
Cifrado de datos en tránsito	204
Uso de la API	2
Uso de certificados	204
Requisitos previos	204
Procedimiento	205
Tienda de credenciales	21

Pasos de configuración	208
Uso de SAP Credential Store con el SDK	210
Solucionar problemas	214
Error de importación	214
Restricción de ubicación no especificada	214
Errores de SSL	215
Perfil de configuración	216
Autorización de IAM	217
Autorización de acciones	217
Escenario activo	34
Caracteres especiales	218
Conectividad	218
Temas adicionales	220
Versiones	220
Estrategia de versión	220
Prácticas recomendadas	197
Aplicación de parches en el SDK de SAP ABAP	221
Instalación de un módulo adicional	221
Desinstalación del SDK de SAP ABAP	221
Licencias de SAP	222
Historial de documentos	224
.....	ccxxvi

¿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 BADI de ABAP, informes, transacciones, servicios OData y otros artefactos de ABAP en, Servicios de AWS como Amazon Simple Storage Service (Amazon S3) (Simple Storage Service (Amazon S3)),, etc. Amazon DynamoDB Amazon Translate También puede desarrollar para 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 de](#)

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 la compatibilidad con las principales versiones del SDK y sus dependencias subyacentes, consulte lo siguiente en la [Guía de Referencia de SDK y herramientas de AWS](#):

- [AWS Política de mantenimiento de los SDK y las herramientas](#)

- [AWS Matriz de soporte de SDK y herramientas y versiones](#)

Referencia de la API

Para ver una lista completa de AWS SDK para SAP ABAP las API, consulta AWS SDK para SAP ABAP la [Guía de referencia de las API](#).

Para ver una lista completa de módulos de AWS SDK para SAP ABAP TLA, consulte AWS SDK para SAP ABAP la sección [Lista de módulos](#).

Precios

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

Recursos adicionales de

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

- [SAP sobre la AWS documentación](#)
- [Blog de desarrolladores de AWS](#)
- [AWS foros de desarrolladores](#)
- [AWS Biblioteca de ejemplos de códigos del SDK](#)
- [@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.

Pasos

- [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 .

```
{
  "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. Se trata de 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.

```
{
  "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

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 de perfil recomendados y la configuración del certificado SSL.
- 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.

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 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.

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.

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 está ejecutando.
 - 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 TranslateReadOnly 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.

- Vaya a Configuración de la aplicación → Perfil del SDK.
 - Seleccione Editar para crear un perfil nuevo.
 - Perfil: DEMO.
 - Descripción: Demo profile.
- Seleccione la tecla de flecha derecha situada junto a la entrada creada y vaya 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 está ejecutando.

- Introduzca el espacio de nombres y el nombre clave de las credenciales almacenadas en el almacén de credenciales de SAP.
- 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 situada junto a la entrada creada y vaya a la pestaña IAM Role Mapping.

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 las API de AWS: 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 DEMO 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_xl8unsuppdedlanguage00 INTO DATA(lo_lang).
      WRITE: / 'ERROR' COLOR COL_NEGATIVE,
        'Cannot translate from',
        lo_lang->sourcelanguagecode,
        'to',
        lo_lang->targetlanguagecode.
    CATCH cx_root INTO DATA(lo_root).
      WRITE: / 'ERROR' COLOR COL_NEGATIVE, lo_root->get_text( ).
  ENDTRY.

```

SDK for SAP ABAP - BTP edition

1. 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. Seleccione 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_xl8unsuppdedlanguage00 INTO DATA(lo_lang).
        out->write( |ERROR - Cannot translate from { lo_lang->sourcelanguagecode }
to { lo_lang->targetlanguagecode }| ).
      CATCH cx_root INTO DATA(lo_root).
        out->write( |ERROR - { lo_root->get_text( ) }| ).
    ENDTRY.
  ENDMETHOD.
```

```
ENDCLASS.
```

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

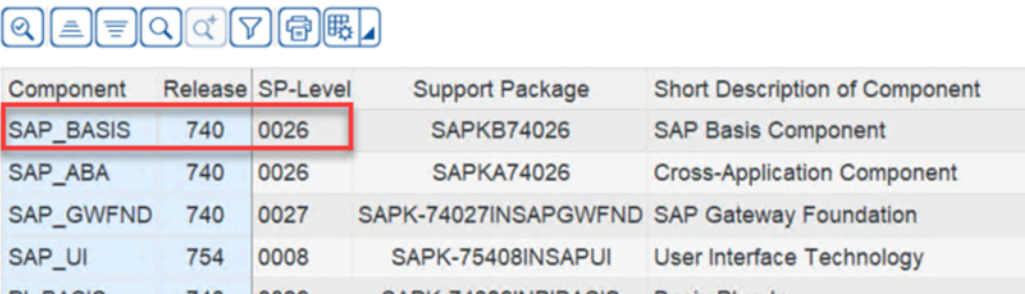
- [Versión de Basis](#)
- [Versión del kernel](#)
- [Parámetros](#)
- [Notas](#)
- [Conectividad saliente](#)
- [Conectividad HTTPS](#)
- [Acceso a los metadatos de la instancia de Amazon EC2](#)

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
PL_BASIS	740	0008	SAPK-74026INSAPBASIS	Basis Plus for

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/#/notes/0002619546>

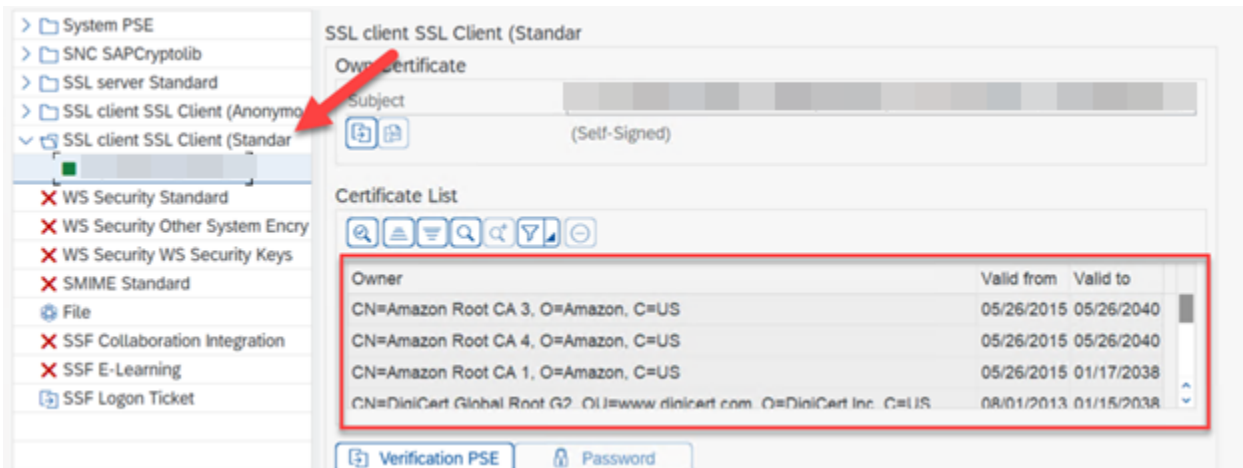
Conectividad saliente

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

1. Vaya a <https://www.amazontrust.com/repository/>.
2. En las CA raíz, 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 la instancia de Amazon EC2

El sistema ABAP establece conexiones HTTP sin cifrar en localhost (<http://169.254.169.254>) para habilitar los metadatos de las instancias de Amazon EC2. El canal HTTP se utiliza únicamente 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 característica 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.

Active Services						
No.	Protocol	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv E:
<input type="checkbox"/>	1	HTTPS		60	600	✓
<input type="checkbox"/>	2	HTTP		60	600	✓

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> [://amazonaws.cn/V1/release/abapsdk-LATEST.zip](https://amazonaws.cn/V1/release/abapsdk-LATEST.zip). sdk-for-sapabap
awsSdkSapabap

```
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-----
```

```

MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAmS3oN3wKBh4HJ0Ga0tye
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwzsf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgjbNmX0hB0Eprg8/nVRHspf3zuisJC21MPkz0
cH0R31MNsMLzm+d/gVklT31R/JwAcFCkXTWvR8/V0WNGZZXdVubefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLjz8dH
Zg25o79U9EJln+lpqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIUURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----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 AWS Support.

AWS Transportes del SDK

Temas

- [Contenido](#)
- [Importación](#)
- [Espacio de nombres](#)

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

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 módulos de los acrónimos (TLA), consulte [AWS SDK para SAP ABAP : lista de módulos](#).

AWS Los transportes del SDK son solicitudes de entorno de trabajo. 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 quiere utilizar las API de Amazon S3 e Amazon Translate importar los transportes de core transporte (que contiene el tiempo de ejecución principal, Amazon S3 y AWS STS los módulos) y de x18 transporte (que contiene el módulo Amazon Translate).

Para ver una lista completa de las API del SDK de SAP ABAP, consulte [SDK de SAP ABAP: Guía de referencia de la API](#).

Los siguientes son puntos clave a la hora de importar los 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 AWS Support.
- 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 AWS Support.
- 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.

Espacio de nombres

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 transacciónSU21, póngase en contacto con nosotros AWS Support 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](#)
- [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 Fiori seleccionando Deploy Product.
2. En Productos > Productos de socios, seleccione **/AWS1/SDK_OMNI**.

Póngase en contacto con nosotros AWS Support si no aparece **/AWS1/SDK_OMNI** después de haber sido aceptado en la versión preliminar para desarrolladores.

3. En Versión de destino, elija la versión del SDK para SAP ABAP (edición BTP) que desee instalar en su sistema.
4. Marque las casillas de verificación de todos los SID en los que desee instalar el SDK.

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](#).

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. Muchos de los ajustes son directamente análogos a los ajustes definidos en los archivos `.INI` de otros SDK.

Las configuraciones del SDK, excepto la configuración de tiempo de ejecución, deben completarse en su 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.

Configurando AWS SDK para SAP ABAP

Para ejecutar la transacción de configuración, escriba `/n/AWS1/IMG` en la barra de comandos de la SAPGUI. Si no tiene permisos para configurar el AWS SDK, consulte [Autorizaciones de SAP](#).

Configuración del AWS 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.

En la aplicación de configuración empresarial personalizada, puede agrupar las configuraciones según el tipo de configuración del SDK. Siga los pasos siguientes 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. 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](#)
- [Temas de actualización, rastreo y telemetría para AWS SDK para SAP ABAP](#)
- [Escenarios de conectividad avanzada](#)

Configuración global

Utilice /n/AWS1/IMG IMG transaction 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.
 - Desactivar los metadatos de EC2: deje este campo en blanco. Este campo es de solo lectura en la edición BTP y está establecido en «Sí» de forma predeterminada.
 - Modo de punto de conexión de metadatos: utilice el punto de conexión de metadatos de 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 ZVA01ZVA02, y ZVA03 están relacionadas con la facturación, se mejoran y se ejecutan Servicios de AWS, como Amazon S3, y AWS Lambda, entonces Amazon SageMaker, 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 /n/AWS1/IMG la transacción 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.
- 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.

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 los SDK envían información de telemetría a para fines de soporte. AWS Puede optar por activar la telemetría mejorada. Esto resulta especialmente útil cuando contactas AWS Support

para identificar el origen de una llamada a la API concreta. 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.

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 ingresar las credenciales, lo que puede requerir la regeneración de una nueva clave de acceso secreta en <https://console.aws.amazon.com/iam/>.
- Si el sistema se autentica con los metadatos de la instancia EC2, 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 AWS Support 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

Los SDK envían información de telemetría a AWS Support. 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 AWS Support.

- 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 permite ayudarle AWS Support mejor. AWS Support 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).

Escenarios de conectividad avanzada

AWS SDK para SAP ABAP consume Servicios de AWS al realizar llamadas HTTPS a los puntos de conexión de AWS. 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 para 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](#)
- [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.

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 de conexión 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 un punto de conexión de VPC de interfaz?](#)

En el siguiente ejemplo, se crea un punto de conexión 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 sesión AWS (que utiliza AWS STS para asumir un rol de IAM) y traducir la clase de API.

```
" 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`.

Acceso a puntos de conexión en varias regiones

El punto de conexión de AWS 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.
```

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.
- API (paquete /AWS1/API y sus subpaquetes): 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 perjudique a otra API. Para ver una lista completa de las API de AWS SDK para SAP ABAP , consulte la [AWS SDK para SAP ABAP : Guía de referencia de la 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](#)
- [Limitaciones](#)

Representación de datos en ABAP

Esta sección abarca los siguientes temas.

Temas

- [Tipos de datos](#)
- [Tipos de datos de AWS](#)

Tipos de datos

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

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

Tipos de datos de AWS	Tipo de datos de ABAP	Comentarios
		de ejecución. Si NaN, Infinity o +Infinity están representados, 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 representar en ABAP, y se utilizan CADENAS en lugar de bigInteger.
bigDecimal	STRING	
Marca de tiempo	TZNTSTMPS	TZNTSTMPS permite el procesamiento con funciones de marca temporal nativas de ABAP.

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

Tipos de datos de AWS	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.
Array (Matriz)	TABLA ESTÁNDAR	

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

Tipos de datos de AWS

Los siguientes enfoques se han integrado para admitir Servicios de AWS en ABAP.

- Algunos tipos de datos de AWS 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.
- Los datos de AWS se representan en el cable como JSON o XML y los valores son opcionales. Por ejemplo, consulte los siguientes ejemplos devueltos por un Servicio de AWS en 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 para 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 en AWS mediante el método definido en el perfil del SDK.
- Su perfil de instancia de Amazon EC2 otorga a su sistema SAP los derechos a `sts:assumeRole` en 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, AWS Lambda y DynamoDB, entonces crea objetos de API a partir de /aws1/cl_s3_factory, /aws1/cl_lmd_factory y /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/ifs3`, 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 SDK de AWS adopta este enfoque orientado a objetos y todos los datos se representan como objetos y tablas. El campo `LastModified` se representa como una marca temporal que se puede convertir en una fecha con el comando `CONVERT TIME STAMP` nativo de ABAP. `GET_SIZE()` devuelve un `INT4` para facilitar las operaciones matemáticas y de formato.

```
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 oTLA. 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 los AWS SDK para SAP ABAP TLA de módulos, 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. Fíjese en este lanzamiento de una instancia de Amazon EC2 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_ec2secgrpiddstringlist_w=>tt_securitygroupidstringlist(  
  ( NEW /aws1/  
cl_ec2secgrpiddstringlist_w( 'sg-12345678' ) )  
  ( NEW /aws1/  
cl_ec2secgrpiddstringlist_w( 'sg-55555555' ) )  
  ( NEW /aws1/  
cl_ec2secgrpiddstringlist_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

Lo [Clases de API](#) descrito en la sección anterior refleja con precisión las API de AWS servicio y las representa 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](#)

Configuración mediante programación

Utilice `/n/AWS1/IMG` IMG transaction 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

Al trabajar con AWS API asíncronas, es necesario esperar a que un determinado recurso esté disponible antes de tomar medidas adicionales. Por ejemplo, la CREATETABLE() API de Amazon DynamoDB responde inmediatamente 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 = |example-bucket| ).
```

```
“ Wait until the newly created bucket becomes available
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
    iv_bucket = |example-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. Puede utilizar las API paginadas con facilidad y sin entender el mecanismo subyacente de las API que utilizan los 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( ).
    ENDLLOOP.
  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, consulta la Guía de referencia sobre [el comportamiento de los reintentos](#) en AWS los SDK y las herramientas.

Creación de productos con SDK

Un producto o complemento de ABAP que consume Servicios de AWS 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, los ID de producto de los productos Invoice Analyzer, Tax Calculator y Pricing Engine pueden ser INVOICE_ANALYZER, TAX_CALCULATOR y PRICING_ENGINE.

La adición de un ID de producto a la sesión mejora la telemetría a que se envía a 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, AWS Support puede identificar el producto que realiza 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.

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 actualmente no es compatible con. AWS SDK para SAP ABAP
- Los módulos que se basan en las características de streaming de HTTP/2 aún no son compatibles. Algunas operaciones de los servicios que funcionan con transmisiones de eventos aún no se admiten y las operaciones de transmisión multimedia de los servicios, como Amazon Kinesis Video Streams, no funcionarán.

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

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Los ejemplos entre servicios son aplicaciones de muestra que funcionan en varios Servicios de AWS.

Ejemplos

- [Acciones y escenarios con SDK de SAP ABAP](#)

Acciones y escenarios con 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 Servicios de AWS

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Servicios

- [Ejemplos de Amazon Bedrock Runtime con el SDK de SAP ABAP](#)
- [CloudWatch ejemplos de uso del 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 Kinesis que utilizan el SDK de SAP ABAP](#)

- [Ejemplos de Lambda que utilizan SDK de SAP ABAP](#)
- [Ejemplos de Amazon S3 que utilizan SDK de SAP ABAP](#)
- [SageMaker ejemplos de uso del 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 Amazon Textract que utilizan el SDK de SAP ABAP](#)
- [Ejemplos de Amazon Translate que utilizan el SDK de SAP ABAP](#)

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.

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Anthropic Claude](#)
- [Difusión estable](#)

Anthropic Claude

InvokeModel

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

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](#).

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:\n\nTell me a joke\n\nAssistant:\n",
*   "max_tokens_to_sample": 2048,
*   "temperature": 0.5,
*   "top_k": 250,
*   "top_p": 1.0,
*   "stop_sequences": []
* }

DATA: BEGIN OF ls_input,
      prompt                TYPE string,
      max_tokens_to_sample TYPE /aws1/rt_shape_integer,
      temperature           TYPE /aws1/rt_shape_float,
      top_k                 TYPE /aws1/rt_shape_integer,
      top_p                 TYPE /aws1/rt_shape_float,
      stop_sequences        TYPE /aws1/rt_stringtab,
END OF ls_input.

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

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

```

```

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

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

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

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

ENDTRY.

```

Invoque el modelo básico de Anthropic Claude 2 para generar texto utilizando 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.
```

- Para obtener más información sobre la API, consulte el AWS SDK para ver [InvokeModel](#) referencia sobre la API ABAP de SAP.

Difusión estable

InvokeModel

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

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](#).

Crea 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-v0'
    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.

```

Utilice el modelo básico Stable Diffusion XL de Stability.ai para generar imágenes mediante el cliente de alto nivel L2.

```

        TRY.
            DATA(lo_bdr_l2_sd) = /aws1/
            cl_bdr_l2_factory=>create_stable_diffusion_10( 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 [InvokeModel](#) la referencia sobre la API ABAP del AWS SDK.

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

DeleteAlarms

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

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](#).

```
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 [DeleteAlarms](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeAlarms

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

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](#).

```
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 [DescribeAlarms](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DisableAlarmActions

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

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](#).

```
"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 [DisableAlarmActions](#) la referencia sobre la API ABAP del AWS SDK para SAP.

EnableAlarmActions

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

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](#).

```
"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 [EnableAlarmActions](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListMetrics

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

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](#).


```
"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 [ListMetrics](#) la referencia sobre la API ABAP del AWS SDK para SAP.

PutMetricAlarm

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

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](#).

```
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 [PutMetricAlarm](#) la referencia sobre la API ABAP del AWS SDK para 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 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 lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

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

"Create an ABAP internal table for the created alarm."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."

```

```

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

"Describe alarm using the same ABAP internal table."
TRY.
  oo_result = lo_cwt->describealarms(
    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 SDK de AWS para SAP ABAP.
 - [DeleteAlarms](#)
 - [DescribeAlarms](#)
 - [DisableAlarmActions](#)
 - [PutMetricAlarm](#)

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.

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateTable

En el siguiente ejemplo de código, se muestra cómo usar CreateTable.

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](#).

```
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 [CreateTable](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteItem

En el siguiente ejemplo de código, se muestra cómo usar DeleteItem.

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](#).


```
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 [DeleteItem](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteTable

En el siguiente ejemplo de código, se muestra cómo usar DeleteTable.

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](#).

```
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 [DeleteTable](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeTable

En el siguiente ejemplo de código, se muestra cómo usar DescribeTable.

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](#).

```

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 [DescribeTable](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetItem

En el siguiente ejemplo de código, se muestra cómo usar `GetItem`.

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](#).

```
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 [GetItem](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListTables

En el siguiente ejemplo de código, se muestra cómo usar `ListTables`.

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](#).


```
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( ).  
    ENLOOP.  
    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 [ListTables](#) la referencia sobre la API ABAP del AWS SDK para SAP.

PutItem

En el siguiente ejemplo de código, se muestra cómo usar PutItem.

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](#).

```
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 [PutItem](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Query

En el siguiente ejemplo de código, se muestra cómo usar Query.

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](#).

```

TRY.
  " Query movies for a given year .
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
    ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_year }| ) ) ).
  DATA(lt_key_conditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
    ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
      key = 'year'
      value = NEW /aws1/cl_dyncondition(
        it_attributevaluelist = lt_attributelist
        iv_comparisonoperator = |EQ|

```

```

    ) ) ) ).
    oo_result = lo_dyn->query(
      iv_tablename = iv_table_name
      it_keyconditions = lt_key_conditions ).
    DATA(lt_items) = oo_result->get_items( ).
    "You can loop over the results to get item attributes.
    LOOP AT lt_items INTO DATA(lt_item).
      DATA(lo_title) = lt_item[ key = 'title' ]-value.
      DATA(lo_year) = lt_item[ key = 'year' ]-value.
    ENDLLOOP.
    DATA(lv_count) = oo_result->get_count( ).
    MESSAGE 'Item count is: ' && lv_count TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
      MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

```

- 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 usar Scan.

SDK de 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.
    ENDLLOOP.
    DATA(lv_count) = oo_scan_result->get_count( ).
    MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

```

- 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 usar UpdateItem.

SDK de 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 [UpdateItem](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Introducción a tablas, elementos y consultas

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 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](#).

```
" Create an Amazon Dynamo DB table.
```

```
TRY.
```

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

```
DATA(lo_dyn) = /aws1/cl_dyn_factory=>create( lo_session ).
```

```

DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
  ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                        iv_keytype = 'HASH' ) )
  ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                        iv_keytype = 'RANGE' ) ) ).

DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
  ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                   iv_attributetype = 'N' ) )
  ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                   iv_attributetype = 'S' ) ) ).

" Adjust read/write capacities as desired.
DATA(lo_dynprovthroughput) = NEW /aws1/cl_dynprovthroughput(
  iv_readcapacityunits = 5
  iv_writecapacityunits = 5 ).
DATA(oo_result) = lo_dyn->createtable(
  it_keyschema = lt_keyschema
  iv_tablename = iv_table_name
  it_attributedefinitions = lt_attributedefinitions
  io_provisionedthroughput = lo_dynprovthroughput ).
" Table creation can take some time. Wait till table exists before
returning.
lo_dyn->get_waiter( )->tableexists(
  iv_max_wait_time = 200
  iv_tablename      = iv_table_name ).
MESSAGE 'DynamoDB Table' && iv_table_name && 'created.' TYPE 'I'.
" It throws exception if the table already exists.
CATCH /aws1/cx_dynresourceinuseex INTO DATA(lo_resourceinuseex).
  DATA(lv_error) = |"{ lo_resourceinuseex->av_err_code }" -
{ lo_resourceinuseex->av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

" Describe table
TRY.
  DATA(lo_table) = lo_dyn->describetable( iv_tablename = iv_table_name ).
  DATA(lv_tablename) = lo_table->get_table( )->ask_tablename( ).
  MESSAGE 'The table name is ' && lv_tablename TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table does not exist' TYPE 'E'.
ENDTRY.

" Put items into the table.

```

```

    TRY.
      DATA(lo_resp_putitem) = lo_dyn->putitem(
        iv_tablename = iv_table_name
        it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
      ) ).
      lo_resp_putitem = lo_dyn->putitem(
        iv_tablename = iv_table_name
        it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
Wars' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) )
      ) ).
      lo_resp_putitem = lo_dyn->putitem(
        iv_tablename = iv_table_name
        it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Speed' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
      ) ).
      " TYPE REF TO ZCL_AWS1_dyn_PUT_ITEM_OUTPUT

```

```

    MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
  CATCH /aws1/cx_dyncondalcheckfaile00.
    MESSAGE 'A condition specified in the operation could not be evaluated.'
  TYPE 'E'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  CATCH /aws1/cx_dyntransactconflictex.
    MESSAGE 'Another transaction is using the item' TYPE 'E'.
  ENDTRY.

" Get item from table.
TRY.
  DATA(lo_resp_getitem) = lo_dyn->getitem(
    iv_tablename          = iv_table_name
    it_key                = VALUE /aws1/cl_dynattributevalue=>tt_key(
      ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
'1975' ) ) )
      ) ).
  DATA(lt_attr) = lo_resp_getitem->get_item( ).
  DATA(lo_title) = lt_attr[ key = 'title' ]-value.
  DATA(lo_year) = lt_attr[ key = 'year' ]-value.
  DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  ENDTRY.

" Query item from table.
TRY.
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
    ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
  DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
    ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
      key = 'year'
      value = NEW /aws1/cl_dyncondition(
        it_attributelist = lt_attributelist
        iv_comparisonoperator = |EQ|

```

```

    ) ) ) ).
DATA(lo_query_result) = lo_dyn->query(
  iv_tablename = iv_table_name
  it_keyconditions = lt_keyconditions ).
DATA(lt_items) = lo_query_result->get_items( ).
READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
lo_title = lt_item[ key = 'title' ]-value.
lo_year = lt_item[ key = 'year' ]-value.
lo_rating = lt_item[ key = 'rating' ]-value.
MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
ENDTRY.

" Scan items from table.
TRY.
  DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
  lt_items = lo_scan_result->get_items( ).
  " Read the first item and display the attributes.
  READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  ENDTRY.

" Update items from table.
TRY.
  DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
  ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
    key = 'rating' value = NEW /aws1/cl_dynattrvalueupdate(
      io_value = NEW /aws1/cl_dynattributevalue( iv_n = '7.6' )
      iv_action = |PUT| ) ) ) ).
  DATA(lt_key) = VALUE /aws1/cl_dynattributevalue=>tt_key(
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
      key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) )
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(

```

```

        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'1980' ) ) ) ).
    DATA(lo_resp) = lo_dyn->updateitem(
        iv_tablename      = iv_table_name
        it_key             = lt_key
        it_attributeupdates = lt_attributeupdates ).
    MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.
    CATCH /aws1/cx_dyncondalcheckfaile00.
        MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

" Delete table.
TRY.
    lo_dyn->deletetable( iv_tablename = iv_table_name ).
    lo_dyn->get_waiter( )->tablenotexists(
        iv_max_wait_time = 200
        iv_tablename      = iv_table_name ).
    MESSAGE 'DynamoDB Table deleted.' TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dynresourceinuseex.
        MESSAGE 'The table cannot be deleted as it is in use' TYPE 'E'.
    ENDTRY.

```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)

- [Scan](#)
- [UpdateItem](#)

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)

Acciones

AllocateAddress

En el siguiente ejemplo de código, se muestra cómo usar `AllocateAddress`.

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](#).

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 [AllocateAddress](#) la referencia sobre la API ABAP del AWS SDK para SAP.

AssociateAddress

En el siguiente ejemplo de código, se muestra cómo usar AssociateAddress.

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](#).

```

  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 [AssociateAddress](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateKeyPair

En el siguiente ejemplo de código, se muestra cómo usar `CreateKeyPair`.

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](#).

```
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 [CreateKeyPair](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateSecurityGroup

En el siguiente ejemplo de código, se muestra cómo usar `CreateSecurityGroup`.

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](#).

```
TRY.
```

```

        oo_result = lo_ec2->createsecuritygroup(
returned for testing purposes. " oo_result is
        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 [CreateSecurityGroup](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteKeyPair

En el siguiente ejemplo de código, se muestra cómo usar DeleteKeyPair.

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](#).

```

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 [DeleteKeyPair](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteSecurityGroup

En el siguiente ejemplo de código, se muestra cómo usar DeleteSecurityGroup.

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](#).

```
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 [DeleteSecurityGroup](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeAddresses

En el siguiente ejemplo de código, se muestra cómo usar DescribeAddresses.

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](#).

```
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 [DescribeAddresses](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeAvailabilityZones

En el siguiente ejemplo de código, se muestra cómo usar DescribeAvailabilityZones.

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](#).

```

    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 [DescribeAvailabilityZones](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeInstances

En el siguiente ejemplo de código, se muestra cómo usar DescribeInstances.

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](#).

```

TRY.
    oo_result = lo_ec2->describeinstances( ) .
oo_result is returned for testing purposes. "

    " Retrieving details of EC2 instances. "
    DATA: lv_instance_id    TYPE /aws1/ec2string,
           lv_status        TYPE /aws1/ec2instancename,
           lv_instance_type TYPE /aws1/ec2instancetype,
           lv_image_id      TYPE /aws1/ec2string.
    LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
        LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
            lv_instance_id = lo_instance->get_instanceid( ).
            lv_status = lo_instance->get_state( )->get_name( ).
            lv_instance_type = lo_instance->get_instancetype( ).
            lv_image_id = lo_instance->get_imageid( ).
        ENDLLOOP.
    ENDLLOOP.
    MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [DescribeInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeKeyPairs

En el siguiente ejemplo de código, se muestra cómo usar DescribeKeyPairs.

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](#).

```
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 [DescribeKeyPairs](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeRegions

En el siguiente ejemplo de código, se muestra cómo usar DescribeRegions.

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](#).

```
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 [DescribeRegions](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeSecurityGroups

En el siguiente ejemplo de código, se muestra cómo usar DescribeSecurityGroups.

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](#).

```

    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 [DescribeSecurityGroups](#) la referencia sobre la API ABAP del AWS SDK para SAP.

MonitorInstances

En el siguiente ejemplo de código, se muestra cómo usar MonitorInstances.

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 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 [MonitorInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

RebootInstances

En el siguiente ejemplo de código, se muestra cómo usar RebootInstances.

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 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 [RebootInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ReleaseAddress

En el siguiente ejemplo de código, se muestra cómo usar ReleaseAddress.

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](#).

```

    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 [ReleaseAddress](#) la referencia sobre la API ABAP del AWS SDK para SAP.

RunInstances

En el siguiente ejemplo de código, se muestra cómo usar RunInstances.

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](#).

```

" Create tags for resource created during instance launch. "
DATA lt_tag specifications TYPE /aws1/
cl_ec2tag specifications=>tt_tag specifications list.
DATA ls_tag specifications LIKE LINE OF lt_tag specifications.
ls_tag specifications = NEW /aws1/cl_ec2tag specifications(
  iv_resourcetype = 'instance'
  it_tags = VALUE /aws1/cl_ec2tag=>tt_tag list(
    ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
  )
).
APPEND ls_tag specifications TO lt_tag specifications.

TRY.
  " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
  oo_result = lo_ec2->runinstances( " oo_result is
returned for testing purposes. "
  iv_imageid = iv_ami_id
  iv_instancetype = 't2.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 [RunInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StartInstances

En el siguiente ejemplo de código, se muestra cómo usar StartInstances.

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 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 [StartInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StopInstances

En el siguiente ejemplo de código, se muestra cómo usar StopInstances.

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 lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
    APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
lt_instance_ids.

    "Perform dry run"
    TRY.
        " DryRun is set to true. This checks for the required permissions to stop
the instance without actually making the request. "
        lo_ec2->stopinstances(
            it_instanceids = lt_instance_ids
            iv_dryrun = abap_true
        ).
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        " If the error code returned is `DryRunOperation`, then you have the
required permissions to stop this instance. "
        IF lo_exception->av_err_code = 'DryRunOperation'.
            MESSAGE 'Dry run to stop instance completed.' TYPE 'I'.
            " DryRun is set to false to stop instance. "
            oo_result = lo_ec2->stopinstances(           " oo_result is returned for
testing purposes. "
                it_instanceids = lt_instance_ids
                iv_dryrun = abap_false
            ).
            MESSAGE 'Successfully stopped the EC2 instance.' TYPE 'I'.
            " If the error code returned is `UnauthorizedOperation`, then you don't
have the required permissions to stop this instance. "
            ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
                MESSAGE 'Dry run to stop instance failed. User does not have permissions
to stop the instance.' TYPE 'E'.
            ELSE.
                DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
                MESSAGE lv_error TYPE 'E'.
            ENDIF.
        ENDTRY.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [StopInstances](#) la 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.

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateStream

En el siguiente ejemplo de código, se muestra cómo usar CreateStream.

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](#).

```
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_knslimitexceedex .
MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateStream](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteStream

En el siguiente ejemplo de código, se muestra cómo usar DeleteStream.

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](#).

```
TRY.
  lo_kns->deletestream(
    iv_streamname = iv_stream_name
  ).
  MESSAGE 'Stream deleted.' TYPE 'I'.
  CATCH /aws1/cx_knslimitexceedex .
  MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
  CATCH /aws1/cx_knsresourceinuseex .
  MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DeleteStream](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeStream

En el siguiente ejemplo de código, se muestra cómo usar DescribeStream.

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](#).

```
TRY.
    oo_result = lo_kns->describestream(
        iv_streamname = iv_stream_name
    ).
    DATA(lt_stream_description) = oo_result->get_streamdescription( ).
    MESSAGE 'Streams retrieved.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceedex .
    MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DescribeStream](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetRecords

En el siguiente ejemplo de código, se muestra cómo usar GetRecords.

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](#).

```

TRY.
    oo_result = lo_kns->getrecords(           " oo_result is returned for
testing purposes. "
        iv_sharditerator = iv_shard_iterator
    ).
    DATA(lt_records) = oo_result->get_records( ).
    MESSAGE 'Record retrieved.' TYPE 'I'.
CATCH /aws1/cx_knsexpirediteratorex .
    MESSAGE 'Iterator expired.' TYPE 'E'.
CATCH /aws1/cx_knsinvalidargumentex .
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex .
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex .
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex .
    MESSAGE 'KMS key used is in an invalid state.' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex .
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired .
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmsstrottlingex .
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [GetRecords](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListStreams

En el siguiente ejemplo de código, se muestra cómo usar `ListStreams`.

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](#).

```
TRY.  
    oo_result = lo_kns->liststreams(           " oo_result is returned for testing  
purposes. "  
    "Set Limit to specify that a maximum of streams should be returned."  
    iv_limit = iv_limit  
    ).  
    DATA(lt_streams) = oo_result->get_streamnames( ).  
    MESSAGE 'Streams listed.' TYPE 'I'.  
CATCH /aws1/cx_knslimitexceedex .  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListStreams](#) la referencia sobre la API ABAP del AWS SDK para SAP.

PutRecord

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

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](#).

```
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_knskmsstrottlingex .
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [PutRecord](#) la referencia sobre la API ABAP del AWS SDK para SAP.

RegisterStreamConsumer

En el siguiente ejemplo de código, se muestra cómo usar RegisterStreamConsumer.

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](#).

```
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 [RegisterStreamConsumer](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Introducción a flujos de datos

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 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_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_knslimitexceededx .
    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_knskmsstrottingex .
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Create a shard iterator in order to read the record."
TRY.
    lo_sharditerator = lo_kns->getsharditerator(
        iv_shardid = lo_record_result->get_shardid( )
        iv_sharditeratortype = iv_sharditeratortype
        iv_streamname = iv_stream_name
    ).
    MESSAGE 'Shard iterator created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
```

```

    CATCH /aws1/cx_knsprovthruputexcdex .
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourcenotfound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Read the record."
TRY.
    oo_result = lo_kns->getrecords(                " oo_result is returned
for testing purposes. "
        iv_sharditerator = lo_sharditerator->get_sharditerator( )
    ).
    MESSAGE 'Shard iterator created.' TYPE 'I'.
    CATCH /aws1/cx_knsexpirediteratorex .
        MESSAGE 'Iterator expired.' TYPE 'E'.
    CATCH /aws1/cx_knsinvalidargumentex .
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
    CATCH /aws1/cx_knskmsaccessdeniedex .
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
    CATCH /aws1/cx_knskmsdisabledex .
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
    CATCH /aws1/cx_knskmsinvalidstateex .
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
    CATCH /aws1/cx_knskmsnotfoundex .
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
    CATCH /aws1/cx_knskmsoptinrequired .
        MESSAGE 'KMS key option is required.' TYPE 'E'.
    CATCH /aws1/cx_knskmsstrottlingex .
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexcdex .
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_knsresourcenotfoundex .
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Delete stream."
TRY.
    lo_kns->deletestream(
        iv_streamname = iv_stream_name
    ).

```



```
    MESSAGE 'Stream deleted.' TYPE 'I'.
    CATCH /aws1/cx_knslimitexceeddex .
    MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
    CATCH /aws1/cx_knsresourceinuseex .
    MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
    ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [CreateStream](#)
 - [DeleteStream](#)
 - [GetRecords](#)
 - [GetShardIterator](#)
 - [PutRecord](#)

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.

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateFunction

En el siguiente ejemplo de código, se muestra cómo usar CreateFunction.

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](#).

```
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 [CreateFunction](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteFunction

En el siguiente ejemplo de código, se muestra cómo usar DeleteFunction.

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](#).

```
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 [DeleteFunction](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetFunction

En el siguiente ejemplo de código, se muestra cómo usar `GetFunction`.

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](#).

```
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 [GetFunction](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Invoke

En el siguiente ejemplo de código, se muestra cómo usar `Invoke`.

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](#).

```

TRY.
  DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
    `{`  &&
    `  "action": "increment",`  &&
    `  "number": 10`  &&
    `}`
  ).
  oo_result = lo_lmd->invoke(
    " oo_result is returned for
testing purposes. "
    iv_functionname = iv_function_name
    iv_payload = lv_json
  ).
  MESSAGE 'Lambda function invoked.' TYPE 'I'.
  CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
  CATCH /aws1/cx_lmdinvrequestcontex.
    MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
  CATCH /aws1/cx_lmdinvalidzipfileex.
    MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
  CATCH /aws1/cx_lmdrequesttoolargeex.
    MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
  CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
  CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
  CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
  CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
  CATCH /aws1/cx_lmdunsuppmediatyp00.
    MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
  ENDTRY.

```

- 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 usar `ListFunctions`.

SDK de 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 [ListFunctions](#) la referencia sobre la API ABAP del AWS SDK para SAP.

UpdateFunctionCode

En el siguiente ejemplo de código, se muestra cómo usar `UpdateFunctionCode`.

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](#).

```
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_lmdcodestorageexc dex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodeverification00.
        MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvalidcodesigex.
        MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
    CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [UpdateFunctionCode](#) la referencia sobre la API ABAP del AWS SDK para SAP.

UpdateFunctionConfiguration

En el siguiente ejemplo de código, se muestra cómo usar UpdateFunctionConfiguration.

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](#).

```
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 [UpdateFunctionConfiguration](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Comenzar a usar las funciones

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 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](#).

```
TRY.  
  "Create an AWS Identity and Access Management (IAM) role that grants AWS  
Lambda permission to write to logs."  
  DATA(lv_policy_document) = `{` &&  
    ` "Version": "2012-10-17", ` &&  
    ` "Statement": [ ` &&  
      `{` &&  
        ` "Effect": "Allow", ` &&  
        ` "Action": [ ` &&  
          ` "sts:AssumeRole" ` &&  
        ` ], ` &&  
        ` "Principal": { ` &&  
          ` "Service": [ ` &&  
            ` "lambda.amazonaws.com" ` &&  
          ` ] ` &&  
        ` } ` &&  
    ` } ` &&
```

```

        `}` ` &&
        `]` ` &&
    `}``.
TRY.
    DATA(lo_create_role_output) = lo_iam->createrole(
        iv_rolename = iv_role_name
        iv_assumerolepolicydocument = lv_policy_document
        iv_description = 'Grant lambda permission to write to logs'
    ).
    MESSAGE 'IAM role created.' TYPE 'I'.
    WAIT UP TO 10 SECONDS.          " Make sure that the IAM role is ready
for use. "
    CATCH /aws1/cx_iamentityalrddyexex.
        MESSAGE 'IAM role already exists.' TYPE 'E'.
    CATCH /aws1/cx_iaminvalidinputex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_iammalformedplydocex.
        MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
ENDTRY.

TRY.
    lo_iam->attachrolepolicy(
        iv_rolename = iv_role_name
        iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole'
    ).
    MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
    CATCH /aws1/cx_iaminvalidinputex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
    CATCH /aws1/cx_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 = lo_create_role_output->get_role( )->get_arn( )
        iv_handler = iv_handler
        io_code = io_initial_zip_file
        iv_description = 'AWS Lambda code example'
    ).
    MESSAGE 'Lambda function created.' TYPE 'I'.
CATCH /aws1/cx_lmdcodestorageexcdex.
    MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
ENDTRY.

" Verify the function is in Active state "
WHILE lo_lmd->getfunction( iv_functionname = iv_function_name )-
>get_configuration( )->ask_state( ) <> 'Active'.
    IF sy-index = 10.
        EXIT.                " Maximum 10 seconds. "
    ENDIF.
    WAIT UP TO 1 SECONDS.
ENDWHILE.

"Invoke the function with a single parameter and get results."
TRY.
    DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
        `{` &&
        ` "action": "increment",` &&
        ` "number": 10` &&
        `}`
    ).
    DATA(lo_initial_invoke_output) = lo_lmd->invoke(
        iv_functionname = iv_function_name
        iv_payload = lv_json
    ).
    ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
    " ov_initial_invoke_payload is returned for testing purposes. "
    DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
    CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
lo_writer_json.
    DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).

```

```

        MESSAGE 'Lambda function invoked.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdunsuppmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
    ENDRY.

    " Update the function code and configure its Lambda environment with an
environment variable. "
    " Lambda function is updated to perform 'decrement' action also. "
    TRY.
        lo_lmd->updatefunctioncode(
            iv_functionname = iv_function_name
            iv_zipfile = io_updated_zip_file
        ).
        WAIT UP TO 10 SECONDS.          " Make sure that the update is
completed. "
        MESSAGE 'Lambda function code updated.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodestorageexcex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDRY.

    TRY.
        DATA lt_variables TYPE /aws1/
cl_lmdenvironmentvaria00=>tt_environmentvariables.
        DATA ls_variable LIKE LINE OF lt_variables.
        ls_variable-key = 'LOG_LEVEL'.
        ls_variable-value = NEW /aws1/cl_lmdenvironmentvaria00( iv_value =
'info' ).
        INSERT ls_variable INTO TABLE lt_variables.

        lo_lmd->updatefunctionconfiguration(
            iv_functionname = iv_function_name
            io_environment = NEW /aws1/cl_lmdenvironment( it_variables =
lt_variables )

```

```

    ).
    WAIT UP TO 10 SECONDS.          " Make sure that the update is
completed. "
    MESSAGE 'Lambda function configuration/settings updated.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.'
TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
ENDTRY.

"Invoke the function with new parameters and get results. Display the
execution log that's returned from the invocation."
TRY.
    lv_json = /aws1/cl_rt_util=>string_to_xstring(
        `{`  &&
        ` "action": "decrement",`  &&
        ` "number": 10`  &&
        `}`
    ).
    DATA(lo_updated_invoke_output) = lo_lmd->invoke(
        iv_functionname = iv_function_name
        iv_payload = lv_json
    ).
    ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
    " ov_updated_invoke_payload is returned for testing purposes. "
    lo_writer_json = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
    CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
lo_writer_json.
    lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
    MESSAGE 'Lambda function invoked.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvrequestcontex.
    MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdunsuppmediatyp00.
    MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.

```

```
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 'E'.
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 'E'.
    CATCH /aws1/cx_iamplynnotattachableex.
        MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
        MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
ENDTRY.

" Delete the IAM role. "
TRY.
    lo_iam->deleterole( iv_rolename = iv_role_name ).
```

```
        MESSAGE 'IAM role deleted.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
        MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    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 SDK de AWS para SAP ABAP.
 - [CreateFunction](#)
 - [DeleteFunction](#)
 - [GetFunction](#)
 - [Invoke](#)
 - [ListFunctions](#)
 - [UpdateFunctionCode](#)
 - [UpdateFunctionConfiguration](#)

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.

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CopyObject

En el siguiente ejemplo de código, se muestra cómo usar CopyObject.

SDK de SAP ABAP

Note

Hay más información al respecto en 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 [CopyObject](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateBucket

En el siguiente ejemplo de código, se muestra cómo usar CreateBucket.

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](#).


```
TRY.  
  lo_s3->createbucket(  
    iv_bucket = iv_bucket_name  
  ).  
  MESSAGE 'S3 bucket created.' TYPE 'I'.  
CATCH /aws1/cx_s3_bucketalrddyexists.  
  MESSAGE 'Bucket name already exists.' TYPE 'E'.  
CATCH /aws1/cx_s3_bktalrddyownedbyyou.  
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateBucket](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteBucket

En el siguiente ejemplo de código, se muestra cómo usar DeleteBucket.

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](#).

```
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 [DeleteBucket](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteObject

En el siguiente ejemplo de código, se muestra cómo usar DeleteObject.

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](#).

```
  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 [DeleteObject](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetObject

En el siguiente ejemplo de código, se muestra cómo usar GetObject.

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](#).


```
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 [GetObject](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListObjectsV2

En el siguiente ejemplo de código, se muestra cómo usar ListObjectsV2.

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](#).

```
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.

PutObject

En el siguiente ejemplo de código, se muestra cómo usar PutObject.

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](#).

```

"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 [PutObject](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Escenarios

Comenzar a usar buckets y objetos

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 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_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).  
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).  
  
" Create an Amazon Simple Storage Service (Amazon S3) bucket. "  
TRY.  
    lo_s3->createbucket(  
        iv_bucket = iv_bucket_name  
    ).  
    MESSAGE 'S3 bucket created.' TYPE 'I'.  
CATCH /aws1/cx_s3_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 SDK de AWS para SAP ABAP.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

SageMaker 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 SageMaker

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

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateEndpoint

En el siguiente ejemplo de código, se muestra cómo usar CreateEndpoint.

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 lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA oo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.

"Create a production variant as an ABAP object."
"Identifies a model that you want to host and the resources chosen to deploy for
hosting it."
CREATE OBJECT lo_production_variants
  EXPORTING
    iv_variantname      = iv_variant_name
    iv_modelname        = iv_model_name
    iv_initialinstancecount = iv_initial_instance_count
    iv_instancetype     = iv_instance_type.

INSERT lo_production_variants INTO TABLE lt_production_variants.

"Create an endpoint configuration."
TRY.
  oo_ep_config_result = lo_sgm->createendpointconfig(
    iv_endpointconfigname = iv_endpoint_config_name
    it_productionvariants = lt_production_variants
  ).
  MESSAGE 'Endpoint configuration created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

```

```
"Create an endpoint."
TRY.
    oo_result = lo_sgm->createendpoint(      " oo_result is returned for testing
purposes. "
        iv_endpointconfigname = iv_endpoint_config_name
        iv_endpointname = iv_endpoint_name
    ).
    MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateEndpoint](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateModel

En el siguiente ejemplo de código, se muestra cómo usar CreateModel.

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_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.

"Create an ABAP object for the container image based on input variables."
CREATE OBJECT lo_primarycontainer
    EXPORTING
        iv_image          = iv_container_image
        iv_modeldataurl = iv_model_data_url.

"Create an Amazon SageMaker model."
TRY.
```

```

        oo_result = lo_sgm->createmodel(          " oo_result is returned for testing
purposes. "
        iv_executionrolearn = iv_execution_role_arn
        iv_modelname = iv_model_name
        io_primarycontainer = lo_primarycontainer
    ).
    MESSAGE 'Model created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourcecelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateModel](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateTrainingJob

En el siguiente ejemplo de código, se muestra cómo usar CreateTrainingJob.

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

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

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

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

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

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

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

"Create ABAP objects for training data sources."
CREATE OBJECT lo_trn_s3datasource
EXPORTING
  iv_s3datatype           = iv_trn_data_s3datatype
  iv_s3datadistributiontype = iv_trn_data_s3datadistribution
  iv_s3uri                = iv_trn_data_s3uri.

CREATE OBJECT lo_trn_datasource
EXPORTING
  io_s3datasource = lo_trn_s3datasource.

CREATE OBJECT lo_trn_channel
```

```
EXPORTING
  iv_channelname      = 'train'
  io_datasource       = lo_trn_datasource
  iv_compressiontype = iv_trn_data_compressiontype
  iv_contenttype      = iv_trn_data_contenttype.

INSERT lo_trn_channel INTO TABLE lt_input_data_config.

"Create ABAP objects for validation data sources."
CREATE OBJECT lo_val_s3datasource
  EXPORTING
    iv_s3datatype           = iv_val_data_s3datatype
    iv_s3datadistributiontype = iv_val_data_s3datadistribution
    iv_s3uri                = iv_val_data_s3uri.

CREATE OBJECT lo_val_datasource
  EXPORTING
    io_s3datasource = lo_val_s3datasource.

CREATE OBJECT lo_val_channel
  EXPORTING
    iv_channelname      = 'validation'
    io_datasource       = lo_val_datasource
    iv_compressiontype = iv_val_data_compressiontype
    iv_contenttype      = iv_val_data_contenttype.

INSERT lo_val_channel INTO TABLE lt_input_data_config.

"Create an ABAP object for algorithm specification."
CREATE OBJECT lo_algorithm_specification
  EXPORTING
    iv_trainingimage      = iv_training_image
    iv_traininginputmode = iv_training_input_mode.

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

"Create an ABAP object for output data configuration."
CREATE OBJECT lo_output_data_config
  EXPORTING
```

```
        iv_s3outputpath = iv_s3_output_path.

"Create an ABAP object for stopping condition."
CREATE OBJECT lo_stopping_condition
EXPORTING
    iv_maxruntimeinseconds = iv_max_runtime_in_seconds.

"Create a training job."
TRY.
    oo_result = lo_sgm->createtrainingjob( " oo_result is returned for
testing purposes. "
        iv_trainingjobname      = iv_training_job_name
        iv_rolearn              = iv_role_arn
        it_hyperparameters      = lt_hyperparameters
        it_inputdataconfig      = lt_input_data_config
        io_algorithmspecification = lo_algorithm_specification
        io_outputdataconfig     = lo_output_data_config
        io_resourceconfig       = lo_resource_config
        io_stoppingcondition    = lo_stopping_condition
    ).
    MESSAGE 'Training job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateTrainingJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

CreateTransformJob

En el siguiente ejemplo de código, se muestra cómo usar CreateTransformJob.

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_transforminput TYPE REF TO /aws1/cl_sgmtransforminput.  
DATA lo_transformoutput TYPE REF TO /aws1/cl_sgmtransformoutput.  
DATA lo_transformresources TYPE REF TO /aws1/cl_sgmtransformresources.  
DATA lo_datasource TYPE REF TO /aws1/cl_sgmtransformdatasrc.  
DATA lo_s3datasource TYPE REF TO /aws1/cl_sgmtransforms3datasrc.
```

"Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data source."

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

"Create an ABAP object for data source."

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

"Create an ABAP object for transform data source."

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

"Create an ABAP object for resource configuration."

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

"Create an ABAP object for output data configuration."

```
CREATE OBJECT lo_transformoutput
```

```

EXPORTING
  iv_s3outputpath = iv_s3_output_path.

"Create a transform job."
TRY.
  oo_result = lo_sgm->createtransformjob(      " oo_result is returned for
testing purposes. "
  iv_modelname = iv_tf_model_name
  iv_transformjobname = iv_tf_job_name
  io_transforminput = lo_transforminput
  io_transformoutput = lo_transformoutput
  io_transformresources = lo_transformresources
  ).
  MESSAGE 'Transform job created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourceinuse.
  MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceNotFound.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceLimitExcd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateTransformJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteEndpoint

En el siguiente ejemplo de código, se muestra cómo usar DeleteEndpoint.

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](#).

```

"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 [DeleteEndpoint](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteModel

En el siguiente ejemplo de código, se muestra cómo usar DeleteModel.

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](#).

```
    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'.
  ENDRY.

```

- Para obtener más información sobre la API, consulte [DeleteModel](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DescribeTrainingJob

En el siguiente ejemplo de código, se muestra cómo usar DescribeTrainingJob.

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](#).

```

  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'.
  ENDRY.

```

- Para obtener más información sobre la API, consulte [DescribeTrainingJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListAlgorithms

En el siguiente ejemplo de código, se muestra cómo usar ListAlgorithms.

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](#).

```
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 [ListAlgorithms](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListModels

En el siguiente ejemplo de código, se muestra cómo usar ListModels.

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](#).

```
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 [ListModels](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListNotebookInstances

En el siguiente ejemplo de código, se muestra cómo usar ListNotebookInstances.

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](#).

```
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 [ListNotebookInstances](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListTrainingJobs

En el siguiente ejemplo de código, se muestra cómo usar ListTrainingJobs.

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](#).

```
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 [ListTrainingJobs](#) la 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 SageMaker modelo.
- Crear una configuración de punto de conexión.

- Cree un punto de conexión y, a continuación, limpie los recursos.

SDK de 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_sgmalgorithm-spec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lv_model_data_url TYPE /aws1/sgmurl.

lv_model_data_url = iv_s3_output_path && iv_training_job_name && '/output/
model.tar.gz'.

"Create ABAP internal table for hyperparameters based on input variables."
"These hyperparameters are based on Amazon SageMaker built-in algorithm -
XGBoost"
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_max_depth.
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

```

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

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

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

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

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

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

"Create ABAP internal table for data based on input variables."
"Training data."
CREATE OBJECT lo_trn_s3datasource
EXPORTING
  iv_s3datatype           = iv_trn_data_s3datatype
  iv_s3datadistributiontype = iv_trn_data_s3datadistribution
  iv_s3uri                = iv_trn_data_s3uri.

CREATE OBJECT lo_trn_datasource EXPORTING io_s3datasource = lo_trn_s3datasource.

CREATE OBJECT lo_trn_channel
EXPORTING
  iv_channelname      = 'train'
  io_datasource       = lo_trn_datasource
  iv_compressiontype = iv_trn_data_compressiontype
  iv_contenttype      = iv_trn_data_contenttype.
INSERT lo_trn_channel INTO TABLE lt_input_data_config.

"Validation data."
CREATE OBJECT lo_val_s3datasource
```

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

CREATE OBJECT lo_val_datasource EXPORTING io_s3datasource = lo_val_s3datasource.

CREATE OBJECT lo_val_channel
  EXPORTING
    iv_channelname      = 'validation'
    io_datasource       = lo_val_datasource
    iv_compressiontype = iv_val_data_compressiontype
    iv_contenttype      = iv_val_data_contenttype.
INSERT lo_val_channel INTO TABLE lt_input_data_config.

"Create an ABAP object for algorithm specification based on input variables."
CREATE OBJECT lo_algorithm_specification
  EXPORTING
    iv_trainingimage      = iv_training_image
    iv_traininginputmode = iv_training_input_mode.

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

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

"Create an ABAP object for stopping condition."
CREATE OBJECT lo_stopping_condition EXPORTING iv_maxruntimeinseconds =
iv_max_runtime_in_seconds.

TRY.
  lo_sgm->createtrainingjob(
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
    it_inputdataconfig      = lt_input_data_config
    io_algorithmspecification = lo_algorithm_specification
    io_outputdataconfig     = lo_output_data_config
```



```

        io_resourceconfig          = lo_resource_config
        io_stoppingcondition      = lo_stopping_condition
    ).
    MESSAGE 'Training job created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceNotFound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceLimitExcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

    "Wait for training job to be completed."
    lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
    WHILE lo_training_result->get_trainingjobstatus( ) <> 'Completed'.
        IF sy-index = 30.
            EXIT.                "Maximum 900 seconds."
        ENDIF.
        WAIT UP TO 30 SECONDS.
        lo_training_result = lo_sgm->describetrainingjob( iv_trainingjobname =
iv_training_job_name ).
    ENDWHILE.

    "Create ABAP object for the container image based on input variables."
    CREATE OBJECT lo_primarycontainer
    EXPORTING
        iv_image          = iv_training_image
        iv_modeldataurl = lv_model_data_url.

    "Create an Amazon SageMaker model."
    TRY.
        lo_sgm->createmodel(
            iv_executionrolearn = iv_role_arn
            iv_modelname = iv_model_name
            io_primarycontainer = lo_primarycontainer
        ).
        MESSAGE 'Model created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourceLimitExcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

    "Create an endpoint production variant."
    CREATE OBJECT lo_production_variants

```

```

EXPORTING
  iv_variantname      = iv_ep_variant_name
  iv_modelname        = iv_model_name
  iv_initialinstancecount = iv_ep_initial_instance_count
  iv_instancetype     = iv_ep_instance_type.
INSERT lo_production_variants INTO TABLE lt_production_variants.

TRY.
  "Create an endpoint configuration."
  lo_ep_config_result = lo_sgm->createendpointconfig(
    iv_endpointconfigname = iv_ep_cfg_name
    it_productionvariants = lt_production_variants
  ).
  MESSAGE 'Endpoint configuration created.' TYPE 'I'.

  "Create an endpoint."
  oo_ep_output = lo_sgm->createendpoint(           " oo_ep_output is returned for
testing purposes. "
    iv_endpointconfigname = iv_ep_cfg_name
    iv_endpointname       = iv_ep_name
  ).
  MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresource-limit-excd.
  MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Wait for endpoint creation to be completed."
DATA(lo_endpoint_result) = lo_sgm->describeendpoint( iv_endpointname =
iv_ep_name ).
WHILE lo_endpoint_result->get_endpointstatus( ) <> 'InService'.
  IF sy-index = 30.
    EXIT.           "Maximum 900 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  lo_endpoint_result = lo_sgm->describeendpoint( iv_endpointname = iv_ep_name ).
ENDWHILE.

TRY.
  "Delete an endpoint."
  lo_sgm->deleteendpoint(
    iv_endpointname = iv_ep_name
  ).
  MESSAGE 'Endpoint deleted' TYPE 'I'.

```

```
"Delete an endpoint configuration."
lo_sgm->deleteendpointconfig(
  iv_endpointconfigname = iv_ep_cfg_name
).
MESSAGE 'Endpoint configuration deleted.' TYPE 'I'.

"Delete model."
lo_sgm->deletemodel(
  iv_modelname = iv_model_name
).
MESSAGE 'Model deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_endpointconfig_exception).
  DATA(lv_endpointconfig_error) = |"{ lo_endpointconfig_exception-
>av_err_code }" - { lo_endpointconfig_exception->av_err_msg }|.
  MESSAGE lv_endpointconfig_error TYPE 'E'.
ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [CreateEndpoint](#)
 - [CreateEndpointConfig](#)
 - [CreateModel](#)
 - [CreateTrainingJob](#)
 - [DeleteEndpoint](#)
 - [DeleteEndpointConfig](#)
 - [DeleteModel](#)
 - [DescribeEndpoint](#)
 - [DescribeTrainingJob](#)

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateTopic

En el siguiente ejemplo de código, se muestra cómo usar CreateTopic.

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](#).

```
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_snstopicclimitexcdex.  
        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 [CreateTopic](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteTopic

En el siguiente ejemplo de código, se muestra cómo usar DeleteTopic.

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](#).

```
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 [DeleteTopic](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetTopicAttributes

En el siguiente ejemplo de código, se muestra cómo usar GetTopicAttributes.

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](#).

```
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 [GetTopicAttributes](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListSubscriptions

En el siguiente ejemplo de código, se muestra cómo usar `ListSubscriptions`.

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](#).

```
TRY.
    oo_result = lo_sns->lists.subscriptions( ).           " 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 [ListSubscriptions](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListTopics

En el siguiente ejemplo de código, se muestra cómo usar `ListTopics`.

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](#).


```
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 [ListTopics](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Publish

En el siguiente ejemplo de código, se muestra cómo usar Publish.

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](#).

```
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.
```

- Para obtener información acerca de la API, consulte [Publish](#) (Publicar) en la Referencia de la API del SDK de AWS para SAP ABAP.

SetTopicAttributes

En el siguiente ejemplo de código, se muestra cómo usar SetTopicAttributes.

SDK de 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->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 [SetTopicAttributes](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Subscribe

En el siguiente ejemplo de código, se muestra cómo usar Subscribe.

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](#).

Suscribe una dirección de correo electrónico a un tema.

```
TRY.
    oo_result = lo_sns->subscribe(
        iv_topicarn = iv_topic_arn
        iv_protocol = 'email'
        iv_endpoint = iv_email_address
        iv_returnsubscriptionarn = abap_true
    ).
    MESSAGE 'Email address subscribed to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
CATCH /aws1/cx_snssubscriptionlmt00.
    MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
ENDTRY.
```

- Para obtener información acerca de la API, consulte [Suscribirse](#) en la Referencia de la API del SDK de AWS para SAP ABAP.

Unsubscribe

En el siguiente ejemplo de código, se muestra cómo usar Unsubscribe.

SDK de SAP ABAP

Note

Hay más información sobre 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 información acerca de la API, consulte [Cancelar suscripción](#) en la Referencia de la API del SDK de AWS 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 de 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](#).

Cree un tema de FIFO, suscriba una cola FIFO de Amazon SQS al tema y publique un mensaje en un tema de Amazon SNS.

```

" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsm_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsm_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsm_w( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

```

```

    TRY.
      DATA(lo_create_result) = lo_sns->createtopic(
        iv_name = iv_topic_name
        it_attributes = lt_tpc_attributes
      ).
      DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
      ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
      MESSAGE 'FIFO topic created' TYPE 'I'.
      CATCH /aws1/cx_snsstopiclimitexcdex.
        MESSAGE 'Unable to create more topics. You have reached the maximum number
of topics allowed.' TYPE 'E'.
      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 detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [CreateTopic](#)
 - [Publicación](#)
 - [Subscribe](#)

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

CreateQueue

En el siguiente ejemplo de código, se muestra cómo usar CreateQueue.

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](#).

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_sqsqueueexists.
    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_sqsqueuedeletedrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
    CATCH /aws1/cx_sqsqueueexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateQueue](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DeleteQueue

En el siguiente ejemplo de código, se muestra cómo usar DeleteQueue.

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](#).

```

    TRY.
        lo_sqs->deletequeue( iv_queueurl = iv_queue_url ).
        MESSAGE 'SQS queue deleted' TYPE 'I'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [DeleteQueue](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetQueueUrl

En el siguiente ejemplo de código, se muestra cómo usar `GetQueueUrl`.

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](#).

```
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 [GetQueueUrl](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListQueues

En el siguiente ejemplo de código, se muestra cómo usar `ListQueues`.

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](#).

```
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 [ListQueues](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ReceiveMessage

En el siguiente ejemplo de código, se muestra cómo usar ReceiveMessage.

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](#).

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'.
  ENDRY.

```

- Para obtener más información sobre la API, consulte [ReceiveMessage](#) la referencia sobre la API ABAP del AWS SDK para SAP.

SendMessage

En el siguiente ejemplo de código, se muestra cómo usar SendMessage.

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](#).

```

  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'.
  ENDRY.

```

- Para obtener más información sobre la API, consulte [SendMessage](#) la 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 de SAP ABAP

Note

Hay más información al respecto. [Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.](#)

Cree un tema de FIFO, suscriba una cola FIFO de Amazon SQS al tema y publique un mensaje en un tema de Amazon SNS.

```

" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsmmap_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmmap_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmmap_w( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

TRY.
  DATA(lo_create_result) = lo_sns->createtopic(
    iv_name = iv_topic_name
    it_attributes = lt_tpc_attributes
  ).
  DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
  ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
  MESSAGE 'FIFO topic created' TYPE 'I'.
CATCH /aws1/cx_snstopiclimitexcdex.
  MESSAGE 'Unable to create more topics. You have reached the maximum number
of topics allowed.' TYPE 'E'.
ENDTRY.

" Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
topic. "

```

```

" Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
an SNS FIFO topic. "
TRY.
  DATA(lo_subscribe_result) = lo_sns->subscribe(
    iv_topicarn = lv_topic_arn
    iv_protocol = 'sqs'
    iv_endpoint = iv_queue_arn
  ).
  DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
  ov_subscription_arn = lv_subscription_arn.
"
ov_subscription_arn is returned for testing purposes. "
  MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
CATCH /aws1/cx_snsnotfoundexception.
  MESSAGE 'Topic does not exist.' TYPE 'E'.
CATCH /aws1/cx_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 detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [CreateTopic](#)
 - [Publicación](#)
 - [Subscribe](#)

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

AnalyzeDocument

En el siguiente ejemplo de código, se muestra cómo usar `AnalyzeDocument`.

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](#).

```
"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 [AnalyzeDocument](#) la referencia sobre la API ABAP del AWS SDK para SAP.

DetectDocumentText

En el siguiente ejemplo de código, se muestra cómo usar DetectDocumentText.

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](#).

```

"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 [DetectDocumentText](#) la referencia sobre la API ABAP del AWS SDK para SAP.

GetDocumentAnalysis

En el siguiente ejemplo de código, se muestra cómo usar GetDocumentAnalysis.

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](#).

```
"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_texprovthruputexc dex.
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 [GetDocumentAnalysis](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StartDocumentAnalysis

En el siguiente ejemplo de código, se muestra cómo usar StartDocumentAnalysis.

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](#).

```
"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 [StartDocumentAnalysis](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StartDocumentTextDetection

En el siguiente ejemplo de código, se muestra cómo usar StartDocumentTextDetection.

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](#).

```
"Starts the asynchronous detection of text in a document."
"Amazon Textract can detect lines of text and the words that make up a line of
text."

"Create an ABAP object for the Amazon S3 object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
      iv_name   = iv_s3object ).
"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
"Start document analysis."
TRY.
      oo_result = lo_tex->startdocumenttextdetection( io_documentlocation =
lo_documentlocation ).
      DATA(lv_jobid) = oo_result->get_jobid( ).           "oo_result is returned
for testing purposes."
      MESSAGE 'Document analysis started.' TYPE 'I'.
      CATCH /aws1/cx_texaccessdeniedex.
      MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
      CATCH /aws1/cx_texbaddocumentex.
      MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
      CATCH /aws1/cx_texdocumenttoolargeex.
      MESSAGE 'The document is too large.' TYPE 'E'.
```

```
CATCH /aws1/cx_texidempotentprmmis00.  
  MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.  
CATCH /aws1/cx_texinternalservererr.  
  MESSAGE 'Internal server error.' TYPE 'E'.  
CATCH /aws1/cx_texinvalidkmskeyex.  
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.  
CATCH /aws1/cx_texinvalidparameterex.  
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.  
CATCH /aws1/cx_texinvalids3objectex.  
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.  
CATCH /aws1/cx_texlimitexceeddex.  
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.  
CATCH /aws1/cx_texprovthruputexcdex.  
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.  
CATCH /aws1/cx_texthrottlingex.  
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.  
CATCH /aws1/cx_texunsupporteddocex.  
  MESSAGE 'The document is not supported.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [StartDocumentTextDetection](#) la 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 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](#).

```
"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_texpvthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

"Get job ID from the output."
DATA(lv_jobid) = lo_start_result->get_jobid( ).

"Wait for job to complete."
oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).      " oo_result
is returned for testing purposes. "
WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.
  IF sy-index = 10.
    EXIT.                  "Maximum 300 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  oo_result = lo_tex->getdocumentanalysis( iv_jobid = lv_jobid ).
ENDWHILE.

DATA(lt_blocks) = oo_result->get_blocks( ).
LOOP AT lt_blocks INTO DATA(lo_block).
  IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, '.
    MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
  ENDIF.
ENDLOOP.

```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [GetDocumentAnalysis](#)
 - [StartDocumentAnalysis](#)

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 funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código en su contexto.

Temas

- [Acciones](#)
- [Escenarios](#)

Acciones

DescribeTextTranslationJob

En el siguiente ejemplo de código, se muestra cómo usar `DescribeTextTranslationJob`.

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](#).

```
"Gets the properties associated with an asynchronous batch translation job."
"Includes properties such as name, ID, status, source and target languages, and
input/output Amazon Simple Storage Service (Amazon S3) buckets."
TRY.
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
for testing purposes."
    EXPORTING
        iv_jobid      = iv_jobid
    ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
```

```

    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
  ENDTRY.

```

- Para obtener más información sobre la API, consulte [DescribeTextTranslationJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

ListTextTranslationJobs

En el siguiente ejemplo de código, se muestra cómo usar ListTextTranslationJobs.

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](#).

```

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

DATA lo_filter TYPE REF TO /aws1/cl_xl8textxl8tionjobfilt.

"Create an ABAP object for filtering using jobname."
CREATE OBJECT lo_filter
  EXPORTING
    iv_jobname = iv_jobname.

TRY.
  oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned for
testing purposes."
    EXPORTING
      io_filter      = lo_filter
    ).
  MESSAGE 'Jobs retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .

```



```

    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
    CATCH /aws1/cx_xl8invalidfilterex .
    MESSAGE 'The filter specified for the operation is not valid. Specify a
different filter.' TYPE 'E'.
    CATCH /aws1/cx_xl8invalidrequestex .
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [ListTextTranslationJobs](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StartTextTranslationJob

En el siguiente ejemplo de código, se muestra cómo usar StartTextTranslationJob.

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](#).

```
"Starts an asynchronous batch translation job."
```

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

```

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

"Create an ABAP object for the input data config."
CREATE OBJECT lo_inputdataconfig
    EXPORTING
        iv_s3uri          = iv_input_data_s3uri

```

```
        iv_contenttype = iv_input_data_contenttype.

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

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

TRY.
  oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
  EXPORTING
    io_inputdataconfig = lo_inputdataconfig
    io_outputdataconfig = lo_outputdataconfig
    it_targetlanguagecodes = lt_targetlanguagecodes
    iv_dataaccessrolelearn = iv_dataaccessrolelearn
    iv_jobname = iv_jobname
    iv_sourcelanguagecode = iv_sourcelanguagecode
  ).
  MESSAGE 'Translation job started.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
CATCH /aws1/cx_xl8invparamvalueex .
  MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
  MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00 .
  MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [StartTextTranslationJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

StopTextTranslationJob

En el siguiente ejemplo de código, se muestra cómo usar StopTextTranslationJob.

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](#).

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

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

- Para obtener más información sobre la API, consulte [StopTextTranslationJob](#) la referencia sobre la API ABAP del AWS SDK para SAP.

TranslateText

En el siguiente ejemplo de código, se muestra cómo usar TranslateText.

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](#).

```
"Translates input text from the source language to the target language."
TRY.
    oo_result = lo_xl8->translatetext(      "oo_result is returned for testing
purposes."
    EXPORTING
        iv_text          = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode
    ).
    MESSAGE 'Translation completed.' TYPE 'I'.
    CATCH /aws1/cx_xl8detectedlanguage00 .
        MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
    CATCH /aws1/cx_xl8internalserverex .
        MESSAGE 'An internal server error occurred.' TYPE 'E'.
    CATCH /aws1/cx_xl8invalidrequestex .
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
    CATCH /aws1/cx_xl8resourcenotfoundex .
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
    CATCH /aws1/cx_xl8serviceunavailex .
        MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
    CATCH /aws1/cx_xl8textsizegmtexcdex .
        MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex .
        MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
    CATCH /aws1/cx_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 [TranslateText](#) la 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 asíncrono.
- 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."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri      = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

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

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

```

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

TRY.
  DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
    EXPORTING
      io_inputdataconfig = lo_inputdataconfig
      io_outputdataconfig = lo_outputdataconfig
      it_targetlanguagecodes = lt_targetlanguagecodes
      iv_dataaccessrolelearn = iv_dataaccessrolelearn
      iv_jobname = iv_jobname
      iv_sourcelanguagecode = iv_sourcelanguagecode
    ).
  MESSAGE 'Translation job started.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8invparamvalueex .
    MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8invalidrequestex.
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time. '
TYPE 'E'.
  CATCH /aws1/cx_xl8unsuppdedlanguage00 .
    MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
ENDTRY.

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

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

```

```
    lo_des_translation_result = lo_xl8->describetexttranslationjob( iv_jobid =
lv_jobid ).
    ENDWHILE.

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

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
 - [DescribeTextTranslationJob](#)
 - [StartTextTranslationJob](#)

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 certificados con funciones de IAM en cualquier lugar.](#)
- [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 tres métodos de autenticación siguientes, que se seleccionan en la configuración del perfil del SDK enIMG.

AWS EI 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.

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](#)
- [Siguiendo el 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.
- Tienda 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](#).

Siguiente paso

Tras autenticar su sistema SAP AWS, el SDK para SAP ABAP asume automáticamente la función de IAM adecuada para la función empresarial del usuario de SAP. `sts:assumeRole`

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 Amazon EC2](#)
- [Roles de IAM para usuarios de SAP](#)

Práctica recomendada para el perfil de instancia Amazon EC2

La instancia 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.

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

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

El ejemplo anterior permite que el sistema SAP asuma los roles de IAM para el usuario de CFO (Director financiero), AUDITOR o INFORMES. AWS EI SDK elegirá el rol de IAM correcto para el usuario en función del rol de PFCG del usuario en SAP.

El perfil de instancia de Amazon EC2 también se puede utilizar para otras funciones.

- [Agente de AWS Backint 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 el trabajo del usuario: leer una tabla de DynamoDB, invocar Amazon Textract en un objeto PDF en Amazon S3, ejecutar una función AWS Lambda. Todos los SDK de AWS utilizan el mismo modelo de seguridad. 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.

```
{
  "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.

```

{
  "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.

```

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

```

}

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

Las funciones de PFCG se denominan funciones empresariales 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 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 CA en funciones de IAM en cualquier lugar como anclaje de confianza y cree un perfil para especificar los roles/políticas para 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.

- 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.

Pasos

- [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.

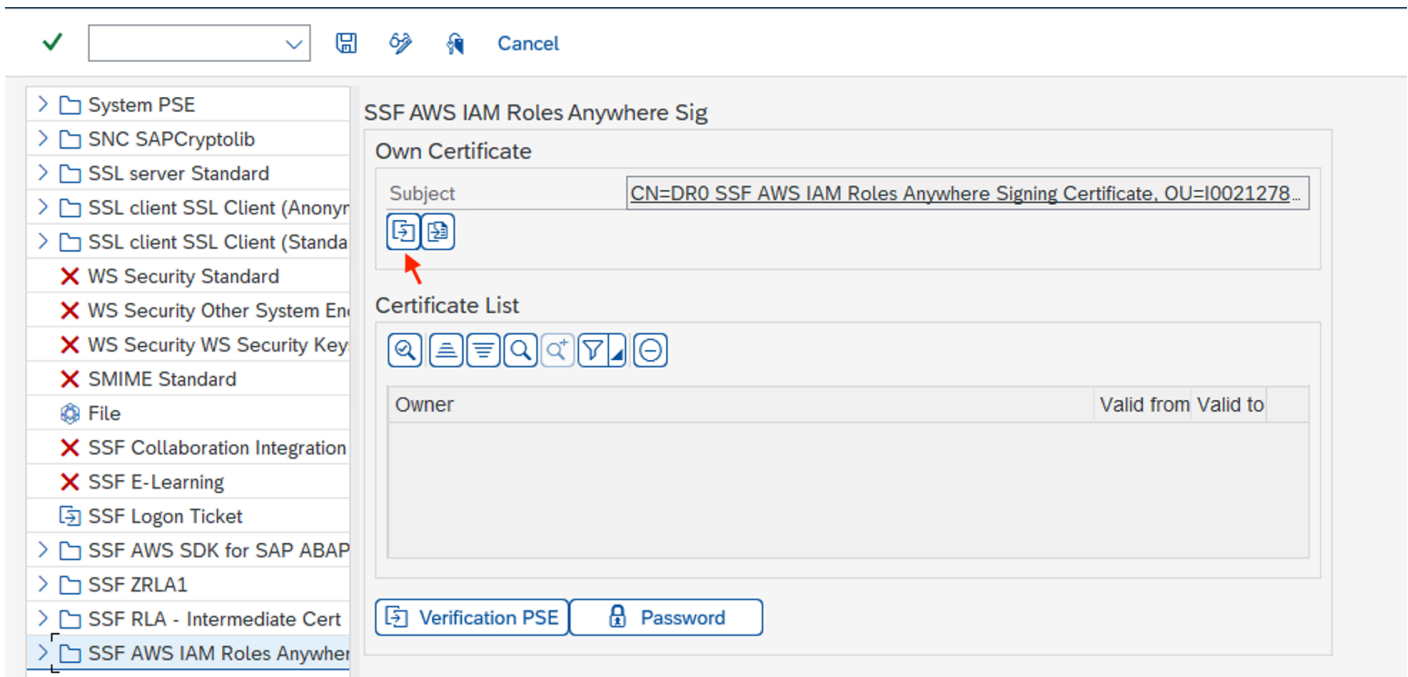
Paso 2: Definir los parámetros de SSF

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 de hash a SHA256 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 transacción /n/AWS1/IMG y seleccione Configuración de AWS SDK para SAP ABAP > Requisitos técnicos previos > Configuraciones de las aplicaciones.
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.

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

Certificate (SSF Application)

Enter your IAM Roles Anywhere details

Trust Anchor ARN

Profile ARN

4. 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 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

Pasos

- [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

Pasos

- [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 `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 clave de servicio creada. Para obtener más información, consulte [the section called “Paso 2: Crear una clave de servicio”](#).
 - 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. Seleccione Seleccionar transporte para seleccionar el transporte mediante la ayuda de valores.
8. Seleccione Guardar.

Solucionar 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_RFC4122_STATIC”

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 región de AWS en el parámetro `io_createbucketconfiguration`.

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 = 'test-bucket'  
  io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )  
).
```

Errores de SSL

Problema: el nombre de host del certificado del servidor SSL no coincide o se ha producido un error en el protocolo de enlace SSL con docs.aws.amazon.com:443: 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: no se pudo encontrar la configuración en el perfil <nombre_de_perfil> con el escenario DEFAULT para <sid>:<cliente>

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
 - Región de AWS
 - Método de autenticación
 - Seleccione el rol de instancia mediante los metadatos para los sistemas SAP que se estén ejecutando en 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 <arn_rol_iam> o el usuario: <arn_usuario> no está autorizado para realizar: sts:AssumeRole en el recurso: <arn_rol_iam>

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: el usuario <arn_usuario> no está autorizado a realizar: <acción> en el recurso: <arn_recurso>

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 hay ningún escenario activo configurado

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): error al conectar directamente a `tla.region.amazonaws.com:443`: 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`.

Reolución: el sistema SAP debe poder establecer la conexión con los puntos de conexión de AWS en el puerto HTTPS 443, ya sea directamente o mediante 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 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 los SDK y 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` Tiempo de ejecución y API, debe aplicar parches al módulo `core` y a todos los demás módulos que haya instalado, incluso si ya no los utiliza. 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.

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 eliminaciones en <https://.aws.amazon.com/V1/release/uninstall-abapsdk-LATEST.zip>. `sdk-for-sapabap`
`awsSdkSapabap`

```
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://.aws.amazon.com/v1/release/uninstall-ABAPSDK-latest.sig](https://.aws.amazon.com/v1/release/uninstall-ABAPSDK-latest.sig). `sdk-for-sapabap` `awsSdkSapabap` 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 para 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 de la Guía para AWS SDK para SAP ABAP desarrolladores

En la siguiente tabla se describen las versiones de la documentación de AWS SDK para SAP ABAP.

Cambio	Descripción	Fecha
Nuevo contenido	Versión preliminar para desarrolladores del SDK para SAP ABAP: edición BTP.	31 de mayo de 2024
Nuevo contenido	Se ha agregado Uso de certificados con funciones de IAM en cualquier lugar .	1 de diciembre de 2023
Nuevo contenido	Se ha agregado Creación de productos con SDK .	1 de diciembre de 2023
Nuevo contenido	Se ha agregado Comportamiento de los reintentos .	1 de diciembre de 2023
Nuevo contenido	Se ha agregado Licencias de SAP .	22 de septiembre de 2023
Versión pública	Relanzamiento inicial de la Guía AWS SDK para SAP ABAP para desarrolladores.	30 de junio de 2023
Nuevo contenido	Se han agregado características de AWS SDK para SAP ABAP .	30 de mayo de 2023
Nuevo contenido	Se agregó Solución de problemas de AWS SDK para SAP ABAP .	17 de febrero de 2023
Vista previa del desarrollador	Vista previa para desarrolladores de la Guía AWS SDK	17 de noviembre de 2022

para SAP ABAP para desarrolladores.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.