



Guida per gli sviluppatori

AWS SDK per SAP ABAP



AWS SDK per SAP ABAP: Guida per gli sviluppatori

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è SDK AWS per SAP ABAP?	1
Caratteristiche di SDK AWS per SAP ABAP	1
Manutenzione	1
Riferimento API	2
Prezzi	2
Risorse	2
Nozioni di base	3
Passaggio 1: prepara il tuo account AWS	3
Ruolo IAM per gli utenti SAP	3
Autenticazione	4
Fase 2: Installare l'SDK	6
Fase 3: Configurazione dell'SDK	6
Fase 4: Configurazione funzionale	8
Fase 5: Autorizzazione degli utenti SAP	10
Passaggio 6: scrivere il codice	12
Passaggio 7: Esegui l'applicazione	15
Configurazione	17
Prerequisiti SAP	17
SDK per SAP ABAP	17
SDK per SAP ABAP - edizione BTP	21
Installazione SDK AWS per SAP ABAP	22
Scaricare l'SDK	22
Verifica il file	22
AWS Trasporti SDK	23
Installazione dell'SDK - edizione BTP	27
Installa SDK per SAP ABAP - edizione BTP	27
Patching SDK per SAP ABAP - edizione BTP	27
Configurazione	28
Impostazioni globali	29
Impostazioni tecniche	29
Configura gli scenari	30
Configurazione dell'applicazione	30
Profilo SDK	31
Risolutore di risorse logiche	32

Esempio	32
Impostazioni di runtime	33
Registra e traccia	33
OPT-IN: telemetria avanzata	33
Scenario attivo	34
Aggiornamento, tracciamento e telemetria	34
Aggiornamento del sistema SAP	34
Traccia	35
Telemetria	36
Scenari di connettività avanzati	36
Connessione tramite un server proxy	37
Connessione tramite un firewall che ispeziona i pacchetti	38
Endpoint gateway	38
Endpoint con interfaccia personalizzata	38
Accesso agli endpoint in più regioni	39
Utilizzo di SDK	41
Rappresentazione dei dati	41
Tipi di dati	42
Tipi di dati AWS	44
Programma di esempio	44
Prerequisiti	45
Codice	45
Sezioni di codice	46
Concetti	49
Classi API	49
Oggetti aggiuntivi	49
Classi di struttura	50
Matrici	52
Mappe	53
Funzioni di livello superiore	53
Funzionalità	1
Configurazione programmatica	54
Waiter	54
Impaginatori	56
Comportamento di ripetizione	57
Prodotti per l'edilizia	57

Impostazione di un ID prodotto	58
Limitazioni	58
Esempi di codice	60
Azioni e scenari	60
Runtime di Amazon Bedrock	61
CloudWatch	66
DynamoDB	74
Amazon EC2	88
Kinesis	104
Lambda	114
Amazon S3	128
SageMaker	137
Amazon SNS	156
Amazon SQS	165
Amazon Textract	172
Amazon Translate	183
Sicurezza	192
Autenticazione del sistema	192
Autenticazione dei metadati	193
Autenticazione con chiave di accesso segreta	193
Autenticazione basata su certificati utilizzando IAM Roles Anywhere	194
Approfondimenti	194
Le migliori pratiche per la sicurezza IAM	195
Procedura consigliata per il profilo di istanza Amazon EC2	195
Ruoli IAM per utenti SAP	196
Autorizzazioni SAP	199
Autorizzazioni per la configurazione	199
Autorizzazioni SAP per gli utenti finali	200
Operazioni sicure	201
Crittografia dei dati a riposo	201
Crittografia dei dati in transito	202
Utilizzo delle API	2
Utilizzo dei certificati	202
Prerequisiti	202
Procedura	203
Archivio credenziali	21

Fasi di configurazione	206
Utilizzo di SAP Credential Store con l'SDK	208
Risoluzione dei problemi	212
Errore di importazione	212
Vincolo di posizione non specificato	212
Errore SSL	213
Configurazione del profilo	214
Autorizzazione IAM	215
Autorizzazione per le azioni	215
Scenario attivo	34
Caratteri speciali	216
Connettività	216
Argomenti aggiuntivi	217
Rilasci	217
Strategia di rilascio	217
Best practice	195
SDK di patch per SAP ABAP	218
Installazione di un modulo aggiuntivo	218
Disinstallazione di SDK per SAP ABAP	218
Licenze SAP	219
Cronologia dei documenti	221
.....	ccxxii

Che cos'è SDK AWS per SAP ABAP?

SDK AWS per SAP ABAP fornisce un'interfaccia ai servizi offerti da AWS nel linguaggio ABAP. Utilizzando l'SDK, puoi implementare BADI ABAP, report, transazioni, servizi OData e altri artefatti ABAP su Servizi AWS, come Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) e altro ancora. Amazon DynamoDB Amazon Translate Puoi anche sviluppare per sistemi basati su ABAP, a partire da SAP 7.4 e in un ambiente NetWeaver SAP Business Technology Platform. Per ulteriori informazioni, consulta [Installazione dell' AWS SDK per SAP ABAP - edizione BTP](#).

Argomenti

- [Caratteristiche di SDK AWS per SAP ABAP](#)
- [Manutenzione e supporto per le versioni principali dell'SDK](#)
- [Riferimento API](#)
- [Prezzi](#)
- [Risorse aggiuntive](#)

Caratteristiche di SDK AWS per SAP ABAP

SDK AWS per SAP ABAP è stato progettato per essere familiare e naturale per gli sviluppatori SAP. Ad esempio, mentre tutti Servizi AWS utilizzano le false stringhe true and per rappresentare dati booleani nelle strutture XML e JSON, SDK per SAP ABAP li converte in valori nativi ABAP e a carattere singolo. 'X' ' ' SDK per SAP ABAP utilizza il più possibile costrutti ABAP nativi, anche nei tipi di dati e nei formati di timestamp. Di conseguenza, il programmatore ABAP non deve preoccuparsi della serializzazione JSON e XML sottostante o del formato wire del protocollo API.

Manutenzione e supporto per le versioni principali dell'SDK

Per informazioni sulla manutenzione e sul supporto per le versioni principali dell'SDK e le relative dipendenze sottostanti, consulta quanto segue nella [Guida di riferimento degli strumenti e degli SDK AWS](#):

- [AWS Politica di manutenzione degli SDK e degli strumenti](#)
- [AWS SDK, strumenti e matrice di supporto delle versioni](#)

Riferimento API

Per visualizzare un elenco completo delle SDK AWS per SAP ABAP API, consulta [SDK AWS per SAP ABAP - Guida di riferimento alle API](#).

Per visualizzare un elenco completo dei moduli SDK AWS per SAP ABAP TLA, vedi [SDK AWS per SAP ABAP - Elenco dei moduli](#).

Prezzi

SDK AWS per SAP ABAP è disponibile senza costi aggiuntivi. Paghi solo per AWS le risorse e i servizi che utilizzi con l'SDK.

Risorse aggiuntive

Oltre a questa guida, sono disponibili le seguenti risorse online per SDK per SAP ABAP.

- [SAP sulla documentazione AWS](#)
- [AWS blog per sviluppatori](#)
- [AWS forum per sviluppatori](#)
- [AWS Libreria di esempi di codice SDK](#)
- [@awsdevelopers](#) (Twitter)

Guida introduttiva con SDK AWS per SAP ABAP

Questa sezione descrive come iniziare a usare l'SDK. Include informazioni sull'installazione dell'SDK, sull'esecuzione della configurazione di base e sulla creazione di un esempio di codice Hello World che traduce una frase da una lingua all'altra. Se non conosci AWS SDK, ti consigliamo di eseguire questi passaggi in un ambiente sandbox.

Fasi

- [Passaggio 1: prepara il tuo account AWS](#)
- [Fase 2: Installare l'SDK](#)
- [Fase 3: Configurazione dell'SDK](#)
- [Fase 4: Configurazione funzionale](#)
- [Fase 5: Autorizzazione degli utenti SAP](#)
- [Passaggio 6: scrivere il codice](#)
- [Passaggio 7: Esegui l'applicazione](#)

Passaggio 1: prepara il tuo account AWS

Per iniziare a usare SDK per SAP ABAP, devi avere un SDK attivo. Account AWS Ne hai bisogno Account AWS anche se il tuo sistema SAP è ospitato in locale, su SAP Business Technology Platform (BTP) o con un altro provider di servizi cloud.

Se il tuo sistema SAP è in esecuzione su AWS Cloud, effettuerai chiamate ai servizi del tuo. AWS Account AWS

Argomenti

- [Ruolo IAM per gli utenti SAP](#)
- [Autenticazione](#)

Ruolo IAM per gli utenti SAP

- Crea un ruolo IAM con le istruzioni fornite nella Guida per l'AWS Identity and Access Management utente. Per ulteriori informazioni, consulta [Creazione di un ruolo per delegare le autorizzazioni](#)

[a un AWS servizio](#). Prendi nota dell'Amazon Resource Name (ARN) del ruolo IAM per un uso successivo.

- Seleziona Amazon EC2 come caso d'uso.
- Usa SapDemoTranslate come nome del ruolo.
- Allega il TranslateReadOnly profilo al ruolo.
- Il ruolo deve avere le seguenti entità per consentire al sistema SAP di assumere il ruolo. Sostituisci **"111122223333"** con il numero di account AWS .

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

Questo esempio mostra che qualsiasi dirigente del Account AWS **"111122223333"** può assumere il ruolo. Si tratta di un'autorizzazione ampia adatta per proof-of-concept. È possibile utilizzare un principio più ristretto per la produzione, come negli esempi seguenti.

- Un utente specifico, quando il sistema SAP utilizza una delle seguenti opzioni:
 - Credenziali crittografate SSF da un sistema SAP locale
 - Credenziali dal servizio SAP Credential Store in ambiente SAP BTP, ABAP
- Un ruolo specifico: quando il sistema SAP si trova su Amazon EC2 e c'è un profilo di istanza.
- Amazon EC2: quando il sistema SAP è su Amazon EC2 e non esiste un profilo di istanza.

Per ulteriori informazioni, consulta [Best practice for IAM Security](#).

Autenticazione

L'autenticazione dipende da dove è ospitato il sistema SAP.

Posizioni

- [Sul cloud AWS](#)
- [In locale, SAP BTP o altro cloud](#)

Sul cloud AWS

Assicurati che l'istanza EC2 su cui è in esecuzione il tuo sistema SAP abbia un profilo di istanza con le seguenti autorizzazioni.

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

Aggiungi l'ARN che hai annotato nel passaggio precedente.

Questa autorizzazione consente al sistema SAP di assumere il SapDemoTranslate ruolo per conto dell'utente ABAP.

In locale, SAP BTP o altro cloud

Se il tuo sistema SAP si trova in locale, su SAP BTP o su un altro cloud, utilizza i seguenti passaggi per stabilire una connessione per l'autenticazione utilizzando una chiave di accesso segreta.

1. Crea un utente IAM. Per ulteriori informazioni, consulta [Creazione di utenti IAM](#) (console).
2. Utilizza SapDemoSID come nome dell'utente IAM. SID è l'ID di sistema del tuo sistema SAP.
3. Assegna SapDemoTranslate un ruolo a questo utente.

Conserva la `manoaccess_key`. `secret_access_key` È necessario configurare queste credenziali nel sistema SAP.

Note

Se il tuo sistema SAP si trova in locale, su SAP BTP o su un altro cloud, puoi autenticarti utilizzando una delle seguenti opzioni.

- [Autenticazione con chiave di accesso segreta](#) tramite SSF o SAP Credential Store
- [Utilizzo di certificati con IAM Roles Anywhere](#)

Fase 2: Installare l'SDK

Consulta le seguenti schede per le istruzioni di installazione.

SDK for SAP ABAP

Importa i trasporti SDK per SAP ABAP nel tuo sistema SAP. Puoi importare i trasporti in qualsiasi client. Per ulteriori informazioni, consulta [Installazione di SDK per SAP ABAP](#).

SDK for SAP ABAP - BTP edition

Installa SDK per SAP ABAP - edizione BTP utilizzando l'applicazione Deploy Product. Per ulteriori informazioni, consulta [Installazione dell'SDK per SAP ABAP - edizione BTP](#).

Fase 3: Configurazione dell'SDK

Consulta le seguenti schede per le istruzioni di configurazione.

SDK for SAP ABAP

Esegui la `/AWS1/IMG` transazione per aprire la Guida all'implementazione di SDK per SAP ABAP. Per eseguire questa transazione, accedi `/n/AWS1/IMG` nella barra dei comandi del tuo sistema SAP, quindi scegli Invio.

Completa le seguenti configurazioni.

- Vai a Prerequisiti tecnici.
 - Controlla i parametri di profilo consigliati e le impostazioni del certificato SSL.
- Vai a Impostazioni globali → Configura scenari.

- Modifica le impostazioni, in base ai consigli in Impostazioni globali.
- Vai a Impostazioni globali → Impostazioni tecniche.
 - Modifica le impostazioni, in base ai consigli in Impostazioni globali.
- Vai a Impostazioni di runtime → Log And Trace.
 - Seleziona Nuove voci.
 - Livello di traccia: Nessuna traccia.
 - Linee di scarico massime:100.
 - OPT-IN: enh telemetry: Lascia vuoto questo campo.
 - Seleziona Salva.
- Vai a Impostazioni di runtime → Scenario attivo.
 - In Nuovo scenario, selezionaDEFAULT.
 - Seleziona Conferma modifica dello scenario.
 - Accetta la richiesta.

SDK for SAP ABAP - BTP edition

Apri l'ambiente ABAP in un browser Web e accedi all'applicazione Custom Business Configurations.

Completa le seguenti configurazioni.

- Vai a Impostazioni globali → Configura scenari.
 - Modifica le impostazioni, in base ai consigli in Impostazioni globali.
- Vai a Impostazioni globali → Impostazioni tecniche.
 - Modifica le impostazioni, in base ai consigli in Impostazioni globali.

Prerequisiti per i sistemi locali

Se il sistema SAP è in esecuzione in locale o in un altro cloud, le credenziali devono essere archiviate nel database SAP. Le credenziali sono crittografate utilizzando SAP SSF e richiedono una libreria crittografica configurata, come quella di SAP. CommonCryptoLib

I passaggi per configurare SSF for SDK per SAP ABAP sono descritti nella transazione. /AWS1/IMG

Note

Il prerequisito precedente non si applica se il sistema SAP è in esecuzione su Amazon EC2. I sistemi SAP in esecuzione su Amazon EC2 recuperano credenziali a rotazione automatica e di breve durata dai metadati delle istanze Amazon EC2.

Fase 4: Configurazione funzionale

Consulta le seguenti schede per le istruzioni di configurazione.

SDK for SAP ABAP

Esegui la transazione /AWS1/IMG (inserisci /n/AWS1/IMG nella barra dei comandi e scegli Invio) per aprire la guida all'implementazione per AWS SDK.

- Vai a Configurazione dell'applicazione → Profilo SDK.
 - Seleziona Nuove voci.
 - Profilo: DEMO.
 - Descrizione: Demo profile.
 - Seleziona Salva.
- Evidenzia la voce che hai creato e fai clic sul ramo dell'albero Autenticazione e impostazioni.
 - Seleziona Nuove voci.
 - SID: l'ID di sistema del sistema SAP in cui ti trovi attualmente.
 - Client: il client del sistema SAP in cui ti trovi attualmente.
 - ID scenario: l'elenco a discesa in cui troverai lo scenario DEFAULT creato dall'amministratore di Basis.
 - AWS Regione: inserisci la AWS regione verso la quale desideri effettuare chiamate. Se il tuo sistema SAP è in esecuzione AWS, inserisci la AWS regione in cui è in esecuzione.
 - Metodo di autenticazione:
 - Seleziona Instance Role tramite metadati se il tuo sistema SAP è in esecuzione su Amazon EC2.
 - Seleziona Credentials from SSF Storage se il tuo sistema SAP è in esecuzione in locale o in un altro cloud.
 - Seleziona Imposta credenziali.

- Inserisci l'ID della chiave di accesso e la chiave di accesso segreta che hai creato nel passaggio precedente.
- Lascia vuoto il campo Disable IAM roles.
- Seleziona Salva.
- Fai clic sul ramo dell'albero IAM Role Mapping.
 - Seleziona Nuove voci.
 - Immettere il numero di sequenza: 010.
 - Inserisci il ruolo Logical IAM: TESTUSER.
 - Inserisci l'ARN del ruolo IAM: inserisci l'arn:aws: del ruolo IAM contenente la TranslateReadOnly policy creata nel passaggio precedente.

SDK for SAP ABAP - BTP edition

Configura l'autenticazione utilizzando SAP Credential Store. Per ulteriori informazioni, consulta [Using SAP Credential Store](#).

Apri l'ambiente ABAP in un browser Web e accedi all'applicazione Custom Business Configurations.

- Vai a Configurazione dell'applicazione → Profilo SDK.
 - Seleziona Modifica per creare un nuovo profilo.
 - Profilo: DEMO.
 - Descrizione: Demo profile.
- Seleziona il tasto freccia destra accanto alla voce creata e vai alla scheda Autenticazione e impostazioni.

Seleziona Nuove voci.

- SID: l'ID di sistema del sistema SAP in cui ti trovi attualmente.
- Client: il client del sistema SAP in cui ti trovi attualmente.
- ID scenario: l'elenco a discesa in cui troverai lo scenario DEFAULT creato dall'amministratore di Basis.
- AWS Regione: inserisci la AWS regione verso la quale desideri effettuare chiamate. Se il tuo sistema SAP è in esecuzione AWS, inserisci la AWS regione in cui è in esecuzione.

- Immettere il namespace e il nome chiave delle credenziali archiviate in SAP Credentials Store.
- Immettere il nome dell'accordo di comunicazione creato per stabilire la comunicazione tra SDK per SAP ABAP - edizione BTP e SAP Credential Store.
- Lascia vuoto il campo Disabilita i ruoli IAM.
- Fai clic con il pulsante destro del mouse sul tasto freccia accanto alla voce creata e vai alla scheda IAM Role Mapping.

Seleziona Nuove voci.

- Immettere il numero di sequenza: 010.
- Inserisci il ruolo Logical IAM: TESTUSER.
- Inserisci l'ARN del ruolo IAM: inserisci l'arn:aws: del ruolo IAM contenente la `TranslateReadOnly` policy creata nel passaggio precedente.

Fase 5: Autorizzazione degli utenti SAP

Gli utenti SAP non sono autorizzati a utilizzare la AWS funzionalità per impostazione predefinita. Gli utenti devono essere autorizzati esplicitamente utilizzando le autorizzazioni SAP. Per ulteriori dettagli, consulta le schede seguenti.

SDK for SAP ABAP

Crea un ruolo PFCG

- Vai alla transazione PFCG
- Inserisci il nome del ruolo ZAWS_SDK_DEMO_TESTUSER e seleziona Crea ruolo singolo.
 - Descrizione:Role for demo AWS SDK functionality.
 - Vai alla scheda Autorizzazioni.
 - Seleziona Modifica dati di autorizzazione e accetta il pop-up informativo.
 - Nel pop-up Scegli modello, seleziona Non selezionare i modelli.
 - Seleziona Aggiungi manualmente dalla barra degli strumenti.
 - Aggiungi i seguenti oggetti di autorizzazione:
 - /AWS1/LROL
 - /AWS1/SESS

- Nell'albero di autorizzazione, inserisci:
 - Profilo per l'accesso alle API AWS: DEMO
 - Ruolo IAM logico: TESTUSER
- Seleziona Salva.
- Seleziona Genera.
- Seleziona Indietro.
- Seleziona Salva per salvare il ruolo.

Assegna il ruolo PFCG agli utenti SAP

Qualsiasi utente a cui è stato assegnato il ZAWS_SDK_DEMO_TESTUSER ruolo sarà autorizzato a utilizzare le funzioni AWS SDK con le impostazioni configurate nel profilo SDK. DEMO L'utente autorizzato assumerà anche il ruolo IAM mappato al ruolo IAM TESTUSER logico in quel profilo.

- Esegui transazione. SU01
 - Inserisci l'ID utente di un utente SAP che testerà la funzionalità AWS SDK.
 - Seleziona Modifica.
 - Vai alla scheda Ruoli e assegna il ZAWS_SDK_DEMO_TESTUSER ruolo all'utente.
 - Seleziona Salva.

SDK for SAP ABAP - BTP edition

Crea un ruolo aziendale

- Apri il tuo ambiente ABAP in un browser Web e accedi all'applicazione Maintain Business Roles.
- Seleziona Crea da modello e inserisci i seguenti dettagli.
 - Modello: scegli **/AWS1/RT_BTP_ENDUSER**.
 - Nuovo ID del ruolo aziendale: inserisci un ID.
 - Descrizione del nuovo ruolo aziendale: inserisci una descrizione.
- Seleziona OK per visualizzare la pagina relativa al ruolo aziendale.
- Nella scheda Dettagli generali del ruolo, vai a Access Categories e imposta il campo Write, Read, Value Help su Restricted.

- Seleziona **Mantieni** restrizioni ed espandi **Tipi di restrizione** assegnati dal riquadro di navigazione a sinistra. Aggiorna il seguente campo nella sezione **Restrizioni e valori**.
- In **Scegli la sessione SDK**, seleziona l'icona a forma di matita accanto a **Profilo SDK** e vai alla scheda **Intervalli**. Entra **DEMO** e seleziona **Aggiungi**.
- In **Scegli il ruolo IAM logico**, seleziona l'icona a forma di matita accanto a **Logical IAM Role** e vai alla scheda **Ranges**. Entra **TESTUSER** e seleziona **Aggiungi**.

Seleziona l'icona a forma di matita accanto a **Profilo SDK** e vai alla scheda **Intervalli**. Entra **DEMO** e seleziona **Aggiungi**

- Torna al modello di ruolo aziendale e apri la scheda **Utenti aziendali**. Seleziona **Aggiungi** per assegnare il ruolo di business appena creato a un utente aziendale SAP che testerà la funzionalità SDK. Seleziona **Salva**.

Qualsiasi utente aziendale assegnato al ruolo di business creato sarà autorizzato a utilizzare le funzioni AWS SDK con le impostazioni configurate nel profilo SDK. **DEMO** L'utente autorizzato assumerà anche il ruolo IAM mappato al ruolo IAM **TESTUSER** logico in quel profilo.

Passaggio 6: scrivere il codice

Consulta le seguenti schede per maggiori dettagli.

SDK for SAP ABAP

1. Transazione aperta. SE38

- Inserisci **ZDEMO_TRANSLATE_HELLO_WORLD** come nome del programma.
- Seleziona **Create**.
- Inserisci **AWS SDK Hello World In Any Language** come titolo.
- Tipo: scegli **Programma eseguibile**.
- Stato: scegli **Test Program**.
- Seleziona **Salva**.
- Salva il programma come oggetto locale.

Aggiungete il codice seguente.

```

*&-----*
*& 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. Fai clic con il pulsante destro del mouse sul pacchetto in cui deve essere creata la classe ABAP, quindi seleziona Nuovo > classe ABAP.
2. Immettete **ZCL_DEMO_XL8_SIMPLE** il nome della classe e aggiungete una descrizione della classe. Seleziona Avanti.

3. Crea o scegli una richiesta di trasporto. Seleziona Fine.

Aggiungi il codice seguente.

```
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.
```

Passaggio 7: Esegui l'applicazione

Per ulteriori dettagli, consulta le schede seguenti.

SDK for SAP ABAP

Esegui l'applicazione inSE38. In caso di successo, il risultato sarà il seguente.

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

Se mancano le autorizzazioni, la configurazione o i prerequisiti di base, è possibile che venga visualizzato un messaggio di errore. Guarda l'esempio seguente.

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

Se il tuo ruolo SAP ti autorizza a utilizzare un profilo SDK e a mapparlo a un ruolo IAM logico mentre le tue autorizzazioni IAM non sono configurate per consentire al sistema SAP di assumere il ruolo IAM, il risultato sarà il seguente.

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

In questo caso, esamina le autorizzazioni IAM e la configurazione di trust sui ruoli IAM, sugli utenti o su entrambi definiti in [the section called “Passaggio 1: prepara il tuo account AWS”](#)

SDK for SAP ABAP - BTP edition

Esegui l'applicazione su Eclipse > Esegui come > Applicazione ABAP (console). In caso di successo, il risultato sarà il seguente.

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

Se mancano le autorizzazioni, la configurazione o i prerequisiti di base, è possibile che venga visualizzato un messaggio di errore. Guarda l'esempio seguente.

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

Se il tuo ruolo SAP ti autorizza a utilizzare un profilo SDK e a mapparlo a un ruolo IAM logico mentre le tue autorizzazioni IAM non sono configurate per consentire al sistema SAP di assumere il ruolo IAM, il risultato sarà il seguente.

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

In questo caso, esamina le autorizzazioni IAM e la configurazione di trust sui ruoli IAM, sugli utenti o su entrambi definiti in [the section called “Passaggio 1: prepara il tuo account AWS”](#)

Configurazione

Questa sezione fornisce informazioni su come configurare l'ambiente di sviluppo da utilizzare SDK AWS per SAP ABAP.

Argomenti

- [Prerequisiti SAP](#)
- [Installazione SDK AWS per SAP ABAP](#)
- [Installazione dell' AWS SDK per SAP ABAP - edizione BTP](#)

Prerequisiti SAP

I seguenti prerequisiti per l'installazione dell'SDK sono applicabili quando i sistemi SAP sono ospitati su. AWS

Argomenti

- [Prerequisiti per SDK per SAP ABAP AWS](#)
- [Prerequisiti per AWS SDK per SAP ABAP - edizione BTP](#)

Prerequisiti per SDK per SAP ABAP AWS

Di seguito sono riportati i prerequisiti per SDK per AWS SAP ABAP.

Argomenti

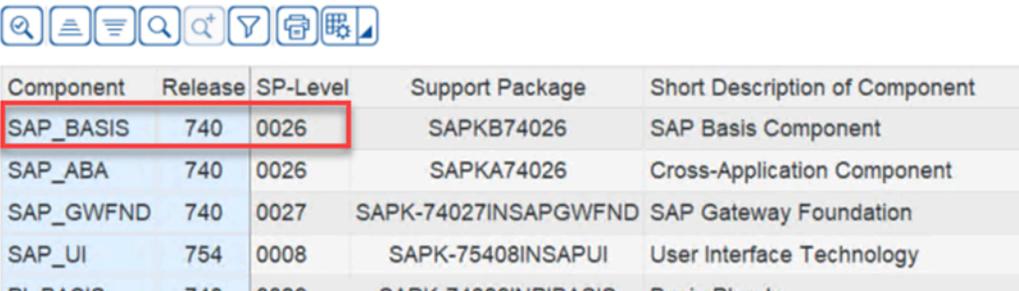
- [Versione base](#)
- [Versione del kernel](#)
- [Parametri](#)
- [Note](#)
- [Connettività in uscita](#)
- [Connettività HTTPS](#)
- [Accesso ai metadati delle istanze Amazon EC2](#)

Versione base

SDK per SAP ABAP è compatibile con SAP NetWeaver 7.4 e versioni successive. SDK per SAP ABAP non riguarda alcuna tabella di applicazioni SAP. È completamente agnostico per quanto riguarda le applicazioni, come SAP Enterprise Resource Planning e SAP Landscape Transformation Replication Server.

Il livello di SP minimo supportato per è. SAP_BASIS 740 SP 0008 Per ulteriori informazioni, vedere [SAP Note 1856171 - Supporto dei campi del modulo con lo stesso nome in CL_HTTP_ENTITY](#) (richiede l'accesso al portale SAP). In base ai requisiti aziendali, è possibile scegliere un livello SP superiore, come mostrato nell'immagine seguente.

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

Non è previsto un requisito minimo di livello SP per SAP_BASIS 750 le release successive.

Versione del kernel

L'SDK per SAP ABAP e gli strumenti che utilizzano Internet Communication Manager (ICM) per la connettività HTTP si affidano al kernel SAP per le sue funzionalità crittografiche, HTTP, XML e JSON. Ti consigliamo di utilizzare l'ultima versione del kernel compatibile con la tua piattaforma SAP. NetWeaver Il requisito minimo è la release 741 del kernel. Per ulteriori informazioni, consulta [SAP Note 2083594 - Versioni del kernel SAP e livelli di patch del kernel SAP](#) (richiede l'accesso al portale SAP).

Se si utilizza la versione 741 o 742 del kernel, sono richiesti i seguenti livelli di patch:

- 741, patch n. 212
- 742, patch 111

Parametri

Il sistema SAP deve supportare Server Name Indication (SNI) come descritto nelle seguenti note SAP (richiede l'accesso al portale SAP).

- [SAP Note 2124480 - ICM/Web Dispatcher: TLS Extension Server Name Indication \(SNI\) come client](#)
- [SAP Note 2582368 - Aggiornamento SAPSSL per l'invio lato client dell'estensione TLS SNI tramite saphttp, sapkprotp, sldreg](#)

Configura il seguente parametro nel file. DEFAULT.PFL

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

Note

Applica la seguente nota SAP al tuo sistema.

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

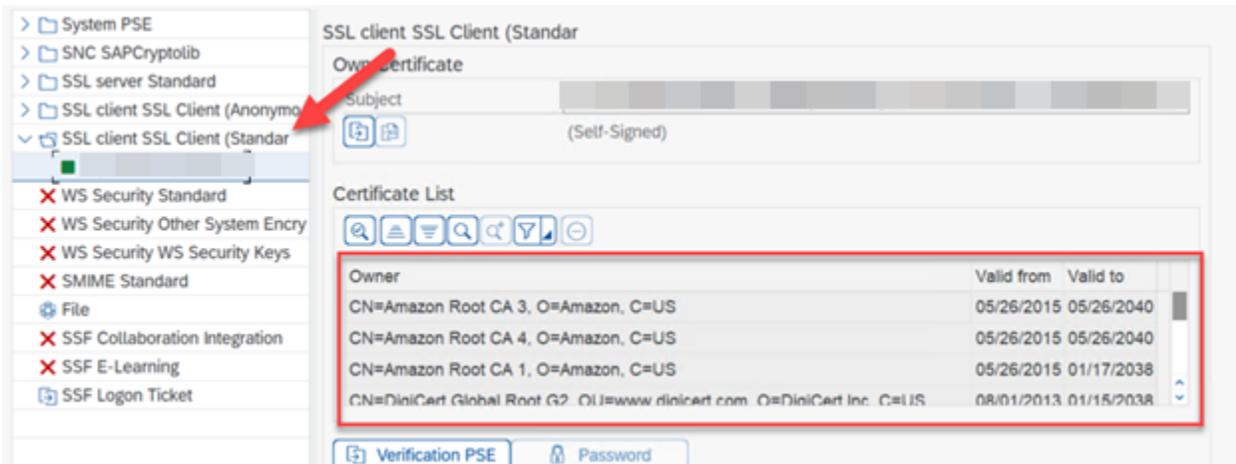
Connettività in uscita

SDK per SAP ABAP è un client HTTPS. Il sistema SAP invia messaggi HTTPS in uscita. Non è richiesta alcuna connettività in entrata.

Connettività HTTPS

Tutte le chiamate AWS API vengono effettuate con canali HTTPS crittografati. Il sistema SAP deve essere configurato in modo da considerare attendibili AWS i certificati per stabilire una connessione HTTPS in uscita.

1. [Vai a https://www.amazontrust.com/repository/](https://www.amazontrust.com/repository/).
2. In Root CA, scarica tutti i certificati utilizzando il link PEM.
3. Importa questi certificati SSL Client (Standard) PSE su ciascuno STRUST dei tuoi sistemi SAP, come mostrato nell'immagine seguente.



Accesso ai metadati delle istanze Amazon EC2

Il sistema ABAP effettua connessioni HTTP non crittografate a localhost (<http://169.254.169.254>) per abilitare i metadati delle istanze Amazon EC2. Il canale HTTP viene utilizzato solo per recuperare AWS le credenziali dal server locale. Il traffico HTTP rimane all'interno dell'host.

I metadati consentono a un sistema SAP di AWS autenticarsi in modo sicuro senza archiviare una chiave segreta nel SAP Secure Store. Questa funzionalità è applicabile solo ai sistemi SAP ospitati su Amazon EC2.

Configura il DEFAULT.PFL file con il seguente parametro per consentire al sistema SAP di effettuare una connessione HTTP in uscita non crittografata.

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

Utilizzate il seguente parametro per abilitare la connessione HTTP in uscita senza abilitare la connessione in entrata.

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

Verifica che il tuo sistema SAP sia configurato per le connessioni HTTP in uscita con i seguenti passaggi:

1. Esegui una transazione SMICM.
2. Vai a Active Services.
3. Verifica di vedere un segno di spunta verde nella riga HTTP, nella colonna Attivo, come mostrato nell'immagine seguente.

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

Prerequisiti per AWS SDK per SAP ABAP - edizione BTP

I seguenti sono gli unici prerequisiti per l'edizione AWS SDK per SAP ABAP - BTP.

Argomenti

- [SAP Landscape Portal — edizione BTP](#)
- [SAP Credential Store — edizione BTP](#)

SAP Landscape Portal — edizione BTP

Questo prerequisito è applicabile solo per l'edizione AWS SDK per SAP ABAP - BTP.

SAP Landscape Portal è l'unico meccanismo supportato per installare componenti aggiuntivi in un ambiente SAP BTP. Assicurati di essere abbonato per utilizzare questo servizio. Per ulteriori informazioni, consulta [Landscape Portal](#).

SAP Credential Store — edizione BTP

Questo prerequisito è applicabile solo per l'edizione AWS SDK per SAP ABAP - BTP.

Nell'anteprima per sviluppatori, l'autenticazione con chiave di accesso segreta è l'unico meccanismo supportato per l'autenticazione dell' AWS SDK per l'edizione SAP ABAP - BTP. L'SDK legge le credenziali dal Credential Store e archivia la chiave di accesso segreta in modo sicuro.

È necessario soddisfare i seguenti prerequisiti.

- Abbonamento a Credential Store.
- Credential Store assegnato come diritto al tuo account secondario BTP. Vedi [Configurazione iniziale per maggiori dettagli](#).
- Un'istanza di servizio con piano standard per Credential Store. Per maggiori dettagli, consulta [Creare un'istanza di servizio](#).

Per ulteriori informazioni, consulta [Using SAP Credential Store](#).

Il servizio SAP Credential Store viene eseguito in SAP BTP al di fuori del sistema ABAP BTP. Vedi [SAP Credential Store](#) per maggiori dettagli.

Installazione SDK AWS per SAP ABAP

Argomenti

- [Scarica SDK per SAP ABAP](#)
- [Verify SDK per il file SAP ABAP \(opzionale\)](#)
- [AWS Trasporti SDK](#)

Scarica SDK per SAP ABAP

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

Una volta completato il download, ti consigliamo di decomprimere il file scaricato in una directory, ad esempio. /tmp/awssdk

Verify SDK per il file SAP ABAP (opzionale)

Questo passaggio facoltativo di convalida della firma del file SDK ti aiuta a confermare che l'SDK non è stato manomesso. Utilizza i seguenti passaggi per verificare il tuo file SDK.

1. Scarica il file SDK SIGNATURE con il seguente comando.

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

2. Copiare la chiave pubblica seguente e salvarla in un file denominato `abapsdk-signing-key.pem`.

```
-----BEGIN PUBLIC KEY-----  
MIICIjANBgkqhkiG9w0BAQEFAAACg8AMIICCGKCAgEAmS3oN3wKBh4HJ0Ga0tye  
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv  
051sTdEJJpfLEWJJZZNK3v9fGWkyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
```

```
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuisJC21MPkz0
cHOR31MNsMLzm+d/gVklT31R/JwAcFCkXTWvR8/V0WNGZZXdVUbefrI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLjz8dH
Zg25o79U9EJ1n+1pqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. Verifica il file ZIP SDK scaricato con il seguente comando. Il comando richiede `openssl` che faccia parte di molte distribuzioni Linux.

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

4. Verificate che l'output del comando precedente sia `Verified OK`
5. Se l'output è `Verification Failure`, ripeti i passaggi precedenti. Se continui a ricevere un output non riuscito, non installare l'SDK e non contattarci. AWS Support

AWS Trasporti SDK

Argomenti

- [Indice](#)
- [Importing \(Importazione\)](#)
- [Spazio dei nomi](#)

Indice

L'installazione di SDK per SAP ABAP viene completata tramite ABAP Transports. È necessario importare questi trasporti nel proprio ambiente di sviluppo o sandbox.

Ogni versione di SDK per SAP ABAP sostituisce completamente quella precedente. Non è necessario applicare trasporti incrementali. I trasporti sono raggruppati in un file ZIP. Di seguito è riportata la struttura del file ZIP.

```
transports/
```

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

La transports cartella contiene una core sottocartella. La core sottocartella contiene i principali trasporti di runtime e una sottocartella per ogni modulo, denominata con l'abbreviazione di tre lettere del modulo. [Per un elenco completo dei moduli TLA, vedi - Elenco dei moduli.SDK AWS per SAP ABAP](#)

AWS I trasporti SDK sono richieste di workbench. A seconda della configurazione dei percorsi TMS, l'SDK potrebbe non essere inoltrato automaticamente alle code di controllo qualità e produzione dopo l'importazione nel sistema precedente. È necessario aggiungerle manualmente alla coda di ciascun sistema.

Quando il progetto è pronto per la fase successiva, l' AWS SDK può essere importato insieme a trasporti separati contenenti il proprio Z codice con funzionalità aziendali. Se utilizzi un sistema di controllo delle modifiche, come SAP Change Request Management (CharM), consulta il tuo amministratore di CharM per una corretta gestione dei trasporti di terze parti.

Importing (Importazione)

Argomenti

- [Indicatori chiave](#)
- [È ora di importare](#)

AWS I trasporti SDK sono indipendenti dal client. Il trasporto principale è obbligatorio e contiene il codice di runtime SDK, l'API per AWS Security Token Service e l'API per Amazon Simple Storage Service. I moduli SDK rimanenti vengono forniti ciascuno in un trasporto separato. Per mantenere ridotte le dimensioni dell'SDK nel sistema, ogni modulo SDK è opzionale. È possibile installare moduli aggiuntivi in un secondo momento, se necessario per la logica aziendale.

Ad esempio, se desideri utilizzare le API per Amazon S3 Amazon Translate e importare core il trasporto (contenente core runtime, Amazon S3 e i moduli) AWS STS e x18 il trasporto (contenente il modulo per) i trasporti. Amazon Translate

Per visualizzare un elenco completo degli SDK per le API SAP ABAP, consulta [SDK](#) per SAP ABAP - Guida di riferimento delle API.

Di seguito sono riportati i punti chiave per l'importazione dei trasporti SDK. AWS

- Ogni trasporto viene consegnato come e Knnnnnn .AWS Rnnnnnn .AWS
 - Knnnnnn .AWS deve essere copiato in /usr/sap/trans/cofiles
 - Rnnnnnn .AWS deve essere copiato in. /usr/sap/trans/data
- Quando si importano i trasporti, è necessario selezionare l'opzione Ignora la versione del componente non valida disponibile in Import Transport Request > Options > Import Options.
- Tutti i trasporti desiderati possono essere importati contemporaneamente.
- Se si importano i trasporti separatamente, è necessario importare core prima il trasporto.
- Il livello di rilascio di tutti i trasporti deve essere identico.

È ora di importare

AWS L'importazione dei trasporti SDK può richiedere molti minuti. I trasporti hanno esito positivo se STMS mostra una luce verde (RC=0) o gialla (). RC=4

- Una luce rossa (RC=8) indica che l'importazione ha avuto un errore di sintassi.
 - Seleziona Richiesta → Visualizza → Registri per esaminare l'errore di importazione.
 - Durante l'importazione, se viene generato un errore a causa di un'interfaccia mancante IF_SYSTEM_UUID_RFC4122_STATIC, assicurati che SAP Note 2619546 sia applicato al sistema. [Per ulteriori informazioni, consulta Note.](#)
 - Se la causa dell'errore è sconosciuta, contatta AWS Support.
- Un fulmine rosso (RC=12) indica che i file di trasporto non sono stati caricati correttamente /usr/sap/trans o non dispongono delle autorizzazioni necessarie.

Indicatori chiave

Di seguito sono riportati i punti chiave per l'importazione AWS dei trasporti SDK.

- Ogni trasporto viene consegnato come e Knnnnnn .AWS Rnnnnnn .AWS

- `Knnnnnn.AWS` deve essere copiato in `/usr/sap/trans/cofiles`
- `Rnnnnnn.AWS` deve essere copiato in `/usr/sap/trans/data`
- Quando si importano i trasporti, è necessario selezionare l'opzione Ignora la versione del componente non valida disponibile in Import Transport Request > Options > Import Options.
- Tutti i trasporti desiderati possono essere importati contemporaneamente.
- Se si importano i trasporti separatamente, è necessario importare core prima il trasporto.
- Il livello di rilascio di tutti i trasporti deve essere identico.

È ora di importare

AWS L'importazione dei trasporti SDK può richiedere molti minuti. I trasporti hanno esito positivo se STMS mostra una luce verde (RC=0) o gialla (). RC=4

- Una luce rossa (RC=8) indica che l'importazione ha avuto un errore di sintassi.
 - Seleziona Richiesta → Visualizza → Registri per esaminare l'errore di importazione.
 - Durante l'importazione, se viene generato un errore a causa di un'interfaccia mancante `IF_SYSTEM_UUID_RFC4122_STATIC`, assicurati che SAP Note 2619546 sia applicato al sistema. [Per ulteriori informazioni, consulta Note.](#)
 - Se la causa dell'errore è sconosciuta, contatta AWS Support.
- Un fulmine rosso (RC=12) indica che i file di trasporto non sono stati caricati correttamente `/usr/sap/trans` o non dispongono delle autorizzazioni necessarie.

Spazio dei nomi

SDK per SAP ABAP utilizza lo spazio dei `/AWS1/` nomi e non modifica gli oggetti SAP o altri oggetti del sistema con la seguente eccezione.

- AWS auth gli oggetti si trovano in una classe di oggetti Auth. Le classi di oggetti Auth sono limitate a quattro caratteri e non supportano i namespace. SDK per SAP ABAP utilizza Auth Object Class `is.YAW1` Se hai già una classe di oggetti di autenticazione `YAW1` nella transazione `SU21`, contatta prima dell'installazione. AWS Support

Installazione dell' AWS SDK per SAP ABAP - edizione BTP

L'edizione BTP è disponibile in anteprima per sviluppatori e può essere installata partecipando all'anteprima. Per installare l'SDK, compila il modulo di partecipazione su [AWS SDK for SAP ABAP - Anteprima per sviluppatori dell'edizione BTP](#).

Prima di installare SDK per SAP ABAP - BTP edition, assicurati di soddisfare i prerequisiti richiesti. [Per ulteriori informazioni, consulta SAP Landscape Portal e SAP Credential Store](#).

Argomenti

- [Installa SDK per SAP ABAP - edizione BTP](#)
- [Patching SDK per SAP ABAP - edizione BTP](#)

Installa SDK per SAP ABAP - edizione BTP

1. Vai alla tua istanza di SAP Landscape Portal e avvia l'applicazione Fiori selezionando Deploy Product.
2. In Prodotti > Prodotti partner, seleziona. **/AWS1/SDK_OMNI**

Contattaci AWS Support se non lo vedi **/AWS1/SDK_OMNI** dopo essere stato accettato nell'anteprima per sviluppatori.

3. In Target Version, scegli la versione di SDK per SAP ABAP - BTP edition che desideri installare sul tuo sistema.
4. Seleziona le caselle di controllo per tutti i SID su cui desideri installare l'SDK.

L'installazione può richiedere 30-45 minuti e include tempi di inattività del sistema. Per maggiori dettagli, consulta [Deploy](#) Product.

Patching SDK per SAP ABAP - edizione BTP

Il processo di patching per SDK per SAP ABAP - edizione BTP è simile al processo di installazione. Se installi l'SDK su un sistema su cui è già installata una versione precedente, l'SDK viene aggiornato sulla nuova versione che hai scelto.

Configurazione SDK AWS per SAP ABAP

Prima dell'uso SDK AWS per SAP ABAP, è necessario configurare l'SDK con le impostazioni tecniche e funzionali necessarie per le operazioni dell'SDK. Alcune impostazioni sono trasportabili, altre sono impostazioni di runtime. Molte impostazioni sono direttamente analoghe alle impostazioni definite nei .INI file per altri SDK.

Le configurazioni SDK, ad eccezione delle impostazioni di Runtime, devono essere completate nell'ambiente di sviluppo. È possibile trasferire le configurazioni al controllo qualità e alla produzione seguendo le consuete regole di trasporto e controllo delle modifiche. La configurazione trasportabile non è consigliata per gli ambienti di produzione.

Configurazione SDK AWS per SAP ABAP

Per eseguire la transazione di configurazione, inserisci `/n/AWS1/IMG` nella barra dei comandi SAPGUI. [Se non disponi delle autorizzazioni per configurare l' AWS SDK, consulta Autorizzazioni SAP.](#)

Configurazione dell' AWS SDK per SAP ABAP - edizione BTP

Utilizza i seguenti passaggi per configurare SDK per l'edizione SAP ABAP - BTP.

1. Apri il tuo ambiente ABAP in un browser web.
2. Passa all'applicazione Custom Business Configurations.

Nell'applicazione Custom Business Configuration, puoi raggruppare le configurazioni in base al tipo di impostazioni SDK. Utilizza i seguenti passaggi per raggruppare le configurazioni.

1. Apri l'ambiente ABAP in un browser Web e accedi all'applicazione Custom Business Configurations.
2. Seleziona Impostazioni > Gruppo e scegli Gruppo di configurazione dall'elenco a discesa. Seleziona OK.
3. Le configurazioni sono ora disponibili in una struttura gerarchica. Per salvare la vista, vedere [Visualizzazioni \(Gestione delle varianti\)](#) - Componenti.

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Impostazioni globali](#)
- [Configurazione dell'applicazione](#)
- [Impostazioni di runtime](#)
- [Argomenti di aggiornamento, tracciamento e telemetria per SDK AWS per SAP ABAP](#)
- [Scenari di connettività avanzati](#)

Impostazioni globali

Utilizza `/n/AWS1/IMG` IMG transaction for AWS SDK per SAP ABAP e l'applicazione Custom Business Configuration for AWS SDK per SAP ABAP - edizione BTP per configurare le impostazioni globali. Questo argomento utilizza IMG e Custom Business Configuration in modo intercambiabile.

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Impostazioni tecniche](#)
- [Configura gli scenari](#)

Impostazioni tecniche

Le impostazioni globali della `/AWS1/IMG` transazione influiscono sul comportamento dell'intero SDK. Queste impostazioni sono generalmente configurate da un amministratore di Basis. È possibile impostare questi valori sulle seguenti impostazioni consigliate.

- Seleziona Nuove voci.
 - Regionalizzazione S3: accedi ai bucket [us-east-1 utilizzando s3.amazonaws.com](#).
 - Regionalizzazione STS: accedi a STS utilizzando un endpoint globale.
 - Disattiva i metadati EC2: lascia vuoto questo campo. Questo campo è di sola lettura nell'edizione BTP ed è impostato su «Sì» per impostazione predefinita.
 - Modalità metadati: utilizza l'endpoint di metadati IPv4. Questo campo è di sola lettura nell'edizione BTP e viene aggiornato automaticamente.
 - Metadata Endpt URL: lascia vuoto questo campo. Questo campo è di sola lettura nell'edizione BTP.
- Seleziona Salva.

Configura gli scenari

Gli scenari consentono a AWS SDK di cambiare le impostazioni in modo più efficiente durante uno scenario di test di emergenza o di disaster recovery in più aree geografiche. Potrebbe non essere necessaria questa funzionalità e sarà invece sufficiente configurare il seguente scenario PREDEFINITO.

- Seleziona Nuove voci.
 - ID dello scenario: DEFAULT
 - Descrizione dello scenario: scenario predefinito
- Seleziona Salva.

Se disponi di una configurazione di disaster recovery multiregionale o di altri casi unici che richiedono una rapida modifica delle impostazioni, puoi configurare più scenari.

- DEFAULT- Funzionamento standard.
- DR- Configurazione speciale se un disastro richiede lo spostamento dell'intero sistema in un'altra regione.
- DR_TEST- Configurazione speciale per simulare un disastro, ad esempio, in un clone temporaneo di produzione.

Configurazione dell'applicazione

La configurazione di SDK per SAP ABAP è simile alla configurazione di altre applicazioni basate su ABAP. È organizzato in diversi profili per raggruppare le impostazioni di vari scenari. Un profilo ABAP SDK definisce le impostazioni richieste per uno scenario applicativo specifico. Ad esempio, se le transazioni ZVA01 e ZVA03 sono transazioni relative alle fatture migliorate ed eseguite su Servizi AWS Amazon S3 e AWS Lambda Amazon SageMaker, è possibile creare un profilo SDK chiamato ZVA02 ZINV0ICE. Questo profilo può raggruppare le impostazioni tecniche, le autorizzazioni SAP e le mappature dei ruoli IAM per le funzionalità relative alla fattura.

Utilizza /n/AWS1/IMG transaction for AWS SDK per SAP ABAP e l'applicazione Custom Business Configuration per SDK per SAP ABAP - edizione BTP per AWS configurare le impostazioni globali. Questo argomento utilizza IMG e Custom Business Configuration in modo intercambiabile.

Argomenti

- [Profilo SDK](#)
- [Risolutore di risorse logiche](#)
- [Esempio](#)

Profilo SDK

Un profilo ABAP SDK definisce quanto segue per ogni SID e client.

Note

Il client è sempre al 100% in ambiente SAP BTP, ABAP.

- La AWS regione predefinita per tutte le chiamate API. Ad esempio, se i sistemi SAP sono in esecuzione nella us-east-1 regione, è probabile che anche le altre AWS risorse si trovino nella stessa regione e questa dovrebbe essere la regione predefinita. Il codice ABAP può sovrascrivere la regione predefinita.
- Metodo di autenticazione
 - Per i sistemi SAP in esecuzione su Amazon EC2, consigliamo vivamente di scegliere i metadati del ruolo dell'istanza per trarre vantaggio dalle credenziali a rotazione automatica di breve durata.
 - Per i sistemi SAP in esecuzione in locale o in un altro cloud, devi scegliere le credenziali dallo storage SSF.
 - Per i sistemi ABAP in esecuzione su SAP BTP, è necessario scegliere le credenziali da SAP Credential Store. Per ulteriori informazioni, vedere [Utilizzo di SAP Credential Store per l'autenticazione](#).
- Una mappatura dei ruoli IAM logici ai ruoli IAM.
 - Questa mappatura è ordinata in ordine di priorità decrescente.
 - Un ruolo IAM con la massima priorità per il quale un utente è autorizzato in un ruolo PFCG verrà selezionato automaticamente per l'utente.

Note

I ruoli PFCG sono chiamati ruoli aziendali in ambiente SAP BTP, ABAP.

Quando un programma ABAP desidera connettersi a un Servizio AWS, specifica un profilo ABAP SDK che richiama le impostazioni necessarie. AUTHORIZATION-CHECKVerrà eseguito un messaggio per confermare che l'utente dispone delle autorizzazioni per accedere al profilo SDK. L'amministratore della sicurezza SAP può definire un ruolo PFCG che consente l'accesso agli utenti appropriati.

Risolutore di risorse logiche

Il risolutore di risorse logiche consente di disporre di una posizione standard in cui archiviare i nomi delle risorse. Viene fornito con SDK per SAP ABAP. La sua azione è simile al modo in cui la FILE transazione associa i nomi di file logici ai nomi di file fisici.

Una risorsa logica definisce il concetto di AWS risorsa, ad esempio il bucket Amazon S3 che contiene le nostre fatture. Questa risorsa logica, ad esempio, può essere denominata ZINVOICES_OUTBOUND e può essere mappata a un nome di bucket fisico diverso, a seconda che il sistema SAP sia di sviluppo, controllo qualità o produzione.

L'SDK per SAP ABAP è configurato in modo tale che un sistema di controllo qualità risolva le risorse logiche in risorse fisiche di controllo qualità, anche dopo un aggiornamento del sistema dalla produzione. La mappatura delle risorse per TUTTI i sistemi viene definita nel sistema SAP di sviluppo e trasferita successivamente. Questo approccio è diverso dalla configurazione usuale nei sistemi SAP in cui la mappatura viene gestita come dati master e impostata in ciascun sistema. Il vantaggio del resolver di risorse logiche offerto da SDK per SAP ABAP è che le possibilità di un trasporto errato dopo gli aggiornamenti del sistema sono quasi nulle.

Esempio

Esistono quattro bucket Amazon S3 separati, uno per sviluppo, produzione e controllo qualità, oltre a un secondo bucket QA per i test di regressione.

Quando l'SDK risolve una risorsa logica come ZINVOICE_OUTBOUND una risorsa fisica, verifica SY-SYSID e chiede su quale SID e client SY-MANDT sto utilizzando? e seleziona automaticamente la risorsa fisica corretta.

Se è necessario modificare la mappatura di una risorsa in produzione, è necessario modificare la mappatura nel sistema IMG di sviluppo e portarla avanti. Ciò garantisce che la riassegnazione AWS delle risorse a un sistema SAP sia soggetta al controllo delle modifiche come per qualsiasi altro trasporto.

Note

Poiché la configurazione dell'SDK dipende dal client, la riassegnazione delle risorse viene trasferita in una richiesta di personalizzazione e il trasporto deve essere importato in ogni client.

Impostazioni di runtime

Questa sezione comprende i seguenti argomenti.

Note

Queste impostazioni non sono trasportabili e sono locali per ogni sistema SAP.

Argomenti

- [Registra e traccia](#)
- [OPT-IN: telemetria avanzata](#)
- [Scenario attivo](#)

Registra e traccia

È possibile attivare una traccia per scopi di debug. Si consiglia di mantenere il livello di traccia su No Trace, a meno che non si diagnostichi un problema tecnico. Per ulteriori informazioni, vedere [Funzionamento sicuro](#).

Queste impostazioni non sono applicabili all'SDK per SAP ABAP - edizione BTP.

OPT-IN: telemetria avanzata

Tutti gli SDK inviano informazioni di telemetria a scopo di supporto. AWS Puoi optare per la telemetria avanzata. Ciò è particolarmente utile quando si contatta AWS Support per identificare l'origine di una particolare chiamata API. Per ulteriori informazioni, consulta [Trace](#) and [Telemetry](#).

Queste impostazioni non sono applicabili all'edizione SDK per SAP ABAP - BTP.

Scenario attivo

Attiva il tuo DEFAULT scenario in questa transazione. Questa attivazione è richiesta una sola volta per ogni sistema e non deve essere modificata a meno che il sistema non sia sottoposto a un ripristino di emergenza in più regioni. In una configurazione multiregionale, è possibile utilizzare questa impostazione per passare il sistema SAP a un ambiente di disaster recovery o a scenari di test di disaster recovery.

Argomenti di aggiornamento, tracciamento e telemetria per SDK AWS per SAP ABAP

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Aggiornamento del sistema SAP](#)
- [Traccia](#)
- [Telemetria](#)

Aggiornamento del sistema SAP

Dopo un aggiornamento del sistema, la sfida principale per un amministratore Basis è assicurarsi che i diversi sistemi non accedano alle rispettive risorse. Ad esempio, potresti voler assicurarti che il tuo sistema QA SAP non acceda alle risorse, come un bucket S3, del tuo ambiente di produzione.

SDK per SAP ABAP offre un approccio attento alla sicurezza delle risorse logiche per affrontare questa sfida. Un analista aziendale può eseguire le seguenti operazioni.

1. Definire una risorsa logica, ad esempio ZINVOICE_OUTBOUND.
2. Mappa tutti i sistemi e i client nel sistema di sviluppo.
3. Trasporta la configurazione di TUTTI i sistemi fino al panorama di produzione.

Passaggi di base dopo un aggiornamento

1. Verifica l'autenticazione

- Se il sistema utilizza l'autenticazione con chiave di accesso segreta, le credenziali crittografate SSF non saranno valide perché sono archiviate nei dati master. [Le credenziali devono essere reinserite, il che potrebbe richiedere la rigenerazione di una nuova chiave di accesso segreta in https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)
- Se il sistema si sta autenticando con i metadati delle istanze EC2, non sono necessari passaggi.

Controlla le impostazioni di tracciamento

- Nel/AWS1/IMG, assicurati che le impostazioni di traccia siano quelle che desideri. Queste impostazioni non sono trasportabili.

Traccia

L'output di traccia è controllato nelle impostazioni di runtime di IMG.

I livelli di traccia che è possibile utilizzare sono:

- Nessuna traccia
- Chiamate API Trace
- Traccia le chiamate API e il payload

Questa opzione contiene informazioni non crittografate sul payload.

- Trace le chiamate API, il payload e la trasformazione XML interna

Questa opzione contiene informazioni non crittografate sul payload.

Se la traccia API è attivata, le tracce vengono scritte DIR_WORK nel `aws1_trace-YYYY-MM-DD.log` file.

Se la traccia del payload è inoltre attivata, `aws1_payload_*` vengono creati file aggiuntivi con lo stesso titolo per ogni chiamata e componente del payload. La lunghezza della traccia del payload può essere limitata con il limite di lunghezza che si applica a ogni singolo errore di traccia del payload.

Le tracce del payload hanno principalmente lo scopo di raccogliere informazioni da fornire AWS Support in caso di errore di serializzazione. Ti consigliamo di scegliere No Trace a meno che tu non stia tentando di diagnosticare un errore SDK.

Note

Le tracce del payload possono contenere informazioni aziendali non crittografate. Ti consigliamo di attivare queste tracce solo per una richiesta da parte di AWS Support per aiutarti a risolvere i problemi. Puoi disattivare queste tracce dopo la risoluzione. Le tracce non vengono eliminate automaticamente e devono essere rimosse dall'amministratore di sistema quando non sono più necessarie.

Queste impostazioni non sono applicabili all'edizione SDK per SAP ABAP - BTP.

Telemetria

Gli SDK inviano informazioni di telemetria a. AWS Support SDK per SAP ABAP raccoglie le seguenti informazioni:

- Versione del sistema operativo e livello di patch
- SAP_BASISversione e livello di patch
- Versione e livello di patch del kernel SAP

Puoi scegliere di inviare le seguenti informazioni a. AWS Support

- SID SAP e nome dell'istanza () `host_sid_nn`
- Client SAP () `SY-MANDT`
- Codice della transazione (`SY-TCODE`) e rapporto () `SY-REPID`

Le informazioni aggiuntive AWS Support consentono di aiutarti meglio. AWS Support è in grado di rilevare il motivo per cui è stata effettuata una determinata chiamata API e di trovare ulteriormente la transazione pertinente in un sistema SAP.

La telemetria è limitata alle versioni SDK e API per SDK per SAP ABAP - edizione BTP.

Scenari di connettività avanzati

SDK AWS per SAP ABAPsi consuma Servizi AWS effettuando chiamate HTTPS verso gli AWS endpoint. In generale, gli AWS endpoint sono accessibili tramite Internet. Un sistema SAP deve

essere in grado di accedere a Internet per stabilire queste connessioni in uscita. SDK per SAP ABAP non richiede mai una connessione in entrata da Internet al sistema SAP.

I seguenti scenari offrono diversi modi per stabilire la connessione in uscita.

Scenari

- [Connessione tramite un server proxy](#)
- [Connessione tramite un firewall che ispeziona i pacchetti](#)
- [Endpoint gateway](#)
- [Endpoint con interfaccia personalizzata](#)
- [Accesso agli endpoint in più regioni](#)

Connessione tramite un server proxy

Per stabilire una connessione tramite un server proxy, utilizzare i seguenti passaggi.

1. Nell'SDK, vai a Transazione **SICF**.
2. Scegliere Execute (Esegui).
3. Nel menu, scegli Client > Server proxy.
4. Imposta l'impostazione del proxy come attiva.
5. Nel campo Nessun proxy per i seguenti indirizzi, elenca le eccezioni separate da punto e virgola.
6. Nei campi Protocollo HTTP e Protocollo HTTPS, specifica i dettagli di connessione per il tuo server proxy.

L'SDK non è a conoscenza del server proxy e non richiede alcuna impostazione per utilizzare la configurazione del server proxy del sistema SAP.

Note

Se utilizzi l'[autenticazione dei metadati delle istanze Amazon EC2](#), il sistema SAP non può utilizzare il server proxy per accedere ai metadati dell'istanza locale su. `http://169.254.169.254` È necessario includere `169.254.169.254` nel campo Nessun proxy per i seguenti indirizzi.

Connessione tramite un firewall che ispeziona i pacchetti

È possibile configurare un firewall che ispeziona i pacchetti per la connessione in uscita. Questi firewall decrittografano il traffico SSL e quindi lo crittografano nuovamente prima di trasmetterlo all'endpoint. Questa configurazione richiede in genere che il firewall rilasci i propri certificati al sistema SAP che utilizza un. Servizio AWS È necessario installare il certificato CA del firewall inSTRUST. Per ulteriori informazioni, consulta [Connettività HTTPS](#).

Endpoint gateway

Alcuni Servizi AWS offrono endpoint gateway per fornire a un VPC un accesso ad alte prestazioni senza Internet. Questi endpoint sono trasparenti per SDK for SAP ABAP e non richiedono alcuna configurazione.

[Per ulteriori informazioni, consulta Gateway Endpoints.](#)

Endpoint con interfaccia personalizzata

Se devi sostituire la risoluzione predefinita dell'endpoint con un endpoint personalizzato, puoi utilizzare un endpoint di interfaccia per fornire al tuo VPC un accesso ad alte prestazioni senza Internet. [Per ulteriori informazioni, consulta Configurare un endpoint di interfaccia.](#)

Quando non si utilizza un DNS privato, questi endpoint hanno i propri indirizzi DNS e un programma ABAP deve sovrascrivere esplicitamente la normale logica di risoluzione degli endpoint. Per ulteriori informazioni, consulta AWS re:Post — [Perché non posso risolvere i nomi di dominio di servizio per un endpoint VPC di interfaccia?](#)

Nell'esempio seguente, viene creato un endpoint di interfaccia per e. AWS STS Amazon Translate Il sistema SAP non utilizza DNS privato e chiama i servizi con un endpoint personalizzato. Le risorse logiche definite in /AWS1/IMG rappresentano gli indirizzi degli endpoint dell'interfaccia fisica, ad esempio. `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com` In questo modo si evita la codifica rigida del DNS nel codice.

Nel codice seguente, le risorse logiche incluse /AWS1/IMG vengono prima risolte in nomi di endpoint fisici. Vengono quindi fornite ai metodi di fabbrica della classe di AWS sessione (che utilizza AWS STS per assumere un ruolo IAM) e alla classe Translate 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-abcdefgh.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-0123456789abcdefgh-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...

```

Come mostrato nell'esempio, qualsiasi chiamata al metodo `go_xl8` va all'endpoint `https://vpce-0123456789abcdefgh-12345567.translate.us-west-2.vpce.amazonaws.com`.

Accesso agli endpoint in più regioni

AWS l'endpoint viene determinato automaticamente in base all'impostazione predefinita Regione AWS definita nel profilo SDK. Puoi anche specificare una regione a livello di codice, sostituendo la regione predefinita. Questo può essere sovrascritto nel `CREATE()` metodo `factory` o successivamente con l'oggetto di configurazione dell'SDK. [Per ulteriori informazioni, consulta Configurazione programmatica.](#)

Nell'esempio seguente, il `CREATE()` metodo `factory` viene utilizzato per impostare la regione ed elencare le code Amazon SQS in entrambe le `us-east-1` regioni. `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( ).
  ENDLLOOP.
ENDLOOP.
```

Usando SDK AWS per SAP ABAP

SDK per SAP ABAP ha due componenti principali.

- SDK Runtime (pacchetto/AWS1/RT): un insieme di oggetti che sono alla base della sicurezza, dell'autenticazione, del tracciamento, della configurazione, della conversione dei dati e di altre funzioni interAPI. I moduli API per Amazon S3 AWS STS, IAM Roles Anywhere e Secrets Manager sono obbligatori.
- API (pacchetto /AWS1/API e relativi pacchetti secondari): un sottopacchetto per ogni API in cui gli oggetti di ciascuna API sono completamente indipendenti l'uno dall'altro, garantendo che una modifica in un'API non comprometta un'altra API. Per un elenco completo delle SDK AWS per SAP ABAP API, consulta [SDK AWS per SAP ABAP](#) - Guida di riferimento delle API.

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Rappresentazione dei dati in ABAP](#)
- [Programma di esempio Amazon S3](#)
- [Concetti SDK per SAP ABAP](#)
- [SDK AWS per SAP ABAP features](#)
- [Creazione di prodotti con SDK](#)
- [Limitazioni](#)

Rappresentazione dei dati in ABAP

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Tipi di dati](#)
- [Tipi di dati AWS](#)

Tipi di dati

Servizi AWS dispongono di un set standard di tipi di dati che devono essere mappati ai tipi di dati ABAP. Per ulteriori dettagli, consulta la tabella seguente.

Tipo di dati AWS	Tipo di dati ABAP	Commenti
booleano	C	Personaggio singolo "X" e " "
Stringa	STRING	
Byte	INT2	INT2 ha un intervallo più ampio di 0-255. La maggior parte Servizi AWS troncherà gli overflow ma questo comportamento non è definito formalmente.
Breve	INT2	
Numero intero	INT4	
Lungo	19 DICEMBRE	INT8 non è disponibile fino all'ABAP 750. DEC19 viene utilizzato per la compatibilità e la coerenza su tutte le piattaforme ABAP supportate.
Blob	CORDA	Rappresenta dati binari
Float	STRING	Sebbene ABAP supporti DecFloat, non può rappresentare valori come NaN, Infinity e -Infinity. AWS SDK li rappresenta internamente come STRINGS e li converte in DECFLOAT16 in fase di esecuzione. Se sono rappresentati NaN, Infinity o
Doppio	STRING	

Tipo di dati AWS	Tipo di dati ABAP	Commenti
		+Infinity, lo sviluppatore può elaborarli in risposta a una serie speciale di eccezioni o mappature.
BigInteger	STRING	Questi valori rappresentano numeri di lunghezza infinita che non possono essere rappresentati in ABAP e vengono utilizzate stringhe al posto di BigInteger.
BigDecimal	STRING	
Time stamp	TZNTSTMP5	TZNTSTMP5 consente l'elaborazione con funzioni di timestamp ABAP native.

Servizi AWS restituiscono anche i seguenti tipi di dati aggregati.

Tipo di dati AWS	Tipo di dati ABAP	Commenti
Struttura	Classe	
Union	Classe	Un'unione è la stessa cosa di una struttura, tranne per il fatto che un'unione non avrà mai più di un set di campi. Tutti gli altri campi verranno impostati su Nessun valore.
Array	TAVOLO STANDARD	
Hash	TABELLA CON HASH	La tabella con hash avrà solo due colonne: una CHIAVE (stringa) e un VALORE (classe).

Tipi di dati AWS

I seguenti approcci sono stati integrati per supportare Servizi AWS in ABAP.

- Certi tipi di dati AWS non possono essere rappresentati in ABAP. Ad esempio, il tipo di dati `float` in ABAP non supporta i valori `NaN`, `Infinity`, oppure `-Infinity`. Pertanto, il tipo di dati `float` è rappresentato come `STRING` ed è tradotto in `DECFLOAT16` in fase di esecuzione.
- I dati AWS sono rappresentati sul cavo come JSON o XML e i valori sono opzionali. Ad esempio, vedere i seguenti esempi restituiti da un Servizio AWS in formato JSON.

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

Se Ana non ha un secondo nome, il servizio restituisce il seguente output.

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

ABAP non fa distinzione tra una stringa di lunghezza 0 e una stringa che non ha valore. Altri linguaggi potrebbero assegnare un valore `NULL` alla stringa o racchiudere la stringa in un costrutto (come `JavaOptional` involucro). Questi non sono supportati in ABAP. Pertanto, SDK per SAP ABAP facilita la distinzione dei valori fornendo varianti di `get` metodo.

Programma di esempio Amazon S3

Questa sezione illustra un semplice programma di esempio per elencare il contenuto di un bucket Amazon S3 chiamando `ListObjectsV2`.

Argomenti

- [Prerequisiti](#)
- [Codice](#)

- [Sezioni di codice](#)

Prerequisiti

È necessario soddisfare i seguenti prerequisiti per eseguire questo programma di esempio.

- Hai un bucket Amazon S3. In questo tutorial, il bucket è denominato `demo-invoices.customer.com`.
- Transazione `/AWS1/IMG`:
 - Ha un profilo SDK definito denominato `DEMO_S3`.
 - Nel profilo SDK, il ruolo logico `IAMTESTUSER` deve essere mappato a un ruolo IAM, ad esempio `arn:aws:iam::111122223333:role/SapDemoFinancequali sovvenzionis3>ListBucket` autorizzazione a elencare il contenuto del tuo bucket Amazon S3.
 - Ha una risorsa logica denominata `DEMO_BUCKET` che è mappato sul tuo bucket Amazon S3 con il SID e il client del tuo sistema SAP.
- L'utente ha un ruolo PFCG che:
 - Autorizza l'utente ad accedere `DEMO_S3` Profilo SDK tramite oggetto di autenticazione `-/AWS1/SESS`.
 - Autorizza l'utente per il ruolo logico `IAMTESTUSER` accesso tramite oggetto di autenticazione `-/AWS1/LROL`.
- Il tuo sistema SAP può autenticarsi su AWS utilizzando il metodo definito nel profilo SDK.
- Il tuo profilo di istanza Amazon EC2 concede al tuo sistema SAP i diritti `dist:s:assumeRole` nel ruolo IAM `arn:aws:iam::111122223333:role/SapDemoFinance` mappato nel profilo SDK.

Codice

Il seguente blocco di codice mostra come apparirebbe il tuo codice.

```
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.

```

Sezioni di codice

Di seguito è riportata una revisione del codice nelle sezioni.

```

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

```

L'utente non può specificare un nome di bucket fisico. Specificano un bucket logico e gli amministratori di sistema (in particolare l'analista aziendale) in coordinamento con AWS l'amministratore mappa i bucket logici sui bucket fisici in /AWS1/IMG. Nella maggior parte degli scenari aziendali, l'utente non ha la possibilità di scegliere il bucket logico: l'ID della risorsa logica è codificato nel codice o configurato in una tabella di configurazione personalizzata.

```

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

```

Questa riga stabilisce una sessione di sicurezza e dichiara che questo programma ABAP prevede di utilizzare il DEMO_S3 Profilo SDK. Questa chiamata è la connessione alla configurazione SDK e inserisce la regione predefinita, le impostazioni di autenticazione e il ruolo IAM desiderato. Una chiamata a AUTHORIZATION-CHECK viene creato automaticamente per garantire che l'oggetto di autorizzazione/AWS1/SESS è soddisfatto. Inoltre, AUTHORIZATION-CHECK verranno effettuate chiamate per determinare il ruolo IAM logico più potente (numero di sequenza inferiore) per cui l'utente è autorizzato, in base all'oggetto di autorizzazione/AWS1/ROLE. L'SDK presuppone che il ruolo IAM sia mappato al ruolo IAM logico per il SID e il client. Quindi, l'oggetto della sessione attiva il tracciamento in base alle impostazioni di traccia nel IMG.

Se l'utente non è autorizzato per il profilo SDK richiesto o per qualsiasi ruolo IAM logico disponibile, verrà sollevata un'eccezione.

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

Questa riga risolve la risorsa logica in un nome di bucket fisico. Se la risorsa logica non può essere risolta perché la configurazione non ha una mappatura per questa combinazione SID/client, verrà sollevata un'eccezione.

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

Questa riga crea un oggetto API per Amazon S3 utilizzando create() metodo di /aws1/cl_s3_factory. L'oggetto restituito è di tipo /aws1/if_s3 che è l'interfaccia per un'API Amazon S3. È necessario creare un oggetto API separato per ogni servizio. Ad esempio, se un programma ABAP utilizza Amazon S3, AWS Lambda e DynamoDB, quindi crea oggetti API da /aws1/cl_s3_factory, /aws1/cl_lmd_factory, e /aws1/cl_dyn_factory.

Esistono alcuni parametri opzionali per il costruttore che consentono di specificare la regione se si desidera sovrascrivere la regione predefinita configurata nel IMG. In questo modo, possono esserci due casi di /aws1/if_s3, uno per us-east-1 e uno per us-west-2, se si desidera copiare oggetti da un bucket in una regione a un bucket in un'altra regione. Allo stesso modo, è possibile creare due diversi oggetti della sessione di sicurezza e utilizzarli per creare due istanze separate di /aws1/cl_s3, se hai bisogno di un report da leggere da un bucket relativo alla finanza e scrivere oggetti in un bucket relativo alla logistica.

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

Questa linea è una chiamata a `ListObjectsV2`. Richiede semplici argomenti di input e restituisce un singolo oggetto. Questi oggetti possono rappresentare dati JSON e XML profondi, deserializzati in un costrutto ABAP orientato agli oggetti. In alcuni casi può essere piuttosto complicato. Ora, è sufficiente elaborare l'output per elencare il contenuto 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.
```

L'accesso ai dati avviene tramite un `GET...()` metodo di stile che nasconde la rappresentazione interna dei dati. `GET_CONTENTS()` restituisce una tabella ABAP e ogni riga contiene un oggetto che rappresenta una singola voce di Amazon S3. Nella maggior parte dei casi, AWSSDK adotta questo approccio orientato agli oggetti e tutti i dati sono rappresentati come oggetti e tabelle. `LastModified` il campo è rappresentato come un timestamp che può essere convertito in una data con ABAP-native `CONVERT TIME STAMP` comando. `GET_SIZE()` restituisce un `INT4` per semplici operazioni matematiche e di formattazione.

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

Tutti gli errori: connessione, client 4xx, server 5xx o qualsiasi errore ABAP, come errori di autorizzazione o configurazione, sono rappresentati come eccezioni. Puoi affrontare ogni eccezione separatamente. È possibile scegliere se gestire un'eccezione come un errore informativo, un nuovo tentativo, un avviso, un breve dump o qualsiasi altro tipo di gestione.

Concetti SDK per SAP ABAP

Questa sezione tratta i concetti di base di. SDK AWS per SAP ABAP

Argomenti

- [Classi API](#)
- [Oggetti aggiuntivi](#)
- [Classi di struttura](#)
- [Matrici](#)
- [Mappe](#)
- [Funzioni di livello superiore](#)

Classi API

A ciascuna Servizio AWS viene assegnato un acronimo di tre lettere o. TLA Il servizio è rappresentato da un'interfaccia nel /AWS1/IF_<TLA> formato. La chiameremo interfaccia di servizio. La classe API è inclusa nel /AWS1/API_<TLA> pacchetto. L'interfaccia di servizio è costituita da un metodo per ogni AWS operazione (chiameremo questi metodi Operation Methods). Per visualizzare un elenco completo dei moduli SDK AWS per SAP ABAP TLA, vedere [SDK AWS per SAP ABAP - Elenco dei moduli](#).

Ogni metodo operativo ha alcuni IMPORTING argomenti e al massimo un RETURNING argomento. Spesso, questi argomenti saranno oggetti con costruttori complicati e una lunga serie di GET...() metodi. In molti casi, gli oggetti conterranno oggetti annidati, riferimenti ricorsivi, tabelle di oggetti, tabelle di tabelle e così via. Questo perché Servizi AWS stanno passando strutture XML e JSON profonde, che non possono essere rappresentate da un insieme semplice di argomenti.

L'RETURNINGargomento è sempre una classe, anche se la classe contiene un solo attributo.

Oggetti aggiuntivi

Oltre a contenere la classe API principale, ogni pacchetto API contiene vari repository e oggetti di dizionario di dati correlati.

- Una classe per ogni oggetto di tipo di struttura.
- Una classe per qualsiasi tipo di dati primitivo che appare in una tabella. Ad esempio, se un servizio restituisce una tabella di stringhe, l'API ABAP la rappresenterà come una tabella di oggetti,

dove ogni oggetto è una classe wrapper che incapsula una stringa. In questo modo la classe wrapper può nascondere i dettagli della rappresentazione di una stringa nulla che non può essere rappresentata nativamente in ABAP.

- Una classe di eccezione per eventuali errori specifici definiti dal servizio.
- Elementi di dati per ogni tipo di dati primitivo. Ogni tipo di dati ha il proprio elemento di dati per poter essere autodocumentato.
- Oggetti aggiuntivi per l'elaborazione interna, come le trasformazioni XSLT per la serializzazione e la deserializzazione dei payload XML e JSON.

Classi di struttura

La maggior parte AWS dei dati, inviati e ricevuti dal servizio, è rappresentata dall' AWS SDK come classi. Queste classi rappresentano strutture di dati e nascondono i dettagli interni dello storage. In particolare, le classi nascondono il modo in cui l'SDK rappresenta questo campo senza valore.

Per ogni campo di una classe di struttura, esistono tre metodi.

GET_field()

Il `GET_field()` metodo

- Restituisce il valore del campo, oppure
- Se il campo non ha alcun valore, restituisce un valore predefinito, che è possibile impostare come parametro opzionale.

Ad esempio, considerate il codice seguente che stampa il vincolo di posizione di un bucket.

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

Se il bucket non ha alcun vincolo di posizione (come nel caso `dius-east-1`), restituirà la stringa `GET_LOCATIONCONSTRAINT()` vuota. È possibile ignorare questo comportamento e specificare il valore desiderato se il campo non ha alcun valore.

```
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' ).
```

Ora il programma scriverà `Bucket Location: assuming us-east-1` se `getbucketlocation()` il risultato non restituisce una posizione.

È possibile chiedere al metodo `GET()` di restituire un risultato specifico se il valore richiesto è completamente mancante, vedere il seguente esempio di codice.

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

In questo caso, se non c'è alcun vincolo di posizione, `GET_LOCATIONCONSTRAINT()` restituirà `NopeNopeNope`.

HAS_field()

`HAS_field()` metodo è un modo per scoprire se il campo ha un valore o meno. Guarda l'esempio seguente.

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

Se si sa che un determinato campo ha sempre un valore, non ci sarà alcun `HAS_field()` metodo.

ASK_field()

Il `ASK_field()` metodo restituisce il valore del campo o solleva un'eccezione se non ha alcun valore. Questo è un modo pratico per elaborare una serie di campi e uscire dalla logica e adottare un approccio diverso se uno qualsiasi dei campi non ha alcun valore.

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

```
ENDTRY.
```

Nota che `/AWS1/CX_RT_VALUE_MISSING` si tratta di un'eccezione statica e riceverai un avviso se scegli di non catturarla.

Best practice

In generale, è possibile utilizzare il `GET_field()` metodo in quanto tratta una stringa nulla come una stringa vuota ed è la più simile ad ABAP delle tre opzioni. Tuttavia, non consente di distinguere facilmente tra situazioni in cui il campo ha un valore vuoto e in cui il campo non ha alcun valore. Se la logica aziendale dipende dalla distinzione tra dati mancanti e dati vuoti, i ASK metodi HAS or consentono di gestire questi casi.

Matrici

Gli array sono rappresentati come tabelle di oggetti standard ABAP.

Un array JSON può contenere valori nulli, come il seguente array: `['cat', 'dog', null, 'horse']` Questo è indicato come array sparso. È rappresentato in ABAP come tabella interna di riferimenti a oggetti e il `null` valore è rappresentato nella tabella come un vero valore `null` ABAP. Quando si esegue un'iterazione su una tabella sparsa, è necessario verificare `null` i valori per evitare di accedere a un `null` oggetto e ottenere un'eccezione. `CX_SY_REF_IS_INITIAL` In pratica, gli array sparsi sono rari nei servizi. AWS

Per inizializzare un array di oggetti, è conveniente utilizzare i nuovi costrutti ABAP 7.40. Prendi in considerazione il lancio di un'istanza Amazon EC2 con diversi gruppi di sicurezza assegnati:

```
ao_ec2->runinstances(  
  iv_imageid           = lo_latest_ami->get_imageid( )  
  iv_instancetype     = 't2.micro'  
  iv_maxcount         = 1  
  iv_mincount        = 1  
  it_securitygroupids = VALUE /aws1/  
cl_ec2secgrpiddstrlist_w=>tt_securitygroupidstringlist(  
  ( NEW /aws1/  
cl_ec2secgrpiddstrlist_w( 'sg-12345678' ) )  
  ( NEW /aws1/  
cl_ec2secgrpiddstrlist_w( 'sg-55555555' ) )  
  ( NEW /aws1/  
cl_ec2secgrpiddstrlist_w( 'sg-99999999' ) )  
)
```

```
iv_subnetid          = ao_snet->get_subnetid( )  
it_tagSpecifications = make_tag_spec( 'instance' )  
)
```

Mappe

Le mappe JSON sono rappresentate in ABAP come `Hashed Tables` se ogni riga della tabella avesse solo due componenti.

- **KEY**— una stringa che è la parte **UNIQUE KEY** della tabella.
- **VALUE**— un oggetto contenente il valore.

Una mappa è uno dei pochissimi casi in cui AWS SDK utilizza una vera struttura anziché una classe. Ciò è necessario perché le tabelle con hash ABAP non possono avere un riferimento a un oggetto come campo chiave e le chiavi della AWS mappa sono sempre stringhe non nulle.

Funzioni di livello superiore

Le API [Classi API](#) descritte nella sezione precedente rispecchiano con precisione le API del AWS servizio e le rappresentano come classi ABAP familiari. In alcuni casi, l'SDK include anche funzioni di livello superiore che si basano sulle classi API per semplificare determinate operazioni. Le funzioni di livello superiore sono incluse per comodità del programmatore e non sostituiscono le classi API di livello inferiore.

Se l'SDK include funzioni di livello superiore per un modulo, sono incluse nello stesso trasporto e sono accessibili tramite una classe factory chiamata `/AWS1/CL_TLA_L2_FACTORY`. La classe factory include metodi per creare vari client di livello superiore per il modulo, documentati insieme al resto dell'API con la documentazione dell'[API](#).

SDK AWS per SAP ABAP features

SDK AWS per SAP ABAP offre le seguenti funzionalità.

Argomenti

- [Configurazione programmatica](#)
- [Waiter](#)
- [Impaginatori](#)
- [Comportamento di ripetizione](#)

Configurazione programmatica

Utilizza `/n/AWS1/IMG IMG transaction for AWS SDK per SAP ABAP` e l'applicazione Custom Business Configuration for AWS SDK per SAP ABAP - edizione BTP per la configurazione programmatica.

Per iniziare la configurazione programmatica, iniziate recuperando un oggetto di configurazione con il comando `get_config()`

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

Ogni oggetto di configurazione implementa `/AWS1/IF_RT_CONFIG` un'interfaccia che include GET termini e SET termini corrispondenti a. IMG Ad esempio, la regione predefinita può essere sostituita. Vedi il seguente comando di esempio.

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

Alcuni oggetti di configurazione non hanno alcuna IMG rappresentazione e possono essere impostati solo a livello di codice, ad esempio il numero massimo di tentativi. Vedi il seguente comando di esempio.

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

L'oggetto di configurazione di Servizi AWS può includere anche metodi specifici del servizio che non sono rappresentati in `/aws1/if_rt_config`. Ad esempio, Amazon S3 può indirizzare un bucket denominato foobucket utilizzando un endpoint `foobucket.s3.region.amazonaws.com` virtuale o uno stile di percorso `s3.region.amazonaws.com/foobucket`. Puoi imporre l'uso dello stile del percorso con il seguente comando di esempio.

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

Per ulteriori informazioni sulle configurazioni dei servizi, vedere [SDK AWS per SAP ABAP — Guida di riferimento delle API](#).

Waiter

Quando si lavora con AWS API asincrone, è necessario attendere che una determinata risorsa diventi disponibile prima di intraprendere ulteriori azioni. Ad esempio, l'`CREATETABLE()` API di

Amazon DynamoDB risponde immediatamente con lo stato della tabella. CREATING È possibile avviare operazioni di lettura o scrittura solo dopo che lo stato della tabella è cambiato in. ACTIVE I camerieri ti danno la possibilità di confermare che AWS le risorse si trovano in uno stato particolare prima di eseguire azioni su di esse.

I camerieri utilizzano le operazioni di servizio per verificare lo stato delle AWS risorse fino a quando la risorsa non raggiunge lo stato previsto o finché non viene stabilito che la risorsa non raggiunge lo stato desiderato. Scrivere il codice per il sondaggio continuo delle risorse può richiedere molto tempo ed è soggetto a errori. AWS I camerieri aiutano a semplificare questa complessità assumendosi la responsabilità di eseguire i sondaggi per vostro conto.

Guarda il seguente esempio di Amazon S3 con un cameriere.

```
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|
).
```

- In questo esempio, il client Amazon S3 viene utilizzato per creare un bucket. Il `get_waiter()` comando viene implementato per specificare quando. `bucketexists`
- È necessario specificare il `iv_max_wait_time` parametro per ogni cameriere. Rappresenta la quantità totale di tempo che un cameriere deve attendere prima del completamento. Nell'esempio precedente, un cameriere può correre per 200 secondi.
- Potrebbe essere necessario fornire input aggiuntivi per i parametri richiesti. Nell'esempio precedente, il nome del bucket Amazon S3 è obbligatorio per il parametro. `iv_bucket`
- `/AWS1/CX_RT_WAITER_FAILURE` l'eccezione indica che il cameriere ha superato il tempo massimo specificato nel parametro. `iv_max_wait_time`
- `/AWS1/CX_RT_WAITER_TIMEOUT` l'eccezione indica che il cameriere si è fermato perché non ha raggiunto lo stato desiderato.

Impaginatori

Alcune Servizio AWS operazioni offrono risposte paginate. Sono suddivise in pagine per restituire una quantità fissa di dati con ogni risposta. È necessario effettuare richieste successive con un token o un marker per recuperare l'intero set di risultati. Ad esempio, l'operazione `ListObjectsV2` Amazon S3 restituisce fino a 1.000 oggetti alla volta. È necessario effettuare richieste successive con il token appropriato per ottenere la pagina successiva di risultati.

L'impaginazione è il processo di invio di richieste successive che riprendono dal punto in cui era stata interrotta una richiesta precedente. I paginatori sono iteratori dei risultati forniti da SDK per SAP ABAP. È possibile utilizzare le API impaginate con facilità e senza comprendere il meccanismo sottostante dell'API utilizzando i token di impaginazione.

Lavorare con gli impaginatori

È possibile creare impaginatori con il `get_paginator()` metodo che restituisce un oggetto paginatore. L'oggetto paginatore richiama l'operazione da impaginare. L'oggetto paginatore accetta i parametri richiesti da fornire all'API sottostante. Questo processo restituisce un oggetto iteratore che può essere utilizzato per iterare su risultati impaginati, utilizzando i metodi `and`. `has_next()` `get_next()`

- `has_next()` restituisce un valore booleano che indica se ci sono più risposte o pagine disponibili per l'operazione chiamata.
- `get_next()` restituisce la risposta dell'operazione.

L'esempio seguente elenca tutti gli oggetti in un bucket S3 recuperati utilizzando paginatore.

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

TRY.
  DATA(lo_paginator) = lo_s3->get_paginator( ).
  DATA(lo_iterator) = lo_paginator->listobjectsv2(
    iv_bucket = 'example_bucket'
  ).
  WHILE lo_iterator->has_next( ).
    DATA(lo_output) = lo_iterator->get_next( ).
    LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).
      WRITE: / lo_object->get_key( ), lo_object->get_size( ).
    ENDLOOP.
```

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

Comportamento di ripetizione

SDK per SAP ABAP consente di configurare il numero massimo di tentativi per le richieste che hanno esito negativo a causa di problemi di limitazione o errori transitori. Servizi AWS Il numero di tentativi consentiti a livello di client di servizio, ovvero il numero di volte in cui l'SDK ritenta l'operazione prima di fallire e generare un'eccezione, è specificato dall'attributo nell'oggetto di configurazione del servizio. AV_MAX_ATTEMPTS Quando viene creato un oggetto client di servizio, l'SDK configura l'AV_MAX_ATTEMPTS attributo su un valore predefinito di 3. L'oggetto di configurazione del servizio può essere utilizzato per impostare a livello di programmazione il numero massimo di tentativi sul valore desiderato. Vedi l'esempio seguente per maggiori dettagli.

```
“ 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

Sebbene l'oggetto di configurazione ABAP SDK consenta di impostare la modalità di riprova con il /AWS1/IF_RT_CONFIG~SET_RETRY_MODE() metodo, l'SDK supporta solo la modalità di riprova. standard Per ulteriori informazioni, consulta la Guida di riferimento per [riprovare gli SDK e gli strumenti AWS](#) .

Creazione di prodotti con SDK

Un prodotto o un componente aggiuntivo ABAP che utilizza Servizi AWS può migliorare ed estendere le funzionalità dell'SDK. È possibile creare tali prodotti da utilizzare con l'SDK.

Argomenti

- [Impostazione di un ID prodotto](#)

Impostazione di un ID prodotto

Si consiglia di impostare un ID prodotto quando si stabilisce una sessione all'interno di un prodotto o di un componente aggiuntivo. Vedi l'esempio seguente per maggiori dettagli.

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

L'ID del prodotto deve contenere solo lettere, numeri e caratteri di sottolineatura senza spazi o caratteri speciali. Puoi abbinarlo al nome tecnico del prodotto o a qualsiasi altro identificatore. Se sviluppi più prodotti o componenti aggiuntivi, l'ID del prodotto deve essere unico per ogni prodotto. Ad esempio, gli ID di prodotto per i prodotti Invoice Analyzer, Tax Calculator e Pricing Engine possono essere INVOICE_ANALYZER, TAX_CALCULATOR e PRICING_ENGINE.

L'aggiunta di un ID di prodotto alla sessione migliora la telemetria inviata con ogni chiamata di servizio. L'ID del prodotto e lo spazio dei nomi dell'oggetto che effettua la chiamata sono inclusi nella telemetria. Con questa telemetria, AWS Support puoi identificare il prodotto che sta effettuando la chiamata nel caso in cui il cliente riscontri problemi con l'SDK. Può aiutare a chiarire che la chiamata viene effettivamente effettuata dal prodotto e non dal codice del cliente.

Limitazioni

SDK AWS per SAP ABAP include moduli SDK per tutti Servizi AWS. Alcuni di questi moduli possono presentare delle limitazioni, come descritto qui.

- I moduli che si basano su collegamenti di MQTT protocollo, ad esempio `iotevents`, non funzioneranno. MQTT non è un protocollo basato su HTTP e attualmente non è supportato da SDK AWS per SAP ABAP.
- I moduli che si basano sulle funzionalità di streaming HTTP/2 non sono ancora supportati. Alcune operazioni dei servizi che funzionano con i flussi di eventi non sono ancora supportate e le operazioni di streaming multimediale dei servizi, come Amazon Kinesis Video Streams, non funzioneranno.

SDK AWS per SAP ABAP presenta le seguenti limitazioni di funzionalità.

- Le seguenti funzionalità di Amazon S3 non sono ancora supportate.
 - Punti di accesso multiregionali
 - Crittografia lato client Amazon S3

AWS L'SDK per SAP ABAP - BTP edition presenta le seguenti limitazioni durante l'anteprima per sviluppatori.

- Alcuni moduli potrebbero non essere disponibili.
- Non può essere disinstallato.
- Viene aggiornato meno frequentemente.

Esempi di codice SDK per SAP ABAP

Gli esempi di codice in questo argomento mostrano come utilizzare l' AWS SDK per SAP ABAP con. AWS

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Esempi

- [Azioni e scenari che utilizzano SDK per SAP ABAP](#)

Azioni e scenari che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con. Servizi AWS

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Servizi

- [Esempi di Amazon Bedrock Runtime con SDK per SAP ABAP](#)
- [CloudWatch esempi che utilizzano SDK per SAP ABAP](#)
- [Esempi di DynamoDB che utilizzano SDK per SAP ABAP](#)
- [Esempi di Amazon EC2 che utilizzano SDK per SAP ABAP](#)
- [Esempi di Kinesis che utilizzano SDK per SAP ABAP](#)

- [Esempi di Lambda con SDK per SAP ABAP](#)
- [Esempi di Amazon S3 che utilizzano SDK per SAP ABAP](#)
- [SageMaker esempi che utilizzano SDK per SAP ABAP](#)
- [Esempi di Amazon SNS che utilizzano SDK per SAP ABAP](#)
- [Esempi di Amazon SQS che utilizzano SDK per SAP ABAP](#)
- [Esempi di Amazon Textract con SDK per SAP ABAP](#)
- [Esempi di Amazon Translate con SDK per SAP ABAP](#)

Esempi di Amazon Bedrock Runtime con SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon Bedrock Runtime.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Anthropic Claude](#)
- [Diffusione stabile](#)

Anthropic Claude

InvokeModel

Il seguente esempio di codice mostra come inviare un messaggio di testo a Anthropic Claude, utilizzando l'API Invoke Model.

SDK per SAP ABAP

 Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Invoca il modello di base Anthropic Claude 2 per generare testo. Questo esempio utilizza funzionalità di /US2/CL_JSON che potrebbero non essere disponibili in alcune versioni.

NetWeaver

```

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

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

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

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

```

```

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

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

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

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

ENDTRY.

```

Invoca il modello di base Anthropic Claude 2 per generare testo utilizzando il client di alto livello 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.

```

- Per i dettagli sulle API, consulta la guida di riferimento [InvokeModel](#) all'API AWS SDK for SAP ABAP.

Diffusione stabile

InvokeModel

Il seguente esempio di codice mostra come richiamare Stability.ai Stable Diffusion XL su Amazon Bedrock per generare un'immagine.

SDK per SAP ABAP

Note

C'è di più su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea un'immagine 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.

```

Invoca il modello di base Stability.ai Stable Diffusion XL per generare immagini utilizzando il client di alto livello 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.

```

- Per i dettagli sulle API, consulta la guida di riferimento [InvokeModel](#) all'API AWS SDK for SAP ABAP.

CloudWatch esempi che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con. CloudWatch

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

DeleteAlarms

Il seguente esempio di codice mostra come utilizzare `DeleteAlarms`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteAlarms](#) consulta AWS SDK for SAP ABAP API reference.

DescribeAlarms

Il seguente esempio di codice mostra come utilizzare. DescribeAlarms

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DescribeAlarms](#) consulta AWS SDK for SAP ABAP API reference.

DisableAlarmActions

Il seguente esempio di codice mostra come utilizzare. DisableAlarmActions

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DisableAlarmActions](#) consulta AWS SDK for SAP ABAP API reference.

EnableAlarmActions

Il seguente esempio di codice mostra come utilizzare. EnableAlarmActions

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [EnableAlarmActions](#) consulta AWS SDK for SAP ABAP API reference.

ListMetrics

Il seguente esempio di codice mostra come utilizzare. ListMetrics

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [ListMetrics](#) consulta AWS SDK for SAP ABAP API reference.

PutMetricAlarm

Il seguente esempio di codice mostra come utilizzare. PutMetricAlarm

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [PutMetricAlarm](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Nozioni di base sugli allarmi

L'esempio di codice seguente mostra come:

- Crea un allarme.
- Disattivare le operazioni di allarme.
- Descrivere un allarme.
- Eliminare un allarme.

SDK per SAP ABAP

 Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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(                                " oo_result is
returned for testing purpose "
    it_alarmnames = lt_alarmnames
    ).
    MESSAGE 'Alarms retrieved' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
    DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
    MESSAGE lv_describealarms_error TYPE 'E'.
ENDTRY.

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

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [DeleteAlarms](#)
 - [DescribeAlarms](#)
 - [DisableAlarmActions](#)
 - [PutMetricAlarm](#)

Esempi di DynamoDB che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con DynamoDB.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateTable

Il seguente esempio di codice mostra come utilizzare CreateTable.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [CreateTable](#) consulta AWS SDK for SAP ABAP API reference.

DeleteItem

Il seguente esempio di codice mostra come utilizzare DeleteItem

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DeleteItem](#) consulta AWS SDK for SAP ABAP API reference.

DeleteTable

Il seguente esempio di codice mostra come utilizzare DeleteTable

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DeleteTable](#) consulta AWS SDK for SAP ABAP API reference.

DescribeTable

Il seguente esempio di codice mostra come utilizzare. DescribeTable

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DescribeTable](#) consulta AWS SDK for SAP ABAP API reference.

GetItem

Il seguente esempio di codice mostra come utilizzare. GetItem

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [GetItem](#) consulta AWS SDK for SAP ABAP API reference.

ListTables

Il seguente esempio di codice mostra come utilizzare. ListTables

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.
```

```

oo_result = lo_dyn->listtables( ).
" You can loop over the oo_result to get table properties like this.
LOOP AT oo_result->get_tablenames( ) INTO DATA(lo_table_name).
  DATA(lv_tablename) = lo_table_name->get_value( ).
ENDLOOP.
DATA(lv_tablecount) = lines( oo_result->get_tablenames( ) ).
MESSAGE 'Found ' && lv_tablecount && ' tables' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [ListTables](#) consulta AWS SDK for SAP ABAP API reference.

PutItem

Il seguente esempio di codice mostra come utilizzare. PutItem

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [PutItem](#) consulta AWS SDK for SAP ABAP API reference.

Query

Il seguente esempio di codice mostra come utilizzare. Query

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per informazioni dettagliate sull'API, consulta [Query](#) nella Documentazione di riferimento dell'API dell'AWS SDK per SAP ABAP.

Scan

Il seguente esempio di codice mostra come usare Scan.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per informazioni dettagliate sull'API, consulta [Scan](#) nella Documentazione di riferimento dell'API dell'AWS SDK per SAP ABAP.

UpdateItem

Il seguente esempio di codice mostra come usare `UpdateItem`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [UpdateItem](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Nozioni di base sull'utilizzo di tabelle, elementi e query

L'esempio di codice seguente mostra come:

- Crea una tabella in grado di contenere i dati del filmato.
- Inserisci, ottieni e aggiorna un singolo filmato nella tabella.
- Scrivi i dati del filmato nella tabella da un file JSON di esempio.
- Esegui una query sui filmati che sono stati rilasciati in un dato anno.
- Cerca i filmati che sono stati distribuiti in diversi anni.
- Elimina un filmato dalla tabella, quindi elimina la tabella.

SDK per SAP ABAP

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_attributevaluelist(
    ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
  DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
    ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
      key = 'year'
      value = NEW /aws1/cl_dyncondition(
        it_attributevaluelist = lt_attributelist
        iv_comparisonoperator = |EQ|
      ) ) ) ).
  DATA(lo_query_result) = lo_dyn->query(
    iv_tablename = iv_table_name
    it_keyconditions = lt_keyconditions ).
  DATA(lt_items) = lo_query_result->get_items( ).
  READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.

```

```

ENDTRY.

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

" Update items from table.
TRY.
  DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
  ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
    key = 'rating' value = NEW /aws1/cl_dynattrvalueupdate(
      io_value = NEW /aws1/cl_dynattributevalue( iv_n = '7.6' )
      iv_action = |PUT| ) ) ) ).
  DATA(lt_key) = VALUE /aws1/cl_dynattributevalue=>tt_key(
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
      key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) )
    ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
      key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'1980' ) ) ) ).
  DATA(lo_resp) = lo_dyn->updateitem(
    iv_tablename      = iv_table_name
    it_key            = lt_key
    it_attributeupdates = lt_attributeupdates ).
  MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.
CATCH /aws1/cx_dyncondalcheckfaile00.
  MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
CATCH /aws1/cx_dynresourcenotfoundex.
  MESSAGE 'The table or index does not exist' TYPE 'E'.
CATCH /aws1/cx_dyntransactconflictex.
  MESSAGE 'Another transaction is using the item' TYPE 'E'.

```

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

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)
 - [Scan](#)
 - [UpdateItem](#)

Esempi di Amazon EC2 che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon EC2.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

AllocateAddress

Il seguente esempio di codice mostra come utilizzare `AllocateAddress`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [AllocateAddress](#) consulta AWS SDK for SAP ABAP API reference.

AssociateAddress

Il seguente esempio di codice mostra come utilizzare `AssociateAddress`

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_ec2->associateaddress(
        iv_allocationid = iv_allocation_id
        iv_instanceid = iv_instance_id
    ).
    MESSAGE 'Associated an Elastic IP address with an EC2 instance.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

" oo_result is returned for testing purposes. "

- Per i dettagli sulle API, [AssociateAddress](#) consulta AWS SDK for SAP ABAP API reference.

CreateKeyPair

Il seguente esempio di codice mostra come utilizzare. CreateKeyPair

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
    MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.

```

" oo_result is returned for testing purposes. "

```

    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.

```

- Per i dettagli sulle API, [CreateKeyPair](#) consulta AWS SDK for SAP ABAP API reference.

CreateSecurityGroup

Il seguente esempio di codice mostra come utilizzare. CreateSecurityGroup

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

    TRY.
      oo_result = lo_ec2->createsecuritygroup(
        iv_description = 'Security group example'
        iv_groupname = iv_security_group_name
        iv_vpcid = iv_vpc_id
      ).
      MESSAGE 'Security group created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
      DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
      MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- Per i dettagli sulle API, [CreateSecurityGroup](#) consulta AWS SDK for SAP ABAP API reference.

DeleteKeyPair

Il seguente esempio di codice mostra come utilizzare. DeleteKeyPair

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteKeyPair](#) consulta AWS SDK for SAP ABAP API reference.

DeleteSecurityGroup

Il seguente esempio di codice mostra come utilizzare DeleteSecurityGroup

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteSecurityGroup](#) consulta AWS SDK for SAP ABAP API reference.

DescribeAddresses

Il seguente esempio di codice mostra come utilizzare. DescribeAddresses

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DescribeAddresses](#) consulta AWS SDK for SAP ABAP API reference.

DescribeAvailabilityZones

Il seguente esempio di codice mostra come utilizzare. DescribeAvailabilityZones

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DescribeAvailabilityZones](#) consulta AWS SDK for SAP ABAP API reference.

DescribeInstances

Il seguente esempio di codice mostra come utilizzare. DescribeInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
ENDLOOP.
MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [DescribeInstances](#) consulta AWS SDK for SAP ABAP API reference.

DescribeKeyPairs

Il seguente esempio di codice mostra come utilizzare `DescribeKeyPairs`

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DescribeKeyPairs](#) consulta AWS SDK for SAP ABAP API reference.

DescribeRegions

Il seguente esempio di codice mostra come utilizzare. DescribeRegions

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DescribeRegions](#) consulta AWS SDK for SAP ABAP API reference.

DescribeSecurityGroups

Il seguente esempio di codice mostra come utilizzare. DescribeSecurityGroups

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.

```

```

DATA lt_group_ids TYPE /aws1/cl_ec2groupidstringlist_w=>tt_groupidstringlist.
APPEND NEW /aws1/cl_ec2groupidstringlist_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.

```

- Per i dettagli sulle API, [DescribeSecurityGroups](#) consulta AWS SDK for SAP ABAP API reference.

MonitorInstances

Il seguente esempio di codice mostra come utilizzare MonitorInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [MonitorInstances](#) consulta AWS SDK for SAP ABAP API reference.

RebootInstances

Il seguente esempio di codice mostra come utilizzare. RebootInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [RebootInstances](#) consulta AWS SDK for SAP ABAP API reference.

ReleaseAddress

Il seguente esempio di codice mostra come utilizzare. ReleaseAddress

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [ReleaseAddress](#) consulta AWS SDK for SAP ABAP API reference.

RunInstances

Il seguente esempio di codice mostra come utilizzare RunInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

" Create tags for resource created during instance launch. "
DATA lt_tag specifications TYPE /aws1/
cl_ec2tag specification=>tt_tag specification list.
DATA ls_tag specifications LIKE LINE OF lt_tag specifications.
ls_tag specifications = NEW /aws1/cl_ec2tag specification(
    iv_resourcetype = 'instance'

```

```

        it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
          ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
        )
      ).
      APPEND ls_tagspecifications TO lt_tagspecifications.

      TRY.
        " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
        oo_result = lo_ec2->runinstances(                                " oo_result is
returned for testing purposes. "
          iv_imageid = iv_ami_id
          iv_instancetype = '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.

```

- Per i dettagli sulle API, [RunInstances](#) consulta AWS SDK for SAP ABAP API reference.

StartInstances

Il seguente esempio di codice mostra come utilizzare. StartInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
    ENDTRY.

```

- Per i dettagli sulle API, [StartInstances](#) consulta AWS SDK for SAP ABAP API reference.

StopInstances

Il seguente esempio di codice mostra come utilizzare. StopInstances

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [StopInstances](#) consulta AWS SDK for SAP ABAP API reference.

Esempi di Kinesis che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Kinesis.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateStream

Il seguente esempio di codice mostra come utilizzare `CreateStream`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_knslimitexceeddex .
  MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
  MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, [CreateStream](#) consulta AWS SDK for SAP ABAP API reference.

DeleteStream

Il seguente esempio di codice mostra come utilizzare DeleteStream

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

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

```
MESSAGE 'The request processing has failed because the resource is in use.'  
TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DeleteStream](#) consulta AWS SDK for SAP ABAP API reference.

DescribeStream

Il seguente esempio di codice mostra come utilizzare. DescribeStream

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_knslimitexceeddex .  
        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.
```

- Per i dettagli sulle API, [DescribeStream](#) consulta AWS SDK for SAP ABAP API reference.

GetRecords

Il seguente esempio di codice mostra come utilizzare. GetRecords

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [GetRecords](#) consulta AWS SDK for SAP ABAP API reference.

ListStreams

Il seguente esempio di codice mostra come utilizzare. ListStreams

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_knslimitexceededException .  
        MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [ListStreams](#) consulta AWS SDK for SAP ABAP API reference.

PutRecord

Il seguente esempio di codice mostra come utilizzare. PutRecord

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [PutRecord](#) consulta AWS SDK for SAP ABAP API reference.

RegisterStreamConsumer

Il seguente esempio di codice mostra come utilizzare. RegisterStreamConsumer

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
CATCH /aws1/cx_sgmresourceinuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresourcefound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, [RegisterStreamConsumer](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Inizia con i flussi di dati

L'esempio di codice seguente mostra come:

- Crea uno stream e inseriscici un record.
- Crea un iteratore shard.
- Leggi il record, quindi ripulisci le risorse.

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_knslimitexceeddex .
    MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
    MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.

"Wait for stream to becomes active."
lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
WHILE lo_stream_description->get_streamstatus( ) <> 'ACTIVE'.
    IF sy-index = 30.
        EXIT.                "maximum 5 minutes"
    ENDIF.
WAIT UP TO 10 SECONDS.
lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).

```

```

    lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
ENDWHILE.

"Create record."
TRY.
    lo_record_result = lo_kns->putrecord(
        iv_streamname = iv_stream_name
        iv_data        = iv_data
        iv_partitionkey = iv_partition_key
    ).
    MESSAGE 'Record created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex .
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex .
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex .
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex .
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex .
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired .
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_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.
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CreateStream](#)
 - [DeleteStream](#)
 - [GetRecords](#)
 - [GetShardIterator](#)
 - [PutRecord](#)

Esempi di Lambda con SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Lambda.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateFunction

Il seguente esempio di codice mostra come utilizzare `CreateFunction`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [CreateFunction](#) consulta AWS SDK for SAP ABAP API reference.

DeleteFunction

Il seguente esempio di codice mostra come utilizzare DeleteFunction

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteFunction](#) consulta AWS SDK for SAP ABAP API reference.

GetFunction

Il seguente esempio di codice mostra come utilizzare. GetFunction

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [GetFunction](#) consulta AWS SDK for SAP ABAP API reference.

Invoke

Il seguente esempio di codice mostra come utilizzare. Invoke

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per informazioni dettagliate sull'API, consulta [Invoke](#) nella documentazione di riferimento dell'SDK AWS per l'API SAP ABAP.

ListFunctions

Il seguente esempio di codice mostra come usare `ListFunctions`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [ListFunctions](#) consulta AWS SDK for SAP ABAP API reference.

UpdateFunctionCode

Il seguente esempio di codice mostra come utilizzare `UpdateFunctionCode`

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_lmd->updatefunctioncode(      " oo_result is returned for
testing purposes. "
        iv_functionname = iv_function_name
        iv_zipfile = io_zip_file
    ).

    MESSAGE 'Lambda function code updated.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodesigningcfgno00.
        MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodestorageexcdex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodeverification00.
        MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvalidcodesigex.
        MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
    CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [UpdateFunctionCode](#) consulta AWS SDK for SAP ABAP API reference.

UpdateFunctionConfiguration

Il seguente esempio di codice mostra come utilizzare. UpdateFunctionConfiguration

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [UpdateFunctionConfiguration](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Nozioni di base sulle funzioni

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM e una funzione Lambda, quindi carica il codice del gestore.
- Richiamare la funzione con un singolo parametro e ottenere i risultati.
- Aggiorna il codice della funzione e configuralo con una variabile di ambiente.
- Richiamare la funzione con nuovi parametri e ottenere i risultati. Visualizza il log di esecuzione restituito.
- Elenca le funzioni dell'account, quindi elimina le risorse.

Per ulteriori informazioni sull'utilizzo di Lambda, consulta [Creare una funzione Lambda con la console](#).

SDK per SAP ABAP

Note

C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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'.
    ENDRY.

    TRY.
        lo_iam->attachrolepolicy(
            iv_rolename = iv_role_name
            iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole'
        ).
        MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
        CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
        CATCH /aws1/cx_iamplynnotattachableex.
            MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
        CATCH /aws1/cx_iamunmodableentityex.
            MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    ENDRY.

    " 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_lmdcodestorageexcde.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDTRY.

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

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

```

```

        DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
        CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
        CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        CATCH /aws1/cx_lmdunsuppmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
        ENDTRY.

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

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

            lo_lmd->updatefunctionconfiguration(
                iv_functionname = iv_function_name

```

```

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

        "Invoke the function with new parameters and get results. Display the
execution log that's returned from the invocation."
        TRY.
            lv_json = /aws1/cl_rt_util=>string_to_xstring(
                `{` ` &&
                ` "action": "decrement", ` ` &&
                ` "number": 10 ` ` &&
                `}` `
            ).
            DATA(lo_updated_invoke_output) = lo_lmd->invoke(
                iv_functionname = iv_function_name
                iv_payload = lv_json
            ).
            ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
            " ov_updated_invoke_payload is returned for testing purposes. "
            lo_writer_json = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
lo_writer_json.
            lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
            CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
            CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
            CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
            CATCH /aws1/cx_lmdunsuppedmediatyp00.

```

```
        MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
    ENDRY.

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

" Delete the IAM role. "
```

```
    TRY.  
        lo_iam->deleterole( iv_rolename = iv_role_name ).  
        MESSAGE 'IAM role deleted.' TYPE 'I'.  
    CATCH /aws1/cx_iamnosuchentityex.  
        MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.  
    CATCH /aws1/cx_iamunmodableentityex.  
        MESSAGE 'Service that depends on the service-linked role is not  
modifiable.' TYPE 'E'.  
    ENDMETHOD.  
  
    CATCH /aws1/cx_rt_service_generic INTO lo_exception.  
        DATA(lv_error) = lo_exception->get_longtext( ).  
        MESSAGE lv_error TYPE 'E'.  
    ENDMETHOD.
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CreateFunction](#)
 - [DeleteFunction](#)
 - [GetFunction](#)
 - [Invoke](#)
 - [ListFunctions](#)
 - [UpdateFunctionCode](#)
 - [UpdateFunctionConfiguration](#)

Esempi di Amazon S3 che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon S3.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CopyObject

Il seguente esempio di codice mostra come utilizzare `CopyObject`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [CopyObject](#) consulta AWS SDK for SAP ABAP API reference.

CreateBucket

Il seguente esempio di codice mostra come utilizzare `CreateBucket`

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [CreateBucket](#) consulta AWS SDK for SAP ABAP API reference.

DeleteBucket

Il seguente esempio di codice mostra come utilizzare DeleteBucket

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteBucket](#) consulta AWS SDK for SAP ABAP API reference.

DeleteObject

Il seguente esempio di codice mostra come utilizzare DeleteObject

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteObject](#) consulta AWS SDK for SAP ABAP API reference.

GetObject

Il seguente esempio di codice mostra come utilizzare GetObject

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [GetObject](#) consulta AWS SDK for SAP ABAP API reference.

ListObjectsV2

Il seguente esempio di codice mostra come utilizzare ListObjectsV2

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sull'API, consulta [ListObjectsV2](#) in AWS SDK per il riferimento all'API SAP ABAP.

PutObject

Il seguente esempio di codice mostra come utilizzare. PutObject

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [PutObject](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Nozioni di base su bucket e oggetti

L'esempio di codice seguente mostra come:

- Crea un bucket e carica un file in tale bucket.
- Scaricare un oggetto da un bucket.
- Copiare un oggetto in una sottocartella in un bucket.
- Elencare gli oggetti in un bucket.
- Elimina il bucket e tutti gli oggetti in esso contenuti.

SDK per SAP ABAP

Note

C'è di più su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_bucketalrddyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

```
"Upload an object to an S3 bucket."
TRY.
  "Get contents of file from application server."
  DATA lv_file_content TYPE xstring.
  OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
  READ DATASET iv_key INTO lv_file_content.
  CLOSE DATASET iv_key.

  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
    iv_body = lv_file_content
  ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
  DATA(lo_result) = lo_s3->getobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
  ).
  DATA(lv_object_data) = lo_result->get_body( ).
  MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
  lo_s3->copyobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    iv_copysource = |{ iv_bucket_name }/{ iv_key }|
  ).
  MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
```

```
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDRY.

" 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.
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

SageMaker esempi che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con. SageMaker

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateEndpoint

Il seguente esempio di codice mostra come utilizzare `CreateEndpoint`.

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [CreateEndpoint](#) consulta AWS SDK for SAP ABAP API reference.

CreateModel

Il seguente esempio di codice mostra come utilizzare `CreateModel`

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [CreateModel](#) consulta AWS SDK for SAP ABAP API reference.

CreateTrainingJob

Il seguente esempio di codice mostra come utilizzare. CreateTrainingJob

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_sgmalgorithmsspec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
```

"Create ABAP internal table for hyperparameters based on input variables."

"These hyperparameters are based on the Amazon SageMaker built-in algorithm, XGBoost."

```
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_sgmresourcecelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

```

- Per i dettagli sulle API, [CreateTrainingJob](#) consulta AWS SDK for SAP ABAP API reference.

CreateTransformJob

Il seguente esempio di codice mostra come utilizzare. CreateTransformJob

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_sgmresourcecelimitexcd.
MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [CreateTransformJob](#) consulta AWS SDK for SAP ABAP API reference.

DeleteEndpoint

Il seguente esempio di codice mostra come utilizzare DeleteEndpoint

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteEndpoint](#) consulta AWS SDK for SAP ABAP API reference.

DeleteModel

Il seguente esempio di codice mostra come utilizzare DeleteModel

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

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

- Per i dettagli sulle API, [DeleteModel](#) consulta AWS SDK for SAP ABAP API reference.

DescribeTrainingJob

Il seguente esempio di codice mostra come utilizzare DescribeTrainingJob

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_sgm->describetrainingjob(      " oo_result is returned for
testing purposes. "
        iv_trainingjobname = iv_training_job_name
    ).
    MESSAGE 'Retrieved description of training job.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [DescribeTrainingJob](#) consulta AWS SDK for SAP ABAP API reference.

ListAlgorithms

Il seguente esempio di codice mostra come utilizzare. ListAlgorithms

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [ListAlgorithms](#) consulta AWS SDK for SAP ABAP API reference.

ListModels

Il seguente esempio di codice mostra come utilizzare. ListModels

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [ListModels](#) consulta AWS SDK for SAP ABAP API reference.

ListNotebookInstances

Il seguente esempio di codice mostra come utilizzare. ListNotebookInstances

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [ListNotebookInstances](#) consulta AWS SDK for SAP ABAP API reference.

ListTrainingJobs

Il seguente esempio di codice mostra come utilizzare. ListTrainingJobs

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [ListTrainingJobs](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Inizia a usare modelli ed endpoint

L'esempio di codice seguente mostra come:

- Inizia un percorso di formazione e crea un SageMaker modello.
- Crea una configurazione endpoint
- Crea un endpoint, quindi ripulisci le risorse.

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_sgmresourcecelimitexcd.
  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(
    iv_endpointconfigname = iv_ep_cfg_name
    iv_endpointname       = iv_ep_name
  ).
  MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
```

```

    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
  ENDRTRY.

  "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'.
  ENDRTRY.

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CreateEndpoint](#)

- [CreateEndpointConfig](#)
- [CreateModel](#)
- [CreateTrainingJob](#)
- [DeleteEndpoint](#)
- [DeleteEndpointConfig](#)
- [DeleteModel](#)
- [DescribeEndpoint](#)
- [DescribeTrainingJob](#)

Esempi di Amazon SNS che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon SNS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateTopic

Il seguente esempio di codice mostra come utilizzare `CreateTopic`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_sns->createtopic( iv_name = iv_topic_name ). " oo_result is  
returned for testing purposes. "  
    MESSAGE 'SNS topic created' TYPE 'I'.  
    CATCH /aws1/cx_snstopiclimitexcdex.  
        MESSAGE 'Unable to create more topics. You have reached the maximum number  
of topics allowed.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [CreateTopic](#) consulta AWS SDK for SAP ABAP API reference.

DeleteTopic

Il seguente esempio di codice mostra come utilizzare DeleteTopic

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [DeleteTopic](#) consulta AWS SDK for SAP ABAP API reference.

GetTopicAttributes

Il seguente esempio di codice mostra come utilizzare. GetTopicAttributes

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [GetTopicAttributes](#) consulta AWS SDK for SAP ABAP API reference.

ListSubscriptions

Il seguente esempio di codice mostra come utilizzare. ListSubscriptions

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.
```

```
        oo_result = lo_sns->listsuscriptions( ).           " oo_result is
returned for testing purposes. "
        DATA(lt_suscriptions) = oo_result->get_suscriptions( ).
        MESSAGE 'Retrieved list of subscribers.' TYPE 'I'.
        CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to list subscribers.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, [ListSubscriptions](#) consulta AWS SDK for SAP ABAP API reference.

ListTopics

Il seguente esempio di codice mostra come utilizzare. ListTopics

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [ListTopics](#) consulta AWS SDK for SAP ABAP API reference.

Publish

Il seguente esempio di codice mostra come utilizzare. Publish

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.
    oo_result = lo_sns->publish(
testing purposes. "           " oo_result is returned for
        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.
```

- Per informazioni dettagliate sulle API, consulta [Publish](#) nella documentazione di riferimento dell'SDK AWS per l'API SAP ABAP.

SetTopicAttributes

Il seguente esempio di codice mostra come usare `SetTopicAttributes`.

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [SetTopicAttributes](#) consulta AWS SDK for SAP ABAP API reference.

Subscribe

Il seguente esempio di codice mostra come utilizzare. Subscribe

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Sottoscrivi un indirizzo email a un argomento.

```

TRY.
    oo_result = lo_sns->subscribe(
        for testing purposes."                                "oo_result is returned
        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.

```

- Per informazioni dettagliate sulle API, consulta [Subscribe](#) nella documentazione di riferimento dell'SDK AWS per l'API SAP ABAP.

Unsubscribe

Il seguente esempio di codice mostra come utilizzare `Unsubscribe`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per informazioni dettagliate sulle API, consulta [Unsubscribe](#) nella documentazione di riferimento dell'SDK AWS per l'API SAP ABAP.

Scenari

Creazione e pubblicazione su un argomento FIFO

Gli esempi di codice seguenti mostrano come creare e pubblicare su un argomento FIFO Amazon SNS.

SDK per SAP ABAP

 Note

C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea un argomento FIFO, sottoscrivi una coda Amazon SQS FIFO all'argomento e pubblica un messaggio su un argomento Amazon SNS.

```

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

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

" Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
topic. "
" Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
an SNS FIFO topic. "
TRY.
  DATA(lo_subscribe_result) = lo_sns->subscribe(
    iv_topicarn = lv_topic_arn
    iv_protocol = 'sqs'
  )

```

```

        iv_endpoint = iv_queue_arn
    ).
    DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
    ov_subscription_arn = lv_subscription_arn.
    "
    ov_subscription_arn is returned for testing purposes. "
    MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
    CATCH /aws1/cx_snssubscriptionlmt00.
    MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
    ENDRTRY.

    " Publish message to SNS topic. "
    TRY.
        DATA lt_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>tt_messageattributemap.
        DATA ls_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>ts_messageattributemap_maprow.
        ls_msg_attributes-key = 'Importance'.
        ls_msg_attributes-value = NEW /aws1/cl_snsmessageattrvalue( iv_datatype =
'String' iv_stringvalue = 'High' ).
        INSERT ls_msg_attributes INTO TABLE lt_msg_attributes.

        DATA(lo_result) = lo_sns->publish(
            iv_topicarn = lv_topic_arn
            iv_message = 'The price of your mobile plan has been increased from $19
to $23'
            iv_subject = 'Changes to mobile plan'
            iv_messagegroupid = 'Update-2'
            iv_messagededuplicationid = 'Update-2.1'
            it_messageattributes = lt_msg_attributes
        ).
        ov_message_id = lo_result->get_messageid( ).
        "
    ov_message_id is returned for testing purposes. "
    MESSAGE 'Message was published to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDRTRY.

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CreateTopic](#)
 - [Pubblicare](#)
 - [Subscribe](#)

Esempi di Amazon SQS che utilizzano SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon SQS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

CreateQueue

Il seguente esempio di codice mostra come utilizzare `CreateQueue`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea una coda standard Amazon SQS.

```

TRY.
    oo_result = lo_sqs->createqueue( iv_queue_name = iv_queue_name ).      "
oo_result is returned for testing purposes. "
    MESSAGE 'SQS queue created.' TYPE 'I'.
    CATCH /aws1/cx_sqsqueuedeletedrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
    CATCH /aws1/cx_sqsqueueexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
ENDTRY.

```

Crea una coda Amazon SQS che attende l'arrivo di un messaggio.

```

TRY.
    DATA lt_attributes TYPE /aws1/cl_sqsqueueattrmap_w=>tt_queueattributemap.
    DATA ls_attribute TYPE /aws1/
cl_sqsqueueattrmap_w=>ts_queueattributemap_maprow.
    ls_attribute-key = 'ReceiveMessageWaitTimeSeconds'.      " Time in
seconds for long polling, such as how long the call waits for a message to arrive
in the queue before returning. "
    ls_attribute-value = NEW /aws1/cl_sqsqueueattrmap_w( iv_value =
iv_wait_time ).
    INSERT ls_attribute INTO TABLE lt_attributes.
    oo_result = lo_sqs->createqueue(      " oo_result is returned
for testing purposes. "
        iv_queue_name = iv_queue_name
        it_attributes = lt_attributes
    ).
    MESSAGE 'SQS queue created.' TYPE 'I'.
    CATCH /aws1/cx_sqsqueuedeletedrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
    CATCH /aws1/cx_sqsqueueexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sull'API, consulta il riferimento [CreateQueue](#) all'API AWS SDK for SAP ABAP.

DeleteQueue

Il seguente esempio di codice mostra come utilizzare DeleteQueue

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

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

- Per i dettagli sulle API, [DeleteQueue](#) consulta AWS SDK for SAP ABAP API reference.

GetQueueUrl

Il seguente esempio di codice mostra come utilizzare GetQueueUrl

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [GetQueueUrl](#) consulta AWS SDK for SAP ABAP API reference.

ListQueues

Il seguente esempio di codice mostra come utilizzare. ListQueues

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_sqs->listqueues( ).      " oo_result is returned for  
testing purposes. "  
    MESSAGE 'Retrieved list of queues.' TYPE 'I'.  
ENDTRY.
```

- Per i dettagli sulle API, [ListQueues](#) consulta AWS SDK for SAP ABAP API reference.

ReceiveMessage

Il seguente esempio di codice mostra come utilizzare. ReceiveMessage

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Ricevi un messaggio da una coda 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.

```

Ricevi un messaggio da una coda Amazon SQS utilizzando il supporto per sondaggi lunghi.

```

TRY.
    oo_result = lo_sqs->receivemessage(           " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        iv_waittimeseconds = iv_wait_time      " Time in seconds for long
polling, such as how long the call waits for a message to arrive in the queue
before returning. "
    ).
    DATA(lt_messages) = oo_result->get_messages( ).
    MESSAGE 'Message received from SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsoverlimit.
    MESSAGE 'Maximum number of in-flight messages reached.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, consulta il riferimento [ReceiveMessage](#) all'API AWS SDK for SAP ABAP.

SendMessage

Il seguente esempio di codice mostra come utilizzare. SendMessage

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_sqs->sendmessage(           " oo_result is returned for
testing purposes. "

```

```

        iv_queueurl = iv_queue_url
        iv_messagebody = iv_message
    ).
    MESSAGE 'Message sent to SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsinvalidmsgconts.
        MESSAGE 'Message contains non-valid characters.' TYPE 'E'.
    CATCH /aws1/cx_sqsunsupportedop.
        MESSAGE 'Operation not supported.' TYPE 'E'.
    ENDTRY.

```

- Per i dettagli sulle API, [SendMessage](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Creazione e pubblicazione su un argomento FIFO

Gli esempi di codice seguenti mostrano come creare e pubblicare su un argomento FIFO Amazon SNS.

SDK per SAP ABAP

Note

C'è di più su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea un argomento FIFO, sottoscrivi una coda Amazon SQS FIFO all'argomento e pubblica un messaggio su un argomento Amazon SNS.

```

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

TRY.
    DATA(lo_create_result) = lo_sns->createtopic(

```

```

        iv_name = iv_topic_name
        it_attributes = lt_tpc_attributes
    ).
    DATA(lv_topic_arn) = lo_create_result->get_topicarn( ).
    ov_topic_arn = lv_topic_arn.
ov_topic_arn is returned for testing purposes. "
    MESSAGE 'FIFO topic created' TYPE 'I'.
    CATCH /aws1/cx_snstopicclimitexcdex.
        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.

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [CreateTopic](#)
 - [Pubblicare](#)
 - [Subscribe](#)

Esempi di Amazon Textract con SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon Textract.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

AnalyzeDocument

Il seguente esempio di codice mostra come utilizzare `AnalyzeDocument`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.
```

- Per i dettagli sulle API, [AnalyzeDocument](#) consulta AWS SDK for SAP ABAP API reference.

DetectDocumentText

Il seguente esempio di codice mostra come utilizzare. DetectDocumentText

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DetectDocumentText](#) consulta AWS SDK for SAP ABAP API reference.

GetDocumentAnalysis

Il seguente esempio di codice mostra come utilizzare `GetDocumentAnalysis`

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

"Gets the results for an Amazon Textract"
"asynchronous operation that analyzes text in a document."
TRY.
  oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).
"oo_result is returned for testing purposes."
  WHILE oo_result->get_jobstatus( ) <> 'SUCCEEDED'.
    IF sy-index = 10.
      EXIT.                "Maximum 300 seconds.
    ENDIF.
    WAIT UP TO 30 SECONDS.
    oo_result = lo_tex->getdocumentanalysis( iv_jobid = iv_jobid ).
  ENDWHILE.

DATA(lt_blocks) = oo_result->get_blocks( ).

```

```
LOOP AT lt_blocks INTO DATA(lo_block).
  IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR, '.
    MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
  ENDIF.
ENDLOOP.
MESSAGE 'Document analysis retrieved.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidjobidex.
  MESSAGE 'Job ID is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_textrhrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, [GetDocumentAnalysis](#) consulta AWS SDK for SAP ABAP API reference.

StartDocumentAnalysis

Il seguente esempio di codice mostra come utilizzare. StartDocumentAnalysis

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_texpvthruputexclex.
    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.

```

- Per i dettagli sulle API, [StartDocumentAnalysis](#) consulta AWS SDK for SAP ABAP API reference.

StartDocumentTextDetection

Il seguente esempio di codice mostra come utilizzare. StartDocumentTextDetection SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_txbaddocumentex.
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.
```

- Per i dettagli sulle API, [StartDocumentTextDetection](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Inizia con l'analisi dei documenti

L'esempio di codice seguente mostra come:

- Avvia l'analisi asincrona.
- Ottieni l'analisi dei documenti.

SDK per SAP ABAP

 Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
  CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
  CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
  CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

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

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

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

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [GetDocumentAnalysis](#)

- [StartDocumentAnalysis](#)

Esempi di Amazon Translate con SDK per SAP ABAP

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l' AWS SDK per SAP ABAP con Amazon Translate.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)
- [Scenari](#)

Azioni

DescribeTextTranslationJob

Il seguente esempio di codice mostra come utilizzare `DescribeTextTranslationJob`.

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [DescribeTextTranslationJob](#) consulta AWS SDK for SAP ABAP API reference.

ListTextTranslationJobs

Il seguente esempio di codice mostra come utilizzare ListTextTranslationJobs

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

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

```
DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.
```

"Create an ABAP object for filtering using jobname."

```
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.

```

- Per i dettagli sulle API, [ListTextTranslationJobs](#) consulta AWS SDK for SAP ABAP API reference.

StartTextTranslationJob

Il seguente esempio di codice mostra come utilizzare. StartTextTranslationJob

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [StartTextTranslationJob](#) consulta AWS SDK for SAP ABAP API reference.

StopTextTranslationJob

Il seguente esempio di codice mostra come utilizzare. StopTextTranslationJob

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per i dettagli sulle API, [StopTextTranslationJob](#) consulta AWS SDK for SAP ABAP API reference.

TranslateText

Il seguente esempio di codice mostra come utilizzare TranslateText

SDK per SAP ABAP

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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_xl8textsizefmtexcdex .
        MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex .
        MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
```

```
CATCH /aws1/cx_xl8unsuppelanguage00 .
  MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, [TranslateText](#) consulta AWS SDK for SAP ABAP API reference.

Scenari

Inizia con i lavori di traduzione

L'esempio di codice seguente mostra come:

- Avvia un lavoro di traduzione in batch asincrono.
- Attendi il completamento del processo asincrono.
- Descrivi il lavoro asincrono.

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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.

```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
 - [DescribeTextTranslationJob](#)
 - [StartTextTranslationJob](#)

Sicurezza in SDK AWS per SAP ABAP

La sicurezza del cloud ha AWS la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gira Servizi AWS su Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per maggiori informazioni sui programmi di conformità applicabili SDK AWS per SAP ABAP, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) .
- Sicurezza nel cloud: la tua responsabilità è determinata dal Servizio AWS materiale che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa sezione comprende i seguenti argomenti.

Argomenti

- [Autenticazione del sistema SAP su AWS](#)
- [Le migliori pratiche per la sicurezza IAM](#)
- [Autorizzazioni SAP](#)
- [Operazioni sicure](#)
- [Utilizzo dei certificati con IAM Roles Anywhere](#)
- [Utilizzo di SAP Credential Store](#)

Autenticazione del sistema SAP su AWS

Prima che un sistema SAP possa effettuare chiamate per AWS conto degli utenti SAP, il sistema SAP deve autenticarsi. AWS SDK AWS per SAP ABAP supporta i seguenti tre metodi di autenticazione, selezionati nelle impostazioni del profilo SDK in. IMG

AWS SDK per SAP ABAP - L'edizione BTP può essere autenticato solo con il metodo che utilizza SAP Credential Store. [the section called “Autenticazione con chiave di accesso segreta”](#)

Argomenti

- [Autenticazione dei metadati delle istanze Amazon EC2](#)
- [Autenticazione con chiave di accesso segreta](#)
- [Autenticazione basata su certificati utilizzando IAM Roles Anywhere](#)
- [Approfondimenti](#)

Autenticazione dei metadati delle istanze Amazon EC2

I sistemi SAP in esecuzione su Amazon EC2 possono acquisire credenziali di breve durata e a rotazione automatica dai metadati delle istanze Amazon EC2. Per ulteriori informazioni, consulta [Utilizzo delle credenziali per i metadati delle istanze Amazon EC2](#).

Consigliamo vivamente questo metodo di autenticazione durante l'utilizzo di SDK per SAP ABAP. Per abilitarlo, l'amministratore di Basis deve abilitare la comunicazione HTTP in uscita. Non è richiesta alcuna ulteriore configurazione di Basis.

Note

Questo metodo di autenticazione si applica solo se i sistemi SAP sono in esecuzione su Amazon EC2. I sistemi SAP ospitati in locale o in altri ambienti cloud non possono autenticarsi utilizzando questo metodo.

Autenticazione con chiave di accesso segreta

Con questo metodo, utilizzi un ID chiave di accesso e una chiave di accesso segreta per autenticare il tuo sistema SAP. AWS Il sistema SAP accede AWS utilizzando un utente IAM. Per ulteriori informazioni, consulta [Gestione delle chiavi di accesso per gli utenti IAM](#).

L'amministratore di base riceve un ID chiave di accesso e una chiave di accesso segreta dall'amministratore AWS IAM. Il sistema SAP deve essere configurato per memorizzare l'ID della chiave di accesso e la chiave di accesso segreta.

- Proteggi, archivia e inoltra (SSF)

- Utilizza la funzionalità SSF per autenticare AWS SDK per SAP ABAP. [Per ulteriori informazioni, consulta Firme digitali e crittografia.](#)
- Puoi anche testare gli SSF envelope e le develope funzionalità con il SSF02 rapporto. Per ulteriori informazioni, consulta [Testare l'installazione SSF.](#)
- I passaggi per configurare SSF for SDK per SAP ABAP sono descritti nella transazione. /AWS1/IMG Vai a Prerequisiti tecnici, quindi seleziona Impostazioni aggiuntive per i sistemi locali.
- Archivio credenziali SAP
 - Usa SAP Credential Store per autenticare l' AWS SDK per l'edizione SAP ABAP - BTP. [Per ulteriori informazioni, consulta Cos'è SAP Credential Store?](#)
 - Vedi [Utilizzo di SAP Credential Store](#) per i passaggi di configurazione.

Autenticazione basata su certificati utilizzando IAM Roles Anywhere

Un certificato X.509 emesso dall'autorità di certificazione (CA) può essere utilizzato per l'autenticazione con Roles Anywhere. AWS Identity and Access Management Il certificato deve essere configurato in. STRUST La CA deve essere registrata con IAM Roles Anywhere come trust anchor e deve essere creato un profilo per specificare i ruoli e le politiche che IAM Roles Anywhere assumerebbe. Per ulteriori informazioni, consulta [Creazione di un ancoraggio e di un profilo di fiducia in AWS Identity and Access Management Roles](#) Anywhere.

Per i passaggi dettagliati su come utilizzare IAM Roles Anywhere con SDK per SAP ABAP, consulta [Utilizzo dei certificati con](#) IAM Roles Anywhere.

Note

La revoca dei certificati è supportata solo tramite l'uso di elenchi di revoca dei certificati importati. [Per ulteriori informazioni, vedere Revoca.](#)

Approfondimenti

Dopo aver autenticato il sistema SAP AWS, SDK per SAP ABAP esegue automaticamente l'assunzione del ruolo IAM appropriato `sts : assumeRole` per la funzione aziendale dell'utente SAP.

Le migliori pratiche per la sicurezza IAM

L'amministratore IAM sarà responsabile delle seguenti tre aree chiave.

- Garantire che il sistema SAP possa autenticarsi con i metadati di Amazon EC2 o le credenziali Secret Key.
- Garantire che il sistema SAP disponga delle autorizzazioni necessarie per `sts:assumeRole`.
- Per ogni ruolo IAM logico, creazione di un ruolo IAM per gli utenti SAP con le autorizzazioni necessarie per eseguire le funzioni aziendali (ad esempio, le autorizzazioni necessarie per Amazon S3, DynamoDB o altri servizi). Questi sono i ruoli che gli utenti SAP assumeranno.

Per ulteriori informazioni, vedere [Sicurezza](#) capitolo di SAP Lens:AWSFramework ben architettato.

Argomenti

- [Procedura consigliata per il profilo di istanza Amazon EC2](#)
- [Ruoli IAM per utenti SAP](#)

Procedura consigliata per il profilo di istanza Amazon EC2

L'istanza Amazon EC2 su cui viene eseguito il sistema SAP dispone di una serie di autorizzazioni in base al profilo dell'istanza. In genere, il profilo dell'istanza deve disporre solo delle autorizzazioni per la chiamata `sts:assumeRole`, per consentire al sistema SAP di assumere ruoli IAM specifici dell'azienda in base alle esigenze. Questa elevazione ad altri ruoli garantisce che un programma ABAP possa assumere un ruolo che dia all'utente i privilegi minimi necessari per svolgere il proprio lavoro. Ad esempio, un profilo di istanza potrebbe contenere la seguente dichiarazione.

```
{
  "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"
  ]
}
]
}

```

Questo esempio precedente consente al sistema SAP di assumere i ruoli IAM per l'utente CFO, AUDITOR o REPORTING. AWS SDK sceglierà il ruolo IAM corretto per l'utente in base al ruolo PFCG dell'utente in SAP.

Il profilo di istanza Amazon EC2 può essere utilizzato anche per altre funzioni.

- [AWS Agente di supporto per SAP HANA](#)
- [SAP su AWS Elevata disponibilità con routing di indirizzi IP sovrapposti](#)

Queste soluzioni possono anche richiedere `sts:assumeRole` autorizzazioni per ruoli specifici per il backup o il failover oppure possono richiedere l'assegnazione delle autorizzazioni direttamente al profilo dell'istanza.

Ruoli IAM per utenti SAP

Il programma ABAP necessita delle autorizzazioni per eseguire il lavoro dell'utente: leggere una tabella DynamoDB, richiamare Amazon Textract su un oggetto PDF in Amazon S3, eseguire un'AWS Lambda funzione. Lo stesso modello di sicurezza viene utilizzato in tutti i casi di AWS SDK. Puoi usare un ruolo IAM esistente che è stato utilizzato per un altro caso di AWS SDK.

L'analista aziendale SAP chiederà all'amministratore IAM il ruolo `arn:aws:` di un ruolo IAM per ogni ruolo logico necessario. Ad esempio, in uno scenario finanziario, l'analista aziendale può definire i seguenti ruoli logici IAM.

- CFO
- AUDITOR
- REPORTING

L'amministratore IAM definirà i ruoli IAM per ogni ruolo IAM logico.

CFO

- `arn:aws:iam::0123456789:role/finance-cfo`

- autorizzazioni di lettura e scrittura su un bucket Amazon S3
- autorizzazioni di lettura e scrittura su un database DynamoDB

AUDITOR

- `arn:aws:iam::0123456789:role/finance-auditor`
- autorizzazioni di lettura su un bucket Amazon S3
- autorizzazioni di lettura su un database DynamoDB

REPORTING

- `arn:aws:iam::0123456789:role/finance-reporting`
- autorizzazioni di lettura su un database DynamoDB
- nessuna autorizzazione per il bucket Amazon S3

Il business analyst inserirà i ruoli IAM in una tabella di mappatura per mappare i ruoli IAM logici con i ruoli IAM fisici.

I ruoli IAM per gli utenti SAP devono consentire `sts:assumeRole` per dirigenti fidati. I principi attendibili possono variare in base alla modalità di autenticazione del sistema SAPAWS. Per ulteriori dettagli, vedere [Specificare un preside](#).

Di seguito sono riportati alcuni esempi degli scenari SAP più comuni.

- Sistema SAP in esecuzione su Amazon EC2 con un profilo di istanza assegnato— qui, un profilo di istanza Amazon EC2 è associato a un ruolo IAM.

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

```
]
}
```

- Sistemi SAP in esecuzione su Amazon EC2 senza un profilo di istanza— qui, Amazon EC2 assume ruoli per gli utenti SAP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Principal": {
        "Service": [ "ec2.amazonaws.com" ]
      }
    }
  ]
}
```

- Sistemi SAP in esecuzione in locale— I sistemi SAP eseguiti in locale possono autenticarsi solo utilizzando la chiave di accesso segreta. Per ulteriori informazioni, vedere [Autenticazione del sistema SAP su AWS](#).

Qui, qualsiasi ruolo IAM assunto da un utente SAP deve avere una relazione di fiducia che si fidi dell'utente SAP.

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

Autorizzazioni SAP

L'autorizzazione richiesta per configurare l'SDK dipende dall'edizione SDK.

Argomenti

- [Autorizzazioni per la configurazione](#)
- [Autorizzazioni SAP per gli utenti finali](#)

Autorizzazioni per la configurazione

Per maggiori dettagli, consulta le schede seguenti.

SDK for SAP ABAP

Le seguenti autorizzazioni sono necessarie per configurare SDK per SAP ABAP.

- S_ TCODE
 - TCD = /AWS1/IMG
- S_TABU_DIS
 - ACTVT = 02, 03
 - DICBERCLS

Scegli tra i seguenti gruppi di autorizzazioni.

- /AWS1/CFG- SDK AWS per SAP ABAP v1 - Config
- /AWS1/MOD- SDK AWS per SAP ABAP v1 - Tempo di esecuzione
- /AWS1/PFL- SDK AWS per SAP ABAP v1 - Profilo SDK
- /AWS1/RES- SDK AWS per SAP ABAP v1 - Risorse logiche
- /AWS1/TRC- SDK AWS per SAP ABAP v1 - Traccia

SDK for SAP ABAP - BTP edition

Utilizza i seguenti passaggi per consentire a SDK for SAP ABAP - BTP edition di accedere alla configurazione.

1. Crea un nuovo ruolo aziendale utilizzando il modello di ruolo aziendale.
SAP_BR_BPC_EXPERT Questo modello fornisce l'accesso all'applicazione Custom Business Configuration.
2. In Dettagli generali sul ruolo, vai a Accedi alle categorie e scegli Senza restrizioni per Read, Write, Value Help.
3. Vai alla scheda Business Catalog e assegna il catalogo /AWS1/RTBTP_BCAT aziendale per fornire l'accesso alla configurazione SDK.
4. Vai alla scheda Utenti aziendali e assegna agli utenti aziendali il compito di concedere l'accesso alla configurazione SDK.

Autorizzazioni SAP per gli utenti finali

Prerequisito: definire i profili SDK

Prima che l'amministratore della sicurezza SAP possa definire i propri ruoli, il Business Analyst definirà i profili SDK nella transazione /AWS1/IMG per AWS SDK per SAP ABAP o nell'applicazione Custom Business Configuration for SDK per SAP ABAP - edizione BTP. In genere, un profilo SDK viene denominato in base alla sua funzione aziendale: ZFINANCE, ZBILLING, ZMFG, ZPAYROLL, ecc. Per ogni profilo SDK, Business Analyst definirà ruoli IAM logici con nomi brevi, come CFO, AUDITOR, REPORTING. Questi verranno mappati ai ruoli IAM reali dall'amministratore di sicurezza IAM.

Definisci PFCG o ruoli aziendali

Note

I ruoli PFCG sono chiamati ruoli aziendali in ambiente SAP BTP, ABAP.

L'amministratore della sicurezza SAP aggiungerà quindi un oggetto di autorizzazione /AWS1/SESS per concedere l'accesso a un profilo SDK.

Oggetto di autenticazione /AWS1/SESS

- Campo = /AWS1/PROF ZFINANCE

Gli utenti devono inoltre essere mappati ai ruoli IAM logici per ogni profilo SDK, a seconda della loro funzione lavorativa. Ad esempio, un revisore finanziario con accesso ai report potrebbe essere autorizzato per un ruolo IAM logico chiamato. AUDITOR

Oggetto di autenticazione /AWS1/LROL

- Campo = /AWS1/PROF ZFINANCE
- Campo /AWS1/LROL = AUDITOR

Nel frattempo, il CFO, con autorizzazioni di lettura/scrittura, potrebbe avere un ruolo PFCG che gli autorizza il ruolo logico di. CFO

Oggetto di autenticazione /AWS1/LROL

- Campo = /AWS1/PROF ZFINANCE
- Campo /AWS1/LROL = CFO

In generale, un utente deve essere autorizzato per un solo ruolo IAM logico per profilo SDK. Se un utente è autorizzato per più di un ruolo IAM (ad esempio, se il CFO è autorizzato per entrambi i ruoli IAM CFO AUDITOR logici), AWS SDK risolve il problema garantendo che il ruolo con priorità più alta (numero di sequenza inferiore) abbia effetto.

Come in tutti gli scenari di sicurezza, agli utenti dovrebbe essere concesso il minimo privilegio per svolgere le proprie funzioni lavorative. Una strategia semplice per la gestione dei ruoli PFCG consisterebbe nel denominare i singoli ruoli PFCG in base al profilo SDK e al ruolo logico che autorizzano. Ad esempio, il ruolo consente l'accesso al profilo e al Z_AWS_PFL_ZFINANCE_CFO ruolo IAM logico. ZFINANCE CFO Questi singoli ruoli possono quindi essere assegnati a ruoli composti che definiscono le funzioni lavorative. Ogni azienda ha la propria strategia per la gestione dei ruoli e ti invitiamo a definire una strategia PFCG adatta a te.

Operazioni sicure

Crittografia dei dati a riposo

AWS Le chiavi di accesso segrete vengono utilizzate per l'autenticazione dell'SDK. Sono crittografate utilizzando la funzionalità SSF o Credential Store di SAP.

Crittografia dei dati in transito

Tutte le chiamate a Servizi AWS sono crittografate con HTTPS. SAP ICM gestisce la connessione HTTPS. AWS i certificati devono essere considerati attendibili in STRUST.

Utilizzo delle API

Quando un utente ABAP assume un ruolo utilizzando `sts:assumeRole`, il nome della sessione è intitolato `USERID-SID-MANDT`, dove:

- `USERID` è l'utente ABAP della variabile. `SY-UNAME`
- `SID` è l'ID del sistema ABAP ricavato dalla `SY-SYSID` variabile.
- `MANDT` è il client ABAP della variabile. `SY-MANDT`

Il nome della sessione appare CloudTrail come nome utente. Ciò garantisce che le chiamate API di un utente ABAP possano essere ricondotte al sistema, al client e all'utente che ha avviato la chiamata. Per ulteriori informazioni, consulta [What is AWS CloudTrail?](#)

Utilizzo dei certificati con IAM Roles Anywhere

È possibile autenticare il sistema SAP AWS utilizzando l'autenticazione basata su certificati con Roles Anywhere. AWS Identity and Access Management È necessario configurare il certificato in e configurare il STRUST profilo SDK in. `/AWS1/IMG`

Prerequisiti

I seguenti prerequisiti devono essere soddisfatti prima di iniziare la configurazione per la certificazione.

- Il certificato X.509 emesso dall'autorità di certificazione (CA) deve soddisfare i seguenti requisiti.
 - Il certificato di firma deve essere un certificato v3.
 - La catena non deve superare i 5 certificati.
 - Il certificato deve supportare gli algoritmi RSA o ECDSA.
- Registra la tua CA con IAM Roles Anywhere come trust anchor e crea un profilo per specificare i ruoli/le policy di IAM Roles Anywhere. Per ulteriori informazioni, consulta [Creazione di un ancoraggio e di un profilo di fiducia in Roles Anywhere](#). AWS Identity and Access Management

- I ruoli IAM per gli utenti SAP devono essere creati dall'amministratore IAM. I ruoli devono disporre delle autorizzazioni necessarie per chiamare i ruoli richiesti. Servizi AWS Per ulteriori informazioni, consulta [Best practice for IAM Security](#).
- Crea l'autorizzazione per eseguire /AWS1/IMG la transazione. Per ulteriori informazioni, consulta [Autorizzazioni per la configurazione](#).

Procedura

Segui queste istruzioni per configurare l'autenticazione basata su certificati.

Fasi

- [Passaggio 1: definire un'applicazione SSF utilizzando Secure Store and Forward \(SSF\) di SAP](#)
- [Fase 2 — Impostare i parametri SSF](#)
- [Passaggio 3: creare il PSE e la richiesta di certificato](#)
- [Fase 4: Importazione della risposta al certificato nel PSE pertinente](#)
- [Fase 5 — Configurazione del profilo SDK per utilizzare IAM Roles Anywhere](#)

Passaggio 1: definire un'applicazione SSF utilizzando Secure Store and Forward (SSF) di SAP

1. Esegui il codice di transazione SE16 per definire un'applicazione SSF.
2. Immettete il nome della SSFAPPLIC tabella e selezionate Nuove voci.
3. Immettete un nome per l'applicazione SSF nel APPLIC campo, una descrizione nel DESCRIPT campo e selezionate l'Selected (X) opzione per i campi rimanenti.

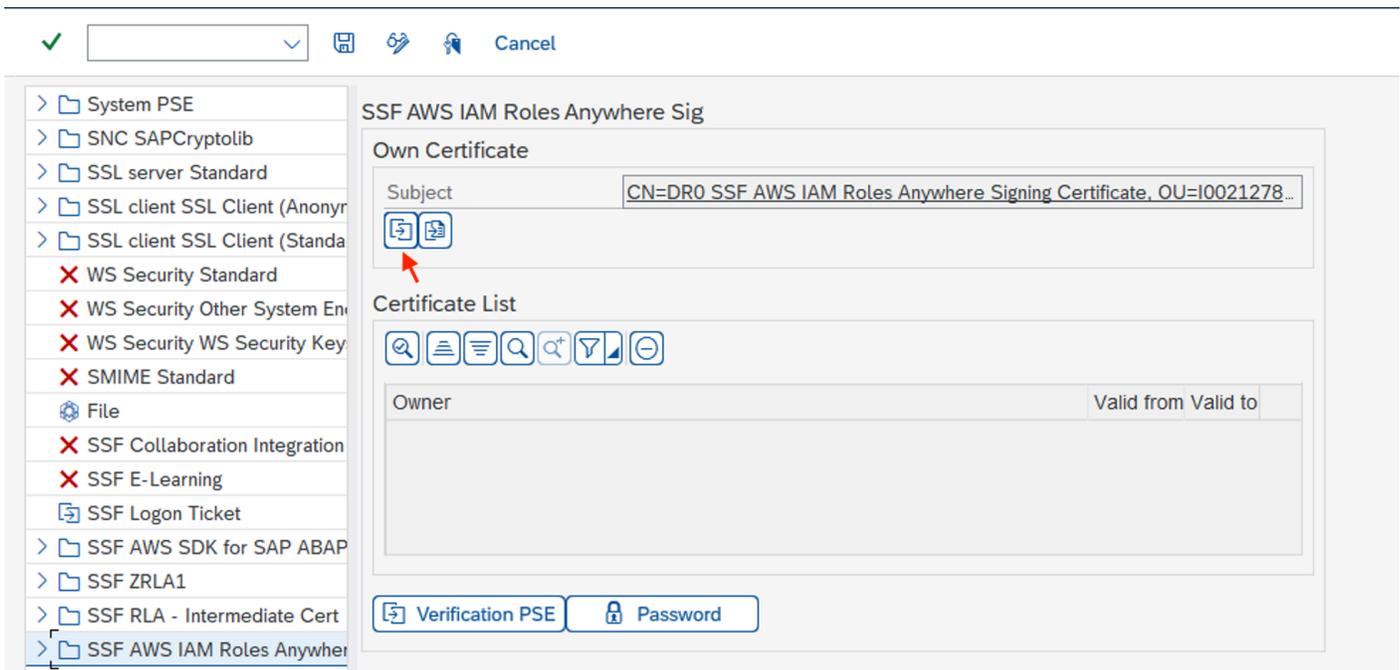
Fase 2 — Impostare i parametri SSF

1. Esegui la Guida /n/AWS1/IMG all' SDK AWS per SAP ABAP implementazione (IMG) per avviare.
2. Seleziona SDK AWS per SAP ABAP Impostazioni > Prerequisiti tecnici > Impostazioni aggiuntive per i sistemi locali.
3. Esegui l'attività IMG Set SSF Parameters.
4. Seleziona Nuove voci e scegli l'applicazione SSF creata nel passaggio precedente. Seleziona Salva.

5. Modificate l'algoritmo hash in SHA256 e l'algoritmo di crittografia in AES256-CBC. Conserva le altre impostazioni come predefinite e seleziona Salva.

Passaggio 3: creare il PSE e la richiesta di certificato

1. Esegui la /n/AWS1/IMG transazione e seleziona SDK AWS per SAP ABAP Impostazioni > Prerequisiti tecnici > Impostazioni aggiuntive per i sistemi locali.
2. Esegui l'attività Create PSE for SSF Application IMG.
3. Seleziona Modifica per la STRUST transazione.
4. Selezionate con il pulsante destro del mouse l'applicazione SSF creata in [the section called "Fase 1"](#) e scegliete Crea. Conservate tutte le altre impostazioni predefinite e selezionate Continua.
5. Seleziona Crea richiesta di certificato. Vedi l'immagine seguente. Conserva le opzioni predefinite e seleziona Continua. Copia o esporta la richiesta di certificato generata e forniscila alla tua CA. La CA verifica la richiesta e risponde con un certificato a chiave pubblica firmato.



Il processo di firma varia in base alla CA e alla tecnologia da essa utilizzata. Per un esempio, vedi [Emissione di certificati privati per entità finali](#) con AWS Private Certificate Authority.

Fase 4: Importazione della risposta al certificato nel PSE pertinente

1. Esegui la `/n/AWS1/IMG` transazione e seleziona SDK AWS per SAP ABAP Impostazioni > Prerequisiti tecnici > Impostazioni aggiuntive per i sistemi locali.
2. Esegui l'attività `Create PSE for SSF Application IMG`.
3. Seleziona Modifica per la `STRUST` transazione.
4. Scegli l'applicazione SSF, quindi seleziona Import Certificate Response nella sezione PSE sotto l'oggetto. Copia e incolla la risposta del certificato nella casella di testo o importa il file dal file system. Seleziona Continua > Salva.
5. I dettagli del certificato possono essere visualizzati selezionando l'oggetto due volte. Le informazioni vengono visualizzate nella sezione del certificato.

Fase 5 — Configurazione del profilo SDK per utilizzare IAM Roles Anywhere

1. Esegui la `/n/AWS1/IMG` transazione e seleziona SDK AWS per SAP ABAP Impostazioni > Prerequisiti tecnici > Configurazioni dell'applicazione.
2. Crea un nuovo profilo SDK e assegnagli un nome.
3. Scegli IAM Roles Anywhere come metodo di autenticazione.
 - Nel riquadro di sinistra, seleziona Autenticazione e impostazioni.
 - Crea una nuova voce e inserisci le informazioni per il tuo sistema SAP e Regione AWS.
 - Seleziona IAM Roles Anywhere per il metodo di autenticazione e seleziona Salva.
 - Seleziona Inserisci dettagli e, nella finestra pop-up, scegli l'applicazione SSF creata in [the section called “Fase 1”](#) Inserisci l'ARN di Trust Anchor e l'ARN del profilo che sono stati creati in [the section called “Prerequisiti”](#) Vedi l'immagine seguente. Seleziona Continua.

☰ SAP ✕

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. Nel riquadro sinistro, seleziona IAM Role Mapping. Inserisci un nome e fornisci l'ARN del ruolo IAM fornito dall'amministratore IAM.

Per ulteriori informazioni, consulta [Configurazione dell'applicazione](#).

Utilizzo di SAP Credential Store

SAP Credential Store viene utilizzato in SAP Business Technology Platform per archiviare in modo sicuro le credenziali per l'autenticazione con chiave di accesso segreta a. AWSÈ necessario disporre di un abbonamento per utilizzare il servizio.

Le seguenti istruzioni presuppongono che tu abbia già configurato un profilo SDK. Per ulteriori informazioni, consulta [Configurazione SDK AWS per SAP ABAP](#).

Prima di iniziare la configurazione, assicuratevi di soddisfare i prerequisiti. Per ulteriori informazioni, consulta [SAP Credential Store](#).

Argomenti

- [Fasi di configurazione](#)
- [Utilizzo di SAP Credential Store con l'SDK](#)

Fasi di configurazione

Fasi

- [Fase 1: Configurare le impostazioni per l'autenticazione](#)
- [Fase 2: Creare una chiave di servizio](#)
- [Passaggio 3: Convertire la chiave di servizio in .p12 formato](#)
- [Fase 4: Connect all'ambiente SAP BTP, ABAP](#)

Fase 1: Configurare le impostazioni per l'autenticazione

Utilizza i seguenti passaggi per configurare le impostazioni del Credential Store per l'autenticazione.

1. Vai alla scheda Impostazioni dell'istanza SAP Credential Store.
2. Seleziona Modifica configurazioni:
 - Scegli Mutual TLS come tipo di autenticazione predefinito.
 - Seleziona Disabilitato per lo stato di crittografia del payload. Il payload è crittografato in transito con HTTPS. Tuttavia, al momento il payload non può essere crittografato due volte.
3. Seleziona Salva.

Fase 2: Creare una chiave di servizio

Utilizza i seguenti passaggi per creare una chiave di servizio per Credential Store.

1. Nel riquadro sinistro dell'applicazione SAP Credential Store, accedi a Service Keys.
2. Seleziona Crea chiave di servizio.
3. Immetti un nome per la chiave di servizio e seleziona Crea.

La chiave di servizio viene creata sulla base del tipo di autenticazione scelto. Scarica la chiave di servizio e conservala al sicuro per un utilizzo successivo.

Passaggio 3: Convertire la chiave di servizio in **.p12** formato

È necessario un certificato client in questo .p12 formato per creare un utente in uscita per il sistema di comunicazione. Utilizza i seguenti passaggi per generare un .p12 certificato dai dettagli del certificato forniti nella chiave del servizio Credential Store.

1. Scarica il certificato SAP Cloud Root CA (richiesto da SAP) da [SAP Trust Center Services](#).

2. Apri il certificato SAP Cloud Root CA in qualsiasi formato di file di testo. Alla fine del file, premi Invio e copia e incolla il certificato dal campo certificato della chiave di servizio. Sostituisci i nuovi caratteri di riga \n con la nuova riga effettiva (Invio) e salva l'intero certificato in `.cer` formato file.
3. Copia la chiave dal campo chiave della chiave di servizio. Questa chiave privata deve essere trattata come dati sensibili. Incollatela in un file di testo e sostituisci i nuovi caratteri di riga \n con una nuova riga effettiva (Invio). Salva la chiave privata in un file di testo.
4. Con il certificato e la chiave privata generati nei passaggi precedenti, esegui il comando seguente per generare un `.p12` certificato.

```
openssl pkcs12 -export -out <.p12_<filename>> -inkey <private_key.key> -in  
<certificate.cer>
```

Il comando richiedeva la verifica della password di esportazione. Conserva la password per un ulteriore utilizzo.

Elimina il file di `.key` testo salvato nella tua chiave privata.

Fase 4: Connect all'ambiente SAP BTP, ABAP

Configura l'ambiente SAP BTP, ABAP per connetterti con SAP Credential Store.

Argomenti

- [Sistema di comunicazione](#)
- [Disposizione della comunicazione](#)

Sistema di comunicazione

Utilizza i seguenti passaggi per creare un sistema di comunicazione che consenta la comunicazione dall'ambiente SAP BTP, ABAP a SAP Credential Store.

1. Apri il launchpad Fiori del sistema ambientale ABAP.
2. Seleziona il riquadro Sistemi di comunicazione per aprire l'applicazione.
3. Selezionare Nuovo.
4. Immettete un nome e un ID per il sistema di comunicazione e selezionate Crea. Ad esempio, puoi assegnare un nome al sistemaZSAP_CREDSTORE.
5. Inserite le altre informazioni richieste:

- Nome host: copia il nome host dall'URL della chiave di servizio. Ad esempio, se l'URL è `https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`, allora il nome host è `credstore.mesh.cf.us10.hana.ondemand.com`.
 - Utenti per le comunicazioni in uscita: selezionare questa opzione + per aggiungere un nuovo utente.
 - a. Seleziona Certificato client SSL come meccanismo di autenticazione.
 - b. Seleziona Carica nuovo certificato:
 - Sfoglia il .p12 certificato generato nel passaggio precedente.
 - Inserisci una descrizione.
 - Inserisci la password di esportazione utilizzata per generare il .p12 certificato.
 - Seleziona Carica.
 - c. Seleziona Crea per creare un utente in uscita.
6. Seleziona Salva.
 7. Elimina la chiave di servizio scaricata nel passaggio precedente.

Disposizione della comunicazione

Utilizzare i passaggi seguenti per creare una disposizione di comunicazione che fornisca uno scenario di comunicazione per la comunicazione in uscita.

1. Apri il launchpad Fiori del sistema ambientale ABAP.
2. Seleziona il riquadro Communication Arrangements per aprire l'applicazione.
3. Selezionare Nuovo.
4. Selezionate lo scenario `/AWS1/CRED_COMM_SCENARIO` di comunicazione e immettete un nome per la disposizione di comunicazione. Ad esempio, `Z_AWS_SDK_TO_SAP_CREDSTORE`.
5. Seleziona Crea.
6. Nel campo Sistema di comunicazione, cercate il sistema di comunicazione creato nel passaggio precedente. Le altre informazioni vengono compilate automaticamente dopo la selezione del sistema.
7. Seleziona Salva.
8. Seleziona Verifica connessione per testare la connessione.

Una volta completata questa configurazione, l'ambiente ABAP può utilizzare la disposizione di comunicazione per utilizzare il servizio SAP Credential Store tramite il servizio in uscita (HTTP).

Utilizzo di SAP Credential Store con l'SDK

Fasi

- [Fase 1: Creare uno spazio dei nomi e delle credenziali](#)
- [Fase 2: Configurazione dell'applicazione Custom Business Configuration](#)

Fase 1: Creare uno spazio dei nomi e delle credenziali

[Crea un namespace e una credenziale in SAP Credential Store con SAP Help: crea, modifica ed elimina una credenziale.](#)

Inserisci i seguenti dettagli per creare una credenziale di tipo Key.

- Namespace: immetti un nome per lo spazio dei nomi e raggruppa le relative credenziali.
- Nome: immettere un nome per la chiave. Consigliamo `aws-0123456789012-username`, dove:
 - `0123456789012` è l' Account AWS ID a cui la credenziale concede l'accesso
 - `username` è il nome utente IAM a cui appartiene la credenziale
- Valore: inserisci una chiave di accesso segreta codificata in base 64. Usa il seguente comando per codificare in base 64 la tua chiave di accesso segreta.

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

Il comando legge la chiave di accesso segreta dallo standard input e la passa a base64 senza una nuova riga finale. Emette sullo schermo la chiave di accesso segreta codificata in base 64. Cancella o chiudi il terminale dopo aver copiato il valore in SAP Credential Store.

- Nome utente: inserisci l'ID della tua chiave di accesso.
- Seleziona Crea.

Viene creato un nuovo spazio dei nomi con una credenziale e le credenziali possono essere aggiunte, eliminate o modificate all'interno di questo spazio dei nomi.

Segui il principio del privilegio minimo per gestire l'accesso alle credenziali archiviate nel namespace.

Fase 2: Configurazione dell'applicazione Custom Business Configuration

Utilizza i seguenti passaggi per configurare l'applicazione Custom Business Configuration per definire le credenziali da utilizzare per l'autenticazione da parte dell'SDK.

1. Apri il launchpad Fiori del sistema ambientale ABAP.
2. Sfoglia il riquadro Custom Business Configuration per aprire l'applicazione.
3. Apri SDK Profile Business Configuration.
4. Seleziona il profilo SDK per il quale devono essere configurate le impostazioni di autenticazione per SAP Credential Store.
5. Nella scheda Autenticazione e impostazioni per il profilo selezionato, seleziona Modifica e inserisci i seguenti dettagli:
 - Metodo di autenticazione: seleziona Credenziali da SAP Credential Store.
 - Namespace: immettere lo spazio dei nomi creato in SAP Credential Store. Per ulteriori informazioni, consulta [the section called “Fase 1: Creare uno spazio dei nomi e delle credenziali”](#).
 - Nome chiave: immettere il nome della chiave di servizio creata. Per ulteriori informazioni, consulta [the section called “Fase 2: Creare una chiave di servizio”](#).
 - Disposizione di comunicazione: immettere il nome della disposizione di comunicazione creata. Per ulteriori informazioni, consulta [the section called “Disposizione della comunicazione”](#).
6. Seleziona Applica per accedere alla schermata del profilo AWS SDK.
7. Seleziona Selezione trasporto per selezionare il trasporto utilizzando il valore help.
8. Seleziona Salva.

Risoluzione dei problemi di SDK AWS per SAP ABAP

Questa sezione fornisce le procedure di risoluzione dei problemi per possibili scenari di errore.

Argomenti

- [Errore di importazione](#)
- [Vincolo di posizione non specificato](#)
- [Errori SSL](#)
- [Configurazione del profilo](#)
- [Autorizzazione IAM](#)
- [Autorizzazione per l'esecuzione delle azioni richieste](#)
- [Scenario attivo](#)
- [Caratteri speciali nel codice](#)
- [Connettività](#)

Errore di importazione

Problema: la classe 'CL_SYSTEM_UUID' non contiene un'interfaccia 'IF_SYSTEM_UUID_RFC4122_STATIC'

Causa: la nota SAP 0002619546 non è presente nel sistema.

Risoluzione: assicurati che [SAP Note 0002619546](#) sia applicato al tuo sistema.

Vincolo di posizione non specificato

Problema: il vincolo di posizione non specificato è incompatibile per l'endpoint specifico a cui è stata inviata la richiesta `region`

Causa: nel bucket Amazon S3 manca il parametro AWS Region nel parametro `io_createbucketconfiguration`

Risoluzione: quando crei un bucket in qualsiasi regione, ad eccezione `us-east-1`, specifica la regione `io_createbucketconfiguration` del bucket Amazon S3 utilizzando il parametro in `createbucket()` Non è necessario specificare un vincolo per `us-east-1`

L'esempio seguente mostra un parametro configurato `io_createbucketconfiguration` correttamente.

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

Errori SSL

Problema: mancata corrispondenza del nome host del certificato del server SSL o handshake SSL con `docs.aws.amazon.com:443` non riuscito: `SSSLERR_NO_SSL_RESPONSE`

icm/HTTPS/client_sni_enabledCausa: il **TRUE** parametro non **DEFAULT** è impostato su nel profilo.

Risoluzione: utilizza i seguenti passaggi per risolvere i problemi specifici o qualsiasi altro problema relativo a SSL.

1. Apri SAPGUI e vai alla barra dei comandi.
2. Esegui transazione. RZ10
3. Vai su Profilo e scegli DEFAULT profilo. La versione viene compilata automaticamente.
4. Nella sezione Modifica profilo, seleziona Manutenzione estesa, quindi seleziona Modifica.
5. Cerca il `icm/HTTPS/client_sni_enabled` parametro.
 - Se il parametro esiste, modificate il valore del parametro e impostatelo su **TRUE**.
 - Se il parametro non esiste, creane uno utilizzando la procedura seguente.

1. Seleziona Parametro.

Note

Assicurati di selezionare il parametro per la creazione e non per la modifica (icona a forma di matita).

2. Immettete `icm/HTTPS/client_sni_enabled` nel campo Nome del parametro.
3. Inserire **TRUE** nel campo Valore del parametro.
4. Seleziona Save (Salva).

6. Salva queste modifiche nel DEFAULT profilo ed esci.

Configurazione del profilo

Problema: impossibile trovare la configurazione nel profilo <profile_name> con lo scenario DEFAULT per <sid>: <client>

Cause: <profile_name> non è corretto o non è stato configurato.

Risoluzione: utilizza i seguenti passaggi per configurare il profilo.

1. Apri SAPGUI ed esegui la transazione. /n/AWS1/IMG
2. Vai a Configurazione dell'applicazione > Profilo SDK.
 - Se il tuo profilo è configurato, verifica che il nome del profilo sia corretto.
 - Se il tuo profilo non è configurato, segui i passaggi per configurare un profilo.
3. Seleziona Nuove voci.
 - a. Inserisci un nome e una descrizione per il profilo.
 - b. Seleziona Save (Salva).
4. Scegli la voce che hai creato nel passaggio precedente, quindi seleziona Autenticazione e impostazioni.
5. Seleziona Nuove voci, inserisci i seguenti dettagli, quindi seleziona Salva.
 - SID
 - Client
 - ID dello scenario
 - Regione AWS
 - Metodo di autenticazione
 - Seleziona Instance Role tramite Metadata per i sistemi SAP in esecuzione in AWS
 - Seleziona Credenziali da SSF Storage per sistemi SAP in esecuzione in locale o in altro cloud.
6. Seleziona IAM Role Mapping > New Entries, inserisci i seguenti dettagli e seleziona Salva.
 - Numero sequenza
 - Ruolo logico IAM

- Ruolo IAM ARN

Autorizzazione IAM

Problema: impossibile assumere il ruolo <iam_role_arn> o l'utente: <user_arn> non è autorizzato a eseguire: sts: AssumeRole sulla risorsa: <iam_role_arn>

Cause: le possibili cause di questo errore possono essere le seguenti.

- È stato specificato un ARN del ruolo IAM errato
- L'utente IAM non è autorizzato ad accedere al ruolo IAM
- Mancanza di relazione di fiducia tra il ruolo IAM assunto e il ruolo IAM o l'utente IAM che assume

Risoluzione: utilizza i seguenti passaggi per garantire che l'ARN del ruolo IAM sia corretto.

1. Apri SAPGUI ed esegui la transazione. /n/AWS1/IMG
2. Vai a Configurazione dell'applicazione > Profilo SDK e scegli il profilo che è stato configurato con il tuo ruolo IAM.
3. Seleziona IAM Role Mapping e verifica o correggi l'ARN del tuo ruolo IAM.
 - Se l'ARN del tuo ruolo IAM è corretto, assicurati che il tuo ruolo IAM sia stato configurato correttamente. Per ulteriori informazioni, consulta [Risoluzione dei problemi dei ruoli IAM](#).

Autorizzazione per l'esecuzione delle azioni richieste

Problema: l'utente <user_arn> non è autorizzato a eseguire: <action> sulla risorsa: <resource_arn>

Causa: l'utente non dispone delle autorizzazioni per eseguire un'azione.

Risoluzione: user_arn deve essere configurato con le autorizzazioni richieste resource_arn per eseguire una determinata operazione. action Per ulteriori informazioni, consulta [Autorizzazioni necessarie per accedere alle risorse IAM](#).

Scenario attivo

Problema: nessuno scenario attivo configurato

Causa: la configurazione dello scenario attivo non è stata eseguita.

Risoluzione: consulta [le impostazioni di runtime](#) per configurare uno scenario attivo.

Caratteri speciali nel codice

Avviso: il carattere 0x00A0 non può far parte di una parola ABAP

Note

Questo avviso può essere preceduto da vari messaggi di errore.

Causa: copiando e incollando codice da fonti diverse è possibile inserire caratteri speciali nel codice.

Risoluzione: quando si incolla un codice nell'editor di codice sorgente ABAP, viene visualizzato il seguente pop-up.

Sono stati rilevati caratteri spaziali che non si dividono. Convertire in spazi?

Scegli Sì per rispondere a questa domanda. Inoltre, ti consigliamo di selezionare il codice per copiarlo, invece di utilizzare il pulsante Copia nelle caselle di codice.

Connettività

Problema — SCLNT_HTTP (411): connessione diretta a tla.region.amazonaws.com:443 non riuscita: NIECONN_REFUSED (-10)

Causa: il sistema SAP non dispone di connettività Internet e non può stabilire una connessione TCP/IP alla porta 443 di tla.region.amazonaws.com.

Risoluzione: il sistema SAP deve essere in grado di stabilire la connessione agli endpoint sulla porta HTTPS 443, direttamente o tramite un server proxy. AWS È possibile stabilire/verificare la connettività Internet con una delle seguenti opzioni.

- Connessione diretta in uscita a Internet tramite un NAT o un gateway Internet
- Connessione tramite un server proxy

Per ulteriori informazioni, vedere [Connessione tramite un server proxy](#).

Argomenti aggiuntivi

Questa sezione comprende i seguenti argomenti.

Argomenti

- [SDK AWS per SAP ABAP rilasci](#)
- [Licenze SAP](#)

SDK AWS per SAP ABAP rilasci

AWS L'SDK per SAP ABAP viene fornito nei trasporti e l' AWS SDK per SAP ABAP - BTP edition viene fornito come componenti aggiuntivi. Il meccanismo per importare trasporti e componenti aggiuntivi è diverso, ma la funzionalità tecnica è la stessa. Per ulteriori informazioni, consulta [Configurazione](#).

Argomenti

- [Strategia di rilascio](#)
- [Best practice](#)
- [SDK di patch per SAP ABAP](#)
- [Installazione di un modulo aggiuntivo](#)
- [Disinstallazione di SDK per SAP ABAP](#)

Strategia di rilascio

La versione 1 di SDK AWS per SAP ABAP viene aggiornata frequentemente. Le nuove patch vengono rilasciate settimanalmente o giornalmente in base alle versioni e agli aggiornamenti di Servizi AWS. Le patch per Servizi AWS possono includere correzioni di bug e altre modifiche che aggiornano il livello di patch dell'SDK. Per ulteriori informazioni, consulta la politica di manutenzione di [AWS SDK](#) e strumenti.

Best practice

Consigliamo di mantenere lo stesso livello di patch di SDK per SAP ABAP per tutti i sistemi SAP (sviluppo, controllo qualità e produzione).

Quando applichi una patch all'SDK, importa la versione più recente nella tua sandbox. Puoi quindi importarla nei sistemi di sviluppo, controllo qualità e produzione, seguendo le normali procedure di controllo delle modifiche.

SDK di patch per SAP ABAP

Ogni versione di SDK per SAP ABAP viene fornita come un set di trasporti cumulativi, incluse tutte le correzioni di bug, le funzionalità e gli aggiornamenti. Non c'è differenza tra una patch e un trasporto di installazione. È necessario importare i trasporti più recenti in Patch SDK per SAP ABAP.

A causa delle dipendenze dei moduli `core` Runtime e API, è necessario applicare una patch al `core` modulo e a tutti gli altri moduli installati, anche se non si utilizzano più tali moduli. Ad esempio, se hai importato i `1md` trasporti `coreec2`, e quando hai installato l'SDK, devi importare i trasporti più recenti per e durante l'applicazione delle `core` patch `ec2.1md`

Installazione di un modulo aggiuntivo

Importa il trasporto per il nuovo modulo allo stesso livello di patch del modulo esistente `core` e i moduli per installare un modulo API aggiuntivo nel tuo sistema SAP. Segui le linee guida riportate in [the section called “SDK di patch per SAP ABAP”](#) se desideri importare una versione più recente del modulo. Ciò garantisce che i livelli di patch siano compatibili con tutti i moduli SDK.

Disinstallazione di SDK per SAP ABAP

[Per disinstallare SDK per SAP ABAP, devi scaricare un kit per il trasporto delle cancellazioni da https://aws.amazon.com/V1/release/uninstall-abapsdk-LATEST.zip. sdk-for-sapabap awsSdkSapabap](https://aws.amazon.com/V1/release/uninstall-abapsdk-LATEST.zip)

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

[Puoi scaricare un file di firma da https://aws.amazon.com/v1/release/uninstall-ABAPSDK-Latest.sig. sdk-for-sapabap awsSdkSapabap](https://aws.amazon.com/v1/release/uninstall-ABAPSDK-Latest.sig) Per convalidare il file, [consulta Verify SDK per SAP ABAP](#).

Per ogni modulo SDK installato sul sistema SAP, il trasporto di eliminazione corrispondente deve essere importato dal file ZIP precedente. Puoi rimuovere un singolo modulo senza disinstallare l'intero SDK. Puoi farlo importando solo il trasporto di eliminazione per il modulo che desideri rimuovere. Se state disinstallando l'intero SDK con tutti i relativi moduli, il trasporto di eliminazione principale deve essere importato per ultimo.

Ti consigliamo di testare la disinstallazione in una sandbox prima di provarla nei sistemi di sviluppo, controllo qualità o produzione.

Considerazioni

Prima di disinstallare l'SDK, consulta le seguenti considerazioni.

- Le impostazioni di configurazione dell'SDK di andranno perse. IMGDeve essere riconfigurato al momento dell'installazione.
- Se disponi di programmi Z che si basano sull'SDK, genereranno errori di sintassi dopo la rimozione dell'SDK.
- I ruoli PFCG o Business contenenti riferimenti di autorizzazione SDK avranno autorizzazioni non valide dopo la rimozione dell'SDK. Rimuovi i riferimenti di autorizzazione SDK dai ruoli PFCG prima di disinstallare l'SDK.

Note

AWS L'SDK per SAP ABAP - L'edizione BTP non può essere disinstallato durante l'anteprima per sviluppatori.

Licenze SAP

L'uso del software SAP è soggetto ai termini di SAP. L'utente è responsabile del rispetto dei termini di licenza SAP, inclusa la distribuzione del software e le condizioni di licenza indirette. Le informazioni fornite non costituiscono consulenza legale e non devono essere utilizzate per scopi di conformità delle licenze. In caso di domande sulle licenze o sui diritti relativi al software SAP, rivolgiti al tuo team legale, a SAP e/o al tuo rivenditore SAP.

Domanda: L'utilizzo di SDK per SAP ABAP influirà sulla mia licenza SAP?

Risposta: SDK AWS per SAP ABAP consente di utilizzare il proprio codice Servizi AWS ABAP. Viene utilizzato in scenari di integrazione tra un sistema SAP e. Servizi AWS Qualsiasi scenario in cui i dati del sistema SAP vengono inviati a un sistema di terze parti (non SAP) o creati da tale sistema può avere implicazioni per le licenze indirette. SAP utilizza diversi approcci per definire l'accesso indiretto, ad esempio calcoli basati sull'utente e calcoli basati sui risultati. La metodologia per definire l'accesso indiretto dipende dal contratto stipulato con SAP. È necessario conoscere le linee guida fornite nel contratto con SAP e discuterne ulteriormente con SAP o il relativo rivenditore.

Nel 2018, SAP ha rilasciato due documenti: *Indirect Access Guide for SAP Installed Base Customers* e *SAP ERP Pricing for Digital Age - Addressing Indirect/Digital Access*. Questi documenti sono disponibili sui siti Web SAP e sono esempi di approcci di licenza indiretta. Tuttavia, questi documenti non riflettono il particolare accordo dell'utente con SAP.

Cronologia dei documenti per la Guida per SDK AWS per SAP ABAP gli sviluppatori

La tabella seguente descrive le versioni della documentazione per SDK AWS per SAP ABAP.

Modifica	Descrizione	Data
Nuovo contenuto	Anteprima per sviluppatori di SDK per SAP ABAP - edizione BTP.	31 maggio 2024
Nuovo contenuto	Aggiunto Utilizzo di certificati con IAM Roles Anywhere.	1 dicembre 2023
Nuovo contenuto	Aggiunti prodotti Building con SDK.	1 dicembre 2023
Nuovo contenuto	È stato aggiunto il comportamento Retry.	1 dicembre 2023
Nuovo contenuto	Aggiunte licenze SAP.	22 settembre 2023
Rilascio pubblico	Rilancio iniziale della Developer Guide. SDK AWS per SAP ABAP	30 giugno 2023
Nuovo contenuto	Funzionalità aggiunte.SDK AWS per SAP ABAP	30 maggio 2023
Nuovo contenuto	Aggiunta la risoluzione dei problemi SDK AWS per SAP ABAP.	17 febbraio 2023
Anteprima per sviluppatori	Anteprima per SDK AWS per SAP ABAP sviluppatori della Developer Guide.	17 novembre 2022

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.